

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Мікроконтролерна система спостереження з використанням мікроконтролера ESP32-CAM
Назва теми

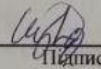
КВРКІ.190141.19.01.20 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

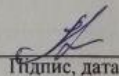
Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент IV курсу, група KI2-19-1


Підпис

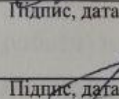
Б.С.Савченко
Ініціали, прізвище

Керівник


Підпис, дата

О.С.Засорнов
Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Т.О. Говорущенко
Ініціали, прізвище

«26» червня 2023 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень рівень бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорушенко

“ 11 ” 01 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Савченко Богдану Сергійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Мікроконтролерна система спостереження з використанням мікроконтролеру ESP32-CAM

Керівник проекту (роботи) Засорнов О.С., д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Мікроконтролери та системи відеоспостереження. Дослідження предметної області та постановка задачі

Аналіз технічних характеристик мікроконтролеру. Вибір компонентів. Проектування програмно-технічного

Програмно-апаратна реалізація та визначення необхідних складових для тестування програмно-технічного засобу

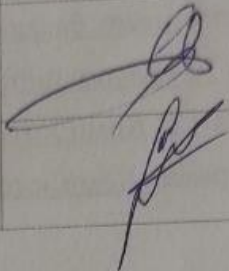

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Загальна блок-схема алгоритму

Схеми з'єднань мікроконтролеру з камерою та для виконання прошивки

Блок-схема поетапної реалізації алгоритму

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 11 » 01 2023 р.

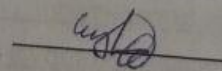
КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2023	виконано
3	Робота над розділом 1 – Огляд відомих методів та їх рішення	01.03.2023	виконано
4	Робота над розділом 2 – Аналіз наявного програмного забезпечення системи моніторингу транспортних засобів у реальному часі	01.04.2023	виконано
5	Робота над розділом 3 – Програмна реалізація та тестування програмно-технічного засобу та висновки	30.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	11.05.2023	виконано
7	Попередній захист ВКР	26.05.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент

Підпис

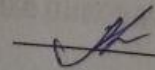
Ініціали, прізвище



Б.С.Савченко

Керівник проекту (роботи)

Підпис



Ініціали, прізвище

І. О. Засорнов

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мікроконтролерна система спостереження з використанням мікроконтролеру ESP32-CAM».

Автор роботи: Савченко Богдан Сергійович.

Керівник роботи: Засорнов Олександр Сергійович.

Пояснювальна записка: 62 с., 20 рис., 40 джерел.

Графічна частина: 3 креслення

Мікроконтролерна система спостереження з використанням мікроконтролеру ESP32-CAM


МІКРОКОНТРОЛЛЕРНА СИСТЕМА, СИСТЕМА ВІДЕОСПОСТЕРЕЖЕННЯ. ПРОШИВКА МІКРОКОНТРОЛЕРУ.

Метою дослідження полягає у проектуванні та розробці програмно-апаратного пристрою для виконання відеоспостереження за допомогою мікроконтролеру ESP32-CAM.

Об'єктом дослідження є програмне забезпечення для мікроконтролерної системи відеоспостереження

Предметом дослідження є Мікроконтролери та системи відеоспостереження

Практичне цінність полягає у проектуванні та реалізації Мікроконтролерної системи відеоспостереження




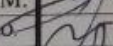


Підпис студента

Дата

ЗМІСТ

ВСТУП	4
1. ОГЛЯД ВІДОМИХ МЕТОДІВ ТА ЇХ РІШЕННЯ	6
1.1 Поняття мікроконтролеру. Складові мікроконтролерів та сфери їх використання.	6
1.2 Загальні відомості про системи відеоспостереження. Компоненти систем відеоспостереження. Класифікація систем відеоспостереження.	9
1.3 Використання мікроконтролерів у системах відеоспостереження ..	17
1.4 Постановка задачі.....	18
1.5 Висновки.	18
2. АНАЛІЗ НАЯВНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ ТРАНСПОРТНИХ ЗАСОБІВ У РЕАЛЬНОМУ ЧАСІ	20
2.1 Технічні характеристики мікроконтролеру.....	20
2.2 Вибір камери для виконання роботи.....	22
2.3 Загальні відомості про мікроконтролер ESP32-CAM	26
2.4 Сфери використання мікроконтролеру ESP32-CAM	27
2.5 Порівняння мікроконтролеру ESP32-CAM з іншими за використанням у сфері відеоспостереження.....	28
2.6 Використання мікроконтролеру ESP32-CAM в системах спостереження. Переваги та недоліки у порівнянні з іншими існуючими системами.....	32
2.7 Існуючі аналоги систем з використанням ESP32-CAM та інших мікроконтролерів.....	34
2.8 Планування Алгоритму	39
2.9 Висновок	40
3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ТА ВИСНОВКИ	43

КВРКІ. 190141.19.01.20ПЗ				
Зм.	Арк.	№докум.	Підпис	Дата
Виконав		Савченко Б.С.		
Перевір.		Засорнов О.С.		
Н.контр.		Лисенко С.М.		
Затвер.		Говорущенко Т.О.		26.05
Мікроконтролерна система спостереження з використанням мікроконтролеру ESP32-CAM Пояснювальна записка				
		Літера	Аркуш	Аркушів
		у	2	62
ХНУ КІ2-19-1				

3.1	Реалізація запланованих функцій для виконання алгоритму	43
3.2	Виконання прошивки мікроконтролеру	44
3.3	Поетапна фрагментація програмно-апаратної реалізації.....	48
3.4	Складові, необхідні для тестування алгоритму та апаратної складової	57
3.5	Теоретичні можливості для покращення алгоритму.....	58
3.6	Висновок	61
ВИСНОВКИ		63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....		65
ДОДАТОК А.....		69
	Загальна блок-схема алгоритму.....	69
ДОДАТОК Б.....		70
	Схеми з'єднань мікроконтролеру з камерою та для виконання прошивки мікроконтролеру.....	70
ДОДАТОК В.....		71
	Блок-схема поетапної реалізації алгоритму	71
ДОДАТОК Г.....		72
	Лістинг коду для виконання відеоспостереження	72
ДОДАТОК Д.....		77
	Лістинг коду для опрацювання та порівняння обличчя на зображеннях.....	77

ВСТУП

У сучасному світі безпека та контроль стали надзвичайно важливими аспектами у різних сферах життя, включаючи публічні приміщення, комерційні об'єкти, офіси та навіть особисті житлові простори. Зростаюча потреба в ефективному спостереженні та відстеженні подій спонукає до розвитку нових технологій, спрямованих на забезпечення безпеки та зниження ризиків.

В останні роки спостереження за допомогою відеокамер здобуває все більшу популярність, адже це ефективний інструмент для контролю та відстеження подій в реальному часі. Однак, традиційні системи відеоспостереження часто обмежені передачею відеопотоку на відеомонітор або зберіганням записів на зовнішньому пристрої. Це може призвести до пропуску важливих подій та ускладнити аналіз зібраної інформації.

З метою поліпшення ефективності та зручності систем відеоспостереження, у цій дипломній роботі розглядається розробка мікроконтролерної системи відеоспостереження з використанням мікроконтролера ESP32-CAM. ESP32-CAM - це невеликий та потужний мікроконтролер, що поєднує в собі функції Wi-Fi, камери та обробки зображень. Використання цього мікроконтролера дозволяє створити компактну та незалежну систему спостереження, яка може надсилати повідомлення про виявлення певних подій напряму на смартфон користувача.

Основні цілі даної роботи полягають у наступному:

1. Розробка апаратної частини системи, включаючи підключення камери до мікроконтролера ESP32-CAM та розробку алгоритмів обробки зображень.
2. Розробка програмного забезпечення, яке дозволить виконувати обробку зображень та виявлення певних подій.

За власної ініціативи, було вирішено додати до функціоналу даної системи відеоспостереження можливість віддаленого сповіщення користувача в разі виконання певних умов.

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

Дана робота буде включати огляд існуючих методів та технологій відеоспостереження, детальний опис апаратної та програмної реалізації системи, а також експериментальне дослідження її функціональності та продуктивності. Результати цієї роботи можуть знайти застосування в різних сферах, де необхідно забезпечити ефективне відеоспостереження та оперативне реагування на події.

Розвиток мікроконтролерної системи відеоспостереження на базі мікроконтролера ESP32-CAM може значно спростити та покращити процес контролю та безпеки. В цій дипломній роботі розглядаються ключові аспекти розробки такої системи та розглядаються теоретичні варіанти її майбутнього удосконалення за допомогою інших технологій.

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1. ОГЛЯД ВІДОМИХ МЕТОДІВ ТА ЇХ РІШЕННЯ

1.1 Поняття мікроконтролеру. Складові мікроконтролерів та сфери їх використання.

Мікроконтролер - це компактна інтегральна схема, яка поєднує в собі функції мікропроцесора, пам'яті і периферійних пристроїв (таких як таймери, засоби введення-виведення, аналого-цифрові та цифро-аналогові перетворювачі, комунікаційні інтерфейси тощо). Мікроконтролери зазвичай використовуються для керування різними пристроями або системами, де вони виконують вимірювання, обробку сигналів, керування введенням-виведенням та виконання інших функцій.

Перший контролер був розроблений в 1971 році Г.Буном. У 1980 році фірма Intel випускає мікроконтролер i8048, і того ж року пізніше - i8051 [1]. Вдалий набір периферійних пристроїв, можливість гнучкого вибору зовнішньої або внутрішньої програмної пам'яті і прийнятна ціна забезпечили цьому мікроконтролеру успіх на ринку. З погляду технології мікроконтролер i8051 був для свого часу дуже складним виробом — у кристалі було використано 128 тисяч транзисторів, що у чотири рази перевищувало кількість транзисторів в 16-розрядному мікропроцесорі i8086. [2].

Оскільки мікроконтролери мають вбудовану пам'ять і периферійні пристрої, вони можуть працювати автономно і виконувати завдання без необхідності зовнішнього з'єднання з комп'ютером або іншими пристроями. Їх невеликі розміри, низька вартість та низький рівень споживання енергії роблять їх ідеальними для застосувань, де обмежені ресурси, обмежена потужність або простір.

Мікроконтролер, незалежно від його можливостей у функціоналі, зазвичай складається з наступних основних компонентів [3-4]:

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Мікропроцесор (CPU), обчислювальний блок мікроконтролера, який виконує операції обробки даних і керування. У його можливий функціонал входить виконання арифметичних, логічних та керуючих операцій.

2. Однією з основних складових мікроконтролеру є наявність вбудованої пам'яті для збереження програмного коду, даних та конфігураційної інформації. Дана пам'ять поділяється на два типи та зазвичай використовуються обидва: програмна пам'ять, задача якої полягає у зберіганні програмного коду і даних, які змінюються під час роботи, та оперативна для тимчасового зберігання даних під час виконання програми.

3. Мікроконтролер має різні периферійні пристрої для забезпечення зовнішнього зв'язку і взаємодії з навколишнім середовищем. Серед можливих прикладів подібних пристроїв можна навести таймери, лічильники, пристрої введення-виведення, перетворювачі, універсальний асинхронний приймач-передавач, серійний периферійний інтерфейс, шина взаємозв'язку, прямий доступ до пам'яті та багато інших.

4. Мікроконтролер може мати вбудований годинник або таймер, які дозволяють відстежувати час і генерувати вимикання або переривання з заданою частотою або після певного інтервалу часу.

5. Деякі мікроконтролери підтримують різні інтерфейси зв'язку, такі як USB, CAN (керуюча мережа автомобіля), Ethernet, Wi-Fi, Bluetooth тощо. Це дозволяє мікроконтролеру спілкуватися з іншими пристроями або мережами.

Крім вищезгаданих, мікроконтролери можуть мати інші додаткові складові, але вони виключно залежать від того, який подальший функціонал може виконувати той чи інший мікроконтролер, а також від необхідних розробнику технічних рамок.

Мікроконтролери досить активно використовуються у сучасному світі в різних сферах діяльності. Серед основних, слід зазначити серед наступних складових[5-8]:

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

1. У різних споживчих електронних пристроях, таких як смартфони, телевізори, програвачі музики, побутові прилади тощо.

2. У системах автомобілів, включаючи двигун, електронну панель приладів, системи безпеки, системи керування передачами та інші.

3. У медичних пристроях, таких як монітори серцевого ритму, імплантовані прилади, прилади для вимірювання кров'яного тиску та інші.

4. У системах автоматизації та керування в промисловості. Вони забезпечують керування різними процесами, моніторинг параметрів, контроль над пристроями та збір даних.

5. У розробці пристроїв Інтернету речей. Вони забезпечують підключення до мережі Інтернет, обмін даними з хмарними сервісами, контроль за підключеними пристроями та взаємодію зі зовнішнім середовищем.

6. У керуванні роботами, від простих побутових роботів до складних промислових роботів. Вони забезпечують керування рухом, сприймання даних з давачів, прийняття рішень та взаємодію з оточенням.

7. У системах керування енергопостачанням, сонячних панелях, енергозберігаючих пристроях та системах "розумний будинок". Вони дозволяють ефективно використовувати енергію, моніторинг та контроль за енергозберігаючими процесами.

8. У системах відеоспостереження, беручи на себе повне її керування та контроль за камерами, обробкою і аналізом отриманих даних, запис та зберігання відеоданих, взаємодія з іншими підсистемами в системі, наприклад сигналізацією, а також керування живленням.

На цьому список їх сфер використання не завершується і з плином часу кількість систем, що використовують мікроконтролери росте завдяки високій доступності мікроконтролерів у ціні, їх компактності, можливостями, які ті здатні реалізовувати, а також додатковими складовими та давачами, з якими вони можуть працювати у синергії, що значно збільшує і без цього великий потенціал мікроконтролерів незалежно від необхідної сфери використання.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Загальні відомості про системи відеоспостереження. Компоненти систем відеоспостереження. Класифікація систем відеоспостереження.

Системи відеоспостереження (CCTV - Closed Circuit Television) використовуються для відстеження, спостереження та запису подій у реальному часі. Початок активного використання даних систем з метою безпеки почався у 1970-х роках у вигляді побутових відеомагнітофонів. Поява даного винаходу надала доступ до відеозапису як приватним, так і офіційним системам. [9].

Відповідно до швидкого розповсюдження, відбувався і швидкий розвиток даних систем. Можливості камер зростали, якість зображення також, а згодом з'явилися і мультиплексори – прилади, що дозволяли передавати зображення з кількох камер одночасно на один монітор та записувати на одну касету.

У 1980-х роках, найбільшого розвитку у системах відеоспостереження зазнали саме відеокамери. Їх електронно-променеві трубки замінили на прилади із зарядовим зв'язком. Роздільна здатність таких камер була значно нижчою, однак вони були значно меншими за розмірами та більшими за світлочутливістю.

На заміну розповсюдженим аналоговим системам відеоспостереження, у 1990-х стали з'являтися перші цифрові системи відеоспостереження. Відеореєстратори, оснащені твердими дисками, які мали змогу записувати зображення по колу (безперервно, тобто коли закінчується місце на диску — відеозображення починає записуватись спочатку). А також з'явилися перші детектори відеоаналітики— запис з відеокамер відбувався з початком руху в кадрі. Відеоспостереження у 1990-ті роки охопило банкомати, на ринок було випущено першу «відео-няню», а також з'явилася нова частина будь-якого сучасного комп'ютера – веб-камера. [10].

Основні компоненти систем відеоспостереження [11-12]:

1. Камери відеоспостереження, що знімають відео в режимі реального часу і передають його до системи моніторингу або запису. В залежності від виду

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

системи відеоспостереження, вони використовують різні типи камер, серед їх основними є аналогові, цифрові, IP-камери та безпроводові камери.

2. Кабелі, які використовуються для передачі відеосигналу з камер до системи моніторингу або запису. Залежно від типу системи, можуть використовуватись коаксіальні кабелі, волоконно-оптичні кабелі або мережеві кабелі. Для бездротових систем використовуються передавачі, що виконують передачу відеосигналу через радіохвилі.

3. Системи відеоспостереження часто мають можливість зберігати записи відео на довготривалих носіях даних. У сучасних варіантах подібних систем основними у якості носіїв використовують жорсткі диски, відео-сервери або хмарні сховища, залежно від виду системи або її потреб. Це дозволяє зберегти відеодані на певний час і використовувати їх для відновлення або аналізу.

4. Система моніторингу або запису, яка отримує відеосигнал з камер та відображає його на моніторах або записує на хмарне сховище або жорсткий диск для майбутнього його зберігання. Система моніторингу може мати додаткові функції, такі як аналіз відео, сповіщення про події, дистанційний доступ та інші.

5. Монітори використовуються для відображення відеозображення, що надходить з камер. Це можуть бути монітори на контрольних пунктах, робочих станціях операторів або монітори на віддалених місцях для, відповідно, віддаленого спостереження за об'єктом.

6. Системи відеоспостереження можуть включати функції управління, такі як панелі керування, джойстики, програмне забезпечення для управління камерами, налаштування параметрів тощо. Також можуть використовуватись аналітичні інструменти, що дозволяють автоматично виявляти рух, розпізнавати обличчя, відстежувати об'єкти тощо.

По мірі розвитку історії, відбувався і розвиток систем спостереження. Почавши розвиток у вигляді аналогових систем, вони поступово були вдосконалені, а з часом – витіснені цифровими системами. На даний момент розвиток не зупинився, а цифрові системи додатково стали поділятися на IP-

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

системи, бездротові та хмарні. Відповідно до даної класифікації, необхідно визначитись з ключовими особливостями та перевагами даних систем одна над одною:

1. Аналогові системи являють собою традиційний тип систем відеоспостереження, де аналогові камери знімають відео та перетворюють його на аналоговий сигнал, який передається за допомогою коаксіального кабелю до аналогового записувального пристрою (DVR - Digital Video Recorder)[13]. Типова схема побудови подібної системи зображена на рисунку 1.1. DVR виконує роль центрального пристрою. Він приймає аналогові відеосигнали з камер, конвертує їх в цифровий формат для запису, зберігання або виводу відеозображення на екран монітору. DVR може мати різну кількість входів для підключення камер, можливості запису в реальному часі, аналітичні функції та зручність управління.



Рисунок 1.1 – Схема побудови типової аналогової системи відеоспостереження

Основними особливостями при використанні даної системи слід зазначити наступні: аналогові системи є простими у встановленні та налаштуванні; вони мають відносно низьку вартість необхідного обладнання; даний вид систем легко

може бути використаний разом з більш старими варіантами систем відеоспостереження, адже є більш сумісним з ними, ніж інші види; використання аналогових систем відеоспостереження є найбільш оптимальним при реалізації спостереження в межах невеликого масштабу; серед основних мінусів даного виду слід зазначити досить обмежені можливості до виконання аналітики а також реалізації віддаленого доступу до даної системи[14].

2. Цифрові системи використовують цифрові камери, які знімають відео з використанням сенсорів і перетворюють його в цифровий сигнал. Цифрові сигнали передаються по мережевому кабелю (наприклад, Ethernet) до цифрового записуючого пристрою (NVR - Network Video Recorder). Наступним чином, даний пристрій записує отримане відеозображення на жорсткий диск та зберігає його. Але для перегляду отриманого відеозображення та його майбутнього опрацювання чи використання, необхідно використовувати спеціальне програмне забезпечення, яке зазвичай поставляється разом із самим NVR[13]. Цифрові системи забезпечують вищу якість зображення, більші можливості, такі як віддалений доступ та аналітику, але можуть вимагати більшої пропускної здатності мережі. Приклад побудови подібної системи зображено на рисунку 1.2.

Основними можливостями, якими відрізняється даний вид систем відеоспостереження від інших слід зазначити наступне: дані системи мають високу якість зображення, а також високу роздільну здатність; дані системи є більш гнучкими у налаштування, а також мають доступ до більших можливостей у реалізації кращого контролю параметрів камер; обробка даних виконується значно краще та швидше у порівнянні з іншими, а також можливості аналітики є розширеними, що збільшує спектр можливостей систем з їх використанням; на відміну від аналогових систем, даний вид, які наступні зазначені, має здатність до керування за допомогою віддаленого доступу через мережу; однак, основним мінусом даної системи є необхідність у використанні мережі з кращою пропускною здатністю, а також повинно використовуватись обладнання більш високої якості[15].

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

3. IP-системи можуть бути розглянуті, як розширена версія цифрових систем, де камери підключаються безпосередньо до мережі через IP адресу. При цьому, кожна камера має свою власну IP-адресу, по даній причині подібні камери нерідко називають IP-камерами, і передає відео-сигнали та аудіо-сигнали через мережу Ethernet, Wi-Fi чи інший аналог до NVR або комп'ютера. IP-системи забезпечують гнучкість, розширюваність та високу якість зображення, а також підтримують різноманітні додаткові функції та чималу кількість доступних системі протоколів. Перевагою даної системи над цифровими системами є можливість виконання передачі даних на більшій відстані, що розширює можливості у використанні подібних систем. Приклад компонентів та їх з'єднань у даному виді системи зображено на рисунку 1.3.

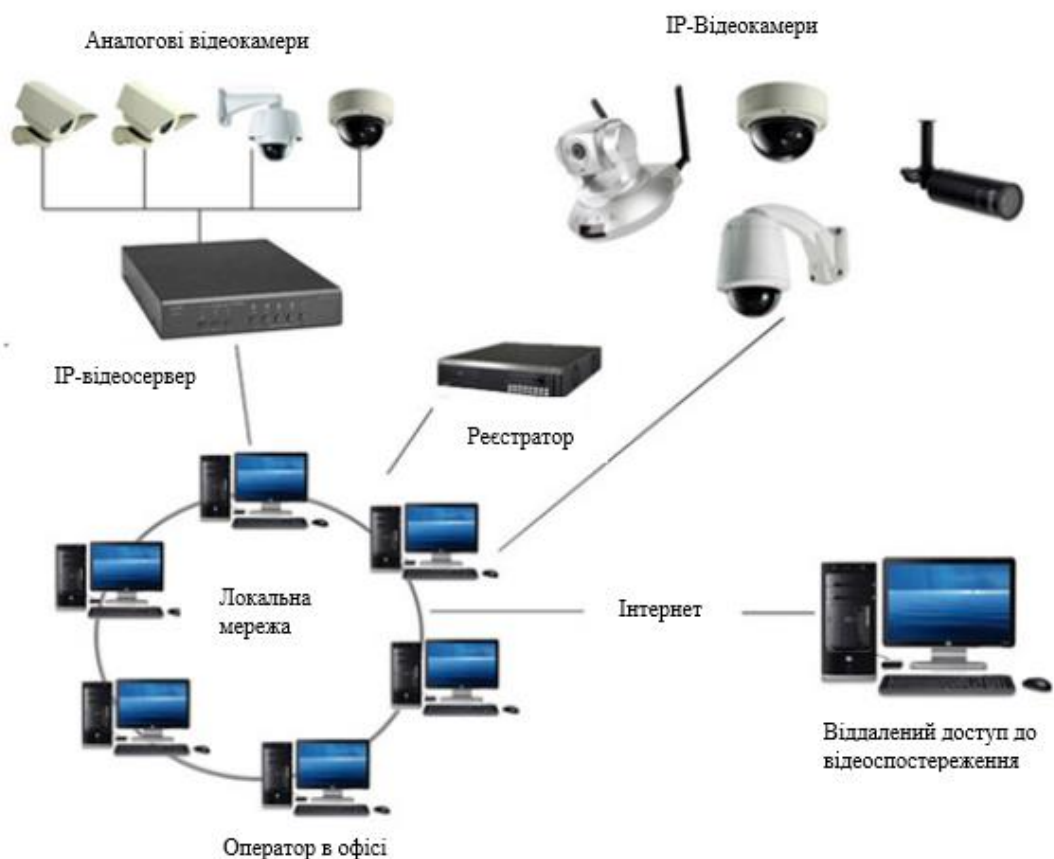


Рисунок 1.2 – Компоненти та їх порядок з'єднань у цифрових системах відеоспостереження

Передовими особливостями серед інших зазначених видів систем відеоспостереження варто згадати доступні: роздільна здатність, як і якість отриманого відеозображення, за допомогою даного виду, здатна бути конкурентноспроможною у порівнянні з цифровими системами відеоспостереження; даний вид систем є досить гнучким у масштабованості та має достатньо відкритих можливостей для майбутнього розширення відповідно до необхідних користувачеві рамок та функціоналу; використання даних систем є досить простим, адже вони мають легкий доступ до відеоданих через мережу; при використанні даних систем, переживання щодо безпеки транспортування та зберігання відеоданих зійдуть на ні, адже дана система має високий рівень безпеки та виконує надійне шифрування даних; основним мінусом даної системи є необхідність у високошвидкісному інтернет-підключенні та у спеціальному мережевому обладнанні[14].



Рисунок 1.3 – Компоненти та порядок з'єднань компонентів IP-систем відеоспостереження

Зм.	Арк.	№ докум.	Підпис	Дата

4. Бездротові системи використовують, відповідно, бездротові камери, які передають отриманий відеосигнал через радіочастоти, Wi-Fi чи інші бездротові технології до записувального пристрою. Здебільшого, дані системи використовують саме технологію Wi-Fi, адже вона здатна забезпечувати більш швидку та стабільну передачу даних на відстань в межах даної бездротової мережі. Наступним чином, приймач або записувальний пристрій приймає отриманий бездротовий сигнал від камер та зберігає відеозаписи. Даний же пристрій може бути і використаним з метою перегляду відеозаписів або може мати внутрішній жорсткий диск для їх зберігання. Керування подібними системами зазвичай виконується за допомогою систем віддаленого доступу або мобільного додатку. Бездротові системи є зручним варіантом для встановлення в місцях, де проводів підключення не є можливими або зручними. Вони забезпечують мобільність та легкість налаштування, але можуть бути більш вразливими до перешкод та обмеженням дальності передачі[16].

5. Особливості даного виду систему не сильно відрізняються від IP-камер і загалом, єдиною відмінністю є вимога до використання високоякісного саме бездротового сигналу відповідно до того, який його варіант використовується у протипагу ідентичній необхідності з'єднання IP-систем до Ethernet з високою швидкістю сигналу та пропускнуою здатністю[17].

6. Використання хмарних систем вважається найбільш сучасним підходом до систем відеоспостереження, де відеозаписи зберігаються в хмарних серверах та обробляються у разі необхідності, а доступ до них реалізовується через Інтернет. Хмарні системи дозволяють зручний віддалений доступ, спільний доступ до відеозаписів та резервне копіювання даних. Крім того, використання хмарних серверів забезпечує найкращу масштабованість та надійність серед інших видів. Приклад подібної системи зображено на рисунку 1.4.

Як і бездротові системи, системи з використанням хмарного доступу можуть бути підпорядковані дистанційному керуванню за допомогою мобільного додатку. Але основною особливістю даних систем є здатність до адаптування до будь-яких

потреб користувача, якими б масштабними вони не були. Даний вид систем добре інтегрується з іншими системами, не пов'язаними з відеоспостереженням, а також відкриті до детального аналізу відеозображень, будь це пошук обличчя та їх порівняння, будь це використання на відеозображенні більш новітніх технологій[18].

Серед особливостей даної системи особливо варто бути зазначеним те, що використання хмарного серверу дозволяє мати доступ до збережених відеоданих дистанційно. Крім цього, сам обсяг збережених даних досить великий та не має потреби у локальному їх зберіганні. Даний вид систем може скласти серйозну конкуренцію бездротовим та IP-системам у сфері забезпечення безпеки зберігання даних, але на відміну від них, даний вид дозволяє використовувати інші хмарні інтеграції для опрацювання інформації, не пересилаючи її за рамки хмарного місця збереження. Але, відповідно, даний вид систем також потребує у стабільному та швидкому Інтернет-підключенні[18].

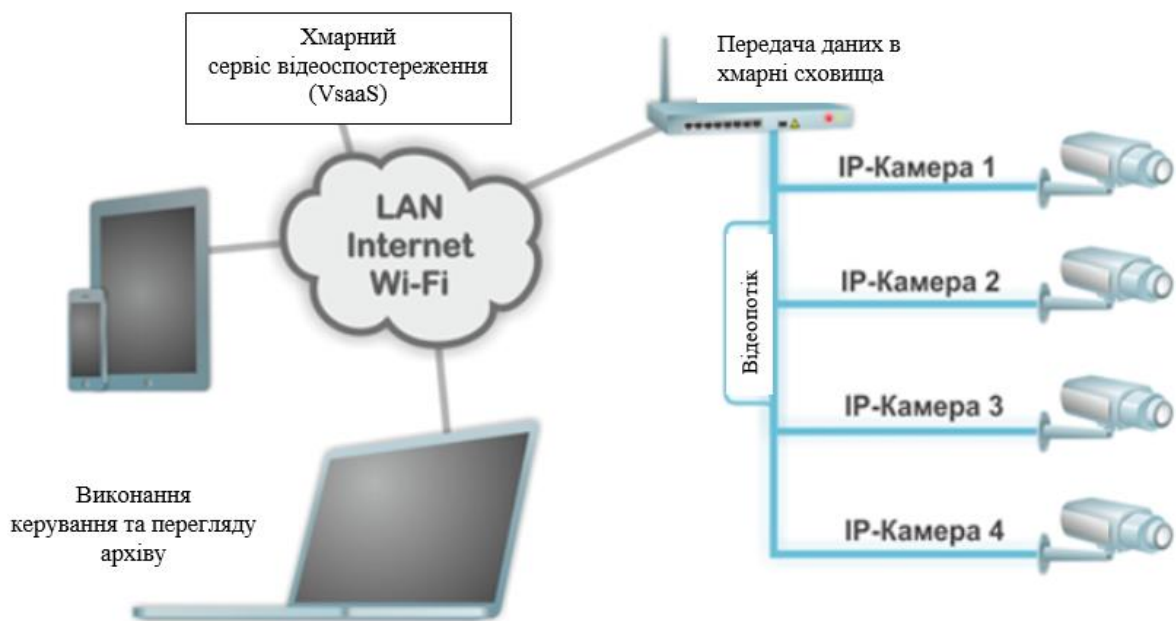


Рисунок 1.4 – Приклад виконання обробки та збереження даних хмарної системи відеоспостереження

1.3 Використання мікроконтролерів у системах відеоспостереження

У системах відеоспостереження, мікроконтролери можуть брати на себе різний функціонал і таким чином забезпечувати повний контроль над тим, що відбувається у системі. Серед доступного функціоналу є:

1. Можливість у використанні для керування рухом камер, зміни налаштувань, таких як зум, фокус, експозиція, а також для здійснення панорамного обзору і поворотів камери. Вони дозволяють програмовано керувати камерами і виконувати певні дії залежно від потреб системи.

2. Обробка та аналіз відеоданих, таких як виявлення руху, відстеження об'єктів, розпізнавання образів, виявлення облич, вимірювання відстаней тощо може бути реалізовано за допомогою мікроконтролеру. Вони можуть виконувати алгоритми обробки в реальному часі та приймати рішення на основі аналізу відеоданих.

3. Керування процесом запису та зберігання відеоданих на пам'ять також може бути реалізовано за допомогою мікроконтролеру. Серед можливих місць збережень можуть бути використані SD-карти або відеосервери, адже зазвичай мікроконтролери здатні з ними працювати, хоча це може відрізнитись від моделі або потреб користувача. Вони дозволяють встановлювати параметри запису, розмір файлів, часові мітки та інші відомості, пов'язані зі зберіганням відеоданих.

4. Виконання функцій у взаємодії з іншими системами, такими як доступ до мережі, передача даних на сервери або управління сигналізацією та системами безпеки. Вони можуть відправляти сповіщення, керувати вимикачами або виконувати інші дії, що залежать від умов і потреб системи відеоспостереження.

5. Керування живленням системи відеоспостереження. Вони можуть контролювати режими сну, ввімкнення та вимкнення пристроїв, управління енергозберігаючими функціями та виконувати інші завдання, що допомагають зменшити споживання енергії та продовжити час роботи системи.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

1.4 Постановка задачі

Під постановкою завдання полягає розробка програми для відеоспостереження з функціональною системою сповіщення користувача:

1. Реалізація виконання відеоспостереження.
2. Функціонування системи знаходження лиць під час відеоспостереження.
3. Розпізнавання форм обличчя та перевірка даного обличчя з іншими зазначеними у певному середовищі для зберігання інформації.

Здебільшого, у даному випадку рекомендується використовувати базу даних, але у рамках поставленого завдання, можна обійтись без її використання. Тому, було вирішено зберігати обличчя для порівнювання у окремій папці. При виконанні порівняння, алгоритм буде порівнювати обличчя з камери зі збереженими.

4. Сповіщення користувача, якщо було помічено обличчя, не збережене у папці.

1.5 Висновки.

Опрацювавши інформаційну базу, було в загальному визначено рамки поняття мікроконтролеру. Для кращого опрацювання даної теми, було охарактеризовано особливості мікроконтролерів, їх складові, а також розглянуто приклади та сфери з їх постійним використанням під час виробництва або дотримання рамок даної діяльності.

Паралельно, було опрацьовано поняття системи відеоспостереження. Для більш детального ознайомлення з даною темою, було розглянуто класифікацію існуючих систем відеоспостереження за принципом їх роботи. Відносно їх особливостей, було визначено їх основні переваги та умови, що необхідні для їх використання. А також, для повноцінного розуміння даних систем як цілісних структур, було визначено їх основні складові та те, який функціонал кожна з них виконує.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

Відповідно до опрацьованої інформації, була сформована постановка необхідних етапів алгоритму для виконання поставленої задачі.

Основним завданням для наступного розділу є створення базового принципу роботи необхідного алгоритму.

Додатково, є необхідним опрацювання технічних особливостей мікроконтролеру ESP32-CAM, необхідного для виконання поставленого завдання. Складовою даного завдання також є вибір необхідних компонентів у разі наявності серед них можливих альтернатив.

Наостанок, необхідно опрацювати системи, що використовують мікроконтролер ESP32-CAM або можуть з ним взаємодіяти, з метою визначення відмінностей даної системи від існуючих аналогів, пошуку їх сильних та слабких сторін, а також з метою пошуку можливих способів реалізації певних функцій з кращою ефективністю або оптимізацією.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

2. АНАЛІЗ НАЯВНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ ТРАНСПОРТНИХ ЗАСОБІВ У РЕАЛЬНОМУ ЧАСІ

2.1 Технічні характеристики мікроконтролеру

Модуль ESP32-CAM це поєднання Wi-Fi модуля з високопродуктивним мікроконтролером ESP32, відеокамерою OV2640 або OV7670, з'єднувачем під мікро-SD карту пам'яті і додатковою оперативною пам'яттю 4 МБайт. Поєднання всіх перерахованих компонентів на платі розмірами 27 x 39 мм дозволяє створювати портативні пристрої для охоронних систем, систем відеоспостереження і систем "розумний будинок". Зовнішній вигляд даного мікроконтролеру можна знайти на рисунку 2.1.



Рисунок 2.1 – Зовнішній вигляд мікроконтролеру ESP32-CAM з під'єднаною камерою OV2640

1. Модуль ESP32-CAM має такі характеристики[19].
2. Контролер: ESP32, 2 ядра, 32-біт.

					КвРКІ. 190141.19.01.20 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Робоча частота процесора: 240 МГц / 600 DMIPS.
4. Оперативна пам'ять: вбудована: 520 КБ / зовнішня: 4 Мбайт.
5. Підтримувані інтерфейси: UART/SPI/іес/PWM/ADC/DAC.
6. Вбудований Lwip і FreeRTOS.
7. Вбудований Wi-Fi модуль.
8. Підтримка STA/AP/STA + AP робочий режим.
9. Підтримка Smart Config/AirKiss розподіленої мережі.
10. Напруга живлення: 5В.
11. Роздільна здатність: 2 Мп OV2640.
12. Підтримка камер: OV2640 і OV7670.
13. Зберігання даних: міcro-SD.
14. Розмір: 27 x 39 мм.
15. Вага: 10 г.

Відповідно даних технічних характеристик мікроконтролеру, слід зазначити наступні сильні сторони у порівнянні з іншими мікроконтролерами, що використовуються при відеоспостереженні та мікроконтролерами загалом:

1. Мікроконтролер ESP32, який взятий за основу мікроконтролеру ESP32-CAM має два ядра з частотою до 240 МГц, що дозволяє обробляти великий обсяг даних та виконувати складні завдання в реальному часі.

2. ESP32-CAM має вбудований Wi-Fi модуль, що дозволяє зв'язати його з мережею бездротового зв'язку для передачі відео- та аудіоданих через мережу або для дистанційного керування.

3. Даний мікроконтролер підтримує камери OV2640 і OV7670, що дозволяє застосовувати його для захоплення високоякісних зображень та відео.

4. ESP32-CAM має підтримку міcro-SD карти, що дозволяє зберігати значно більший обсяг даних без необхідності підключення до зовнішнього сховища.

5. ESP32-CAM має різноманітні вбудовані інтерфейси, такі як UART, SPI, I2C, PWM, ADC, DAC, що дозволяє підключати різноманітні пристрої та сенсори для розширення можливостей системи відеоспостереження.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

Однак, у даного мікроконтролера є фактори, що роблять його не ідеальним в загальному, що може за певних обставин привести до зміни вибору мікроконтролера. У даному випадку дана заміна є неможливою, але слабкі сторони модулю варті бути зазначеними:

1. ESP32-CAM має вбудовану оперативну пам'ять об'ємом лише 520 КБ, що може бути недостатньо для обробки великого обсягу даних.
2. ESP32-CAM вимагає напруги живлення 5В, що може бути не зручно при використанні в певних випадках або в системах, де низька напруга живлення бажана.
3. Хоча ESP32-CAM підтримує камеру з роздільною здатністю 2 Мп, це значення може бути нижчим у порівнянні з деякими іншими камерами, що використовуються в системах відеоспостереження.

2.2 Вибір камери для виконання роботи

Як було зазначено у минулому підрозділі, мікроконтролер ESP32-CAM підтримує дві моделі камер – OV2640 та OV7670. Відповідно до цього, необхідно порівняти обидва варіанти та визначити з них модель камери, яка більше всього підходить для реалізації необхідного для отриманого завдання функціоналу. На рисунку 2.2 розміщено зовнішній вигляд даної камери для візуального ознайомлення та оцінки розмірів.

Почнемо з аналізу моделі OV7670, а точніше її технічних характеристик[20]:

1. Максимальна роздільна здатність камери OV7670 становить 640 x 480 пікселів (VGA).
2. Модель OV7670 підтримує різні формати виведення даних, включаючи YUV/ҮСЬСг 4:2:2, RGB565/555 та 8-бітний компресований формат.
3. Камера має великий динамічний діапазон, що дозволяє захоплювати деталі як в яскравих, так і в темних областях зображення.

4. Камера може захоплювати зображення з різними частотами кадрів, включаючи 30 кадрів на секунду (fps) для VGA роздільності та до 60 fps для менших роздільностей.

5. Камера підтримує паралельний інтерфейс зв'язку, такий як SCCB (Serial Camera Control Bus) або I2C (Inter-Integrated Circuit), для передачі даних і керування камерою.

6. Камера має можливість налаштовувати параметри зображення, такі як експозиція, контраст, насиченість, баланс білого та інші.

7. Камера працює при напрузі живлення від 2.7V до 3.3V.

8. Фізичні розміри камери OV7670 становлять приблизно 32 x 32 мм.

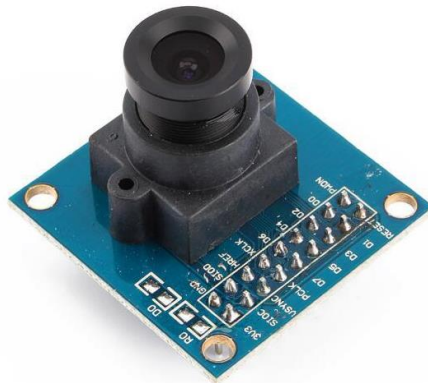


Рисунок 2.2 – Зовнішній вигляд камери OV7670

Для повноцінного порівняння, необхідно також ознайомитись з технічними характеристиками моделі OV2640[21], зображеної на рисунку 2.3:

1. Матриця зображення: 1/4" CMOS.
2. Роздільна здатність: 2 мегапікселі (1600x1200 пікселів).
3. Розмір пікселя: Приблизно 2,2 мкм.
4. В залежності від конфігурації та інтерфейсу, частота кадрів може досягати до 60 кадрів в секунду (FPS) при роздільності 320x240 пікселів.
5. Підтримка форматів зображень JPEG та YUV.

6. Підтримка різних режимів роботи, включаючи відеозапис, захоплення зображень, режим попереднього перегляду та інші.
7. Підтримка паралельного, серійного та двостороннього інтерфейсів зв'язку, таких як SCCB, SPI і DVP.
8. Камера дозволяє контролювати різні параметри зображення, такі як експозиція, баланс білого, розмір кадру, насиченість, контраст та інші.
9. Підтримка апаратного стиснення зображення в форматі JPEG, що дозволяє зменшити розмір файлу збережених зображень.
10. Вбудовані алгоритми автоматичного балансу білого, автоматичної експозиції та автофокусу для покращення якості зображення.



Рисунок 2.3 – Зовнішній вигляд камери OV2640

Відповідно до даної інформації, слід зазначити, що кожна з даних камер має власні переваги та недоліки у порівнянні одна з одною, які варті бути зазначеними для кращого розуміння того, яка саме камера найкраще підходить для виконання необхідного завдання.

Почнемо з переваг моделі OV7670: сама камера зазвичай є більш доступним варіантом та широко використовується у проектах з обмеженим бюджетом завдяки її власній низькій ціні, тому дану камеру можна легко знайти у продажу в багатьох електронних магазинах; фізичні розміри камери значно менші, що полегшує інтеграцію даної камери у різні пристрої та системи.

Однак недоліки даної полягають у технічних характеристиках, а саме: максимальна роздільна здатність камери OV7670 досить низька, що може бути недостатньо для деяких додатків або вимог до якості отриманого зображення;

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

спектр функції, доступних даних моделі досить низький, що значно звужує можливості у використанні даної камери робить; можливості налаштування камери та параметрів її зображення також досить обмежені, що також може не відповідати необхідним параметрам проекту.

В противагу особливостям камери OV7670, слід зазначити, що камера OV2640 має наступні особливості, що кардинально протиставляються: максимальна роздільна здатність камери в два мегапікселі дозволяє захоплювати значно вищу якість зображення та більше її деталей у порівнянні з OV7670; набір функцій камери значно більший, навіть у порівнянні з іншими моделями, серед яких можна відмітити автоматичний баланс білого, автоекспозицію та підтримку різних режимів фокусування, що робить дану модель більш гнучкою для різних додатків та потреб.

Однак перед вибором до використання даної моделі, варто зазначити, що дана модель є більш дорожчою, ніж OV7670, що необхідно враховувати при активному виробництві систем з камерою OV2640 серед складових. Крім цього, розміри її значно більші, що ускладнює її інтеграцію з деякими пристроями та може не підходити до використання у місцях з обмеженим простором.

Проаналізувавши можливості вищезазначених моделей, для кращої реалізації поставленого завдання, серед обох моделей камер було обрано саме варіант камери моделі OV2640.

Незважаючи на те, що даний вибір значно зменшить доступність даного алгоритму, а також може мати труднощі у реалізації за причиною більших розмірів камери, дана модель має значно більший функціонал, що дозволить не тільки краще проводити відеоспостереження, а і збільшить якість роботи, а також реалізує можливість доповнити існуючий функціонал потрібного алгоритму новими функціями.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

2.3 Загальні відомості про мікроконтролер ESP32-CAM

Мікроконтролер ESP32-CAM має свої переваги у порівнянні з іншими відомими мікроконтролерами зі схожим функціоналом. Серед основних слід зазначити наступні: даний мікроконтролер завдяки вбудованим модулям Wi-Fi та Bluetooth може бути підключеним до мереж інтернету безпроводово, а також таким же чином зв'язуватись та працювати з іншими пристроями; незалежно від того, яку камеру використовує мікроконтролер ESP32-CAM, дана камера є вбудованою, що полегшує з'єднання та налаштування мікроконтролеру; двоядерний процесор має досить високу тактову частоту, що може забезпечити достатню обчислювальну потужність, якщо цілі використання потребують чималого функціоналу; підтримка широко спектру інтерфейсів, таких як GPIO, UART, SPI, I2C, ADC та інші, надає мікроконтролеру можливість підключати та використовувати різноманітні сенсори та пристрої для розширення функціоналу систему до необхідних користувачеві потреб.

Однак, окрім переваг даного мікроконтролеру, слід враховувати і його загальні недоліки та недоліки, що виникають при порівнянні з іншими розповсюдженими моделями мікроконтролерів, а саме: кількість флеш-пам'яті та оперативної пам'яті мікроконтролеру досить обмежена, що може створити труднощі у використанні мікроконтролера з ціллю постійного опрацювання великої кількості інформації; порівняно з іншими моделями, рівень споживання енергії мікроконтролером ESP32-CAM є досить високий, що обов'язково слід враховувати при його використанні, даний фактор сильно звужує сфери можливого використання даної моделі, адже для нього не підходить робота, що потребує живлення лише від батареї; вартість даної моделі мікроконтролеру може перевищувати ціну деяких інших моделей, що слід враховувати при оцінюванні бюджету майбутнього проекту.

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4 Сфери використання мікроконтролеру ESP32-CAM

Мікроконтролер ESP32-CAM має широкі можливості застосування в різних сферах. Серед основних, варто зазначити наступні[22-24]:

1. Дана модель мікроконтролеру може використовуватись як самостійна система відеоспостереження з можливістю передачі відео по мережі. Це дозволяє створювати системи відеонагляду з дистанційним доступом та віддаленим керуванням.

2. ESP32-CAM може використовуватись для створення систем розумного дому або як її складова. Будучи частиною даної системи, він може взаємодіяти з іншими сенсорами та пристроями, контролювати освітлення, температуру, безпеку тощо. Крім того, він може функціонувати як веб-сервер для керування та моніторингу різних пристроїв у домашній мережі.

3. Завдяки своїй вбудованій підтримці Wi-Fi та можливості підключення до хмарних сервісів, ESP32-CAM використовується для розробки різноманітних проектів Інтернету Речей, таких як моніторинг рослин, системи безпеки, вимірювання середовища та багато інших.

4. Даний мікроконтролер може бути використаний як управляючий блок для роботів та автономних систем. Його можливість зчитувати дані з датчиків та взаємодія з ними може дозволити мікроконтролеру керувати рухом та взаємодіяти з навколишнім середовищем.

5. ESP32-CAM може використовуватись для потокової передачі відео в режимі реального часу на віддалені сервери або платформи для дистанційного спостереження, трансляції подій або відеоконференцій.

6. Мікроконтролер даної моделі може бути використаний для створення інтерактивних ігор, віртуальних додатків доповненої реальності, розпізнавання облич, віртуальних турів та інших проектів у сфері розваг.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

2.5 Порівняння мікроконтролера ESP32-CAM з іншими за використанням у сфері відеоспостереження

На ринку існує кілька аналогів мікроконтролера ESP32-CAM у сфері відеоспостереження. Ось декілька з них, разом з порівнянням використання та технічними характеристиками щодо ESP32-CAM[25-33]:

1. Raspberry Pi Camera Module є популярним рішенням для відеоспостереження, яке може бути використано з Raspberry Pi. Використання Raspberry Pi з камерою надає великі можливості в налаштуванні системи відеоспостереження. Хоча Raspberry Pi має більше обчислювальних ресурсів порівняно з ESP32-CAM, воно вимагає більшої енергії та коштів. Дана камера зображена на рисунку 2.4.



Рисунок 2.4 – Вигляд Raspberry Pi Camera Module

2. Arducam пропонує широкий вибір камер та модулів для вбудованих систем, включаючи відеоспостереження. Вони мають модулі камер, які можуть бути використані з різними мікроконтролерами, включаючи ESP32-CAM. Технічні характеристики та використання Arducam можуть різнитися в залежності від конкретної моделі. До прикладу, камера Arducam MINI-IMS477, зображена на рисунку 2.5, здатна значно ефективніше виконувати функціонал камери OV2640

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

завдяки більшій роздільній здатності, здатності до використання різних об'єктів відповідно до потреб користувача та інших. Однак у сфері автоматизації та яка складову системи “розумний будинок” камеру MINI-IMS477 буде використовувати менш доцільно, адже ESP32-CAM має власні переваги у віддаленому моніторингу та мережевому підключенні.



Рисунок 2.5 – Модель Arducam MINI-IMS477

3. ESP-EYE є спеціалізованим модулем відеоспостереження на базі ESP32, розробленим самою компанією Espressif. Він має вбудовану камеру OV2640, а також мікрофон та роз'єм для підключення додаткових датчиків. ESP-EYE також підтримує розробку з використанням фреймворку ESP-IDF. Відповідно до даного переліку, можна прийти до висновку, що ESP-EYE є більш доповненою версією мікроконтролера ESP32-CAM і відповідно цього, слід зазначити, що і можливості використання даного мікроконтролера більш розширені. Але за причиною відсутності заміни у процесорі та пам'яті, слід враховувати, що деякі з функцій можуть виконуватись менш ефективно у порівнянні. Даний модуль зображено на рисунку 2.6.

4. Wemos D1 Mini Camera є компактним модулем на базі ESP8266 з вбудованою камерою. Він має меншу потужність порівняно з ESP32-CAM, але може бути використаний для значно простіших проектів відеоспостереження з

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

6. BeagleBone Black - це ще один одноплатний комп'ютер, який може бути використаний для відеоспостереження. Він має вбудований відеопроцесор та HDMI-вихід, що дозволяє підключати зовнішні монітори. BeagleBone Black також має вбудований Ethernet-порт та підтримку різних камерних модулів. Даний мікрокомп'ютер зображено на рисунку 2.9.

7.



Рисунок 2.8 – Камера Orange Pi на 13 мегапікселів для одноплатного мікрокомп'ютера



Рисунок 2.9 – Одноплатний комп'ютер з під'єднаною камерою Pixy2

8. NVIDIA Jetson Nano є потужною платформою для вбудованих систем, яка забезпечує великі можливості в обробці відео. Вона має багатофункціональну

					КвРКІ. 190141.19.01.20 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

камеру та графічний процесор CUDA, який дозволяє виконувати складні алгоритми обробки відео в реальному часі. Однак, Jetson Nano вимагає більшої потужності та має вищу ціну порівняно з ESP32-CAM. З ціллю ознайомлення, зовнішній вигляд даного приладу можна знайти на рисунку 2.10.

Наведені приклади аналогів ESP32-CAM демонструють те, що здебільшого вони використовуюється у значно меншому обсязі та у більш специфічних рамках. Крім того, кожен з вищезгаданих аналогів мікроконтролеру ESP32-CAM має значно гірші характеристики, будь це збільшена енергозатратність у Raspberry Pi Camera Module, або менша потужність у Wemos D1, або вищу ціну та необхідну потужність для NVIDIA Jetson Nano.

Таким чином, це забезпечує для ESP32-CAM більшу долю ринку та розповсюдженість у системах відеоспостереження, створених крупними компаніями для загального використання або у значно менших проектах.

Але при цьому, серії мікроконтролерів та камер Arducam, Orange Pi та модуль ESP-EYE можуть виступати основними конкурентами завдяки більшому асортименту можливостей, які можуть виконувати дані моделі. Хоча у випадку серій мікроконтролерів, поодиноці кожна з моделей значно програє ESP32-CAM у розповсюдженості.



Рисунок 2.10 – Мікрокомп'ютер NVIDIA Jetson Nano з під'єднаною камерою ARO330 MIPI-CSI

2.6 Використання мікроконтролеру ESP32-CAM в системах спостереження.

Переваги та недоліки у порівнянні з іншими існуючими системами.

Мікроконтролер ESP32-CAM часто використовується у системах відеоспостереження, особливо в проектах, які потребують невеликого розміру, низької вартості і низького енергоспоживання. Основна причина цього полягає у сильній комбінації, що використовується у даному мікроконтролері, а саме поєднання мікроконтролеру ESP32 з модулем камери OV2640. Хоч дана модель і підтримує також модель камери OV7670, дана модель використовується більш розповсюджено.

Технічні особливості мікроконтролеру ESP32-CAM та камери, що він використовує, забезпечують йому широке використання, однак з метою кращого визначення функціоналу, як частини системи, необхідно враховувати слабкі та сильні сторони при його використанні. До того ж, окрім вищезазначених особливостей у порівнянні з іншими мікроконтролерами, варто враховувати і особливості, що виникають при саме при використанні у даній сфері у порівнянні з іншими системами відеоспостереження.

Серед переваг ESP32-CAM у сфері відеоспостереження слід зазначити наступні його особливості: даний мікроконтролер пропонує потужний набір функцій, таких як бездротове підключення Wi-Fi, можливість запису відео та фотографування, підтримка мережевих протоколів і додаткових модулів зв'язку, що забезпечує даній моделі широкий функціонал; наявність вже інтегрованої системи в камеру може значно зменшити фінансові витрати при проектуванні даної системи, хоча дані затрати навіть так є досить малими у порівнянні з іншими системами відеоспостереження, що робить його привабливим рішенням для проектів відеоспостереження з низьким бюджетом; даний мікроконтролер має вбудовану здатність до обробки зображень, серед функціоналу якого є можливість розпізнавання обличь, виявлення руху та інші; розробка систем з використанням мікроконтролеру ESP32-CAM виконується досить легко, адже даний

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

мікроконтролер має широку підтримку програмних бібліотек та середовищ розробки.

Основні ж недоліки даного мікроконтролеру серед систем відеоспостереження майже залишилися без змін відносно порівняння ESP32-CAM з іншими мікроконтролерами. Роздільна здатність камери OV2640 залишається досить низькою для використання у системах відеоспостереження, що не рекомендує використовувати даний мікроконтролер у разі необхідності високої якості відеозображення. Ключовим недоліком ESP32-CAM є фактична відсутність вбудованого сховища даних, що вимагає від користувача чи розробника враховувати необхідність у його покупці та налаштуванні. Даний факт може призвести до труднощів у розробці системи відеоспостереження, а також значно зменшити обсяг проектів, у яких він використовується. І наостанок, необхідність у стабільному джерелі живлення для безперебійної роботи не дозволяє використовувати даний мікроконтролер у автономних системах відеоспостереження, а також обмежує здатність до роботи даного мікроконтролеру в разі аварійного вимкнення енергії[34].

2.7 Існуючі аналоги систем з використанням ESP32-CAM та інших мікроконтролерів

На ринку існує кілька систем відеоспостереження, які використовують мікроконтролер ESP32-CAM. Серед відомих їх аналогів можна визначити декілька основних з них:

1. Home Assistant, логотип якого зображено на рисунку 2.11, є відкритим програмним забезпеченням для домашньої автоматизації, яке підтримує інтеграцію з мікроконтролером ESP32-CAM для створення системи відеоспостереження в домашніх умовах. З допомогою Home Assistant можна налаштувати відеострім з камери ESP32-CAM та отримувати сповіщення при виявленні руху або інших подіях [35].

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 34
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.11 – Логотип Home Assistant

Можливі способи використання Home Assistant у поєднанні з ESP32-CAM

1. Мікроконтролер ESP32-CAM може використовуватись як IP-камера для відеоспостереження. Він може передавати відеопотік на Home Assistant, що дозволяє переглядати наживо відео з камери через інтерфейс Home Assistant або зберігати записи відео на сервері Home Assistant[36].

2. ESP32-CAM може підключатися до різних датчиків, наприклад, датчиків руху, датчиків температури та вологості, датчиків освітлення тощо. Ви можете програмувати ESP32-CAM для зчитування даних з цих датчиків та передавати їх до Home Assistant. Це дозволить реалізувати моніторинг стану різних параметрів домашнього середовища на момент перевірки.

3. За допомогою ESP32-CAM та Home Assistant можна реалізувати керування домашніми приладами. Наприклад, можна підключити ІК-передатчик до ESP32-CAM і використовувати його для керування телевізором, кондиціонером або іншими пристроями через Home Assistant.

Повертаючись до поставленого завдання, варто зазначити, що найбільш ймовірно, технологія Home Assistant, при взаємодії з даним алгоритмом, лише доповнить список доступних сфер його використання, адже повідомляюча система відеоспостереження може бути досить практичною у використанні в системах “розумного будинку” для покращення надійності системи безпеки. Хоча при цьому, за допомогою реалізації зв’язку з різними системами датчиків, можна

реалізувати покращену безпеку, додавши до системи відеоспостереження постійну роботу давачі руху або активацію відеоспостереженні, якщо даний давач щось помітив. Або використати з'єднання з давачем освітлення, якщо є необхідність, щоб відеоспостереження починались у нічний період доби.

2. OpenCV (Open Source Computer Vision Library) є бібліотекою комп'ютерного зору, яка підтримує ESP32-CAM. За допомогою OpenCV можна реалізувати різноманітні функції відеоспостереження, включаючи виявлення об'єктів, слідкування руху, розпізнавання облич та багато іншого[37] Для кращого запам'ятовування, було розроблено логотип для даної бібліотеки, який зображено на рисунку 2.12.

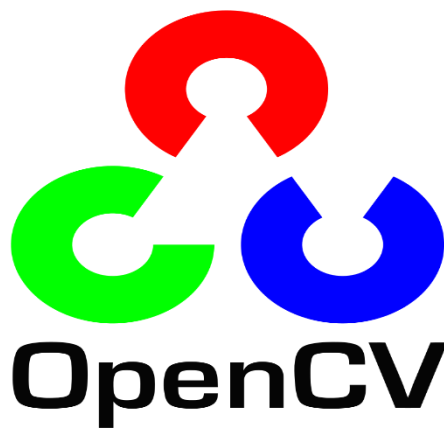


Рисунок 2.12 – Логотип OpenCV

У поєднанні з мікроконтролером ESP32-CAM, дана бібліотека може реалізувати обробку отриманого відеозображення. Сама обробка може бути виконана кількома способами, а саме на окремому сервері чи комп'ютері, або на самому мікроконтролері, якщо обчислювальні можливості є обмеженими або у випадку необхідності автономного функціонування без залучення зовнішнього комп'ютера[38].

У рамках даної роботи, було вирішено використати елементи даної бібліотеки у коді для реалізації системи знаходження лиць у кадрі, визначення форми обличчя за допомогою рамки для подальшого розпізнавання, а також порівняння з лицями на зображеннях у папці, підключеної до системи

відеоспостереження. Обробка отриманого відеозображення буде відбуватись саме на комп'ютері.

3. MotionEyeOS є операційною системою, спеціально розробленою для систем відеоспостереження на базі одноплатних комп'ютерів та мікроконтролерів, включаючи ESP32-CAM. Вона надає веб-інтерфейс для налаштування камери та збереження відеозаписів, підтримує виявлення руху, сповіщення та багато інших функцій[39]. Дана операційна система має власний логотип, зображений на ризунку 2.13.

Серед можливостей, які надає система MotionEyeOS є чимала кількість, яка могла б бути використаною для покращення реалізації необхідного алгоритму, але найбільш доцільною серед них була саме система сповіщення. Тому для її реалізації було рішено використати в майбутньому матеріали саме даної системи або хоча б взяти їх за можливий орієнтир у напрямку принципу роботи системи сповіщень.



Рисунок 2.13 – Логотип MotionEyeOS

4. TinyCam Monitor є додатком для мобільних пристроїв (Android та iOS), який підтримує використання ESP32-CAM як камери в системі відеоспостереження. Додаток дозволяє переглядати відеострім з ESP32-CAM, здійснювати запис відео, встановлювати режими сповіщень та багато іншого[40]. З

метою достовірного знаходження в PlayMarket та AppStore, даний додаток має логотип, зображений на рисунку 2.14.



Рисунок 2.14 – Логотип додатку TinyCam Monitor в PlayMarket

Технологія TinyCam Monitor також може бути використана для покращення необхідного алгоритму. За її допомогою, може бути створений мобільний додаток для віддаленого спостереження у разі необхідності. Таким чином, відправлення повідомлення на пошту може повністю замінити на отримання сповіщення у додатку. При отриманні сповіщення, людина може відкрити додаток та особисто побачити що відбувається у місці спостереження.

Або, при відповідному доповненні функціоналу, можна реалізувати активацію запису відеозображення, якщо камера помітила обличчя, яке не є внесеним у папку чи базу даних, якщо вона використовується і після чого, зберігала відеозапис на телефоні власника з можливістю переглянути даний запис. Але даний функціонал поки було вирішено залишити серед можливих доповнень та не буде внесено до коду даної роботи.

2.8 Планування Алгоритму

Темою даної роботи є реалізація повідомляючої системи відеоспостереження з використанням мікроконтролеру ESP32-CAM, тому для виконання даного

					КвРКІ. 190141.19.01.20 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

завдання було обрано наступний алгоритм реалізації, продемонстрований на даній блок-схемі на рисунку 2.15.

Етапи блок-схеми:

1. Запуск мікросхеми - виконується активація роботи мікросхеми та алгоритму відеоспостереження. Запуск основних функцій та перевірка необхідної для порівняння папки на її наявність. У випадку її відсутності, відбувається її автоматичне створення. Паралельно відбувається перевірка на працездатність камери та перевірка необхідного для подальшої роботи функціоналу – інтерфейсу для перевірки обличчя, його конвертація для порівняння, а також функцій знаходження обличчя та їх порівняння з доданими до папки.

2. Виконання відеоспостереження - До моменту натискання клавіші "Q", виконується цикл, протягом якого відбувається відеоспостереження. Відеоспостереження спеціально виконується у сірих відтінках для покращення розпізнавання обличчя у кадрі, а також для кращого розпізнавання незалежно від рівня освітлення фрагментів освітлення чи інших чинників, що можуть негативно вплинути на роботу алгоритму чи створити труднощі у розпізнаванні.

3. Чи є обличчя в кадрі? - Протягом відеоспостереження виконується перевірка на наявність обличчя в кадрі. Поки його немає, спостереження продовжується. Як тільки, обличчя з'являється в кадрі, алгоритм встановлює рамку навколо нього і таким чином відділяє решту кадру від обличчя для виконання наступного етапу. При цьому, паралельно, камера продовжує відеоспостереження.

4. Чи є помічене обличчя у папці? - Виконується перевірка на наявність поміченого обличчя серед відомих алгоритму. Якщо обличчя наявне у папці, система перестає фіксувати дане обличчя та продовжує відеоспостереження. Якщо ж людина у ній відсутня, у системі залишається запис щодо його лиця, після чого відеоспостереження продовжується далі.

5. Вимкнення мікросхеми - Після натискання клавіші Q, цикл відеоспостереження обривається, камера припиняє свою роботу, як і решта

алгоритмів, а для того, щоб камера і надалі оперативно виконувала свій функціонал, збережені дані видаляються.

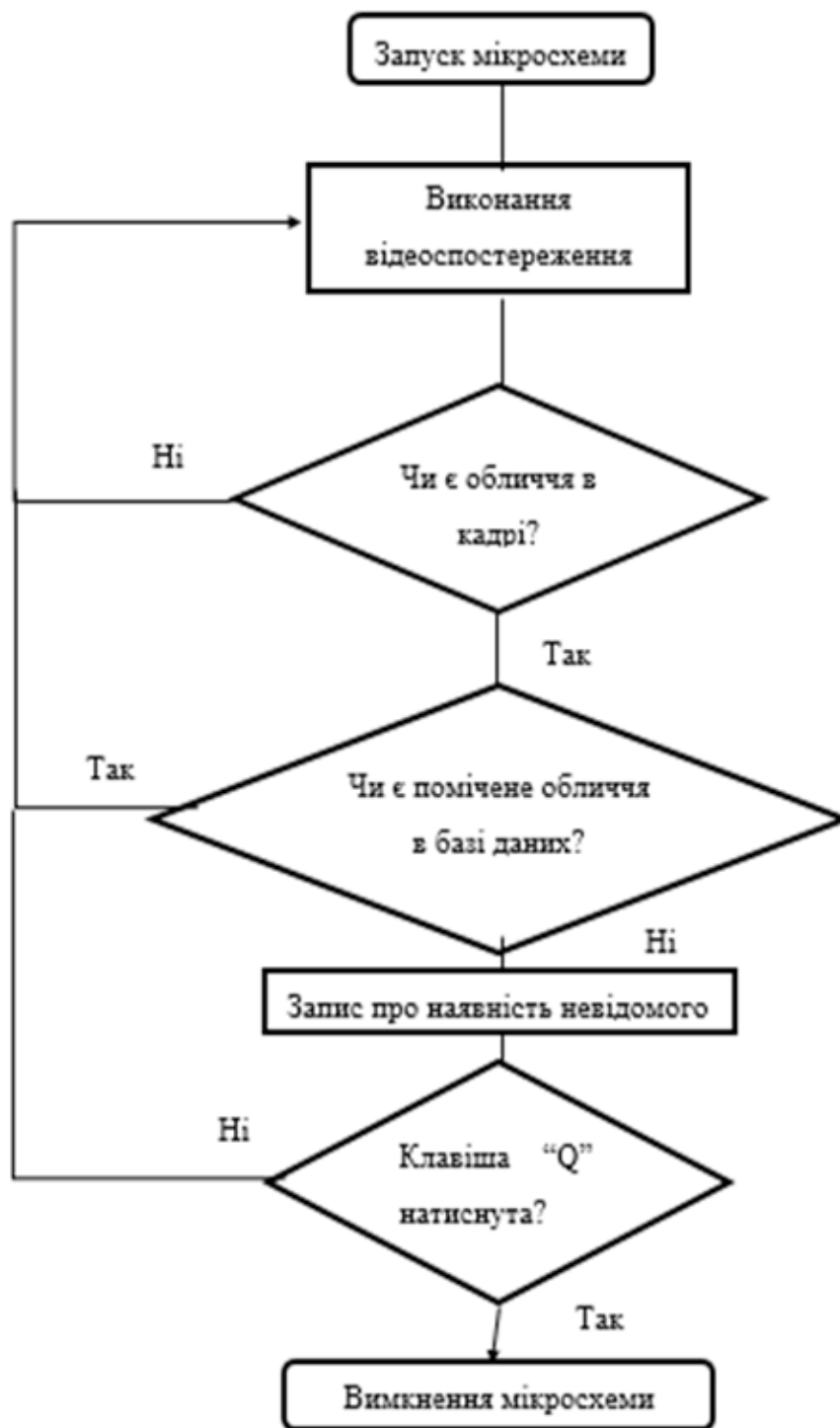


Рисунок 2.15 – Загальна блок-схема алгоритму перевірки та сповіщення

2.9 Висновок

Під час роботи над другим розділом даної дипломної роботи було проаналізовано технічні особливості мікроконтролера ESP32-CAM та його основні складові.

Відповідно до можливостей необхідного для завдання мікроконтролеру, було проаналізовано дві доступні у використанні моделі камер, а саме OV7670 та OV2640 та обрано серед них варіант, який найкраще підходить для виконання поставленої задачі – OV2640, а також з можливістю до покращення у майбутньому створеного алгоритму для збільшення сфер та можливостей подальшого використання у різних сферах, особливо у технології “Розумний будинок”.

При опрацюванні даного розділу, було визначено технічні особливості, переваги та недоліки мікроконтролеру ESP32-CAM та порівняно їх з відповідними складовими інших розповсюджених моделей мікроконтролерів. Окрім цього, було проаналізовано відповідні особливості мікроконтролеру, як складової системи відеоспостереження з метою більш точного визначення можливостей.

З метою визначення основних конкурентів мікроконтролеру ESP32-CAM на ринку систем відеоспостереження, було охарактеризовано найбільш розповсюджені моделі мікроконтролерів, серед яких основними є ESP-EYE, Arducam та серія мікроконтролерів Orange Pi.

Під час аналізу існуючих систем відеоспостереження, було знайдено додаткові технічні можливості та проаналізовано їх можливе додавання до алгоритму та, відповідно до необхідних умов, було додано або розглянуто деякі з них, а саме систему повідомлень з системи MotionEyeOS та обробку відеозображення за допомогою бібліотек OpenCV. Додатково, було визначено можливості до удосконалення майбутньої системи за допомогою функціоналу, який надають системи Home Assistant, для вмонтування системи в повноцінну систему “Розумного будинку” та TinyCam Monitor з ціллю створення більш зручного у використанні мобільного додатку.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

Відповідно до опрацьованої інформації, було проаналізовано та створено базовий алгоритм виконання алгоритму у вигляді блок-схеми для кращого розуміння принципу роботи програмно-технічного засобу перед його майбутньою розробкою програмно-апаратної реалізації, а також визначено основні функції, які потрібні для реалізації.

Основним завданням на наступний розділ є створення програмно-апаратної складової для реалізації алгоритму. Окрім цього, необхідним є визначення основних складових створеної системи для її майбутнього використання.

Додатковою задачею є більш детальний аналіз можливих способів майбутнього вдосконалення створеної системи.

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ТА ВИСНОВКИ

3.1 Реалізація запланованих функцій для виконання алгоритму

Для виконання заданого завдання, а саме повідомляючої системи відеоспостереження з використанням мікроконтролера ESP32-CAM, було створено наступні функції:

1. `sendEmail` – Дана функція реалізовує повідомляючий функціонал алгоритму, який є необхідним відповідно до завдання роботи. Якщо при відеоспостереженні було помічене обличчя, якого немає у папці для порівняння, дана функція конвертує дане обличчя у формат Base64, після чого надсилає повідомлення на необхідну електронну пошту, яка внесена в алгоритм. Відправка відбувається за допомогою бібліотеки CURL. Крім самого надсилання, функція реалізовує перевірку на відсутність помилок у повідомленні перед його надсиланням.

2. `compareImages` – Дана функція отримує зображення знайденого під час відеоспостереження невідомого обличчя та встановлює зв'язок з папкою `CheckBase`. Наступним етапом, вона встановлює алгоритм порівняння зображень на основі `Naar Cascade`. Встановивши рамки порівняння та умови встановлення неспівпадіння, виконується порівняння знайденого обличчя з зображеннями, занесеними у папку.

3. `Main` – Дана функція є основною у програмі. У циклі `while` вона отримує кадри від алгоритму відеоспостереження і у разі появи невідомого обличчя, викликає функцію `compareImages`. Якщо обличчя не знайдене у папці. Алгоритм викликає функцію `sendEmail` та виконується відправка електронної пошти з кадром обличчя. Програма також відображає кадр з прямокутниками навколо обличчя на екрані. Вимкнення програми відбувається при натисканні клавіші 'q', при її натисканні вимикається і мікроконтролер.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім виконання основного алгоритму, необхідно було також створити алгоритм для виконання необхідних для реалізації поставленого завдання функцій самим мікроконтролером.

Для реалізації мікроконтролером необхідно функціоналу, було реалізовано наступні функції:

1. `setup`: Ця функція викликається один раз при запуску програми і виконує початкову налаштування. В ній відбувається ініціалізація дисплея, завантаження класифікатора для розпізнавання обличчя та моделі `Eigenfaces`. Також ініціалізується камера `ESP32-CAM`.

2. `loop`: Ця функція викликається безперервно після `setup()`. В основному циклі функції відбувається отримання кадру з камери, його обробка та передача на розпізнавання облич. Кадр зчитується з камери, конвертується в зображення `OpenCV` і шукаються обличчя за допомогою класифікатора. Якщо знайдено хоча б одне обличчя, воно обробляється алгоритмом `Eigenfaces`, результат відображається на дисплеї, а знайдене обличчя передається на подальшу обробку.

3. `detectFaces`: Ця функція приймає зображення кадру, класифікатор обличчя та порожній вектор для збереження знайдених облич. Функція перетворює зображення у відтінки сірого, вирівнює його гістограму і застосовує класифікатор для пошуку облич. Знайдені обличчя зберігаються у векторі `faces`.

4. `processFace`: Ця функція приймає зображення обличчя як аргумент і виконує його подальшу обробку або передачу в інший алгоритм. У даному коді вона просто передає дані обличчя через `Serial`.

3.2 Виконання прошивки мікроконтролеру

Перед створенням та завантаженням програмної складової мікроконтролеру є необхідним виконати прошивку даної моделі для реалізації взаємодії з модулем камери `OV2640`.

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

Прошивка ESP32-CAM виконується за допомогою UART, тому було вирішено використовувати для зручності USB-UART на базі PL2303. Даний перехідник відповідає нормам даного мікроконтролера – 3.3 В.

Підключення виконується наступним чином:

1. GPIO1 – U0TXD – підключається до Rx UART;
2. GPIO3 – U0RXD – підключається до Tx UART.

Також необхідно заземлити GPIO0. По закінченню, підключення повинно виглядати ідентично до рисунку 3.1.

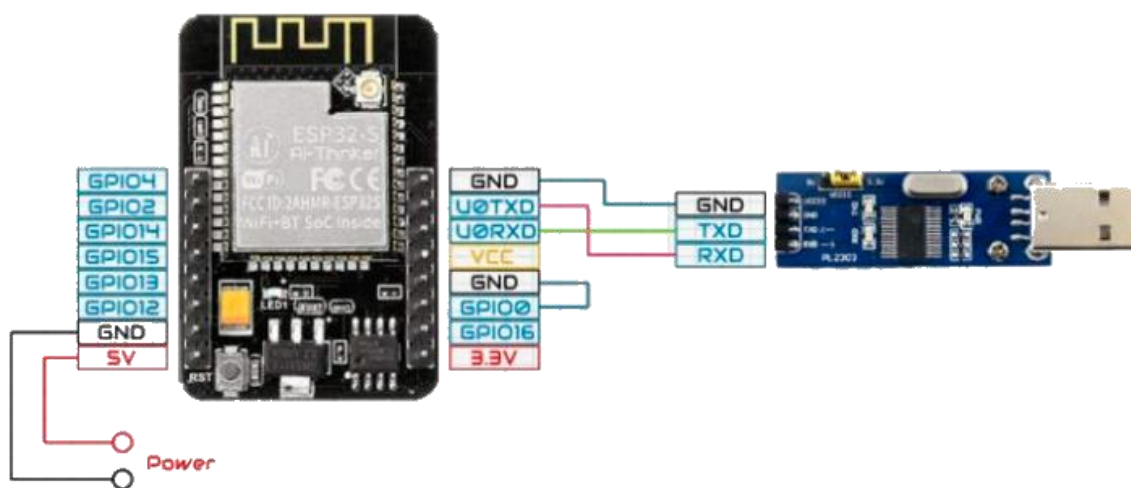


Рисунок 3.1 – Схема з'єднань для виконання прошивки

При виконанні підключення живлення за допомогою 3.3 В, виникають технічні труднощі – прошивка виконується через раз або модуль взагалі не працював. Тому було вирішено використовувати живлення в 5 В, таким чином дані труднощі зникають. Дані зміни живлення також зображені на рисунку 3.1. При цьому, живлення логічних рівнів на входах залишається на рівні 3.3 В, щоб не виникло поломок.

Прошивка була виконана в наступній послідовності:

1. Запуск Arduino IDE. В налаштуваннях за маршрутом File – Preferences записано у вікно Additional Boards Manager URIs наступне повідомлення: https://dl.espressif.com/dl/package_esp32_index.json.”.

2. В меню Tools – Board – Boards Manager введено в пошуку ESP32 та, отримавши там один варіант, він був завантажений, як на рисунку 3.2. Коли завантаження було завершено, в меню Tools – Board – ESP32 Arduino було обрано необхідну для даної роботи схему.

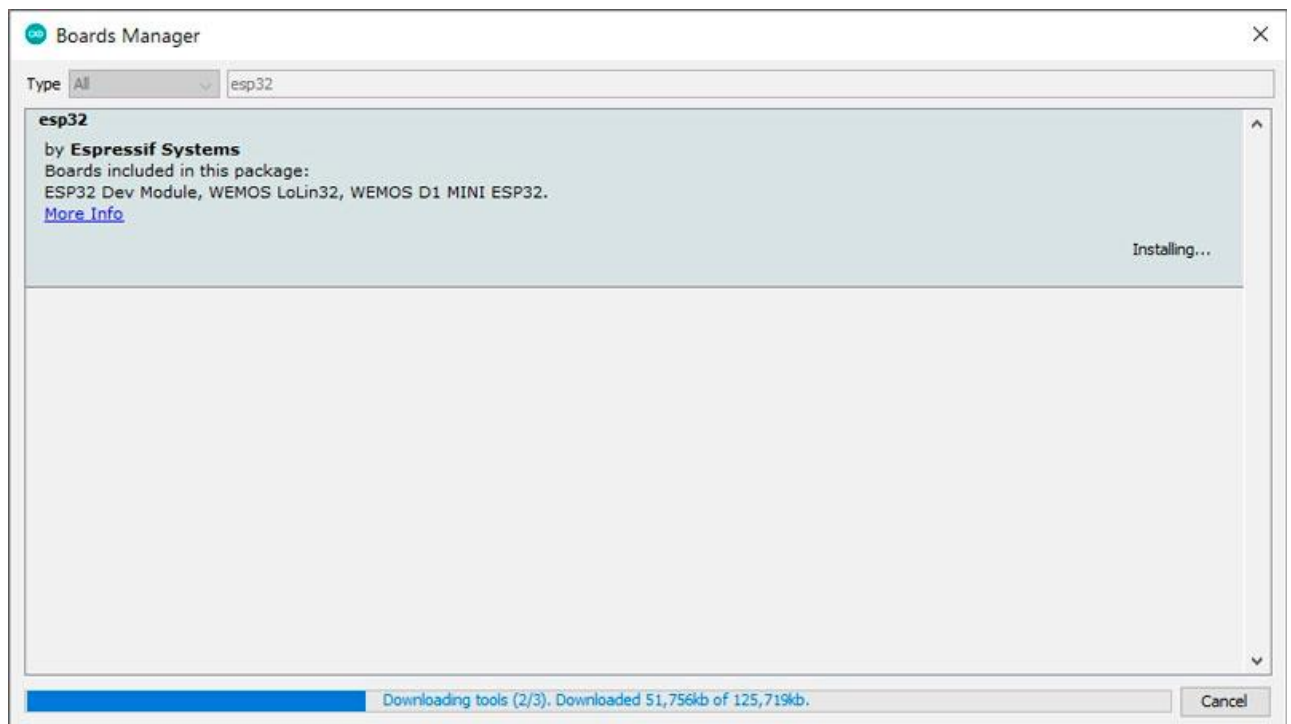


Рисунок 3.2 – Виконання завантаження

3. За маршрутом File – Examples – ESP32 – Camera – CameraWebServer було відкрито приклад роботи з камерою.

4. У отриманому коді, зайві моделі камер було зазначено як прокоментовані, таким чином серед залишилась лише необхідна модель.

```
// Select camera model  
// #define CAMERA_MODEL_WROVER_KIT  
// #define CAMERA_MODEL_ESP_EYE  
// #define CAMERA_MODEL_M5STACK_PSRAM
```

```
//#define CAMERA_MODEL_M5STACK_WIDE
```

```
#define CAMERA_MODEL_AI_THINKER;
```

5. В код прикладу внесено назву мережі Wi-Fi та пароль до неї:

```
const char* ssid = "home";
```

```
const char* password = "pass1234";
```

6. У меню Tools вносено актуальні параметри плати, тобто лише номер COM-порту, адже модель плати вже була обрана на попередніх етапах. Результат налаштувань зображено на рисунку 3.3.

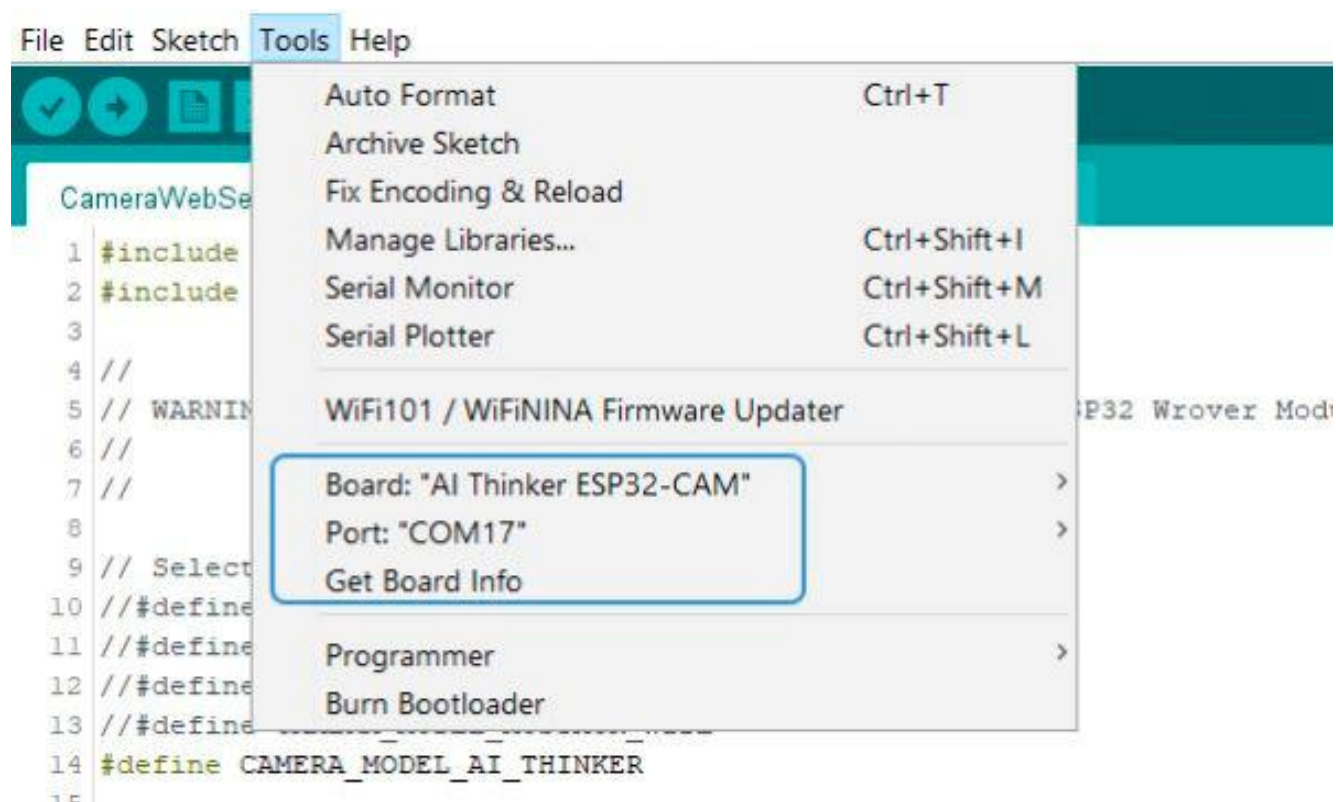


Рисунок 3.3 – Зображення налаштувань з'єднання

7. Натиснувши клавішу Upload відбувся запуск прошивки.

8. Після завершення прошивки, підтяжка GPIO0 була вимкнена та запущено Serial Monitor за маршрутом Tools – Serial Monitor.

Після виконання прошивки, плата була перезапущена за допомогою клавіші Reset і на екрані з'явилися строки, що сигналізують про успішне підключення до мережі Wi-Fi:

“WiFi connected

Starting web server on port: '80'

Starting stream server on port: '81'

Camera Ready! Use 'http://192.168.0.104' to connect”

Перейшовши по зазначеному IP за допомогою браузеру тепер можливо наглядно побачити результат роботи камери. Натиснувши у меню на клавішу Start Stream, на екран буде виводитись зображення з камери OV2640. Таким чином, прошивка була завершена та можна переходити до розробки програмно-апаратної реалізації для систем спостереження та сповіщення користувача.

3.3 Поетапна фрагментація програмно-апаратної реалізації

Етапи виконання функціоналу мікроконтролеру

1. Ініціалізація бібліотек для роботи з TFT дисплеєм Adafruit ST7735 та камерою ArduCAM OV2640:

```
ArduCAM myCAM(OV2640, CS);
```

```
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
```

2. Виконання функції setup(), у функціонал якої входить:

1) ініціалізація Serial комунікації з швидкістю 9600 бод:

```
Serial.begin(9600);
```

2) ініціалізація TFT дисплею:

```
tft.initR(INITR_BLACKTAB);
```

```
tft.fillScreen(ST7735_BLACK);
```

3) завантаження класифікатора для розпізнавання обличчя (haarcascade_frontalface_default.xml):

```
if (!faceCascade.load("haarcascade_frontalface_default.xml")) {
```

```
Serial.println("Не вдалося завантажити класифікатор обличчя.");
```

```
while (1);
```

```
}
```

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

4) завантаження моделі Eigenfaces для розпізнавання обличчя (eigenfaces_model.dat):

```
eigenfaces.loadModel("eigenfaces_model.dat");
```

5) ініціалізація камери ArduCAM OV2640:

```
myCAM.initCAM();
```

```
myCAM.clear_fifo_flag();
```

```
myCAM.write_reg(ARDUCHIP_FRAMES, 0x00);
```

```
delay(100);
```

```
myCAM.write_reg(ARDUCHIP_FRAMES, 0x01);
```

```
delay(100);
```

```
}
```

Основною задачею даної функції є попередня підготовка та запуск модулів, та ініціалізація необхідних моделей та класифікаторів, що будуть використовуватись протягом періоду роботи мікроконтролеру.

3. Виконання функції loop(), у функціонал якої входить:

1) отримання кадру з камери ArduCAM OV2640:

```
myCAM.flush_fifo();
```

```
myCAM.clear_fifo_flag();
```

```
myCAM.start_capture();
```

```
while (!myCAM.get_bit(ARDUCHIP_TRIG, CAP_DONE_MASK)) {}
```

2) зчитування даних з камери та передача їх через Serial комунікацію:

```
-myCAM.CS_LOW();
```

```
myCAM.set_fifo_burst();
```

```
for (int i = 0; i < length; i++) {
```

```
    Serial.write(myCAM.read_fifo());
```

```
}
```

```
myCAM.CS_HIGH();
```

3) застосування класифікатора (haarcascade) для пошуку обличчя на кадрі:

```

void detectFaces(Mat frame, CascadeClassifier faceCascade, vector<Rect>&
faces) {
    Mat grayFrame;
    cvtColor(frame, grayFrame, COLOR_BGR2GRAY);
    equalizeHist(grayFrame, grayFrame);
    faceCascade.detectMultiScale(grayFrame, faces, 1.1, 2, 0 |
CASCADE_SCALE_IMAGE, Size(30, 30));
}

```

4) обробка та розпізнавання знайдених облич, та відображення результату розпізнавання на TFT дисплеї:

```

// Перевірка облич та відображення повідомлення на дисплеї
if (faces.size() > 0) {
    // Обробка знайдених облич
    for (const auto& face : faces) {
        // Витягнення обличчя з кадру
        Mat detectedFace = grayFrame(face);
        // Застосування алгоритму Eigenfaces для розпізнавання обличчя
        int predictedLabel = eigenfaces.predict(detectedFace);
        // Відображення результату розпізнавання на дисплеї
        tft.fillScreen(ST7735_BLACK);
        tft.setCursor(0, 0);
        tft.setTextColor(ST7735_WHITE);
        tft.setTextSize(2);
        tft.print("ID: ");
        tft.println(predictedLabel);
    }
}
else {
    tft.fillScreen(ST7735_BLACK);
}

```

```
}
```

5) введення функції `detectFaces` для виконання розпізнавання обличчя:

```
void detectFaces(cv::Mat frame, CascadeClassifier faceCascade,
std::vector<cv::Rect>& faces) {
    cv::Mat grayFrame;
    cvtColor(frame, grayFrame, COLOR_BGR2GRAY);
    equalizeHist(grayFrame, grayFrame);
    // Застосування класифікатора для пошуку облич
    faceCascade.detectMultiScale(grayFrame, faces, 1.1, 2, 0 |
CASCADE_SCALE_IMAGE, Size(30, 30));
}
```

б) введення функції `processFace` для передачі знайденого обличчя до алгоритму його порівняння:

```
void processFace(cv::Mat face) {
    // Відправка обличчя через Serial
    byte* data = face.data;
    int dataSize = face.total() * face.elemSize();
    for (int i = 0; i < dataSize; i++) {
        Serial.write(data[i]);
    }
}
```

Функція `loop()` є основним циклом програми, за допомогою якого і виконується відеоспостереження та сканування обличчя, а також, при прямому доступі спостерігача до дисплею схеми, виведення результату на екран.

Етапи виконання функціоналу перевіряючої та повідомляючої системи

1. Включення необхідних бібліотек та використання просторів імен:

- 1) `iostream` - для введення/виведення даних;
- 2) `opencv2/opencv.hpp` - для роботи з обробкою зображень та відео;
- 3) `curl/curl.h` - для взаємодії з API електронної пошти;

- 4) filesystem - для роботи з файловою системою;
- 5) vector - для використання векторів;
- 6) algorithm - для використання алгоритмів STL;
- 7) openssl/bio.h - для роботи з кодуванням base64.

2. Оголошення функції sendEmail, яка відправляє електронну пошту з вкладеним зображенням:

- 1) конвертація зображення у формат base64:

```
// Конвертація зображення у формат base64
vector<uchar> buffer;
imencode(".jpg", frame, buffer);
string imageData = base64_encode(buffer.data(), buffer.size());
```

2) налаштування параметрів електронної пошти, таких як адреса відправника, отримувача, тема та повідомлення:

```
string from = "your_email@example.com";
string to = "recipient@example.com";
string subject = "Unknown Person Detected";
string message = "Unknown face has been detected.";
```

3) формування параметрів запиту POST для відправки пошти через API Mailgun:

```
string url = "https://api.mailgun.net/v3/your_domain.com/messages";
string data = "from=" + from + "&to=" + to + "&subject=" + subject + "&text=" +
message + "&attachment=" + imageData;
```

4) налаштування CURL для виконання запиту POST і перевірка наявності помилок під час виконання запиту:

```
// Налаштування CURL
CURL *curl;
CURLcode res;
curl = curl_easy_init();
if (curl) {
```

```

// Встановлення URL для запиту POST
curl_easy_setopt(curl, CURLOPT_URL, url.c_str());
// Встановлення параметрів запиту POST
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, data.c_str());
// Виконання запиту POST
res = curl_easy_perform(curl);
// Перевірка наявності помилки під час виконання запиту
if (res != CURLE_OK) {
    fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
}
// Звільнення ресурсів CURL
curl_easy_cleanup(curl);
}
}

```

3. Оголошення функції compareImages, яка порівнює два зображення:

1) завантаження класифікатора облич Haar Cascade:

```

CascadeClassifier faceCascade;
if (!faceCascade.load("haarcascade_frontalface_default.xml")) {
    cout << "Помилка завантаження класифікатора облич." << endl;
return false;
}

```

2) зміна кольорової схеми зображень на сіро-білу:

```

Mat grayImage1, grayImage2;
cvtColor(image1, grayImage1, COLOR_BGR2GRAY);
cvtColor(image2, grayImage2, COLOR_BGR2GRAY);

```

3) виявлення облич на зображеннях за допомогою класифікатора Haar Cascade:

```

vector<Rect> faces1, faces2;
faceCascade.detectMultiScale(grayImage1, faces1, 1.3, 5);

```

```
faceCascade.detectMultiScale(grayImage2, faces2, 1.3, 5);
```

4) перевірка наявності облич на зображеннях:

```
if (faces1.size() == 0 || faces2.size() == 0) {
```

```
    return false; // Якщо хоча б одне зображення не має облич, вони не  
співпадають
```

5) відображення прямокутників навколо облич на зображеннях:

```
rectangle(image1, faces1[0], Scalar(255, 0, 0), 2);
```

```
rectangle(image2, faces2[0], Scalar(255, 0, 0), 2);
```

6) розрахунок різниці між координатами облич:

```
int dx = faces1[0].x - faces2[0].x;
```

```
int dy = faces1[0].y - faces2[0].y;
```

```
int dw = faces1[0].width - faces2[0].width;
```

```
int dh = faces1[0].height - faces2[0].height;
```

7) встановлення порогового значення для визначення співпадіння облич:

```
int threshold = 10;
```

8) перевірка на співпадіння облич і повернення відповідного результату.

```
if (abs(dx) <= threshold && abs(dy) <= threshold && abs(dw) <= threshold &&  
abs(dh) <= threshold) {
```

```
    return true; // Зображення співпадають
```

```
    }else{
```

```
    return false; // Зображення не співпадають
```

```
    }
```

4. Оголошення функції `getCheckBasePath`, яка повертає шлях до папки `CheckBase`:

1) визначення шляху до папки `CheckBase`:

```
std::string checkBasePath = "CheckBase";
```

2) перевірка існування папки `CheckBase`:

```
if (!fs::exists(checkBasePath) || !fs::is_directory(checkBasePath))
```

```
{
```

3) створення папки CheckBase, якщо вона не існує.

```
fs::create_directory(checkBasePath);  
}
```

5. Оголошення головної функції main:

1) отримання шляху до папки CheckBase за допомогою функції getCheckBasePath.

2) зчитування зображень з папки CheckBase:

```
std::string checkBasePath = getCheckBasePath();  
std::vector<std::string> imagePaths;  
for (const auto& entry : fs::directory_iterator(checkBasePath)) {  
    std::string imagePath = entry.path().string();  
    imagePaths.push_back(imagePath);  
}
```

3) отримання зображення з алгоритму Video.ino та збереження його у змінну videoImage:

```
Mat videoImage;
```

4) перевірка співпадіння отриманого зображення з обличчям та зображень з папки CheckBase за допомогою функції compareImages:

```
bool matchFound = false;  
for (const auto& imagePath : imagePaths) {  
    Mat baseImage = imread(imagePath);  
    if (!baseImage.empty()) {  
        if (compareImages(videoImage, baseImage)) {  
            matchFound = true;  
            break;  
        }  
    }  
}
```

5) відправка електронної пошти з отриманим зображенням у разі відсутності співпадінь за допомогою функції sendEmail:

```
if (!matchFound) {
    sendEmail(videoImage);
    cout << "Електронна пошта надіслана." << endl;
}
// Відправка електронної пошти
sendEmail(frame);
cout << "Електронна пошта надіслана." << endl
// Збереження обличчя у файл
string filename = "unknown_face.jpg";
string filePath = fs::path(checkBasePath) / filename;
imwrite(filePath, frame);
```

б) Завершення програми:

```
// Вимикання програми при натисканні клавіші 'q'
if (waitKey(1) == 'q') {
    break;
}
```

Відповідно до нової, більш точної поетапної системи принципу роботи алгоритму, блок-схема, що була зазначена на рисунку 2.15, тепер зображає собою виконання двох паралельних процесів. Ліва частина блок-схеми відповідає за виконання алгоритму, пов'язаного з функціоналом мікроконтролеру та виконанням відеоспостереження. Права частина блок-схеми відповідає саме за алгоритм порівняння обличчя з зображенням та за функцію сповіщення користувача на вказану електронну пошту. Доповнена блок-схема паралельного виконання алгоритмів зображена на рисунку 3.4.

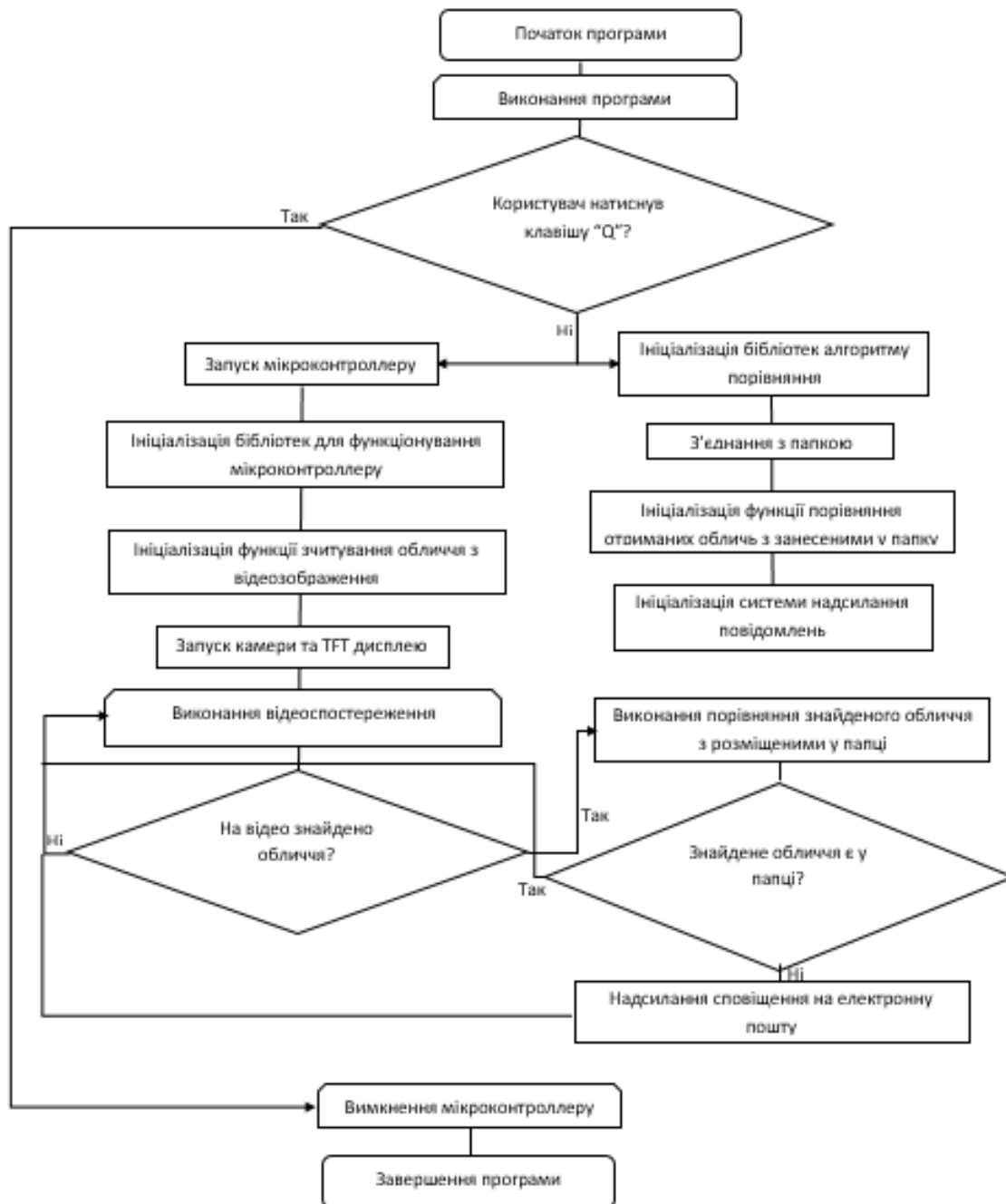


Рисунок 3.4 – Поетапна блок-схема виконання спостереження та сповіщення користувача

3.4 Складові, необхідні для тестування алгоритму та апаратної складової

Для тестування функціоналу цих двох алгоритмів знадобиться наступне обладнання та програмне забезпечення:

Обладнання:

1. Мікроконтролер ESP32-CAM.

2. Камера OV2640.
3. Комп'ютер з встановленою Arduino IDE та необхідними бібліотеками.
4. Відеопотік або зображення з обличчями для тестування алгоритму обробки.

Програмне забезпечення:

1. Arduino IDE для завантаження коду на мікроконтролер ESP32 CAM;
2. Бібліотеки Arduino: Adafruit_GFX, Adafruit_PCD8544, Wire, Adafruit_MLX90614, Servo, SoftwareSerial, EEPROM, Adafruit_ST7735 та ArduCAM.
3. Бібліотеки OpenCV для обробки зображень та розпізнавання обличчя Haar Cascade.
4. Бібліотека CURL для відправки електронної пошти.

3.5 Теоретичні можливості для покращення алгоритму

Опрацьовуючи теоретичний матеріал на тему даної роботи, було знайдено декілька можливостей покращити функціонал даного алгоритму у рамках даного завдання та частково за його межами. Серед основних серед них слід зазначити наступні:

1. З метою забезпечення більш надійної безпеки для жителів та покриття більшої зони для її забезпечення, дана система спостереження може бути оснащена можливістю використовувати інші датчики. Мікроконтролер ESP32-CAM здатний з ним працювати і таким чином систему відеоспостереження можна удосконалити та перетворити на систему автономної безпеки будинку, яка у випадку появи будь-якої небезпеки сповістить жителів будинку або, навіть, необхідні для знищення небезпеки інстанції. Особливо серед ключових можливих датчиків слід зазначити датчик руху та диму.

Датчик диму, підключений до керування мікроконтролером ESP32-CAM здатен забезпечувати безпеку у будинку навіть якщо його жителі знаходяться

всередині нього, але на ділянці, на якій неможливо дану небезпеку відслідкувати. Таким чином, дана система зможе захищати будинок не тільки від зловмисників чи сторонніх, а й від руйнувань, пов'язаних з технічними несправностями або пожежами, що виникли за людським фактором.

Давач руху, в свою чергу, може добре співпрацювати з камерою навіть у випадку відсутності небезпеки. Залежно від розміщення давачу, він може відслідкувати і отримання людиною пошти, і відкриття тих чи інших дверей або вікон, або більш динамічно відслідкувати що відбувається у зоні спостереження камери та поза нею. Однак, використання даного давачу може призвести і до збільшення обсягу некоректного спрацювання даної системи, особливо за наявності у домі тварин.

Окрім зупинення критичних випадків небезпеки, доступ до інших давачів може забезпечити користувачеві й інформаційну користь. Теоретично, за допомогою давачів, підключених до ESP32-CAM можна аналізувати інформацію щодо стану повітря у кімнаті та мати можливість впливати на нього при взаємодії з системою провітрювання або кондиціонування приміщення.

Наостанок, за допомогою подібних датчиків можна реалізувати і автоматизацію певних процесів, які є для користувача щоденною рутинною, таких як полив рослин.

Головне, при пошуку можливостей як ще можна використати мікроконтроллер у місці його використання, враховувати можливості самого мікроконтроллера до виконання стількох дій та одночасне опрацювання настільки великої кількості інформації.

2. Спроектований у даній роботі алгоритм може використовуватись з метою забезпечення безпеки у житлових будинках. Сповіщення працюють у момент, якщо у зоні спостереження алгоритму було помічене невідоме обличчя і забезпечують таким чином безпеку від можливих зловмисників.

З використанням бази даних, дана система частково видозмінює свої основні можливості та доповнює їх. В даному випадку, функціонал може бути доповнений

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

розпізнаванням хто саме прибув до будинку, що буде більш корисним не лише з метою захисту власного будинку, а й у великих підприємств, щоб відслідковувати коли і яка людина прибула на місце роботи.

Окрім використання на місцях роботи, подібні системи можуть використовуватись з метою пошуку злочинців при наявності відомого їх обличчя. Дані системи відеоспостереження встановлюються в людних та інших місцях, а до бази даних занесено обличчя, яке на даний момент знаходиться у розшуку. І у разі необхідності, даний алгоритм можна переробити таким чином, щоб сповіщення спрацьовувало у разі саме знаходження обличчя у базі і після чого, сповіщення надсилалось би до правоохоронних органів та інших подібних інстанцій.

Але основною проблемою створення даної системи буде виникнення спаму сповіщеннями, за причиною якого сповіщення про невідому людину, яка увійшла у приміщення, буде досить складно знайти серед десятків та сотень сповіщень про працівників, що прибули або, якщо необхідно, покинули місце роботи на даний момент.

3. Розглянувши системи, що можуть співпрацювати з мікроконтролером ESP32-CAM, була розглянута система під назвою TinyCam Monitor, у основний перелік можливостей якої входить реалізація керування відеоспостереженням за допомогою мобільного додатку.

Реалізувавши подібну систему для даного завдання, є можливим замінити систему сповіщення користувача на електронну пошту на ідентичне сповіщення, але вже на його телефон за допомогою спеціального мобільного додатку.

Таким чином, сповіщення будуть швидше досягати користувача, адже відкрити панель сповіщень на телефоні або додаток для даної системи буде значно швидше, ніж отримати доступ власної електронної пошти. Крім того, таким чином сповіщення буде відбуватись з меншою кількістю проміжних етапів і завдяки цьому, повідомлення швидше досягне користувача.

4. Після отримання сповіщення про невідоме обличчя на вході у власний дім, найбільш ймовірно, що користувач буде відчувати паніку щодо того, що не

знає, що саме вкоїть можливий зловмисник, увійшов він у будинок чи ні, чи це був поштар, або взагалі людина, яка проходила повз.

За допомогою можливості до передачі відеозображення у режимі реального часу, людина зможе особисто впевнитись в тому, що відбувається у зоні спостереження, чи є причини для паніки та сповіщення необхідних інстанцій.

Реалізація даної системи може відбуватись двома способами. Першим способом буде реалізація подібного функціоналу на монітор у будинку. Таким чином, людина зможе забезпечити власну безпеку, якщо вона застала її всередині будинку, або, на крайній випадок, використати дану систему замість домофону.

Іншим варіантом реалізації може бути поєднання даної системи з раніше згаданою можливістю створення мобільного додатку. Таким чином, людина може у разі необхідності переконатись, що на даний момент відбувається на вході у будинок або у іншій зоні під спостереженням підключеної до додатку системи, що зможе забезпечити їй максимальний спокій відносно безпеки та цілісності її майна або надати кращу можливість переконатись у наявності ризику його виникнення.

3.6 Висновок

Протягом виконання програмно-апаратної реалізації та тестування програмно-технічного засобу, була реалізована більш детальна блок-схема реалізації необхідного для виконання поставленого завдання алгоритму.

Наступним чином, була реалізована програмна складова алгоритму окремо для виконання відеоспостереження та пошуку невідомих обличь, окремо для порівняння знайдених першим алгоритмом обличь із зображеннями, занесеними до папки. Для повної функціональності системи, був реалізований зв'язок між двома файлами алгоритмів та папкою для зображень.

Наостанок, були розглянуті теоретичні можливості для вдосконалення даної системи для реалізації можливих встановлених цілей:

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Розширення спектру відслідковування даною системою за допомогою нових давачів з метою забезпечення повної безпеки.

2. Поєднання системи порівняння з базою даних замість папки для оптимізації перевірки та уточнення особистості знайденого обличчя, що відкриває можливості для аудиту відвідування працівників та розшуку злочинців.

3. Створення мобільного додатку з метою реалізації доступу до цілісності території під спостереженням на відстані.

4. Реалізація відслідковування за територію в реальному часі з монітору комп'ютеру або створеного мобільного додатку.

Підсумовуючи, дана система є теоретично функціональною та може бути активно використовуватись у майбутньому з метою різнопланового забезпечення безпеки користувача, не тільки будучи повністю самостійною системою відеоспостереження, а й частково- або повнофункціональною складовою системи “розумний будинок”, виконуючи крім спостереження ще й інші функції.

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

ВИСНОВКИ

У даній дипломній роботі була проведена детальна аналітична робота в галузі систем відеоспостереження з використанням мікроконтролера ESP32-CAM.

Перший Розділ надає загальні відомості про мікроконтролери та їх сфери застосування. Було розглянуто загальні відомості про системи відеоспостереження та їх класифікацію, а також використання мікроконтролерів у таких системах.

Протягом виконання другого розділу, було визначено технічні особливості мікроконтролеру ESP32-CAM та його складові. При їх аналізі була знайдена можливість альтернативного використання камери OV7670, однак було визначено, що початкова модель OV2640 краще відповідає потребам для виконання завдання.

З метою кращого ознайомлення з можливостями використання необхідного мікроконтролеру, у даному розділі було проаналізовано переваги та недоліки ESP32-CAM порівняно з іншими мікроконтролерами, а також з особливостями даного мікроконтролеру, як складової системи відеоспостереження.

У третьому розділі була проведена програмно-апаратна реалізація та тестування програмно-технічного засобу. Були реалізовані функції для алгоритму та визначено необхідні компоненти для тестування системи та її майбутнього використання. Були також детально розглянуті теоретичні можливості для покращення алгоритму.

Загальною метою цієї дипломної роботи було створення повідомляючої системи відеоспостереження з використанням мікроконтролера ESP32-CAM. Робота показала, що ESP32-CAM є потужним та ефективним мікроконтролером для застосування у системах відеоспостереження.

Завдяки виконанню даної дипломної роботи було розроблено функціональний програмно-технічний засіб, який може бути використаний в системах відеоспостереження з мікроконтролером ESP32-CAM. Проектування та реалізація системи відеоспостереження з використанням мікроконтролера ESP32-CAM виявилися складними завданнями, але їх успішне виконання сприяє

					КвРКІ. 190141.19.01.20 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

забезпеченню безпеки та контролю у різних сферах, таких як домашня безпека, комерційні приміщення, громадські місця та багато інших.

Результати досліджень і розробок, проведених у цій дипломній роботі, можуть бути використані для подальшого розширення та вдосконалення систем відеоспостереження з використанням мікроконтролера ESP32-CAM, а також для проведення подальших наукових досліджень у цій області.

					КВРКІ. 190141.19.01.20 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Історія створення та розвитку мікропроцесорних засобів автоматизації.
URL: <https://elprivod.nmu.org.ua/ua/interesting/hmpa/Історія%20створення%20та%20розв%20итку%20МП%20засобів%20авт.pdf> (дата звернення: 01.03.2023).
2. Архітектура мікроконтролерів з CISC архітектурою. URL: http://radio-vtc.inf.ua/mps/lesson/L16_2012.htm (дата звернення: 01.03.2023).
3. Що таке мікропроцесор, мікроконтролер та програмований логічний контролер. URL: https://elprivod.nmu.org.ua/ua/interesting/what_is_mp_mc_plc.php (дата звернення: 01.03.2023).
4. Все о микроконтроллерах. URL: https://elektrovesti.net/interesting/642_vse-o-mikrokontrollerakh. (дата звернення: 01.03.2023).
5. Глухов О.В., Кравчук О.О., Левченко Є.В. Вивчення властивостей мікроконтролерів і електронних систем на базі платформи Ардуіно: навч. посіб. Харків : ХНУРЕ, 2019. 192 с.
6. Конспект лекцій з дисципліни «Мікропроцесорна техніка». URL: <http://eprints.kname.edu.ua/43462/1/92Л%20Конспект%20МПТ%202015.pdf>. (дата звернення: 01.03.2023).
7. Переваги використання мікроконтролерів у портативних медичних пристроях. URL: https://mts.nure.ua/wp-content/uploads/2021/03/aert-2019_web_14-16.pdf. (дата звернення: 01.03.2023).
8. Про системи керування промисловими роботами. URL: <https://ela.kpi.ua/bitstream/123456789/46800/1/Page%20171-174.pdf>. (дата звернення: 01.03.2023).
9. Системи відеоспостереження та їх використання. URL: <https://valtek.com.ua/ua/system-integration/security-control-system/video-surveillance/video-surveillance-employment>. (дата звернення: 01.03.2023).
10. Історія відеоспостереження. URL: <https://zhuk.ua/istoriyi-ta-fakty/istoriia-videosposterezhennia>. (дата звернення: 01.03.2023).

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

11. “З чого складається система відеоспостереження?”. URL: <https://ott.net.ua/ua/z-choho-skladayetsya-systema-videosposterezhennya>. (дата звернення: 01.03.2023).

12. “Що входить до комплекту відеоспостереження?”. URL: www.volynnews.com/news/all/shcho-vkhodyt-do-komplektu-videosposterezhennia. (дата звернення: 01.03.2023).

13. Відеоспостереження: типи систем і застосування. URL: <https://vv24.in.ua/videosposterezhennya-typy-system.php?lang=1>. (дата звернення: 01.03.2023).

14. “Цифрові IP камери відеоспостереження або аналогові відеокамери безпеки?”. URL: <https://smartel.ua/ua/articles/ip-kamery-videonablyudeniya-analogovye-videokamery-bezopasnosti>. (дата звернення: 01.03.2023).

15. “Аналогове чи цифрове відеоспостереження – що вибрати?”. URL: <http://www.golos.com.ua/news/177774>. (дата звернення: 01.03.2023).

16. “Як працює бездротова камера відеоспостереження?”. URL: <https://pravda.if.ua/kak-rabotaet-besprovodnaya-kamera-videonablyudeniya>. (дата звернення: 01.03.2023).

17. Переваги бездротових систем відеоспостереження. URL: <https://odessa-life.od.ua/article/promo/preimushhestva-besprovodnyh-sistem-videonabljudeniya>. (дата звернення: 01.03.2023).

18. Хмарне відеоспостереження. URL: <https://alliancesafety.com.ua/uk/blog-ukr/hmarne-videosposterezhennja>. (дата звернення: 01.03.2023).

19. Ai-Thinker Модуль Wi-Fi ESP32-CAM с камерой 2MP. URL: <https://arduino.ua/ru/prod4218-modyl-wi-fi-esp32-s-kameroi-2mp>. (дата звернення: 01.04.2023).

20. Камера OV7670. URL: <https://iarduino.ru/shop/Expansion-payments/kamera.html>. (дата звернення: 01.04.2023).

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

21. Модуль камери 2Мп OV2640 для ESP32-CAM (FOV160). URL: <https://arduino.ua/ru/prod4501-modyl-kameri-2mp-ov2640-dlya-esp32-cam-fov160>. (дата звернення: 01.04.2023).
22. ESP32-CAM, Camera Module Based On ESP32, OV2640 Camera and ESP32-CAM-MB adapter Included. URL: <https://www.waveshare.com/esp32-cam.htm>. (дата звернення: 01.04.2023).
23. ESP32-CAM Projects. URL: <https://electronicschacks.com/esp32-cam-projects>. (дата звернення: 01.04.2023).
24. Camera Module Based on ESP32. URL: <https://grobotronics.com/camera-module-based-on-esp32.html>. (дата звернення: 01.04.2023).
25. Working with Raspberry Pi Camera Board. URL: <https://www.mathworks.com/help/supportpkg/raspberrypiio/ref/working-with-raspberry-pi-camera-board.html>. (дата звернення: 01.04.2023).
26. Офіційний сайт Arducam. URL: <https://www.arducam.com>. (дата звернення: 01.04.2023).
27. Arducam 12MP 477P Mini High Quality Camera Module for Raspberry Pi and Pi zero. URL: <https://www.arducam.com/product/arducam-12mp-imx477-mini-high-quality-camera-module-for-raspberry-pi>. (дата звернення: 01.04.2023).
28. ESP-EYE плата для розробки ESP32 з підтримкою розпізнавання обличь, голосового пробудження. URL: <https://cnx-software.ru/2019/01/16/esp-eye-plata-dlya-razrobotki-esp32-ai-s-podderzhkoj>. (дата звернення: 01.04.2023).
29. WeMos D1 mini Модуль. URL: <https://uamper.com/WeMos-D1-mini-модуль-быстрой-отладки-esp8266>. (дата звернення: 01.04.2023).
30. Orange Pi 13MP Camera. URL: <https://arduino.ua/ru/prod5631-orange-pi-13mp-camera>. (дата звернення: 01.04.2023).
31. BeagleBone Black Rev C 4GB. URL: <https://arduino.ua/ru/prod413-beaglebone-black-rev-c-4gb>. (дата звернення: 01.04.2023).

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

32. Pixy2 CMUcam5 Smart Vision Sensor. URL: <https://uamper.com/Pixy2-CMUcam5-сенсор-умного-видео-слежения-машинное-зрение>. (дата звернення: 01.04.2023).

33. NVIDIA Jetson Nano. Возможности современного ИИ для миллионов устройств. URL: <https://www.nvidia.com/ru-ru/autonomous-machines/embedded-systems/jetson-nano>. (дата звернення: 01.04.2023).

34. ESP32 Camera Dev Boards Review and Comparison (Best ESP32-CAM). URL: <https://makeradvisor.com/esp32-camera-cam-boards-review-comparison>. (дата звернення: 01.04.2023).

35. Офіційний сайт Home Assistant. URL: <https://www.home-assistant.io>. (дата звернення: 01.04.2023).

36. ESP32 Camera Component. URL: https://esphome.io/components/esp32_camera.html. (дата звернення: 01.04.2023).

37. Офіційний сайт OpenCV. URL: <https://opencv.org>. (дата звернення: 01.04.2023).

38. ESP32 CAM Object Detection & Identification with OpenCV. URL: <https://iotprojectsideas.com/esp32-cam-object-detection-identification-with-opencv>. (дата звернення: 01.04.2023).

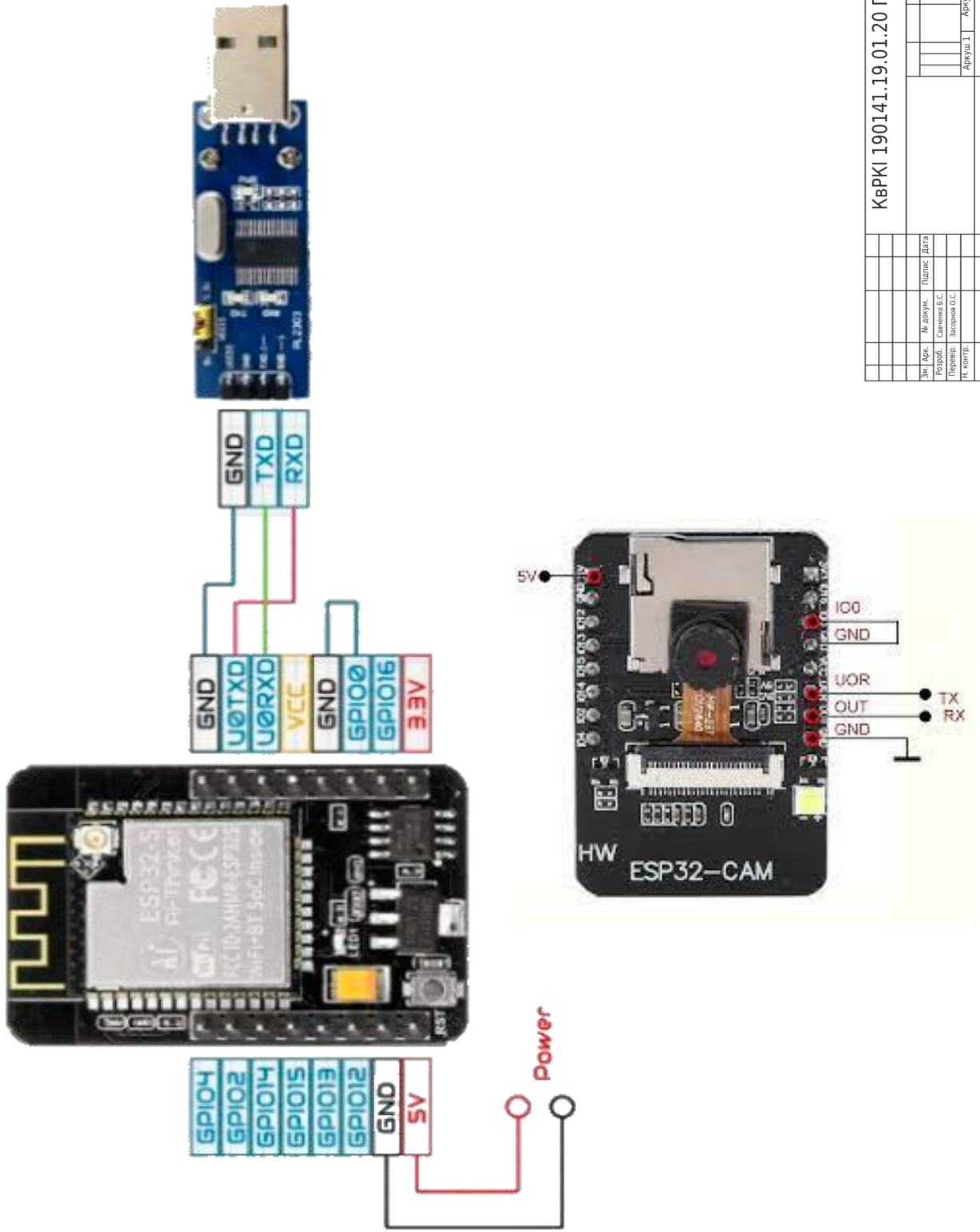
39. MotionEyeOS на GitHub. Приклад використання даної системи для перетворення комп'ютера у систему відеоспостереження. URL: <https://github.com/motioneye-project/motioneyeos>. (дата звернення: 01.04.2023).

40. Додаток TinyCam Monitor для Android на Google Play. URL: <https://play.google.com/store/apps/details?id=com.alexvas.dvr&hl=ru&gl=US&pli=1>. (дата звернення: 01.04.2023).

					КВРКІ. 190141.19.01.20 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК Б

Схеми з'єднань мікроконтролера з камерою та для виконання прошивки мікроконтролера
(Обов'язкова)



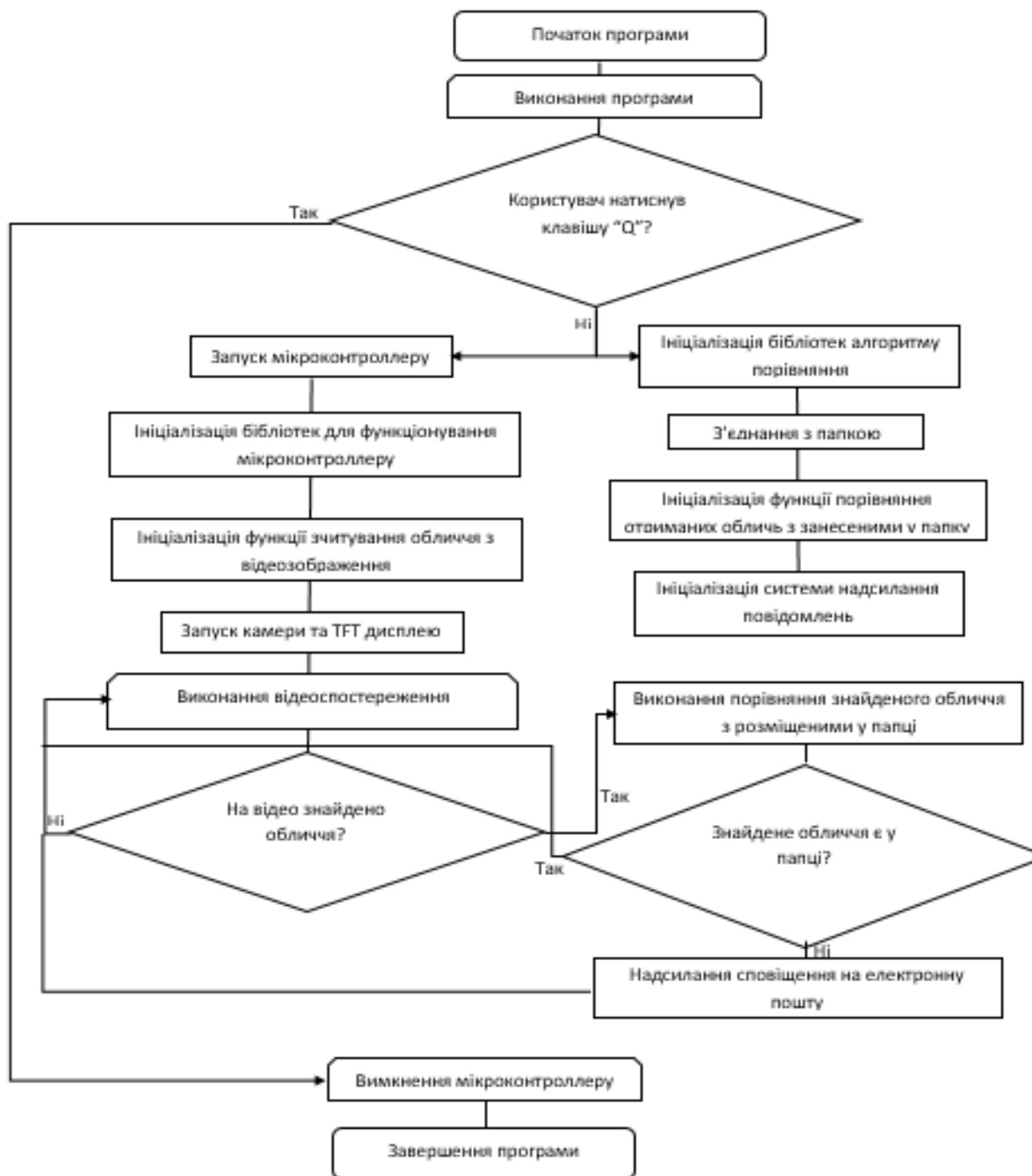
КвРКІ 190141.19.01.20

КвРКІ 190141.19.01.20 ПЗ			
Зк. Акт.	№ докум.	Підпис	Дата
Розроб.	Савченко Є.С.		
Перевір.	Зварова О.С.		
Н. контро.			
Т. розроб.	Лисенко О.А.		
ЗРБ.			
			Кі2-19-1

ДОДАТОК В

Блок-схема поетапної реалізації алгоритму

(Обов'язкова)



ПЗ

Знак	Аркуш 1	Аркуш 2	Аркуш 3
№ докум.	КІ2-19-1		
Підпис			
Дата			
Розроб.	Саченко В.С.	Згорюха О.С.	
Перевір.			
Т. констр.			
Т. констр.	Лисенко С.К.	Голубенко Г.О.	
ЗСТВ.			

КерКІ 1901.41.19.01.20

ДОДАТОК Г

Лістинг коду для виконання відеоспостереження

(обов'язковий)

```
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
#include <Wire.h>
#include <Adafruit_MLX90614.h>
#include <Wire.h>
#include <Adafruit_MLX90614.h>
#include <Servo.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>
#include <ArduCAM.h>

#define CS 10
#define RESET 9
#define MOSI 11
#define MISO 12
#define SCK 13

ArduCAM myCAM(OV2640, CS);

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

CascadeClassifier faceCascade;
Eigenfaces eigenfaces;
```

```

void setup() {
  Serial.begin(9600);

  // Ініціалізація дисплея
  tft.initR(INITR_BLACKTAB);
  tft.fillScreen(ST7735_BLACK);

  // Завантаження класифікатора для розпізнавання обличчя
  if (!faceCascade.load("haarcascade_frontalface_default.xml")) {
    Serial.println("Не вдалося завантажити класифікатор обличчя.");
    while (1);
  }

  // Завантаження моделі Eigenfaces
  eigenfaces.loadModel("eigenfaces_model.dat");

  // Ініціалізація камери
  myCAM.initCAM();
  myCAM.clear_fifo_flag();
  myCAM.write_reg(ARDUCHIP_FRAMES, 0x00);
  delay(100);
  myCAM.write_reg(ARDUCHIP_FRAMES, 0x01);
  delay(100);
}

void detectFaces(cv::Mat frame, CascadeClassifier faceCascade,
std::vector<cv::Rect>& faces) {
  cv::Mat grayFrame;
  cvtColor(frame, grayFrame, COLOR_BGR2GRAY);
  equalizeHist(grayFrame, grayFrame);
}

```

```

// Застосування класифікатора для пошуку облич
faceCascade.detectMultiScale(grayFrame, faces, 1.1, 2, 0 |
CASCADE_SCALE_IMAGE, Size(30, 30));
}

```

```

void processFace(cv::Mat face) {
// Опрацювання обличчя і передача його даних в інший алгоритм
// Відправка обличчя через Serial
byte* data = face.data;
int dataSize = face.total() * face.elemSize();
for (int i = 0; i < dataSize; i++) {
Serial.write(data[i]);
}
}

```

```

void loop() {
// Отримання кадру з камери
myCAM.flush_fifo();
myCAM.clear_fifo_flag();
myCAM.start_capture();

while (!myCAM.get_bit(ARDUCHIP_TRIG, CAP_DONE_MASK)) {}

// Отримання розмірів отриманого кадру
uint32_t length = myCAM.read_fifo_length();

// Зчитування даних з камери
myCAM.CS_LOW();
myCAM.set_fifo_burst();
for (int i = 0; i < length; i++) {

```

```

Serial.write(myCAM.read_fifo());
}
myCAM.CS_HIGH());

// Конвертація отриманого кадру в зображення OpenCV
cv::Mat frame = cv::imdecode(cv::Mat(length, 1, CV_8UC1, myCAM.fb),
cv::IMREAD_COLOR);

// Пошук облич та відображення повідомлення на дисплеї
std::vector<cv::Rect> faces;
detectFaces(frame, faceCascade, faces);

if (faces.size() > 0) {
// Обробка знайдених облич
for (const auto& face : faces) {
// Витягнення обличчя з кадру
cv::Mat detectedFace = frame(face);

// Застосування алгоритму Eigenfaces для розпізнавання обличчя
int predictedLabel = eigenfaces.predict(detectedFace);

// Відображення результату розпізнавання на дисплеї
tft.fillScreen(ST7735_BLACK);
tft.setCursor(0, 0);
tft.setTextColor(ST7735_WHITE);
tft.setTextSize(2);
tft.print("ID: ");
tft.println(predictedLabel);

// Передача знайденого обличчя на опрацювання в інший алгоритм
"CheckModule.cs"

```

```
    processFace(detectedFace);  
  }  
}  
else {  
  tft.fillScreen(ST7735_BLACK);  
}  
  
  delay(1000);  
}
```

ДОДАТОК Д

Лістинг коду для опрацювання та порівняння обличь на зображеннях
(обов'язковий)

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include <curl/curl.h>
#include <filesystem>
#include <vector>
#include <algorithm>
#include <openssl/bio.h>

using namespace std;
using namespace cv;
namespace fs = std::filesystem;

void sendEmail(const Mat& frame) {
    // Конвертація зображення у формат base64
    vector<uchar> buffer;
    imencode(".jpg", frame, buffer);
    string imageData = base64_encode(buffer.data(), buffer.size());

    // Налаштування параметрів електронної пошти
    string from = "your_email@example.com";
    string to = "recipient@example.com";
    string subject = "Unknown Person Detected";
    string message = "Unknown face has been detected.";

    // Формування параметрів запиту POST
    string url = "https://api.mailgun.net/v3/your_domain.com/messages";
```

```
string data = "from=" + from + "&to=" + to + "&subject=" + subject +  
"&text=" + message + "&attachment=" + imageData;
```

```
// Налаштування CURL  
CURL *curl;  
CURLcode res;  
curl = curl_easy_init();  
if (curl) {  
    // Встановлення URL для запиту POST  
    curl_easy_setopt(curl, CURLOPT_URL, url.c_str());  
  
    // Встановлення параметрів запиту POST  
    curl_easy_setopt(curl, CURLOPT_POSTFIELDS, data.c_str());  
  
    // Виконання запиту POST  
    res = curl_easy_perform(curl);  
  
    // Перевірка наявності помилки під час виконання запиту  
    if (res != CURLE_OK) {  
        fprintf(stderr, "curl_easy_perform() failed: %s\n",  
curl_easy_strerror(res));  
    }  
  
    // Звільнення ресурсів CURL  
    curl_easy_cleanup(curl);  
}  
  
// Функція для порівняння двох зображень  
bool compareImages(const Mat& image1, const Mat& image2) {  
    // Завантаження класифікатора облич Haar Cascade
```

```

CascadeClassifier faceCascade;
if (!faceCascade.load("haarcascade_frontalface_default.xml")) {
    cout << "Помилка завантаження класифікатора облич." << endl;
    return false;
}

// Зміна кольорової схеми зображень з BGR на сіро-білу
Mat grayImage1, grayImage2;
cvtColor(image1, grayImage1, COLOR_BGR2GRAY);
cvtColor(image2, grayImage2, COLOR_BGR2GRAY);

// Виявлення облич на зображеннях
vector<Rect> faces1, faces2;
faceCascade.detectMultiScale(grayImage1, faces1, 1.3, 5);
faceCascade.detectMultiScale(grayImage2, faces2, 1.3, 5);

// Перевірка наявності облич на зображеннях
if (faces1.size() == 0 || faces2.size() == 0) {
    return false; // Якщо хоча б одне зображення не має облич, вони не
співпадають
}

// Відображення прямокутників навколо облич на зображеннях (опційно)
rectangle(image1, faces1[0], Scalar(255, 0, 0), 2);
rectangle(image2, faces2[0], Scalar(255, 0, 0), 2);

// Відображення зображень з обличчями (опційно)
imshow("Image 1", image1);
imshow("Image 2", image2);
waitKey(0);

```

```

// Розрахунок різниці між координатами облич
int dx = faces1[0].x - faces2[0].x;
int dy = faces1[0].y - faces2[0].y;
int dw = faces1[0].width - faces2[0].width;
int dh = faces1[0].height - faces2[0].height;

// Встановлення порогового значення для визначення співпадіння облич
int threshold = 10;

// Перевірка на співпадіння облич
if (abs(dx) <= threshold && abs(dy) <= threshold && abs(dw) <= threshold
&& abs(dh) <= threshold) {
    return true; // Зображення співпадають
} else {
    return false; // Зображення не співпадають
}
}

// Функція для отримання шляху до папки CheckBase
std::string getCheckBasePath() {
    // Шлях до папки CheckBase
    std::string checkBasePath = "CheckBase";

    // Перевірка існування папки CheckBase
    if (!fs::exists(checkBasePath) || !fs::is_directory(checkBasePath)) {
        // Створення папки CheckBase, якщо вона не існує
        fs::create_directory(checkBasePath);
    }

    int main() {
        // Зчитування зображень з папки CheckBase

```

```

std::string checkBasePath = getCheckBasePath();
std::vector<std::string> imagePaths;
for (const auto& entry : fs::directory_iterator(checkBasePath)) {
    std::string imagePath = entry.path().string();
    imagePaths.push_back(imagePath);
}
// Зчитування зображення з алгоритму Video.ino
Mat videoImage; // Зображення з алгоритму Video.ino
// Отримання зображення із алгоритму Video.ino і збереження у змінну
videoImage

// Перевірка співпадіння отриманого зображення з обличчям та зображень
з папки CheckBase
bool matchFound = false;
for (const auto& imagePath : imagePaths) {
    Mat baseImage = imread(imagePath);
    if (!baseImage.empty()) {
        if (compareImages(videoImage, baseImage)) {
            matchFound = true;
            break;
        }
    }
}

// Відправка електронної пошти у разі відсутності співпадінь
if (!matchFound) {
    sendEmail(videoImage);
    cout << "Електронна пошта надіслана." << endl;
}

// Відправка електронної пошти
sendEmail(frame);

```

```

cout << "Електронна пошта надіслана." << endl;

// Збереження обличчя у файл
string filename = "unknown_face.jpg";
string filePath = fs::path(checkBasePath) / filename;
imwrite(filePath, frame);
}

// Відображення прямокутника навколо обличчя на відеопотоці
rectangle(frame, face, Scalar(255, 0, 0), 2);
}

// Показ кадру з відображеними обличчями
imshow("Розпізнавання облич", frame);

// Вимикання програми при натисканні клавіші 'q'
if (waitKey(1) == 'q') {
    break;
}
}

// Закриття з'єднання та очищення ресурсів
delete con;
cap.release();
destroyAllWindows();

return 0;
}

```

Ім'я користувача:
Кафедра КІ

Дата перевірки:
23.06.2023 08:34:38 EEST

Дата звіту:
23.06.2023 08:35:09 EEST

ID перевірки:
1015680935

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Савченко_Мікроконтролерна система спостереження з використанням мікроконтролеру ES...
Кількість сторінок: 70 Кількість слів: 12042 Кількість символів: 93953 Розмір файлу: 2.51 MB ID файлу: 1015325218

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

4.05%

Схожість

Найбільша схожість: 1.29% з Інтернет-джерелом (<http://elar.khmnu.edu.ua/jspui/bitstream/123456789/13860/1/%d0%a0>).

3.89% Джерела з Інтернету	129	Сторінка 72
1.83% Джерела з Бібліотеки	132	Сторінка 72

2.58% Цитат

Цитати	3	Сторінка 73
Посилання	1	Сторінка 73

0%

Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування	13 сторінок
------------------------	-------------

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 13%

ID: 117826 Назва: БКР Мікроконтролерна система спостереження з використанням мікроконтролеру ESP32-CAM Додано в БД: 2023-06-23 Автора: Б.С.Савченко Керівники: О.С.Засорьов Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	81447	621	1529 (2%)	16 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Савченко Богдан Сергійович

Тема: Мікроконтролера система спостереження з використанням мікроконтролеру ESP32-CAM

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 62

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є проектування та розробка програмно-апаратного пристрою для виконання відеоспостереження за допомогою мікроконтролеру ESP32-CAM.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області, надані загальні відомості про мікроконтролери та сфери їх застосування, про системи відеоспостереження та їх класифікацію і використання мікроконтролерів у таких системах. Окрім того виконана постановка задачі дослідження. В другому розділі кваліфікаційної роботи було визначені технічні особливості мікроконтролеру ESP32-CAM, здійснено вибір складових для системи відеоспостереження, визначено що відеокамера моделі OV2640 краще відповідає потребам завдання. З метою визначення можливостей використання мікроконтролера ESP32-CAM було проаналізовані його переваги та недоліки а також виконане порівняння його з іншими мікроконтролерами, В третьому розділі кваліфікаційної роботи була проведена програмно-апаратна реалізація та тестування програмно-технічного засобу, а також реалізовані функції для алгоритму та визначено необхідні компоненти для тестування системи та її майбутнього використання, окрім того були розглянуті теоретичні можливості для покращення алгоритму.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: у роботі недостатня увага приділена тестуванню системи що була розроблена.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно стандартів що до оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.


8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: добре/с/ч.д

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Степан

Мельник Васильович, асистент викладача кафедри кібербезпеки,
д-т філософії

"26" травня 2023 р.

 (підпис)

Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорущенко Т. О.

Савченко Богдана Сергійовича

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-19-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 травня 2023 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Мікроконтролерна система відеоспостереження з використанням мікроконтролеру ESP32-CAM

Автор: Савченко Богдан Сергійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Засорнов Олександр Сергійович, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

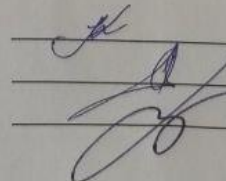
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 3.81% і адресується до 209 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС



О.С. Засорнов

С.М. Лисенко

Т. О. Говорущенко