

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень


Мікроконтролерна система аварійного керування замком з використанням мікро-  
контролера ESP32-CAM  
Назва теми


КВРКІП 2001141.20.17.01 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»  
Назва

Виконала: студентка IV курсу, група KI2c-20-1  М. І. Самойлюк  
Підпис Ініціали, прізвище

Керівник  О. С. Засорнов  
Підпис, дата Ініціали, прізвище

Нормоконтролер  С.М. Лисенко  
Підпис, дата Ініціали, прізвище

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

  
Підпис

Т.О. Говорущенко  
Ініціали, прізвище

«01» 06 2023 р.

Хмельницький 2023

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній 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

Самойлюку Максиму Ігоровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM

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

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

Затверджена наказом ректора університету від 01.03.2023 р. № 18

2. Строк подання студентом проекту (роботи) на кафедру 02.06.2023 р.

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

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

Огляд існуючих систем для розв'язання завдання

Обґрунтування вибору компонентів та середовища реалізації

Реалізація системи

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


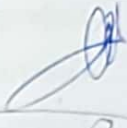


Схеми з'єднань пристрою

Інтерфейси програмно-апаратного засобу

Блок-схеми програм

Лістинг коду для моделювання в Packet Tracer

6. Консультанти розділів дипломного проекту (роботи)

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

7. Дата видачі завдання « 01 » 03 2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	20.02.2022	ВИКОНАНО
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.03.2023	ВИКОНАНО
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	10.03.2023	ВИКОНАНО
4	Робота над розділом 2 – проектування підсистеми	20.04.2023	ВИКОНАНО
5	Робота над розділом 3 – програмно-апаратна реалізація підсистеми	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

М.І. Самоїлюк  
Ініціали, прізвище

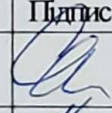
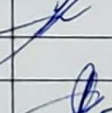
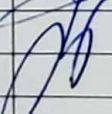
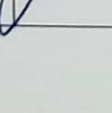
**Керівник проекту (роботи)**

  
Підпис

О.С. Засорнов  
Ініціали, прізвище

№ р я д к а	ф о р м а т	Позначення	Найменування	К і л - л и с т і в	№ екз	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРКІП 2001141.20.17.01 ПЗ	Пояснювальна записка	81		
			<u>Графічні матеріали</u>			
2		КвРКІП 2001141.20.17.01 Е8	Схеми апаратних з'єднань	1		
3		КвРКІП 2001141.20.17.01 Е8	Інтерфейси програмно-апаратного засобу	1		
4		КвРКІП 2001141.20.17.01 Е8	Блок-схеми програм	1		
5		КвРКІП 2001141.20.17.01 Е8	Лістинг коду для моделювання в Packet Tracer	1		

КвРКІП 2001141.20.17.01 ВП

Зм	Арк	№ докум	Підпис	Дата				
Розробив		Самойлок			Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Засорнов				У	1	1
Н. контр.		Лисенко				ХНУ, КІ2с-20-1		
Затв.		Говорущенко		01.06				

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM».

Автор роботи: Самойлюк Максим Ігорович.

Керівник роботи: Засорнов Олександр Сергійович.

Пояснювальна записка: 55 с., 41 рис., 1 табл., 4 дод., 59 джерел.

Графічна частина: 3 креслення.

МІКРОКОНТРОЛЕРНА СИСТЕМА, КЕРУВАННЯ, АВАРІЙНЕ, МІКРОКОНТРОЛЕР, ВИКОРИСТАННЯ ESP32-CAM.

Метою кваліфікаційної роботи є розроблення мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM.

Об'єктом дослідження є система керування з використанням мікроконтролера ESP32-CAM.

Предметом дослідження є Мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM.

Для досягнення поставленої мети використовуються такі методи дослідження, як методи синтезу, аналізу та моделювання процесів, принципи системного аналізу, теоретико-множинні підходи.

Практичне значення має спроектований та реалізована мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM.

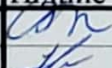
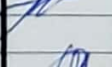


  
Підпис студента

01.05.2023

Дата

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ .....</b>	<b>4</b>
<b>ВСТУП .....</b>	<b>5</b>
<b>1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ РОЗВ'ЯЗАННЯ ЗАВДАННЯ.....</b>	<b>7</b>
1.1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ВИЯВЛЕННЯ НАЯВНИХ ПРОБЛЕМ І ЗАВДАНЬ .....	7
1.2 ПОРІВНЯЛЬНИЙ АНАЛІЗ ПЕРЕВАГ ТА НЕДОЛІКІВ ІСНУЮЧИХ РІШЕНЬ .....	9
1.3 ПІДХОДИ ДО ВИРІШЕННЯ ЗАДАЧІ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ.....	11
1.4 ВИСНОВКИ ТА ПОСТАНОВКА ЗАДАЧІ.....	14
<b>2 ОБГРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ .....</b>	<b>16</b>
2.1 АПАРАТНЕ СЕРЕДОВИЩЕ РЕАЛІЗАЦІЇ .....	16
2.1.1 Опис ESP32-CAM .....	17
2.2 ПРИНЦИП РОБОТИ КОНТАКТНОГО МАГНІТНОГО ЗАМКА .....	22
2.4 ВИБІР МЕТОДІВ І СЕРЕДОВИЩА ПРОГРАМУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	29
2.5 ВИБІР МЕТОДІВ І СЕРЕДОВИЩА ПРОЄКТУВАННЯ .....	32
2.6 ВИСНОВКИ .....	36
<b>3 РЕАЛІЗАЦІЯ СИСТЕМИ.....</b>	<b>37</b>
3.1 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВАРІЙНОГО КЕРУВАННЯ ЗАМКОМ З ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРА ESP32-CAM.....	37
3.2 АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ АВАРІЙНОГО КЕРУВАННЯ ЗАМКОМ З ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРА ESP32-CAM.....	53
3.3 ВИСНОВКИ .....	58
<b>ВИСНОВКИ .....</b>	<b>59</b>

<b>КВРКІП 2001141.20.17.01 ПЗ</b>				
<b>Зм.</b>	<b>Арк</b>	<b>№докум.</b>	<b>Підпис</b>	<b>Дата</b>
Виконав		Самойлюк М.І.		
Перевір.		Засорнов О.С.		
Н.контр.		Лисенко С.М.		
Затвер.		Говорущенко Т.О.		21.06
Мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM				
		<b>Літера</b>	<b>Аркуш</b>	<b>Аркушів</b>
			2	
ХНУ, КІ2с-20-1				

<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....</b>	<b>60</b>
<b>ДОДАТОК А схеми апаратних з'єднань.....</b>	<b>66</b>
<b>ДОДАТОК Б інтерфейси програмно-апаратного засобу .....</b>	<b>67</b>
<b>ДОДАТОК В блок-схеми програм .....</b>	<b>68</b>
<b>ДОДАТОК Г лістинг коду для моделювання в packet tracer.....</b>	<b>69</b>

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

КС – комп'ютерна система

WI-FI – Wireless Fidelity

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		4

## ВСТУП

Останні дослідження свідчать про стабільний розвиток інформаційних технологій, що в подальшому будуть автоматизовані.

Відеосистеми широко використовуються в різних галузях, включаючи системи спостереження на об'єктах, вуличний відеонагляд, охоронні системи та наукові дослідження.

Це стимулює постійний розвиток нових форм відеосистем та експериментів з їхніми характеристиками.

Актуальність даної роботи полягає в необхідності розробки робототехнічної системи аварійного керування замком на основі мікроконтролера ESP32-CAM.

Плата ESP32-CAM добре підходить для побудови відеосистем завдяки вбудованому модулю камери та бездротовому підключенню. Ось деякі можливості для створення відеосистем з використанням ESP32-CAM:

Потокове відео в реальному часі: ESP32-CAM може захоплювати відеокадри і транслювати їх в режимі реального часу через з'єднання Wi-Fi. Налаштувавши веб-сервер на ESP32-CAM, ви можете отримати доступ до потокового відео з веб-браузера або спеціального додатку на віддаленому пристрої.

Камери безпеки. Завдяки своєму модулю камери і можливостям Wi-Fi, ESP32-CAM можна використовувати для побудови систем відеоспостереження, які можна зробити своїми руками. Ви можете налаштувати алгоритми виявлення руху, щоб запустити відеозапис або захоплення зображення при виявленні руху, а потім зберегти або передати отримані дані для моніторингу або аналізу.

Системи відеоспостереження: Об'єднавши кілька модулів ESP32-CAM, ви можете створити розподілену систему відеоспостереження. Кожна камера ESP32-CAM може бути встановлена в різних зонах, захоплюючи відеоматеріали і передаючи їх в централізоване місце для моніторингу та аналізу.

Віддалений моніторинг: ESP32-CAM можна використовувати для моніторингу віддалених місць або об'єктів. Розмістивши плату в певному місці і підключивши

					КвРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		5

її до Інтернету, ви можете отримати віддалений доступ до відеопотоку для моніторингу навколишнього середовища, наприклад, для перевірки стану віддаленого об'єкта нерухомості або спостереження за природним середовищем існування.

Взаємодія з відео. Можна інтегрувати ESP32-CAM з іншими пристроями і датчиками для створення інтерактивних відеосистем. Наприклад, ви можете використовувати алгоритми комп'ютерного зору для виявлення певних об'єктів або жестів і запускати дії або сповіщення на основі аналізу відеопотоку.

Завдання включає дослідження функціонування мікроконтролера та замка, аналіз сфери застосування, опис структури та базової моделі, а також опис існуючих механізмів реалізації та ідентифікацію проблем та їх вирішення.

На основі цього дослідження мають бути сформульовані функціональні та нефункціональні вимоги до системи та розроблена модель функцій, які має виконувати система.

Метою кваліфікаційної роботи є побудова мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM.

Об'єктом дослідження є процес керування побутовими приладами у кіберфізичній системі «Розумний будинок».

Предметом дослідження є програмно-технічний засіб керування побутовими приладами у кіберфізичній системі «Розумний будинок».

Для досягнення поставленої мети використовуються такі методи дослідження, як методи синтезу, аналізу та моделювання процесів, принципи системного аналізу, теоретико-множинні підходи.

Практичне значення має спроектована та реалізована мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM.

# 1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ РОЗВ'ЯЗАННЯ ЗАВДАННЯ

## 1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Замки є важливим елементом забезпечення безпеки будинків, квартир та офісних приміщень.

Сучасні системи замків можуть бути автоматизовані та управлятися з допомогою різних технологій, включаючи мікроконтролери.

Однак, на сьогоднішній день існує декілька проблем з існуючими системами аварійного керування замком, які можуть призвести до необхідності заміни замка або до виникнення ризику вторгнення в будинок [3-6].

Одна з головних проблем існуючих систем аварійного керування замком полягає в тому, що вони часто потребують додаткового обладнання, наприклад, давача руху або магнітного контакту.

Це призводить до збільшення вартості системи та складнощів з її встановленням.

Крім того, датчики можуть вимагати підтримки акумулятора, що може призвести до зменшення часу роботи системи або до необхідності регулярної заміни батареї [4].

Іншою проблемою є можливість вторгнення у будинок при відсутності власника ключів.

Це може статися у разі, якщо ключі були вкрадені або загублені. У такому випадку, є необхідність відкрити замок з допомогою аварійної системи керування об'єктом.

Однак, існуючі системи аварійного керування замком можуть бути ненадійними або неефективними, що може призвести до злому замка або вторгнення в будинок [7-9].

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		7

З метою вирішення цих проблем, можна розробити мікроконтролерну систему аварійного керування замком з використанням мікроконтролера ESP32-CAM [10-12].

Така система може бути більш ефективною та надійною, оскільки може працювати з допомогою вбудованих датчиків, які відслідковують рух та зміни у середовищі.

Застосування мікроконтролера ESP32-CAM дозволяє також використовувати зображення з камери для визначення присутності людини біля дверей та забезпечення їх безпеки.

Це може бути корисним для людей, які часто перебувають далеко від дому або офісу, оскільки вони можуть отримати доступ до камери через мережу Інтернет і перевірити, чи немає неправильних дій біля вхідних дверей [13-17].

Іншою перевагою мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM є те, що вона може бути легко налаштована та оновлена з використанням спеціального системного програмного забезпечення.

Це дає можливість користувачам змінювати налаштування системи, щоб вона відповідала їхнім потребам, а також отримувати оновлення без необхідності використовувати нову апаратуру [18].

Отже, мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM може вирішити декілька проблем, пов'язаних з існуючими системами керування замком.

Це може зробити будинки, квартири та офіси безпечнішими та зручнішими для користувачів [19].

Однак, перед тим, як розпочати розробку такої системи, необхідно провести детальний аналіз предметної області, щоб ідентифікувати проблеми та завдання, які можна розв'язати з допомогою цієї системи.

Одна з головних проблем існуючих систем керування замком полягає в їх недостатній безпеці.

Зазвичай, системи керування замком мають недоліки, які можуть бути використані хакерами або зловмисниками, щоб отримати незаконний доступ до будинку або офісу.

Наприклад, більшість існуючих систем керування замком працюють з використанням магнітних карт або кодів доступу, які можуть бути вкрадені або скопійовані [20].

Застосування мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM дозволяє створити більш безпечну систему керування замком, яка може використовувати біометричні дані, такі як відбитки пальців, для ідентифікації користувача та надання доступу.

Іншою проблемою існуючих систем керування замком є їх обмежена функціональність.

Більшість систем керування замком мають дуже простий функціонал, який дозволяє лише відкривати та закривати замок.

Застосування мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM дозволяє додати додатковий функціонал, такий як система автоматичного відкривання дверей для людей з обмеженими можливостями або система автоматичного відключення замка в разі пожежі або іншої надзвичайної ситуації.

## 1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Звичайні системи керування замком мають свої переваги та недоліки.

Основні переваги таких систем - це їх простота, доступність та низька вартість.

Однак, такі системи мають декілька недоліків, таких як низький рівень безпеки та обмежена функціональність.

Порівняно з існуючими рішеннями, мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM має кілька суттєвих переваг.

Основна перевага - це високий рівень безпеки, завдяки використанню біометричних даних для ідентифікації користувачів та надання доступу [21].

Біометричні дані, такі як відбитки пальців, є унікальними та не можуть бути скопійовані або вкрадені, що робить систему більш безпечною в порівнянні з традиційними системами керування замком.

Ще одна перевага мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM полягає в її функціональності.

Завдяки можливостям мікроконтролера ESP32-CAM, система може мати додатковий функціонал, такий як система автоматичного відкривання дверей для людей з обмеженими можливостями або система автоматичного відключення замка в разі пожежі або іншої надзвичайної ситуації.

Це робить систему більш привабливою для користувачів з різними потребами та може додатково збільшити її безпеку [22].

Однак, мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM також має свої недоліки.

Основним недоліком є висока вартість порівняно з традиційними системами керування замком.

Вартість мікроконтролера ESP32-CAM є вищою, ніж вартість звичайного замка та інших компонентів, що використовуються для розробки такої системи.

Також, для користування мікроконтролерною системою потрібні спеціальні знання та навички програмування, що може становити складність для деяких користувачів [23-24].

Іншою проблемою, пов'язаною з мікроконтролерними системами, є їхній обмежений ресурс процесора та пам'яті.

Це може стати перешкодою для розширення функціональності системи та робить її менш гнучкою в порівнянні зі звичайними відомими системами керування замком.

Отже, після порівняльного аналізу можна зробити висновок, що мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM має більше переваг, ніж недоліків порівняно зі звичайними системами керування замком.

Однак, висока вартість та обмеженість ресурсів мікроконтролера можуть стати чинниками, які потрібно враховувати при виборі такої системи [25].

### 1.3 Підходи до вирішення задачі за темою дослідження

Робота сучасних пристроїв залежить від пристроїв керування, які називаються мікроконтролерами.

Ці мікроконтролери слугують інтерфейсом між кінцевим користувачем та його пристроєм керування, як правило, мобільним телефоном.

У сучасному світі мобільні телефони стали надійними помічниками для кожного, пропонуючи всі необхідні функції користувача для взаємодії з навколишнім середовищем за допомогою програмних додатків.

Ці додатки дозволяють виконувати такі завдання, як система сповіщень, що реалізується через радіоканал за допомогою вбудованих мікросхем мобільного телефону.

Зв'язок між мобільним телефоном та іншими пристроями встановлюється за допомогою бездротових дистанційних технологій, таких як Wi-Fi або Bluetooth.

Є декілька підходів до вирішення задачі розробки мікроконтролерної системи аварійного керування замком з використанням саме мікроконтролера ESP32-CAM.

Розглянемо кілька з них.

Розробка системи з нуля.

					КвРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		11

Цей підхід передбачає розробку всіх елементів системи від початку до кінця, включаючи обладнання та програмне забезпечення [26-27].

Цей підхід може бути досить часовим та ресурсозатратним, але він дозволяє створити систему з високою рівнем налаштування та гнучкістю в зміні функціональності в майбутньому.

Використання готових компонентів.

Цей підхід полягає в тому, щоб використовувати готові компоненти, такі як замки та мікроконтролери, та поєднувати їх в систему.

Цей підхід може бути менш ресурсозатратним та швидким, але може мати обмеження в налаштуванні та гнучкості.

Використання готових рішень.

Цей підхід полягає в тому, щоб використовувати готові системи керування замком та додавати до них додатковий функціонал за допомогою мікроконтролера ESP32-CAM.

Цей підхід може бути найшвидшим та найменш ресурсозатратним, але може бути обмеженим в можливостях налаштування та гнучкості.

Кожен з цих підходів має свої переваги та недоліки, тому вибір залежить від конкретної ситуації та потреб користувачів.

Наприклад, якщо час та ресурси обмежені, то варто розглянути підхід з використанням готових рішень, а якщо важлива гнучкість та налаштування, то потрібно розглянути підхід з розробки системи з нуля [28].

Крім того, під час вирішення задачі розробки мікроконтролерної системи аварійного керування замком можна використовувати різні методики та підходи до проектування, такі як:

Методика системного аналізу.

Цей підхід полягає в тому, щоб дослідити всі аспекти системи, включаючи її структуру, функції та взаємодії, з метою виявлення недоліків та пропозиції оптимальних рішень.

Методика розробки програмного забезпечення.

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		12

Цей підхід полягає в тому, щоб розробити програмне забезпечення для мікроконтролера, використовуючи різні технології та методики, такі як Agile, Scrum тощо.

Методика гнучкої розробки.

Цей підхід передбачає постійне змінення та вдосконалення системи, відповідно до змін потреб користувачів та умов використання.

Кожна з цих методик має свої переваги та недоліки та може бути використана для вирішення конкретних завдань при розробці мікроконтролерної системи аварійного керування замком.

Узагальнюючи, для вирішення задачі розробки мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM можна використовувати різні підходи та методики, залежно від специфіки проекту та потреб користувачів.

Крім того, важливо розглянути порівняльний аналіз існуючих рішень та їх переваг та недоліків, щоб зробити правильний вибір технологій та підходів до розробки системи [23,29-30].

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		13

## 1.4 Висновки та постановка задачі

Постановка задачі є важливим етапом розробки мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-CAM.

На даному етапі визначаються вимоги до системи, обґрунтовується необхідність її розробки та визначаються основні завдання, які повинні бути вирішені в ході проектування та розробки системи.

Основна мета даного проекту - розробити мікроконтролерну систему аварійного керування замком, яка дозволить забезпечити безпеку користувачів та їх майна.

За допомогою цієї системи користувачі зможуть контролювати стан замку та керувати ним.

Основні завдання, які повинні бути вирішені в ході розробки системи, наступні:

1. Розробити апаратну частину системи, яка складатиметься з мікроконтролера ESP32-CAM, електронної схеми та необхідних компонентів.
2. Розробити програмне забезпечення, яке буде відповідати за роботу системи. Програмне забезпечення повинно забезпечувати зчитування даних з камери, керування отриманою інформацією та взаємодію з користувачем.
3. Реалізувати можливість віддаленого керування системою з використанням мобільного пристрою та безпечну передачу даних з мікроконтролера до мобільного пристрою.
4. Протестувати розроблену систему на різних етапах розробки та після завершення проекту.

Постановка задачі дозволяє визначити основні вимоги до системи та визначити основні задачі, які повинні бути вирішені в ході розробки системи. Також в процесі постановки задачі необхідно визначити обмеження, які повинні бути враховані при проектуванні та розробці системи. Наприклад, обмеженнями можуть

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		14

бути часові та бюджетні рамки, а також обмеження з використання технологій та обладнання.

Важливо також визначити очікувані результати від проекту, тобто що повинна забезпечити розроблена система, які цілі вона повинна допомогти досягти та які проблеми вона повинна вирішити.

Постановка задачі також дозволяє визначити потенційних користувачів та їх потреби, що дозволяє зрозуміти, наскільки відповідає розроблена система вимогам та потребам користувачів.

У процесі постановки задачі також необхідно визначити методики, які будуть використані під час проектування та розробки системи, а також методики, які будуть використані для тестування та оцінки розробленої системи.

Отже, постановка задачі є важливим етапом в розробці мікроконтролерної системи аварійного керування замком з використанням мікроконтролера ESP32-SAM, який дозволяє визначити вимоги до системи, визначити обмеження та очікувані результати, а також визначити методики, які будуть використані в процесі проектування та розробки.

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		15

## 2 ОБГРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ

### 2.1 Апаратне середовище реалізації

Мікроконтролерна система аварійного керування замком була розроблена з використанням ESP32-CAM, яка об'єднує модуль камери, а також усі можливі функційні можливості вказаного мікропроцесора на чіпі.

Зараз на ринку існує багато різних методів та апаратних систем ідентифікації осіб, таких як використання підписів, відбитків пальців, голосу, геометрії руки тощо.

Однак, незважаючи на це розмаїття методів, жодна з них не здатна ефективно виявляти та розпізнавати особи в громадських місцях, таких як аеропорти, роздрібні магазини та залізничні станції, за винятком систем розпізнавання обличчя з точки зору аварійного відкриття замками з урахуванням власника квартири, офісу тощо.

Таким чином, автоматизовані системи розпізнавання обличчя можуть використовуватися не тільки в цілях безпеки для розпізнавання осіб у різних сферах та місцях.

Такі системи також можуть використовуватися для цілей відвідуваності в офісах і школах, та можуть допомогти відкрити замок як віддалено так безпосередньо біля оселі з урахування аварійної ситуації з одного боку, та не відкрити замок у випадку присутності зловмисника біля дверей.

З метою вирішення задачі в кваліфікаційній роботі було створено систему аварійного керування замком, яка включає в себе такі підсистеми:

- підсистема розпізнавання обличчя;
- підсистема передачі даних;
- підсистема роботи з сервером.

					КвРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		16

Підсистема розпізнавання обличчя застосовує використання мікроконтролер ESP32-CAM.

Крім того, вона може виконувати функцію підсистеми безпеки з мікроконтролером ESP32-CAM.

Це дозволить розпізнавати обличчя неавторизованих осіб при спробі відкриття замка.

ESP32-CAM є компактним модулем камери, який має чіп ESP32-S.

За допомогою модуля ESP32-CAM може бути створено систему розпізнавання обличчя, не потребуючи складного програмування та додаткових компонентів.

### 2.1.1 Опис ESP32-CAM

В роботі було використано модуль AI-Thinker ESP32-CAM.

Він має в комплекті чіп ESP32-S, а також дуже компактну відео камеру OV2640.

Модуль також має слот для карт micro SD.

Слот для карт Micro SD може використовуватись для зберігання зображень, знятих камерою, а також для зберігання файлів.

Використаний модуль ESP32-CAM має широкі можливості застосування в різних сферах Інтернету речей (IoT).

Він може використовуватися як система розпізнавання облич у офісах, школах та інших приватних приміщеннях.

Також він може використовуватися для бездротового моніторингу, бездротової ідентифікації QR-кодів та багатьох інших додатків IoT.

Модуль ESP32-CAM можна програмувати за допомогою ESP-IDF або Arduino IDE.

Модуль ESP32-CAM також має кілька контактів GPIO для підключення зовнішнього обладнання.

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		17

ESP32-CAM не має роз'єму USB, тому для програмування модуля потрібна плата FTDI.\

ESP32 є серією відносно недорогих чіпів.

Вони розроблені компанією Espressif Systems.

Вказаний чіп включає в себе систему на чіпі з інтегрованими радіоконтролерами:

- Wi-Fi;
- Bluetooth;
- Thread.

Серії ESP32 і ESP32-S використовують процесорні ядра під назвою «Tensilica».

В той серії ESP32-C і ESP32-H використовують ядро з мікроархітектурою типу RISC-V.

У мікропроцесорі наявний вже вбудований радіочастотний тракт.

Він включає в себе:

- балансуєчий трансформатор;
- вбудовані антенні перемикачі;
- компоненти радіочастоти;
- малошумний підсилювач;
- підсилювач потужності;
- фільтри та модулі керування потужністю.

Модуль ESP32 був розроблений і продовжує вироблятися компанією Espressif Systems, а його виготовлення здійснюється TSMC.

Технологія виготовлення складає 40 нм і 28 нм.

Особливості:

1. Найменший модуль для зв'язку 802.11b/g/n з модулем Wi-Fi та BT SoC (система на кристалі).
2. Енергоефективний 32-розрядний процесор, також може служити процесором різних прикладних додатків.

					КвРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		18

3. Тактова частота модуля складає до 160 МГц. Таким чином, його сумарна обчислювальна потужність може досягати до 600 DMIPS при пікових навантаженнях.

4. Модуль містить вбудовану пам'ять розміром 520 КБ SRAM, а також зовнішня 4MPSRAM.

5. Модуль підтримує наступний набір можливостей: UART/SPI/I2C/PWM/ADC/DAC.

6. Модуль отримав підтримку відеокамер OV2640 і OV7670, а також вбудований спалах.

7. Модуль також отримав підтримку завантаження отриманих зображень по протоколу Wi-Fi.

8. Модуль отримав підтримку TF карт (пам'ять для збереження отриманих зображень).

9. Для забезпечення енергоефективності модуль отримав кілька режимів сну.

10. Модуль отримав вбудовану підтримку протоколів Lwip і FreeRTOS для передачі даних.

11. Модуль також отримав підтримку таких режимів роботи як STA/AP/STA+AP.

12. Модуль має підтримку технології Smart Config/AirKiss.

13. Забезпечена підтримка локального та віддаленого оновлення системної прошивки через послідовний порт (FOTA).

Розглянемо технічні характеристики модуля :

1. SPI Flash: за замовчуванням обсяг становить 32 Мб використовуваної пам'яті;

2. Максимальний обсяг використовуваної оперативної пам'яті становить 520 КБ SRAM + 4 МБ PSRAM;

3. Максимальний обсяг використовуваної засобами карт TF становить 4Gb;

Зм.	Арк.	№докум.	Підпис	Дата

КВРКІП 2001141.20.17.01 ПЗ

Арк

19

4. Доступні вбудовані інтерфейси: UART, SPI, I2C, PWM;
5. Доступні формати виведення та збереження зображення: JPEG, BMP, GRAYSCAL;
6. Кількість портів ІО: 9 штук;
7. Живлення: 5 В.

Необхідні компоненти

- ESP32-CAM;
- Програматор FTDI.

Рисунок 2.1 демонструє схему приєднання модуля мікроконтролера до програматора FTDI.

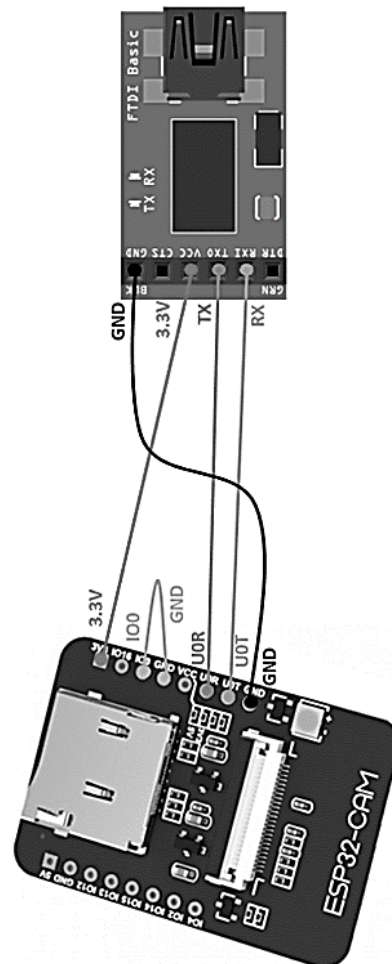


Рисунок 2.1 – Схема приєднання модуля до програматора

Для створення камери безпеки ESP32-CAM нам знадобиться модуль камери ESP32 і програматор FTDI для його програмування.

Зм.	Арк.	№докум.	Підпис	Дата

Оскільки ESP32-CAM не має роз'єму USB, для завантаження коду на нього потрібен програматор FTDI, як показано на зображенні вище.

Контакти Vcc і GND на ESP32 з'єднані з контактами Vcc і GND на платі FTDI. Крім того, контакти Tx і Rx на ESP32 з'єднані з контактами Rx і Tx на платі FTDI.

Зверніть увагу: перед завантаженням коду підключіть контакт IO0 до землі. Контакт IO0 визначає, чи ESP32 перебуває у режимі миготіння. Коли GPIO 0 підключений до GND, ESP32 знаходиться у режимі миготіння.

Після програмування ESP32 я від'єднав плату FTDI і підключив модуль до джерела живлення 3,3 В за допомогою регулятора напруги 7805.

Необхідний для реалізації кваліфікаційної роботи ESP32-CAM OV2640 є модулем, що включає в себе спеціальну камеру ov2640.

За допомогою цієї камери можна здійснювати знімки об'єктів, які розташовані поруч з роздільною здатністю 2 МПкс (рис.2.2).

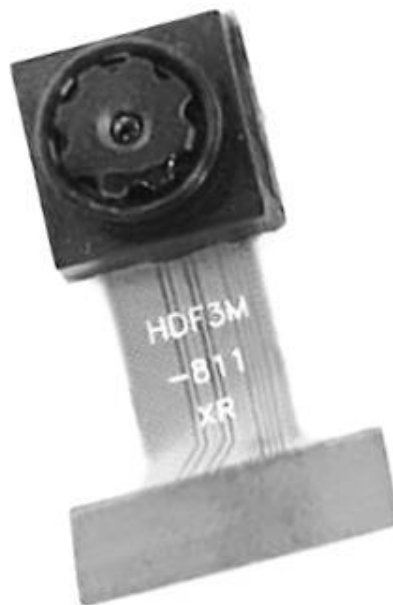


Рисунок 2.2 – Модуль камери

Це дозволяє чітко відтворювати об'єкти, що перебувають у полі зору, і зберігати чіткість без пікселізації або розмиття для подальшого аналізу отриманої інформації.

Також модуль можна застосувати для здійснення відео стріму.

Зм.	Арк.	№докум.	Підпис	Дата

Можна здійснювати підключення мікроконтролерів до необхідних датчиків, приводів та інших пристроїв, використовуючи контакти GPIO, які підтримують різні протоколи зв'язку [31] (рис.2.3).



Рисунок 2.3 – Мікроконтролер [26]

## 2.2 Принцип роботи контактної магнітної замка

Для вирішення поставленої задачі було використано контактний магнітний замок.

Електромагнітний замок, магнітний замок - це запірний пристрій, який складається з електромагніту і арматурної пластини, який використовується для контролю доступу дверних замків.

Прикріплюючи пристрій до дверної рами та пластину арматури до дверей, струм, що проходить через електромагніт, притягує пластину арматури, утримуючи дверний замок контролю доступу закритим.

На відміну від дверного замка з електрозамком контролю доступу, магнітний замок не має сполучних частин.

Зм.	Арк.	№докум.	Підпис	Дата

Можна обійти замок, відключивши подачу електроенергії. Крім того, він може працювати з мережевою системою контролю доступу, яка включає електричний замок, блок живлення, контролер доступу, кнопку виходу, дверний дзвінок або RFID-карти.

Особливості роботи замка:

1. Режим роботи: безвідмовний (блокування при включенні, розблокування при вимкненні).

1. Світлодіодний індикатор стану: вмикається під час блокування, вимикається під час розблокування (деякі режими без світлодіодного індикатора стану замка).

2. Червоний дріт є анодним «+» електродом, чорний дріт є катодним «-» електродом.

Акумуляовані дані мікроконтролера подано в таблиці 2.1

Таблиця 2.1 – Дані про мікроконтролер

Модулі передачі даних	SoC стандарту 802.11b/g/n, модуль WiFi, модуль Bluetooth
Розрядність центрального процесора	32 біти
Номінальна тактова частота центрального процесора	160.0 МГц

Кінець таблиці 2.1 – Дані про мікроконтролер

Основна вбудована системна пам'ять	520 КБ типу SRAM
Наявні системні інтерфейси	UART / SPI / I2C / виділений АЦП / виділений ЦАП
Вбудований модуль камера	OV2640 та OV7670
Завантаження зображень	Через WiFi
Карта флеш пам'яті	Стандарт - TF
Нявна підтримка режиму сну	+
Мережні стеки	Протоколи Lwip + FreeRTOS
Доступні робочі режими	STA-AP-STA-AP

На рисунку 2.4 подано можливий магнітний замок.

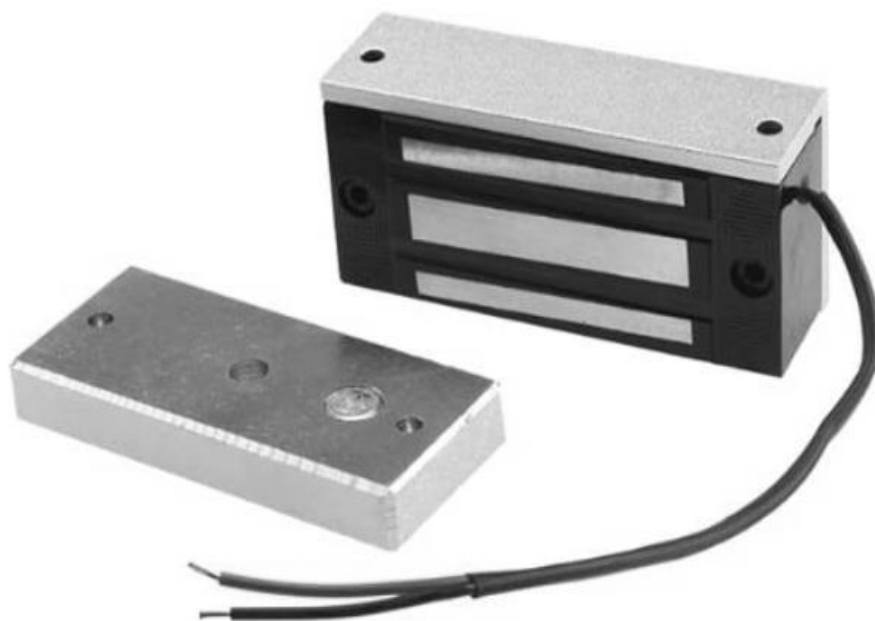


Рисунок 2.4 – Магнітний замок

На рисунку 2.5 подано піни мікроконтролера.

Зм.	Арк.	№докум.	Підпис	Дата

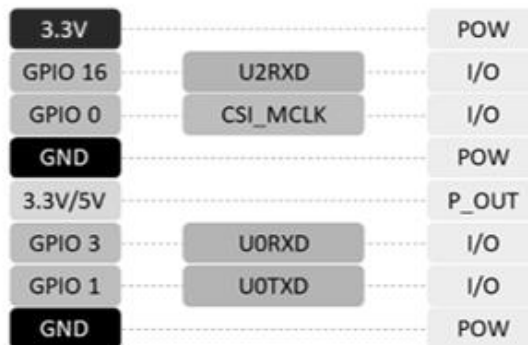
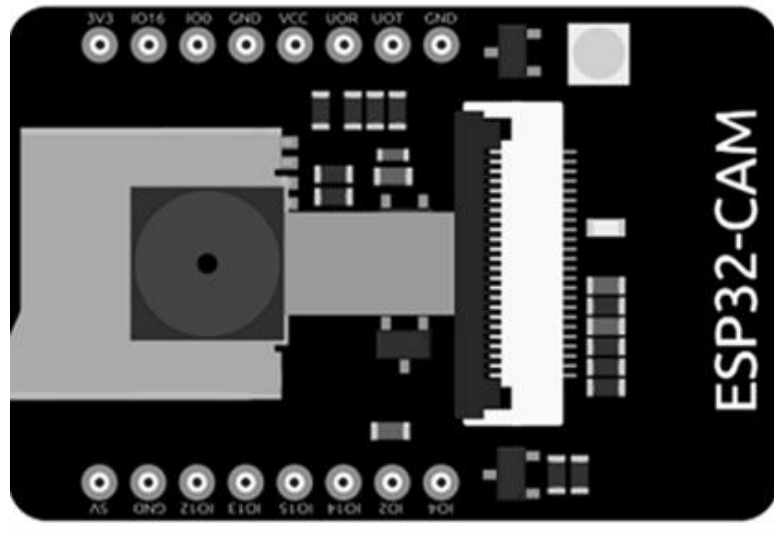
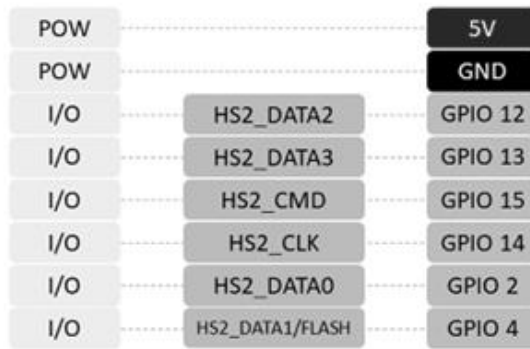


Рисунок 2.5 – Піни мікроконтролера

Також для проєкту необхідно застосувати реле (рисунок 2.6).

1. Технічні характеристики:
2. Розмір замка: 80Lx37.3Wx23.3H (мм) .
3. Арматурна пластина: 74Lx32Wx11H (мм) .

4. Сила тримання: 60 кг .
5. Режим: NC (блокується під час роботи джерела живлення), безвідмовний режим .
6. Напруга: DC12V .
7. Споживання струму: 0,11 А ~ 0,15 А .
8. Функція безпеки: вбудована система від стрибків напруги .
9. Режим відкривання: поворотні двері на 90 градусів .

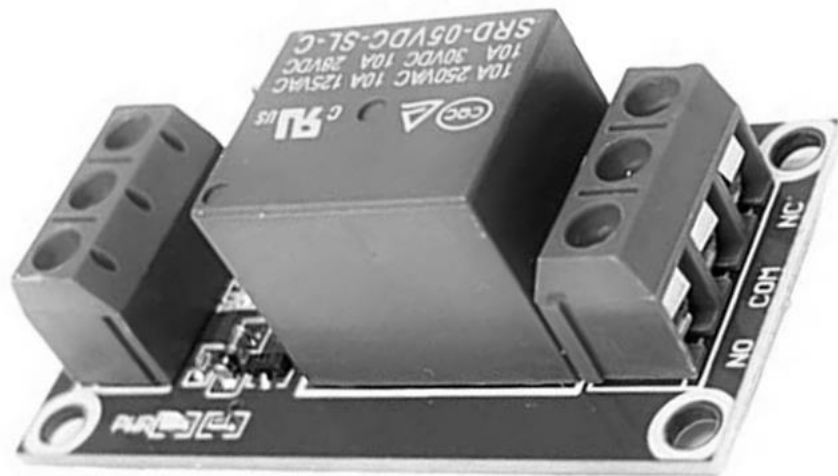


Рисунок 2.6 - Реле

Також важливо переконатися, що між магнітом і перемикачем немає жодних перешкод для передачі сигналу.

Правильне розташування компонентів для коректної роботи показано на схематичному рисунку 2.7.

Зм.	Арк.	№докум.	Підпис	Дата

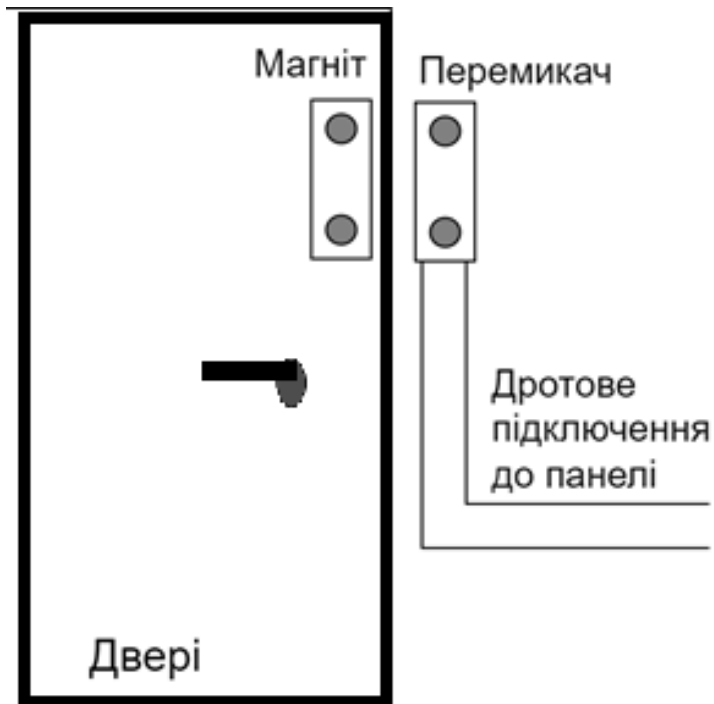


Рисунок 2.7 – Принцип роботи магнітного замка для дверей

Всередині перемикача знаходяться магнітні контакти, які створюють електричний коло.

Дане електричне коло може перебувати у двох різних станах: закритому та відкритому.

Магніт безпосередньо впливає на коло. Стан кола впливає на положення перемикача, яке визначає подію, а саме, чи двері закриті чи відкриті.

На рисунку 2.8 зображено вплив магніту, його поля, на положення перемикача.

Якщо двері зачинено то ланцюг з'єднано і передається сигнал, що позначений змінною  $R$  та дорівнює нулю [38].

При здійсненні відкривання дверей ланцюг роз'єднується, сигнал приймає значення одиниці та викликає подію, а саме про відчинені двері.

Якщо двері було спочатку відчинено то електричний ланцюг перебуває в стані роз'єднано.

У цьому випадку сигнал дорівнює одиниці.

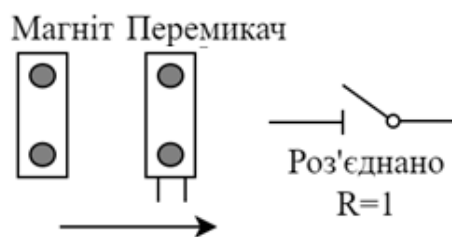
Зм.	Арк.	№докум.	Підпис	Дата

В ситуації, коли відбувається здійснення зачинення дверей – ланцюг з'єднується і сигнал переходить в стан нуля.

Даний перехід станів зображено на рисунках 2.9-2.10.



а)

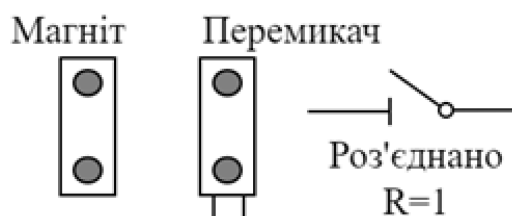


б)

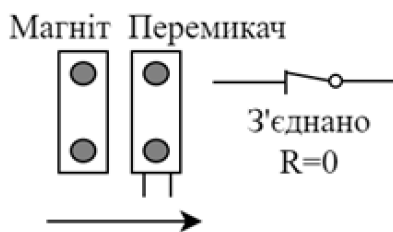
Рисунок 2.9 – Стан зачинених дверей:

а) з'єднано

б) роз'єднано



а)



б)

Рисунок 2.10 – Стан зачинених дверей: а) з'єднано; б) роз'єднано

Зм.	Арк.	№докум.	Підпис	Дата

## 2.4 Вибір методів і середовища програмування програмного забезпечення

Для роботи з мікроконтролером ESP32-CAM можна використати програмне середовище ESP-IDF, а код писати мовою програмування C або обрати середовище Arduino і мову програмування C++.

Можна використати для реалізації системи і інші програми, але краще середовище Arduino, щоб мати можливість писати код з ефективним використанням пам'яті, мати ширший вибір бібліотек коду та мати більшу спільноту розробників, щоб знайти інформацію для розробки.

Дане середовище програмування використовує компілятор gcc, який можна використовувати окремо від його інтегрованого середовища розробки (IDE) із системою збірки make [41].

Це інтегроване середовище розробки, яке підтримується на різних операційних системах, таких як Windows, Mac OS X і Linux.

У програмному забезпеченні Arduino (IDE) програми, написані користувачем, називаються скетчами.

Скетчі зберігаються в текстовому редакторі з розширенням .ino.

Редактор має функції вирізання/вставки та пошуку/заміни тексту програмного коду.

В окремій області повідомлень відображаються повідомлення про збереження, експорт, а також повідомлення про помилки.

Консоль відображає текст, який генерується програмним забезпеченням Arduino (IDE), включаючи повні повідомлення про помилки та іншу використовувану інформацію.

Інформація про вибрану плату та послідовний порт відображається в нижньому правому куті вікна.

Кнопки на панелі інструментів дозволяють:

- перевіряти програми;
- завантажувати програми;

- створювати скетчі;
- , відкривати скетчі;
- зберігати скетчі
- відкривати послідовний монітор.

Для роботи з мікроконтролером, який базується на мікросхемі ESP32-CAM, можна використовувати програмне середовище ESP-IDF та писати системний код мовою програмування C.

Також можливим є обрання середовища Arduino IDE та використовувати мову для системного програмування C++.

Arduino IDE є програмним забезпеченням з відкритим вихідним кодом, розробленим компанією Arduino.cc.

Воно призначене переважно для написання, компіляції та завантаження коду на модулі Arduino.

Arduino IDE подано на рисунку 2.11.

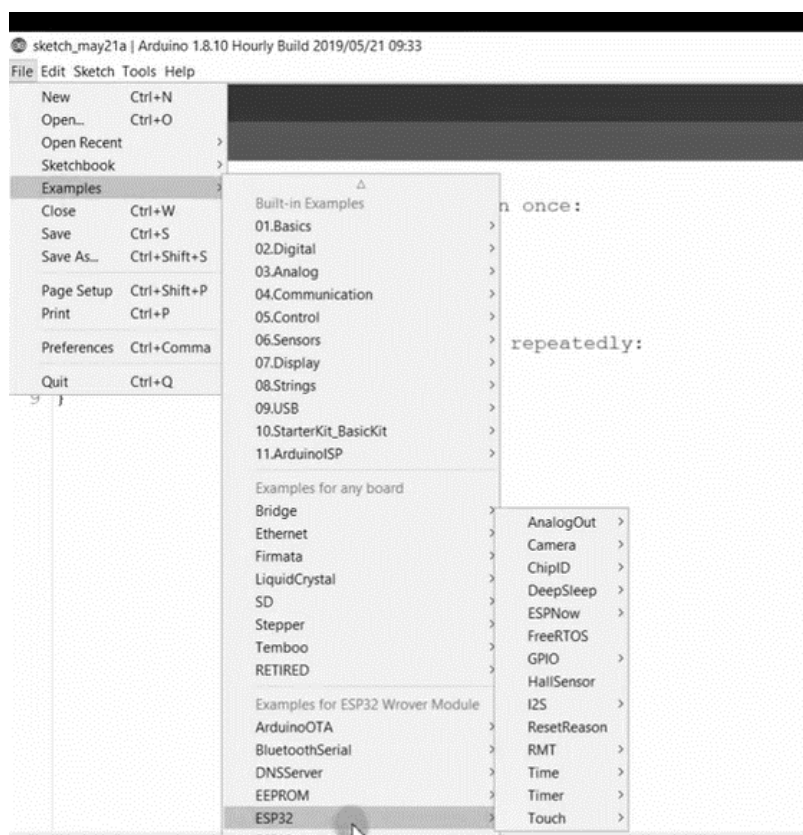


Рисунок 2.11 – Arduino IDE

Зм.	Арк.	№докум.	Підпис	Дата

Arduino IDE є офіційним програмним забезпеченням Arduino, яке зробило процес компіляції коду настільки простим, що навіть людина без технічних знань може розпочати навчання.

Воно доступне для всіх операційних систем, таких як macOS, Windows і Linux, і працює на платформі Java з вбудованими функціями і командами, які відіграють важливу роль у налагодженні, редагуванні та компіляції коду.

Arduino IDE підтримує різні модулі Arduino, включаючи Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro та багато інших.

Кожен з цих модулів містить мікроконтролер на платі, який може бути програмований для обробки інформації у вигляді коду.

Основний код, також відомий як "скетч", створюється на платформі Arduino IDE і в кінцевому підсумку генерує Hex-файл. Цей файл потім передається та завантажується в мікроконтролер на платі Arduino.

Середовище Arduino IDE складається з двох основних частин: Редактора і Компілятора.

Редактор використовується для написання необхідного коду, а Компілятор - для компіляції та завантаження коду на модуль Arduino.

Arduino IDE підтримує мови програмування C та C++.

Хоча для реалізації системи аварійного керування замком з використанням ESP32-CAM доступні й інші програмні середовища та засоби, в кваліфікаційній роботі було рекомендовано обрати середовище Arduino.

Основним аргументом є:

- його ефективне використання системної пам'яті
- наявність великої колекції системних програмних бібліотек для вирішення різноманітних задач;
- велика спільнота розробників, що полегшує пошук інформації та підтримки для потреб у розробці;
- середовище для системного програмування використовує безкоштовний компілятор gcc;

- компілятор можна використовувати незалежно від інтегрованого середовища розробки (IDE), використовуючи систему збірки програмного коду make [41].

Існує інший варіант використання PlatformIO, який забезпечує взаємодію з багатьма інтегрованими середовищами розробки (IDE) та апаратними платформами. Воно також надає функції, які не залежать від конкретної платформи, такі як налагодження, тестування та аналіз коду.

Arduino також підтримує можливість оновлення по повітрю (OTA), якщо є достатньо місця воно використовується для нової версії програмного забезпечення [42].

Тому, планується розробити програмне забезпечення в середовищі Arduino з використанням ESP32CAM Arduino IDE.

Це середовище надає:

- програмний інтерфейс (API);
- можливість для підключення до Wi-Fi мережі
- забезпечує комунікацію через мережні протоколи TCP та UDP;
- дозволяє створювати HTTP-сервери;
- створювати DNS-сервери;
- працювати з флеш-пам'яттю;
- взаємодіяти з великою множиною I2C-пристроїв [44].

## 2.5 Вибір методів і середовища проектування

Для і створення макетних схем та їх ізуалізації можливим є використання програмного забезпечення Fritzing.

Це відкрите програмне забезпечення, яке спрощує процес створення електронних схем і робить їх доступними для використання.

Fritzing дозволяє віртуальне моделювання електричних схем.

Якщо в стандартній бібліотеці програми відсутній певний елемент схеми, його можна знайти в Інтернеті та успішно використовувати разом з програмним забезпеченням [44-45].

Для проєктування системи доцільним є використання платформи Fritzing. Fritzing – програмне забезпечення EDA та інструмент для дизайнерів пристроїв.

Вкладки Welcome, Breadboard, Schematic, PCB і Code представляють основну навігацію через Fritzing. Макетна дошка дозволяє переглядати макет (експериментальну плату), що дає відчуття фізичного укладання компонентів.

Breadboard – інструмент для створення електричної схеми. Компоненти та модулі, що використовуються в одному з цих переглядів, також автоматично з’являються, а також є можливість створити для них спеціальні посилання (маршрутизувати їх).

PCB – інструмент для створення друкованої електронної плати (рисунок 2.12).

Проєктування друкованої плати є можливим за допомогою вкладки «Друкована плата» (рис. 2.13).

Вкладка «Код» дозволяє писати та змінювати системний код і завантажувати його безпосередньо на плату Arduino (рис.2.14).

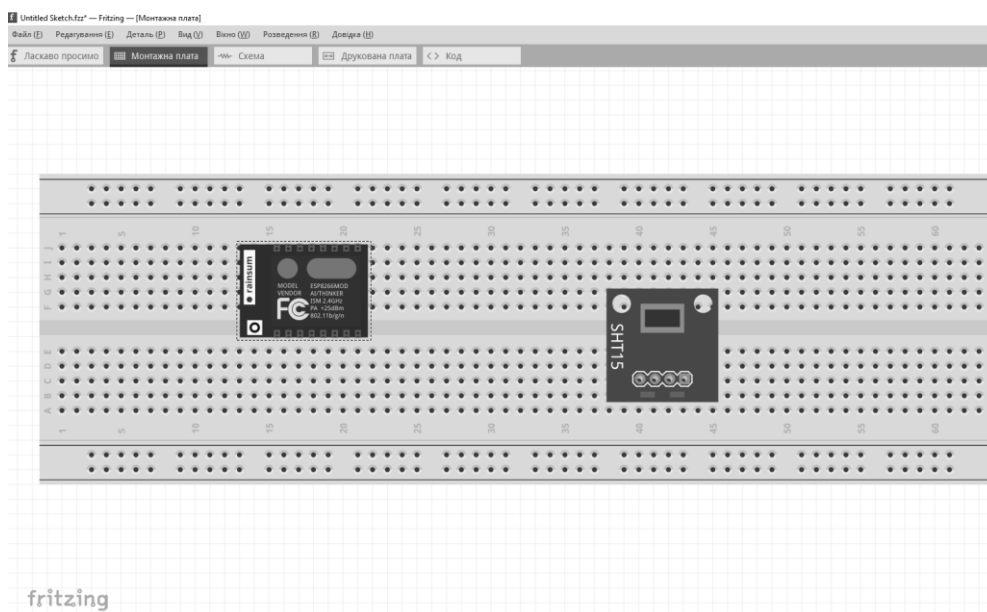


Рисунок 2.12 - Breadboard – інструмент для створення електричної схеми

Зм.	Арк.	№докум.	Підпис	Дата



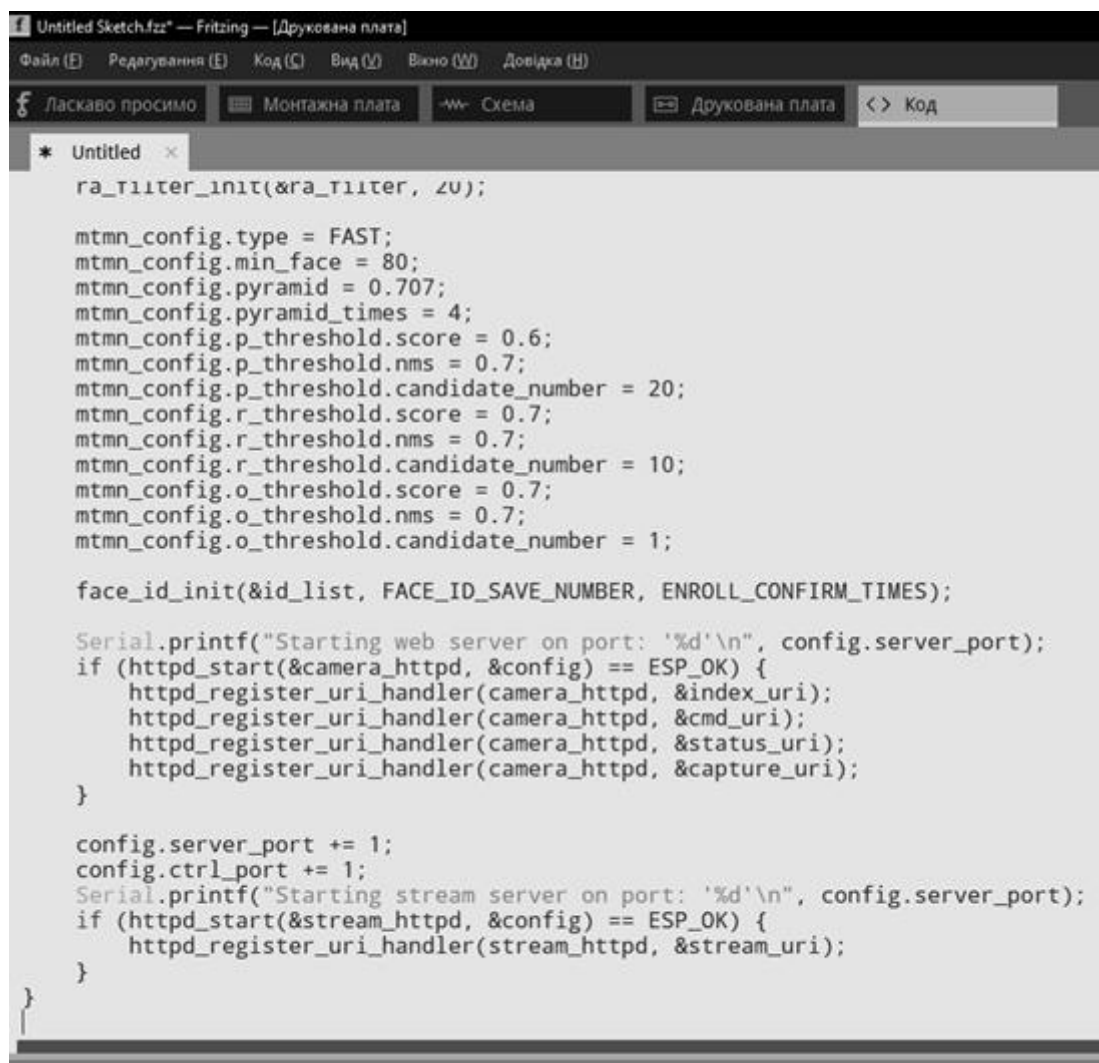
Вкладка «Встановлення нових модулів» Права бічна панель містить усі доступні модулі та компоненти.

В проєкті можна скористатися пошуком необхідної компоненти за допомогою пошукової системи або контейнерів, які насправді є категоріально впорядкованими модулями.

Frtizing поставляється з обмеженою кількістю модулів.

Для окремих продуктів можна знайти посилання для завантаження документів Fritzing

Всю бібліотеку можна завантажити з нашого профілю GitHub.



```
Untitled Sketch.fzz* — Fritzing — [Друкована плата]
Файл (E)  Редагування (E)  Код (C)  Вид (V)  Вікно (W)  Довідка (H)
Ласкаво просимо  Монтажна плата  -w- Схема  Друкована плата  <> Код
* Untitled x
ra_filter_init(&ra_filter, 20);

mtmn_config.type = FAST;
mtmn_config.min_face = 80;
mtmn_config.pyramid = 0.707;
mtmn_config.pyramid_times = 4;
mtmn_config.p_threshold.score = 0.6;
mtmn_config.p_threshold.nms = 0.7;
mtmn_config.p_threshold.candidate_number = 20;
mtmn_config.r_threshold.score = 0.7;
mtmn_config.r_threshold.nms = 0.7;
mtmn_config.r_threshold.candidate_number = 10;
mtmn_config.o_threshold.score = 0.7;
mtmn_config.o_threshold.nms = 0.7;
mtmn_config.o_threshold.candidate_number = 1;

face_id_init(&id_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES);

Serial.printf("Starting web server on port: '%d'\n", config.server_port);
if (httpd_start(&camera_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(camera_httpd, &index_uri);
    httpd_register_uri_handler(camera_httpd, &cmd_uri);
    httpd_register_uri_handler(camera_httpd, &status_uri);
    httpd_register_uri_handler(camera_httpd, &capture_uri);
}

config.server_port += 1;
config.ctrl_port += 1;
Serial.printf("Starting stream server on port: '%d'\n", config.server_port);
if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &stream_uri);
}
}
```

Рисунок 2.15 – Вкладка «Код»

## 2.6 Висновки

В розділі описано апаратне середовище реалізації та опис ESP32-CAM. Також принцип роботи контактного магнітного замка.

Вибір методів і середовища програмування програмного забезпечення та середовища проектування.

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		36

## 3 РЕАЛІЗАЦІЯ СИСТЕМИ

### 3.1 Програмна реалізація системи аварійного керування замком з використанням мікроконтролера ESP32-CAM

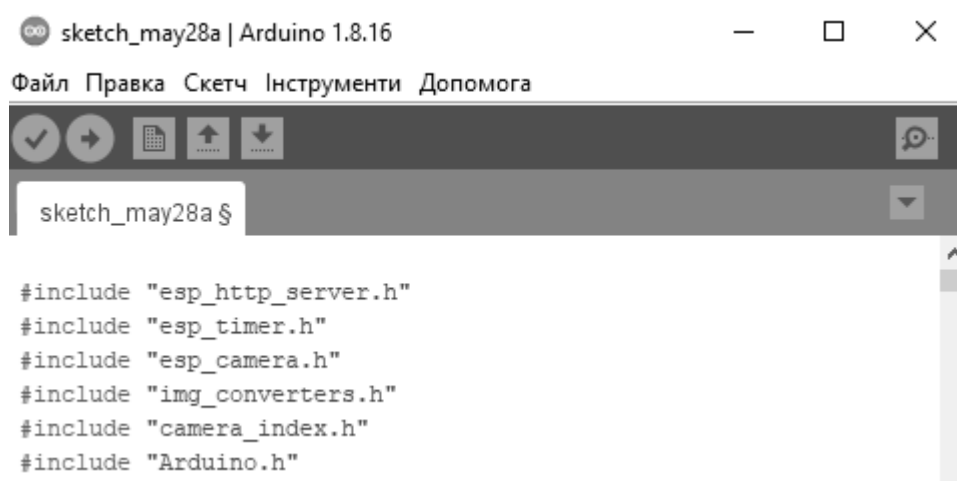
Першим етапом системного програмування є підключення системних бібліотек. Системний код для ESP32-CAM має таку ж структуру, як і відповідний код Arduino:

- код установки;
- код циклу.

Для забезпечення функціонування коду установки базовою функцією є код у функції `setup()`. Він виконується одразу після увімкнення живлення або скидання пристрою. Системний код використовується для ініціалізації змінних, режимів виводів, початку використання бібліотек для системи.

Код циклу виконується одразу після коду установки і може виконуватися багаторазово (нескінченно), і використовується для виконання основної задачі програми.

Ініціалізація системних бібліотек подано на рисунку 3.1.



```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a §
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "camera_index.h"
#include "Arduino.h"
```

Рисунок 3.1 - Ініціалізація системних бібліотек

Зм.	Арк.	№докум.	Підпис	Дата

Базовою функцією роботи мікроконтролера є налаштування роботи усіх портів (рисунок 3.2).

З цією метою виконується ініціалізацію усіх пінів мікроконтролера ESP32-CAM.



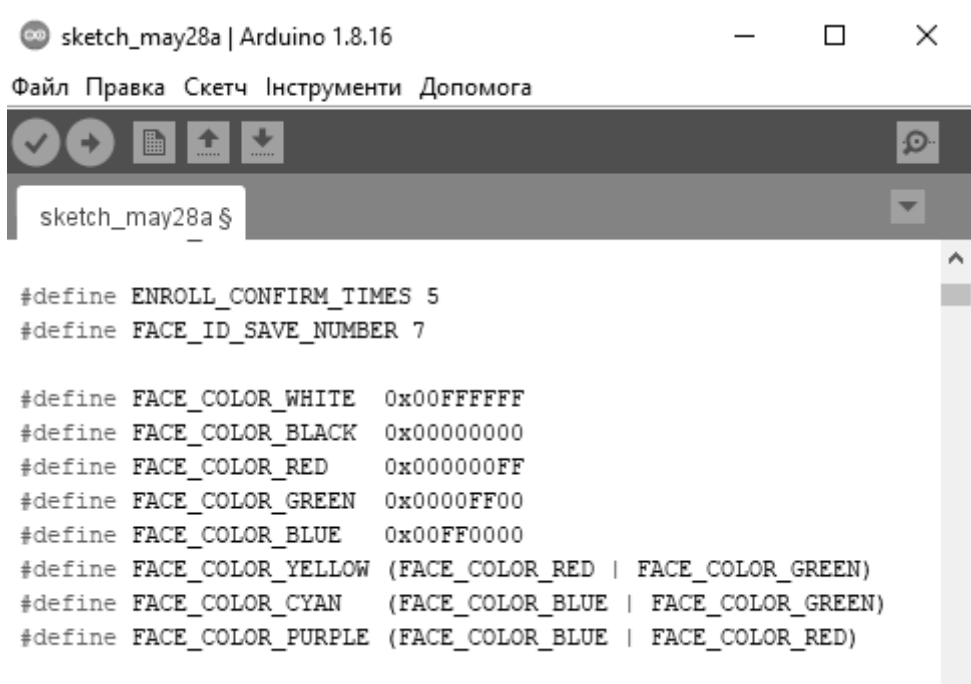
```
void setup() {
  pinMode(Relay, OUTPUT);
  pinMode(Red, OUTPUT);
  pinMode(Green, OUTPUT);
  digitalWrite(Relay, LOW);
  digitalWrite(Red, HIGH);
  digitalWrite(Green, LOW);

  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
```

Рисунок 3.2 - Налаштування роботи усіх портів мікроконтролера

Наступним кроком було здійснено задання змінних для розпізнавання кольорів камери ESP32-CAM (рис .3.3).

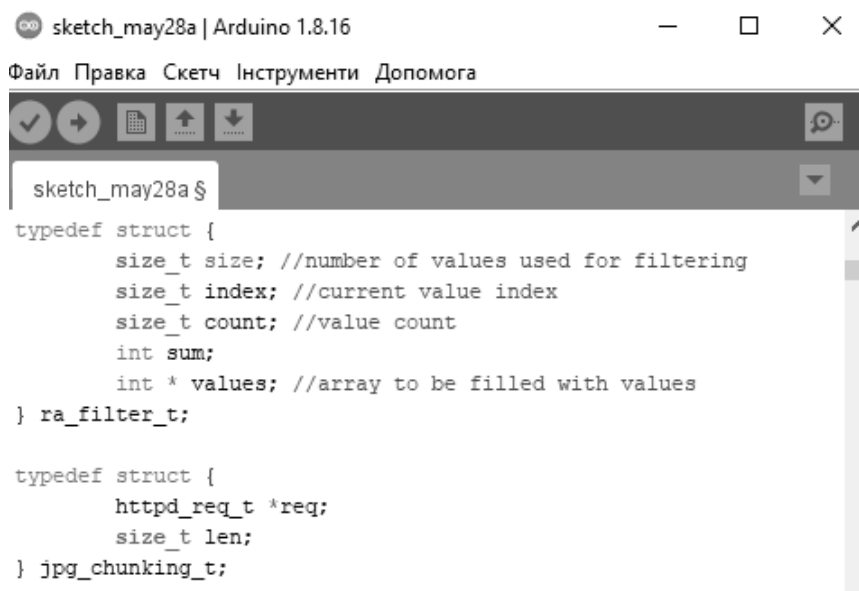


```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a $
#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7

#define FACE_COLOR_WHITE  0x00FFFFFF
#define FACE_COLOR_BLACK  0x00000000
#define FACE_COLOR_RED     0x000000FF
#define FACE_COLOR_GREEN   0x0000FF00
#define FACE_COLOR_BLUE    0x00FF0000
#define FACE_COLOR_YELLOW  (FACE_COLOR_RED | FACE_COLOR_GREEN)
#define FACE_COLOR_CYAN    (FACE_COLOR_BLUE | FACE_COLOR_GREEN)
#define FACE_COLOR_PURPLE  (FACE_COLOR_BLUE | FACE_COLOR_RED)
```

Рисунок 3.3 – Задання змінних для розпізнавання кольорів камери ESP32-CAM

Наступним завданням було задання основної структури роботи із зображеннями (рис .3.4).

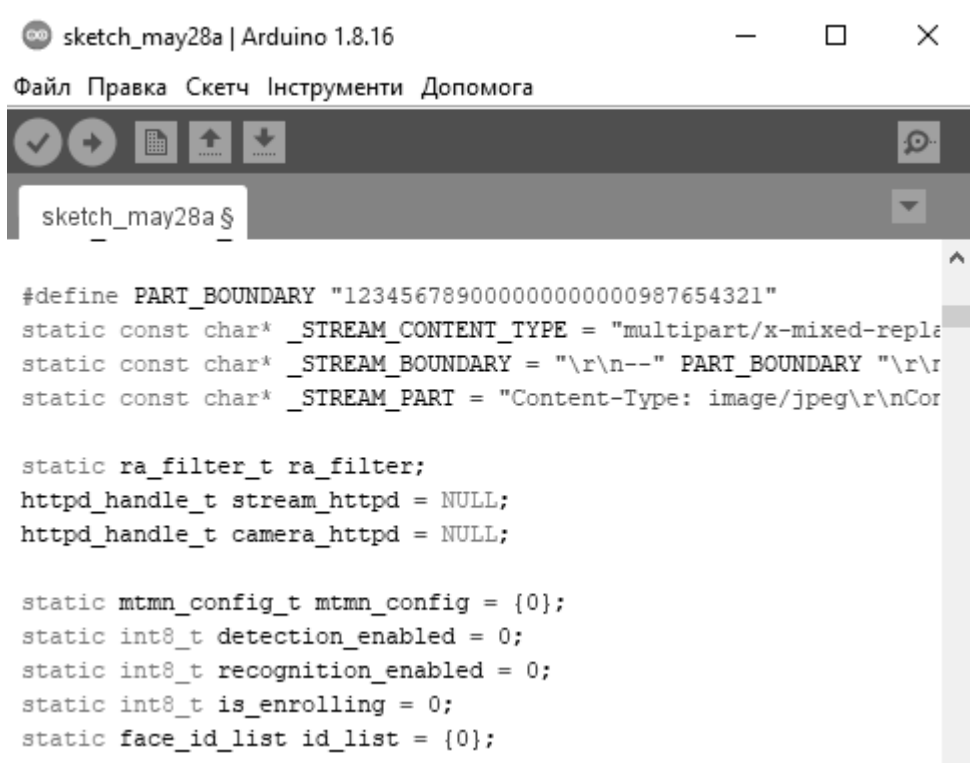


```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a $
typedef struct {
    size_t size; //number of values used for filtering
    size_t index; //current value index
    size_t count; //value count
    int sum;
    int * values; //array to be filled with values
} ra_filter_t;

typedef struct {
    httpd_req_t *req;
    size_t len;
} jpg_chunking_t;
```

Рисунок 3.4 – Задання основної структури роботи із зображеннями

Після цього потрібно задати параметри роботи камери, що відповідають за стрім відеопотоку, граничні значення, параметри фільтрів для камери тощо (рис. 3.5).



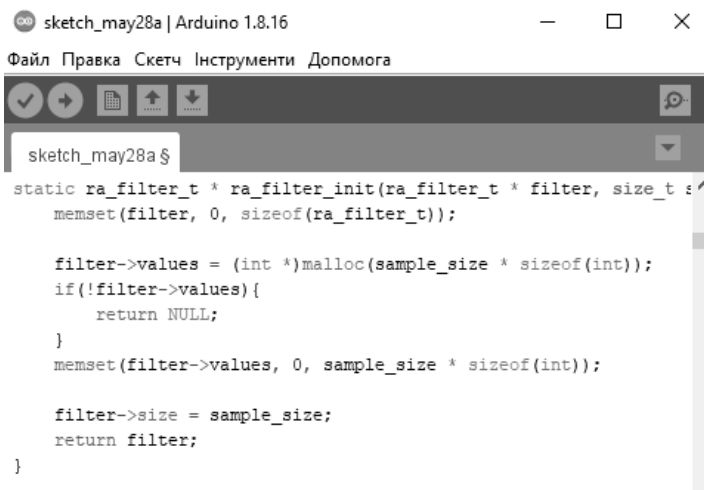
```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a $
#define PART_BOUNDARY "12345678900000000000000987654321"
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-repl
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nCor

static ra_filter_t ra_filter;
httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;

static mtmn_config_t mtmn_config = {0};
static int8_t detection_enabled = 0;
static int8_t recognition_enabled = 0;
static int8_t is_enrolling = 0;
static face_id_list id_list = {0};
```

Рисунок 3.5 – Налаштування параметрів камери

Задамо функцію `ra_filter_t *` для обробки корегуючих фільтрів зображення (рисунок 3.6).



```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a $
static ra_filter_t * ra_filter_init(ra_filter_t * filter, size_t s
memset(filter, 0, sizeof(ra_filter_t));

filter->values = (int *)malloc(sample_size * sizeof(int));
if(!filter->values){
return NULL;
}
memset(filter->values, 0, sample_size * sizeof(int));

filter->size = sample_size;
return filter;
}
```

Рисунок 3.6 – Задання функції `ra_filter_t *` для обробки фільтрів зображення

Зм.	Арк.	№докум.	Підпис	Дата

Блок-схема функції `ra_filter_t *` для обробки фільтрів зображення подано на рисунку 3.7.



Рисунок 3.7 – Блок-схема функції `ra_filter_t *` для обробки фільтрів зображення

Опишемо функцію `rgb_print` для обробки та виведення RGB зображень, що приймає три параметри: матриця кольорів, поточний оброблюваний колір та обробник стану (рисунку 3.8).

Блок-схема функції обробки та виведення RGB зображень подано на рис. 3.9.

```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a $
static void rgb_print(dl_matrix3du_t *image_matrix, uint32_t color) {
    fb_data_t fb;
    fb.width = image_matrix->w;
    fb.height = image_matrix->h;
    fb.data = image_matrix->item;
    fb.bytes_per_pixel = 3;
    fb.format = FB_BGR888;
    fb_gfx_print(&fb, (fb.width - (strlen(str) * 14)) / 2, 10, col
}
```

Рисунок 3.8 – Обробка та виведення RGB зображень

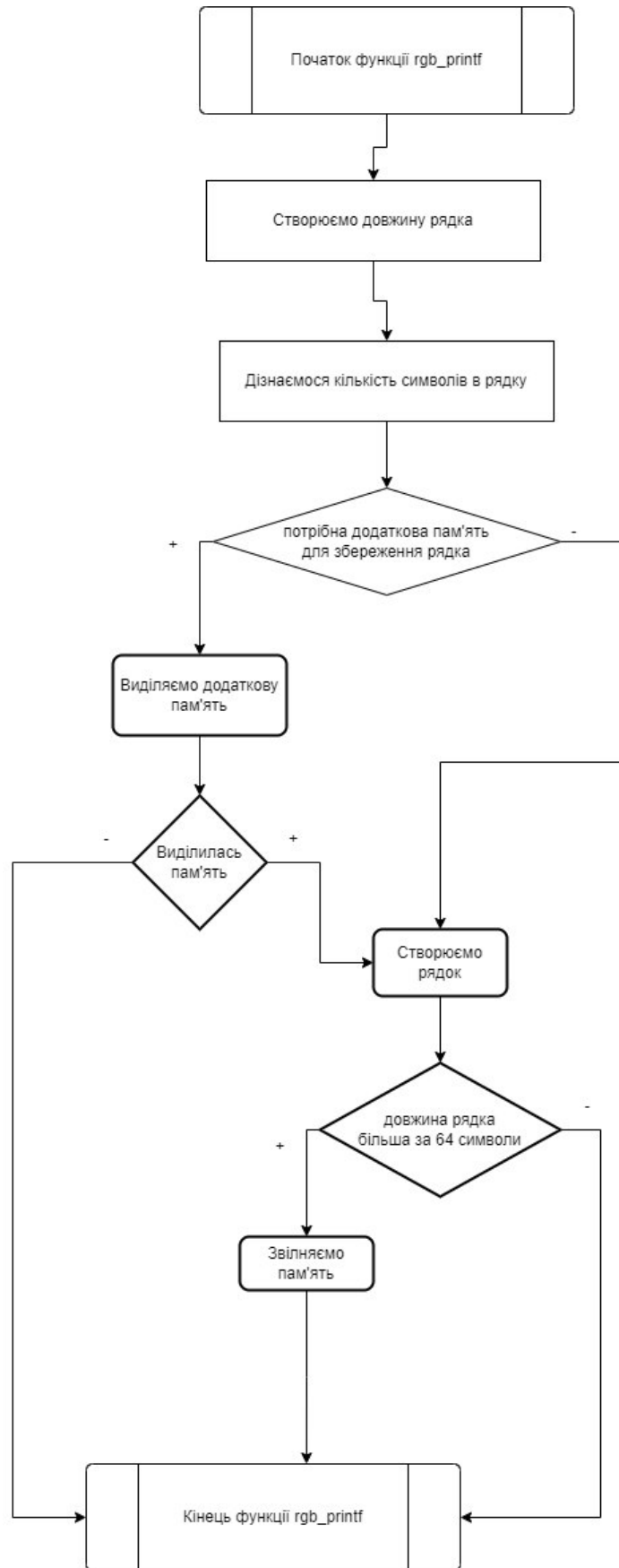
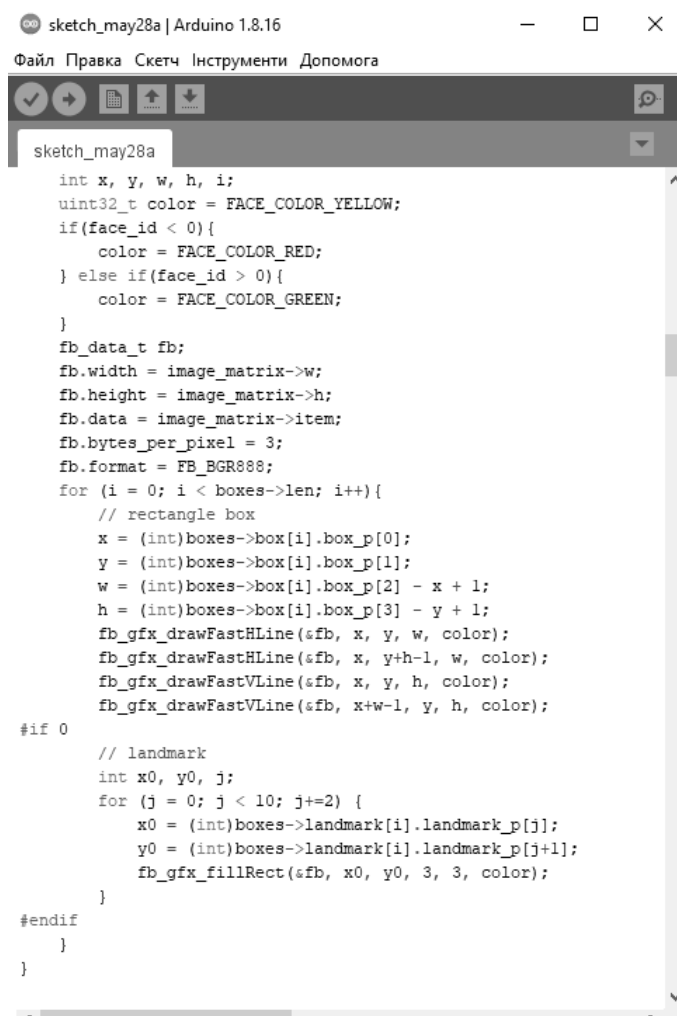


Рисунок 3.9 – Блок-схема функції обробки та виведення RGB зображень

Задамо функцію `draw_face_boxes()` розбиття зображення на блоки для швидкої обробки (рис. 3.10).



```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a
int x, y, w, h, i;
uint32_t color = FACE_COLOR_YELLOW;
if(face_id < 0){
    color = FACE_COLOR_RED;
} else if(face_id > 0){
    color = FACE_COLOR_GREEN;
}
fb_data_t fb;
fb.width = image_matrix->w;
fb.height = image_matrix->h;
fb.data = image_matrix->item;
fb.bytes_per_pixel = 3;
fb.format = FB_BGR888;
for (i = 0; i < boxes->len; i++){
    // rectangle box
    x = (int)boxes->box[i].box_p[0];
    y = (int)boxes->box[i].box_p[1];
    w = (int)boxes->box[i].box_p[2] - x + 1;
    h = (int)boxes->box[i].box_p[3] - y + 1;
    fb_gfx_drawFastHLine(&fb, x, y, w, color);
    fb_gfx_drawFastHLine(&fb, x, y+h-1, w, color);
    fb_gfx_drawFastVLine(&fb, x, y, h, color);
    fb_gfx_drawFastVLine(&fb, x+w-1, y, h, color);
}
// landmark
int x0, y0, j;
for (j = 0; j < 10; j+=2) {
    x0 = (int)boxes->landmark[i].landmark_p[j];
    y0 = (int)boxes->landmark[i].landmark_p[j+1];
    fb_gfx_fillRect(&fb, x0, y0, 3, 3, color);
}
#endif
}
```

Рисунок 3.10 – Розбиття зображення на блоки

Блок-схема функцію `draw_face_boxes()` розбиття зображення на блоки для швидкої обробки подано на рис. 3.11.

Блок-схема функції обробки та виведення RGB зображень представлено на рисунок 3.9.

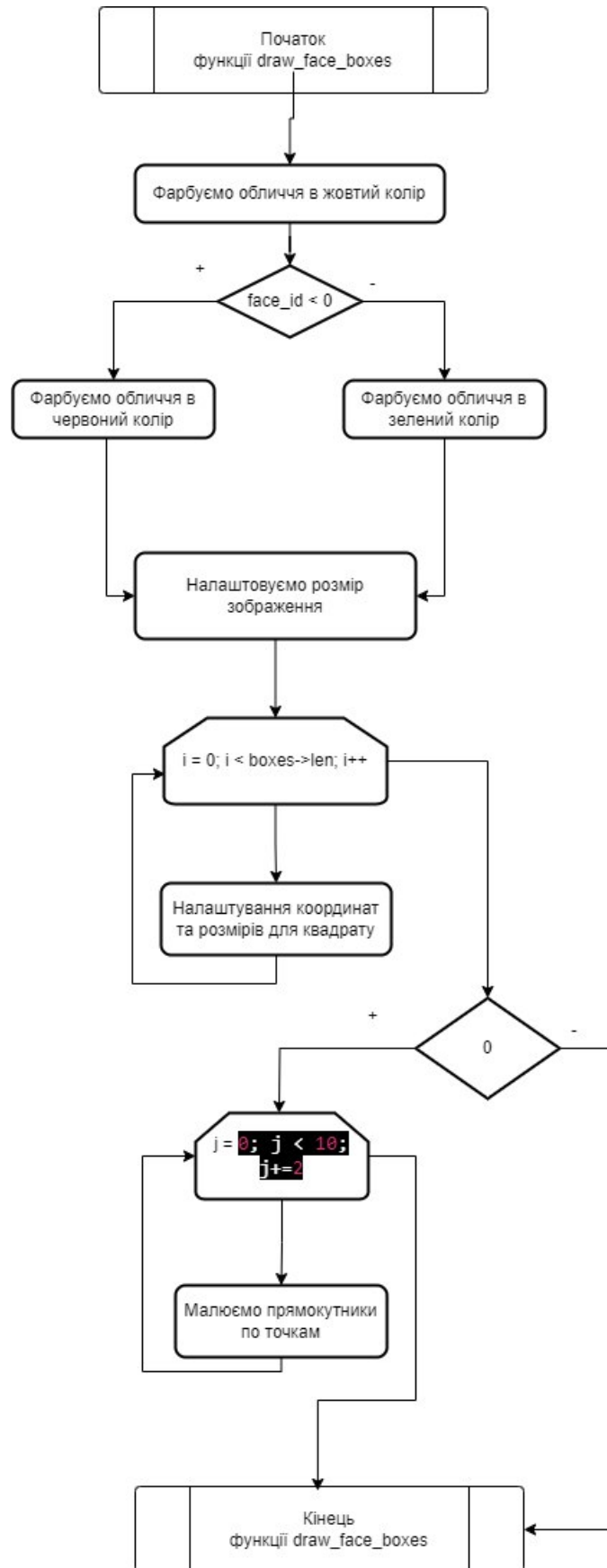


Рисунок 3.11 – Блок-схема функцію draw\_face\_boxes() розбиття зображення на блоки

Безпосередній функціонал здійснення розпізнавання об'єкту на зображенні виконується функцією `static int run_face_recognition()`, блок-схема якої подано на рисунку 3.12.

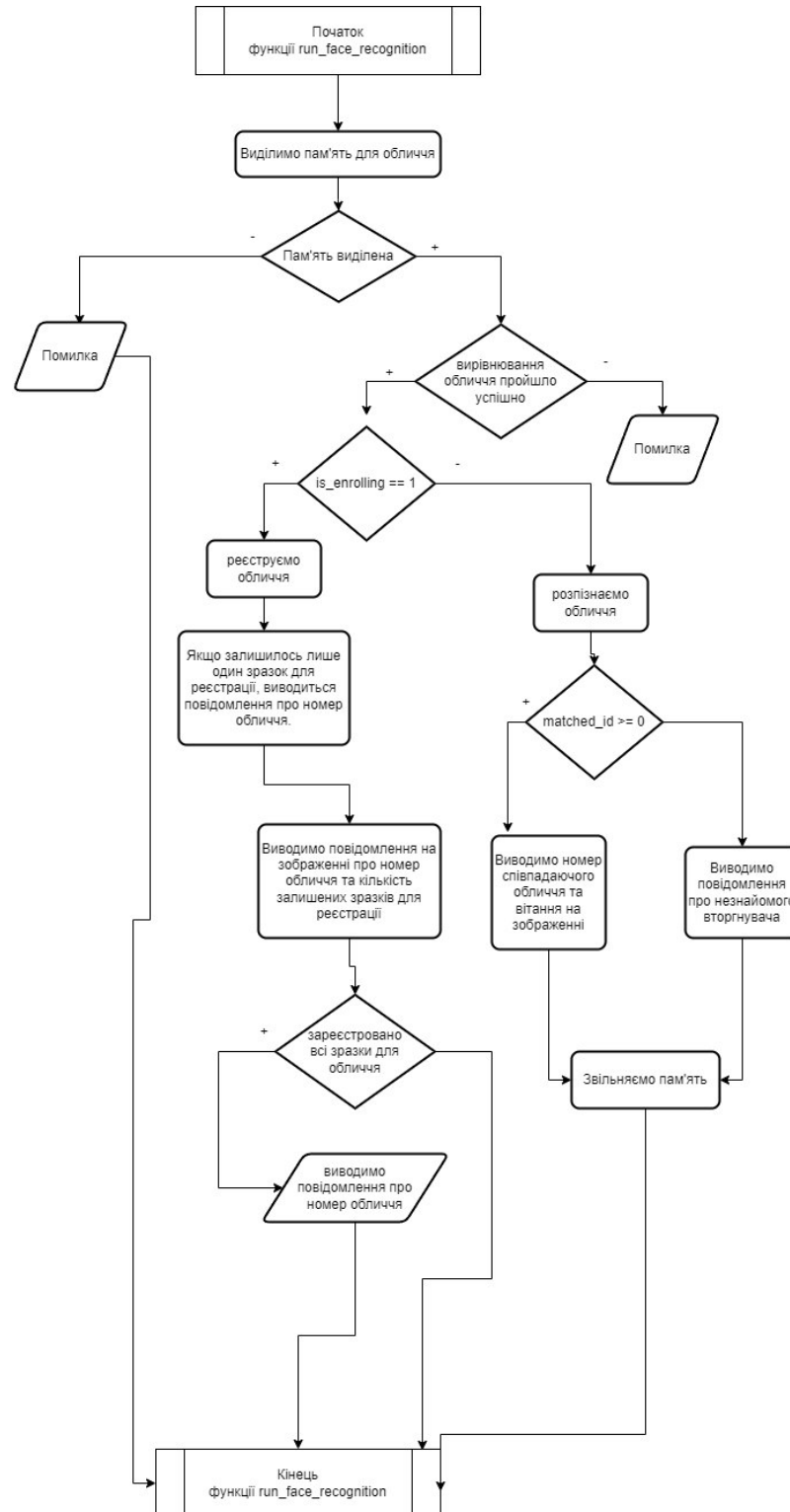
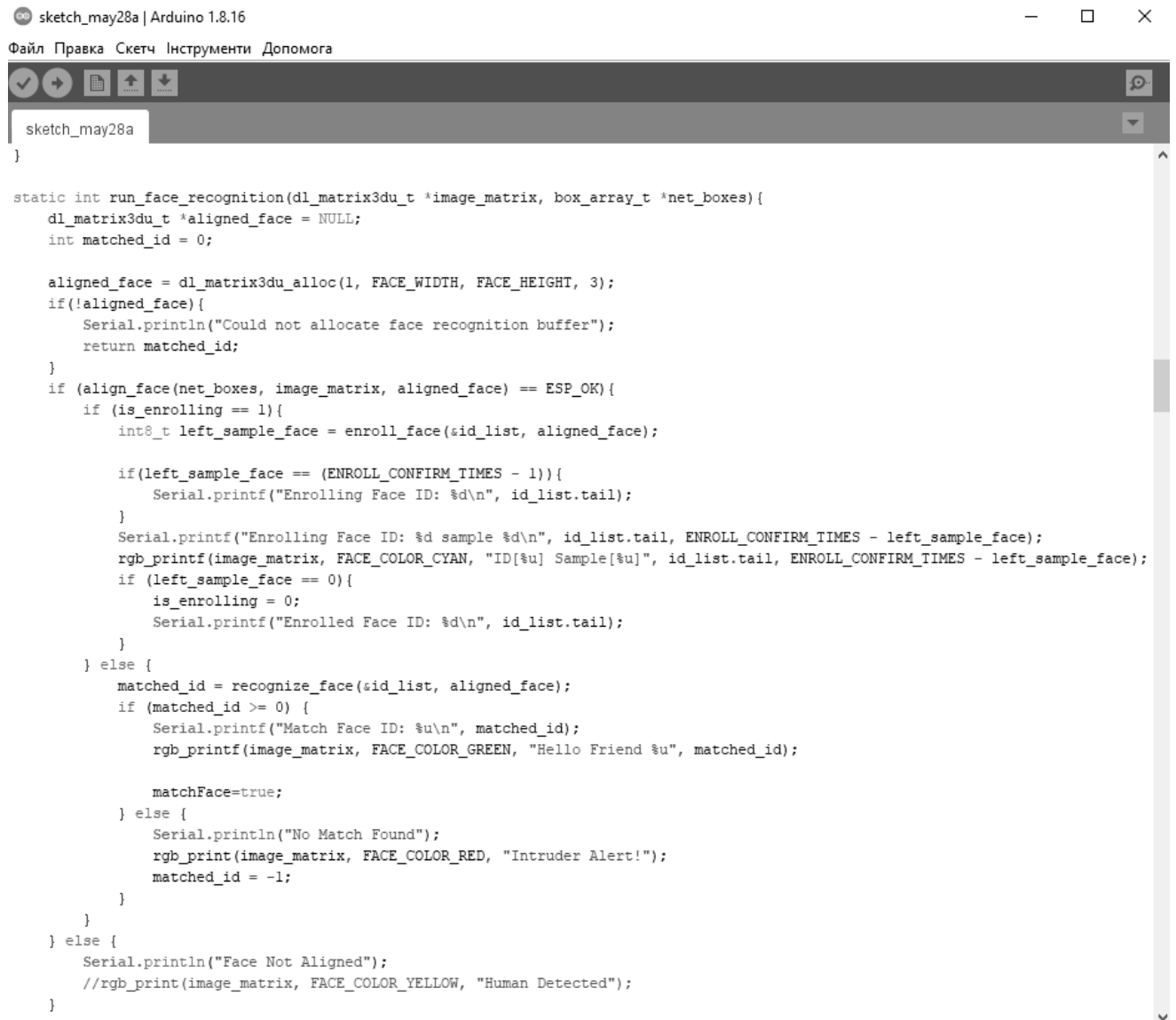


Рисунок 3.12 – Блок-схема функцією `static int run_face_recognition()`

Код функції `static int run_face_recognition()` подано на рисунку 3.13.



```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a
}

static int run_face_recognition(dl_matrix3du_t *image_matrix, box_array_t *net_boxes){
    dl_matrix3du_t *aligned_face = NULL;
    int matched_id = 0;

    aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
    if(!aligned_face){
        Serial.println("Could not allocate face recognition buffer");
        return matched_id;
    }
    if (align_face(net_boxes, image_matrix, aligned_face) == ESP_OK){
        if (is_enrolling == 1){
            int8_t left_sample_face = enroll_face(sid_list, aligned_face);

            if(left_sample_face == (ENROLL_CONFIRM_TIMES - 1)){
                Serial.printf("Enrolling Face ID: %d\n", id_list.tail);
            }
            Serial.printf("Enrolling Face ID: %d sample %d\n", id_list.tail, ENROLL_CONFIRM_TIMES - left_sample_face);
            rgb_printf(image_matrix, FACE_COLOR_CYAN, "ID[%u] Sample[%u]", id_list.tail, ENROLL_CONFIRM_TIMES - left_sample_face);
            if (left_sample_face == 0){
                is_enrolling = 0;
                Serial.printf("Enrolled Face ID: %d\n", id_list.tail);
            }
        } else {
            matched_id = recognize_face(sid_list, aligned_face);
            if (matched_id >= 0) {
                Serial.printf("Match Face ID: %u\n", matched_id);
                rgb_printf(image_matrix, FACE_COLOR_GREEN, "Hello Friend %u", matched_id);

                matchFace=true;
            } else {
                Serial.println("No Match Found");
                rgb_print(image_matrix, FACE_COLOR_RED, "Intruder Alert!");
                matched_id = -1;
            }
        }
    } else {
        Serial.println("Face Not Aligned");
        //rgb_print(image_matrix, FACE_COLOR_YELLOW, "Human Detected");
    }
}
```

Рисунок 3.13 - Здійснення розпізнавання об'єкту на зображенні

Для реалізації можливості декодування відеопотоку (рис. 3.14, рис.3.15), отриманого з модуля камери ESP32CAM, опишемо функцію `jpg_encode_stream()`.

```

sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a

static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t len){
    jpg_chunking_t *j = (jpg_chunking_t *)arg;
    if(!index){
        j->len = 0;
    }
    if(httprd_resp_send_chunk(j->req, (const char *)data, len) != ESP_OK){
        return 0;
    }
    j->len += len;
    return len;
}

```

Рисунок 3.14 – Здійснення декодування відеостріму

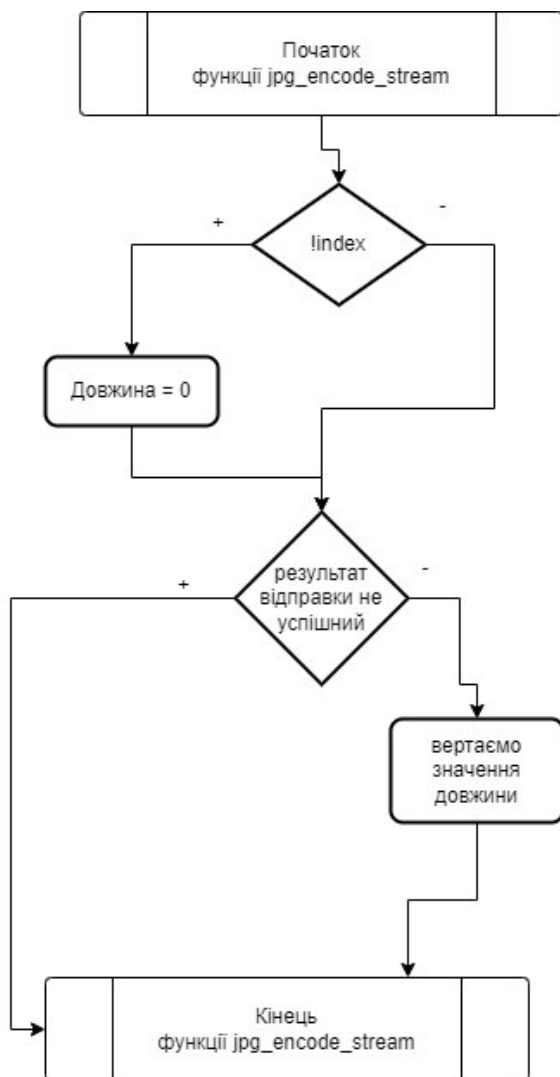


Рисунок 3.15 – Блок-схема функції декодування відеостріму

Важливим етапом системного програмування системи є задання обробника стану та помилок та видача результуючих станів системи користувачу (рис. 3.16). Для цього було описано функцію `esp_err_t capture_handler()`.

Також було розроблено функцію, що обробляє стан та помилки результуючих станів системи щодо обробки відеостріму.

Обробник статусу роботи камери (рис. 3.16, рис.3.17) здійснюється шляхом виконання функції `esp_err_t status_handler()`.

ESP32-CAM буде взаємодіяти з веб-сторінкою (сервером), щоб отримувати й обробляти повідомлення, а також надсилати відповіді на сторінку визначеної адреси (функція `startCameraServer()`), яку подано на рисунку 3.18.

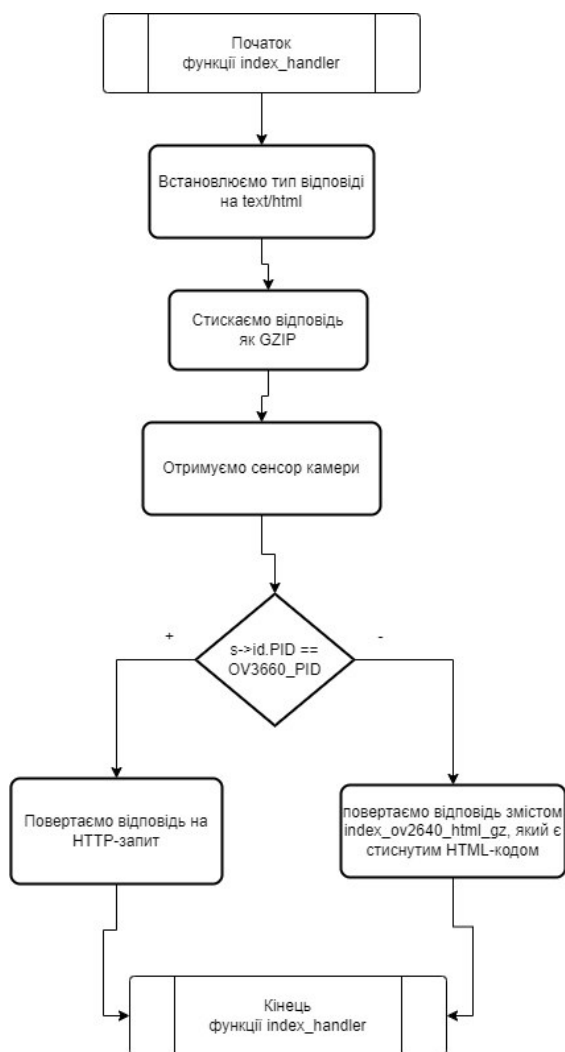


Рисунок 3.16 – блок-схема обробки станів камери

```

static esp_err_t capture_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    int64_t fr_start = esp_timer_get_time();

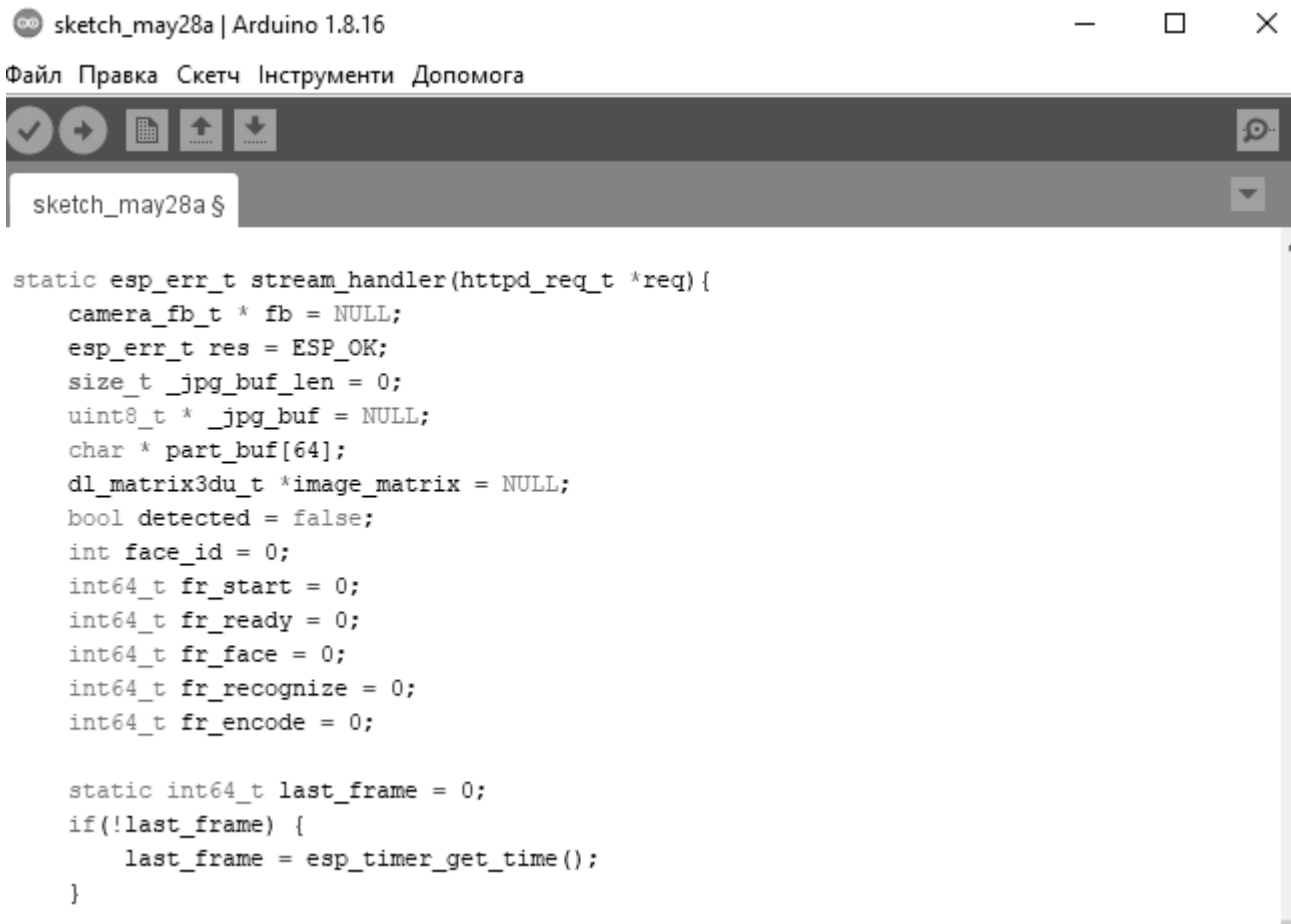
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        httpd_resp_send_500(req);
        return ESP_FAIL;
    }

    httpd_resp_set_type(req, "image/jpeg");
    httpd_resp_set_hdr(req, "Content-Disposition", "inline; filename=capture.jpg");
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");

    size_t out_len, out_width, out_height;
    uint8_t * out_buf;
    bool s;
    bool detected = false;
    int face_id = 0;
    if(!detection_enabled || fb->width > 400){
        size_t fb_len = 0;
        if(fb->format == PIXFORMAT_JPEG){
            fb_len = fb->len;
            res = httpd_resp_send(req, (const char *)fb->buf, fb->len);
        } else {
            jpg_chunking_t jchunk = {req, 0};
            res = frame2jpg_cb(fb, 80, jpg_encode_stream, &jchunk)?ESP_OK:ESP_FAIL;
            httpd_resp_send_chunk(req, NULL, 0);
            fb_len = jchunk.len;
        }
        esp_camera_fb_return(fb);
        int64_t fr_end = esp_timer_get_time();
        Serial.printf("JPG: %uB %ums\n", (uint32_t)(fb_len), (uint32_t)((fr_end - fr_start)/1000));
        return res;
    }
}

```

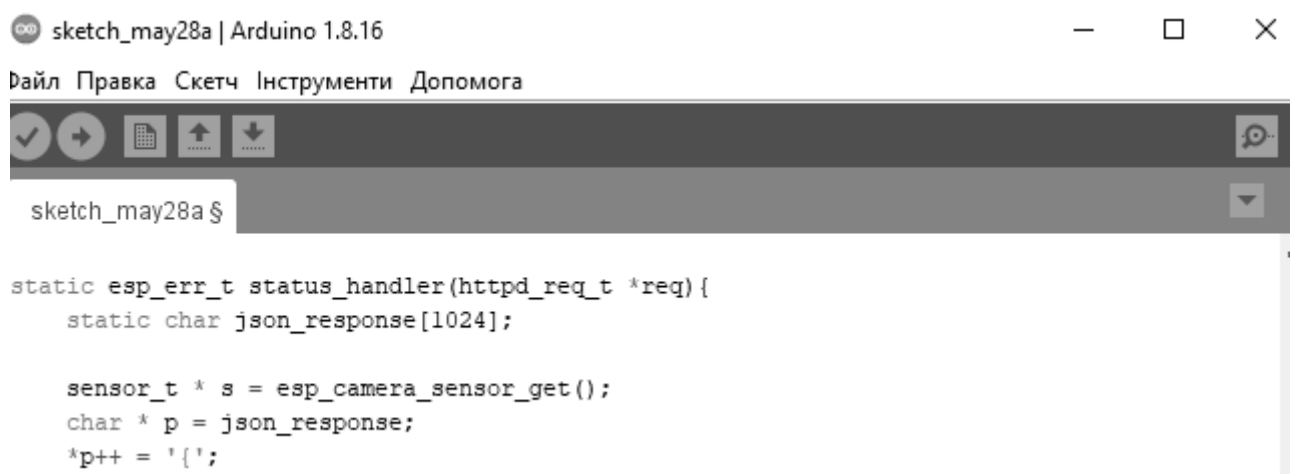
Рисунок 3.17 - Функція-обробник стану та помилок та видача результируючих станів системи щодо обробки зображень



```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a $
static esp_err_t stream_handler(httpd_req_t *req) {
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];
    dl_matrix3du_t *image_matrix = NULL;
    bool detected = false;
    int face_id = 0;
    int64_t fr_start = 0;
    int64_t fr_ready = 0;
    int64_t fr_face = 0;
    int64_t fr_recognize = 0;
    int64_t fr_encode = 0;

    static int64_t last_frame = 0;
    if(!last_frame) {
        last_frame = esp_timer_get_time();
    }
}
```

Рисунок 3.12 – Функція-обробник стану та помилок та видача результуючих станів системи щодо обробки відеостріму



```
sketch_may28a | Arduino 1.8.16
Файл Правка Скетч Інструменти Допомога
sketch_may28a $
static esp_err_t status_handler(httpd_req_t *req){
    static char json_response[1024];

    sensor_t * s = esp_camera_sensor_get();
    char * p = json_response;
    *p++ = '{';
}
```

Рисунок 3.18 – Обробник статусу роботи камери

```
sketch_may28a $
}

void startCameraServer(){
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();

  httpd_uri_t index_uri = {
    .uri      = "/",
    .method   = HTTP_GET,
    .handler  = index_handler,
    .user_ctx = NULL
  };

  httpd_uri_t status_uri = {
    .uri      = "/status",
    .method   = HTTP_GET,
    .handler  = status_handler,
    .user_ctx = NULL
  };

  httpd_uri_t cmd_uri = {
    .uri      = "/control",
    .method   = HTTP_GET,
    .handler  = cmd_handler,
    .user_ctx = NULL
  };

  httpd_uri_t capture_uri = {
    .uri      = "/capture",
    .method   = HTTP_GET,
    .handler  = capture_handler,
    .user_ctx = NULL
  };

  httpd_uri_t stream_uri = {
    .uri      = "/stream",
    .method   = HTTP_GET,
    .handler  = stream_handler,
    .user_ctx = NULL
  };
}
```

Рисунок 3.19 - Взаємодія з веб-сторінкою (сервером)

За передачу даних засобами WI-FI відповідає системний код, поданий на рисунку 3.20.

```

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}

```

Рисунок 3.21 – Системний код підтримки роботи WI-FI

Безпосередній запуск пристрою здійснюється засобами старту внутрішньої функції loop() (рис.3.22).

```

void loop() {
  if(matchFace==true && activateRelay==false)
  {
    activateRelay=true;
    digitalWrite(Relay,HIGH);
    digitalWrite(Green,HIGH);
    digitalWrite(Red,LOW);
    prevMillis=millis();
  }
  if (activateRelay == true && millis()-prevMillis > interval)
  {
    activateRelay=false;
    matchFace=false;
    digitalWrite(Relay,LOW);
    digitalWrite(Green,LOW);
    digitalWrite(Red,HIGH);
  }
}

```

Рисунок 3.22. – Функція loop()

Зм.	Арк.	№докум.	Підпис	Дата

### 3.2 Апаратна реалізація системи аварійного керування замком з використанням мікроконтролера ESP32-CAM

Для апаратної реалізації системи аварійного керування замком з використанням мікроконтролера ESP32-CAM було застосовано середовище проєктування та візуалізації Fritzing.

З цією метою було включено режим вкладки перегляду Breadboard, де на борді було додано усі необхідні компоненти:

1. Безпосередньо модуль ESP32 CAM
2. Модуль реле (Relay Module).
3. Замок (Solenoid Lock).
4. Лампочка спрацювання/стану (LED).
5. Breadboard.
6. Акумулятор номіналом 12v (power Supply / Battery).
7. Регулятор (Regulator).
8. 100uf 16V Capacitor.
9. Проводи з'єднання (wires).

Спроектована схема подана на рисунку 3.23.

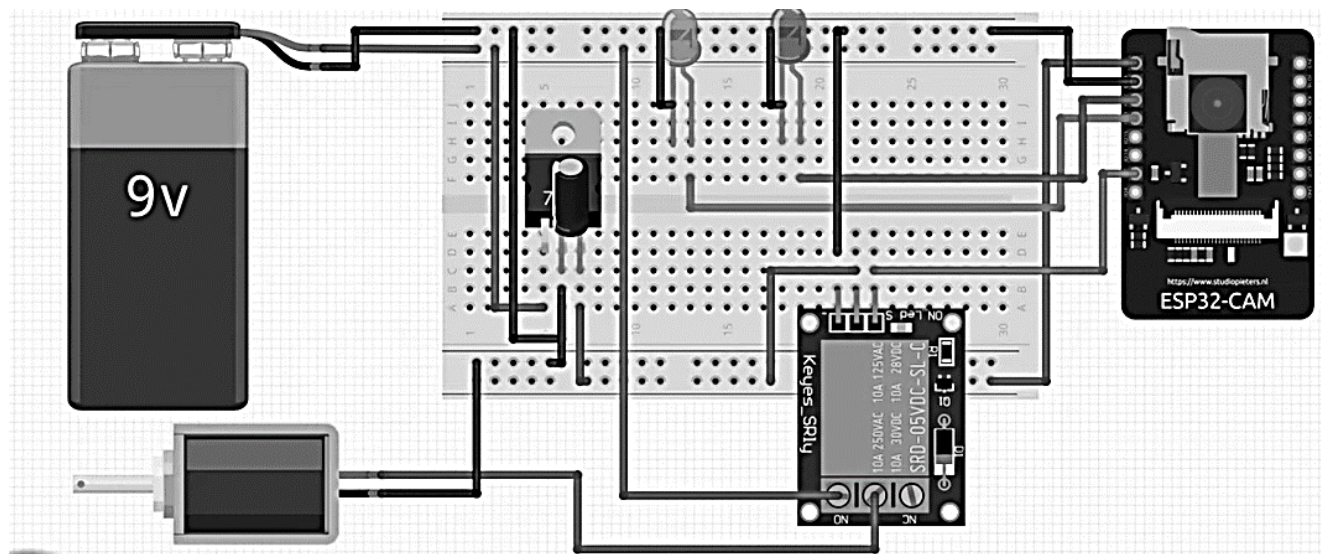


Рисунок 3.23 - Спроектована схема

Збірка системи починається з поєднання модуля мікроконтролера з програматором для програмування (рис. 3.24).

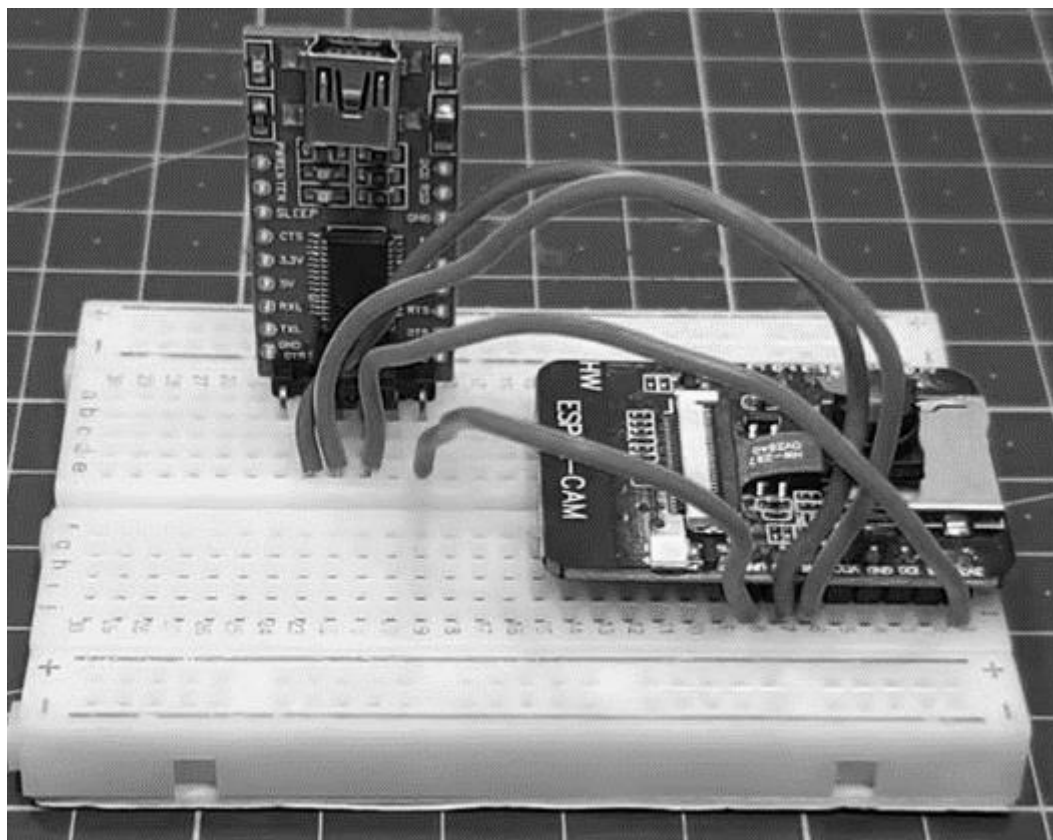


Рисунок 3.24 – Мікроконтролер з програматором

Монтажний прототип системи подбано на рисунку 3.25.

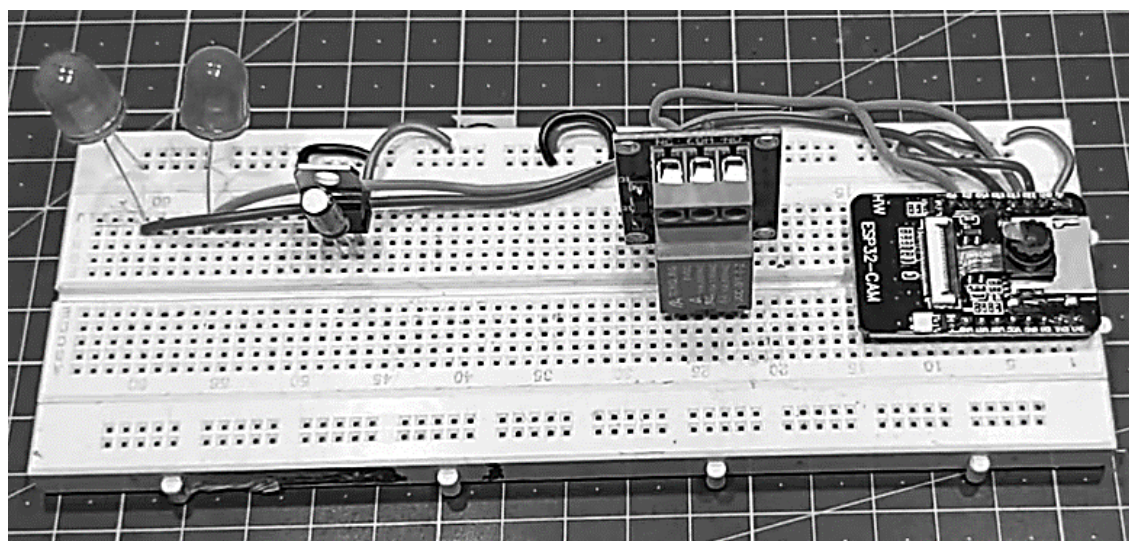


Рисунок 3.25 - Монтажний прототип системи

Зм.	Арк.	№докум.	Підпис	Дата

Зібраний макет системи подано на рисунку 3.26.

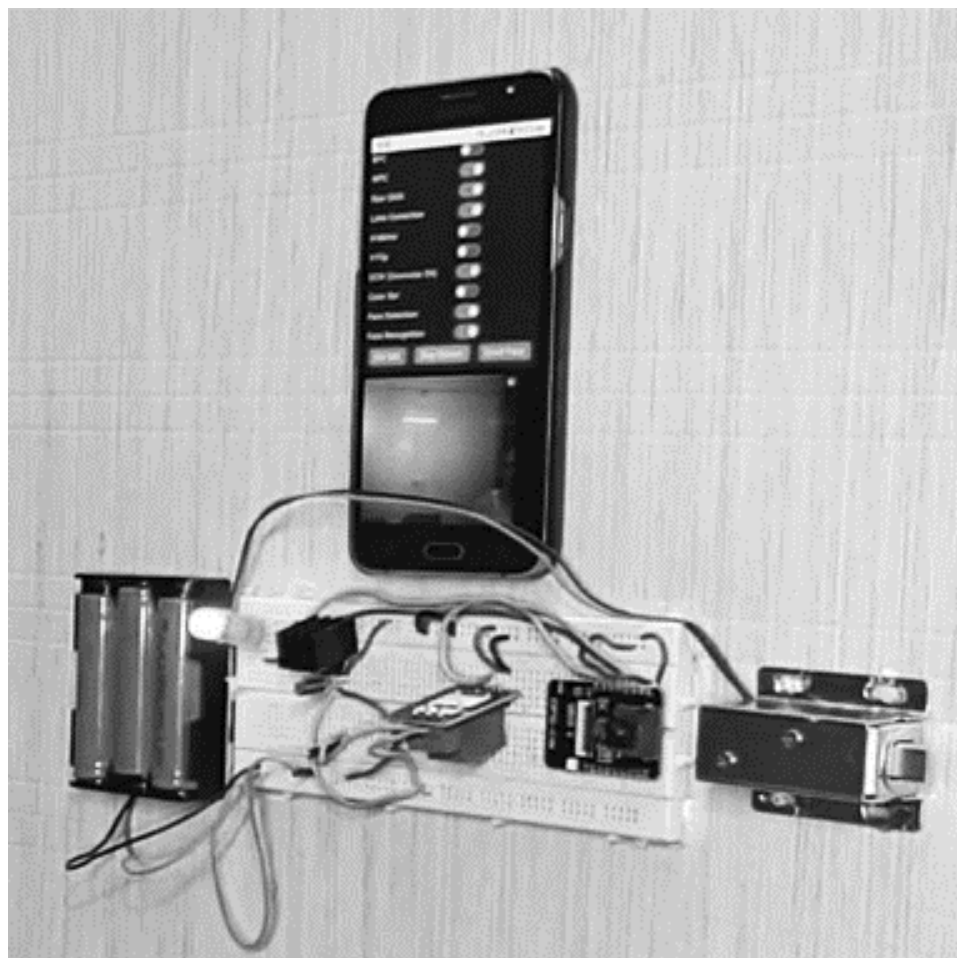


Рисунок 3.26 – Зібраний макет системи подано

Після запуску системного програмного забезпечення, описаного в п. 3.1 здійснюється запуск апаратних компонентів , а також запуск локального сервера, який видає результати роботи камери.

Запущений сервер також містить вкладку з множиною налаштувань роботи камери (рисунок 3.27).

Зм.	Арк.	№докум.	Підпис	Дата

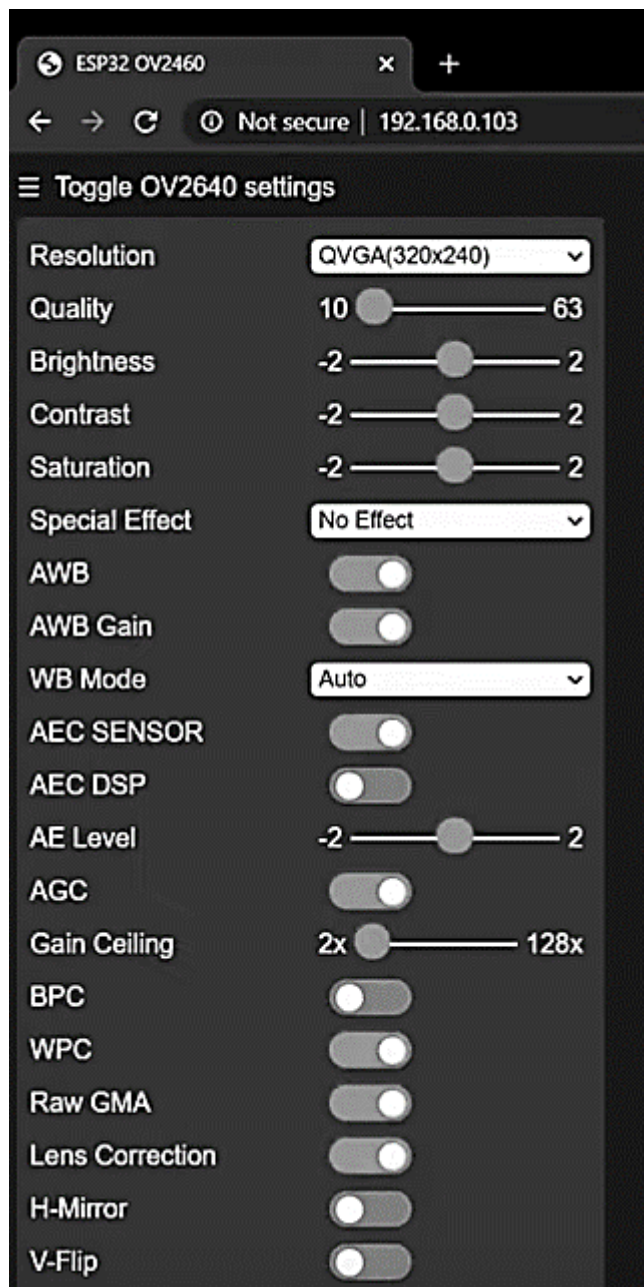


Рисунок 3.27 - Запущений сервер

Вбудовані налаштування, зокрема, дозволяють здійснити такі:

1. Роздільну здатність.
2. Яскравість зображення.
3. Додаткові ефекти.
4. Баланс білого.
5. Raw-режим.

Зм.	Арк.	№докум.	Підпис	Дата

Окремою важливою компонентою налаштувань є включення функцій виявлення та розпізнавання обличчя (рисунок 3.28).

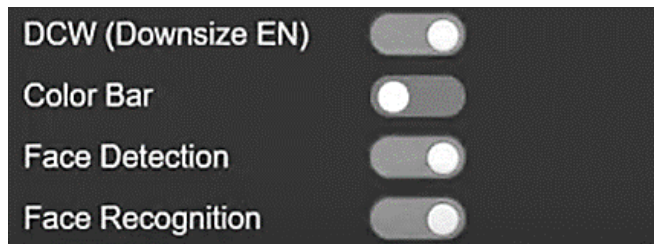


Рисунок 3.28 – Налаштувань функцій виявлення та розпізнавання обличчя в системі

Таким чином після здійснення налаштувань і використавши веб-браузер операційної системи, можна отримати відеопотік за адресою 192.168.0.103 (рисунок 3.29).

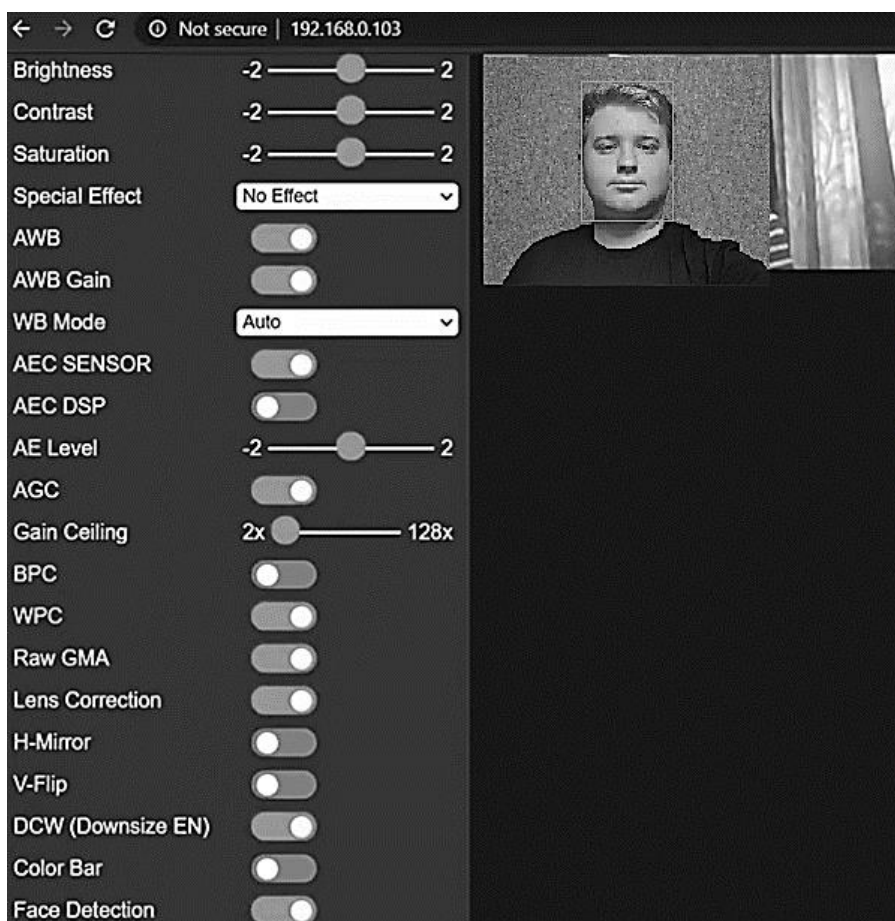


Рисунок 3.29 – Отриманий відеопотік

Зм.	Арк.	№докум.	Підпис	Дата

### 3.3 Висновки

В розділі описано програмна та апаратна реалізація системи аварійного керування замком з використанням мікроконтролера ESP32-CAM

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		58

## ВИСНОВКИ

В результаті кваліфікаційної роботи було розроблено мікроконтролерну систему аварійного керування замком з використанням мікроконтролера ESP32-CAM.

В першому розділі здійснено аналіз предметної області і виявлення наявних проблем і завдань, проведено порівняльний аналіз переваг та недоліків існуючих рішень, описано підходи до вирішення задачі за темою дослідження. Також в розділ зроблено висновки та зроблено постановку задачі.

В другому розділі описано апаратне середовище реалізації та опис ESP32-CAM. Також принцип роботи контактного магнітного замка.

Вибір методів і середовища програмування програмного забезпечення та середовища проектування.

В третьому розділі описано програмна та апаратна реалізація системи аварійного керування замком з використанням мікроконтролера ESP32-CAM

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		59

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ESP32-CAM Remote Controlled Car Robot Web Server, March 13, 2021. URL: <https://randomnerdtutorials.com/esp32-cam-car-robot-web-server/>. (дата звернення: 20.03.2023).
2. Посилання на проєкт Точка доступу на основі ESP32-CAM URL: <https://randomnerdtutorials.com/esp32-cam-access-point-ap-web-server/>. (дата звернення: 29.03.2023).
3. Deborah Lupton. How do data come to matter? Living and becoming with personal data. First Published July 5, Volume: 5 issue: 2, 2018, p. 11. URL: <https://journals.sagepub.com/doi/full/10.1177/2053951718786314>.
4. Agus Kurniawan. Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32. March 30, 2019, p. 252.
5. Panduman F., Fridelin Y. Y., Sukaridhoto S., and Anang T. A Survey of IoT Platform Comparison for Building Cyber- Physical System Architecture. *In - 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. 2019. Pp. 238-243.
6. Проєкт Arduino Project HUB. Discord Security Camera with an ESP32, March 13, 2021. URL: [https://create.arduino.cc/projecthub/WillMakesTV/discord-security-camera-with-an-esp32-7c4621?ref=search&ref\\_id=esp32-cam&offset=0](https://create.arduino.cc/projecthub/WillMakesTV/discord-security-camera-with-an-esp32-7c4621?ref=search&ref_id=esp32-cam&offset=0). (дата звернення: 20.03.2023).
7. Посилання на проєкт Color Detection and Tracking URL: <https://how2electronics.com/color-detection-tracking-with-esp32-cam-opencv/>. (дата звернення: 25.03.2023).
8. Yaqoob I., Ahmed E., Hashem I. A. T., Ahmed A. I. A., Gani A., Imran M., Guizani M. Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. *IEEE wireless communications*. 2017. Pp. 10-16.
9. Munadi R., Rakhman A., Perdana D. Smart Garage Implementation and Design Using Whatsapp Communication Media. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2018. Pp. 1107-1113.

					КВРКІП 2001141.20.17.01 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		60

10. Sawidin S., Pongoh D. S., Ramschie A. A. S. Design of Smart Home Control System Based on Android. *In 2018 International Conference on Applied Science and Technology (iCAST)*. 2018. Pp. 165-170.
11. Nitin N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *In 2017 IEEE international systems engineering symposium (ISSE)*. 2017. Pp. 1-7.
12. Sohan S.M., Maurer F., Anslow C., Robillard M. P. A Study of the Effectiveness of Usage Examples in REST API Documentation. 2017. Pp. 53–61.
13. ShopAbunda. URL: <https://www.shopabunda.com> (дата звернення 14.05.2022).
14. Neil Cameron. Electronics Projects with the ESP8266 and ESP32: Building Web Pages, Applications, and WiFi Enabled Devices. December 18, 2020, p. 723.
15. Tanweer A. IoT-Fog: A Communication Framework using Blockchain in the Internet of Things. *International Journal of Recent Technology and Engineering*. 2019. Vol. 7. Pp. 11-15.
16. D'Alessandro D., Gunderson W., Staten E., Donastien Y. K., Rodriguez P., Bailey R. Integrating Modularity for Mass Customization of IoT Wireless Sensor Systems. *In 2021 Systems and Information Engineering Design Symposium (SIEDS)*. 2021. Pp. 1-5.
17. Hashim H., Salihudin S. F. B., Saad P. S. M. Development of IoT Based Healthcare Monitoring System. *In 2022 IEEE International Conference in Power Engineering Application (ICPEA)*. 2022. Pp. 1-5.
18. Xiaomi.UA. URL: <https://www.xiaomi.ua> (дата звернення 14.05.2022).
19. Zhao W., Yi L. Research on the evolution of the innovation ecosystem of the Internet of Things: A case study of Xiaomi (China). *Procedia Computer Science*, 2022. Pp. 56-62.
20. Song Y., Luximon Y., Leong B. D., Qin Z. The e-commerce performance of internet of things (IoT) in disruptive innovation: Case of Xiaomi. *In Proceedings of the 2019 3rd International Conference on Software and e-Business*. 2019. Pp. 188-192.

21. Ammar M., Russello G., Crispo B. Internet of Things: A survey on the security of IoT frameworks. *J. Inf. Secur.* 2018. Pp. 8-27.
22. Pamungkas H., Akbar S. R., Ichsan M. H. H. Implementasi Sistem Pengenalan Perangkat dan Layanan Sensor dan Aktuator pada Rumah Cerdas Berbasis Arsitektur Publish-Subscribe. *Jurnal Pengembangan Teknologi Informasi & Ilmu Komputer.* 2018. Pp. 2522-2531.
23. Pratama R. P. Implementation and testing of the ESP8266 module with the MQTT-Dash android application on the MQTT network. *Asian Scientific Journal of Information Technology.* 2018. Pp. 157-164.
24. Jabbar W. A., Shang H. K., Hamid S. N., Almohammed A. A., Ramli R. M., Ali M. A. IoT-BBMS: Internet of Things-based baby monitoring system for smart cradle. 2019. Pp. 791-793.
25. Durani H., Sheth M., Vaghasia M., Kotech S. Smart automated home application using IoT with Blynk app. *2018 Second international conference on inventive communication and computational technologies.* 2018. Pp. 393-397.
26. Karuppusamy P. A sensor based IoT monitoring system for electrical devices using Blynk framework. *Journal of Electronics and Informatics.* 2020. Pp. 182-187.
27. Nduka A., Samuel J., Elango S., Divakaran S., Umar U., SenthilPrabha R. Internet of things based remote health monitoring system using arduino. *In 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),* 2019. Pp. 572-576.
28. Mabrouki J., Azrour M., Dhiba D., Farhaoui Y., & El Hajjaji S. IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Mining and Analytics.* 2021. Pp. 25-32.
29. Fritzing. URL: <https://fritzing.org> (дата звернення: 14.05.2022).
30. Twins Chip. URL: <https://www.twinschip.com> (дата звернення: 14.05.2022).

31. Andrews J., Kowsika M., Vakil A., Li J. A motion induced passive infrared (PIR) sensor for stationary human occupancy detection. *In 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2020. Pp. 1295-1304.
32. Surantha N., Wicaksono W. R. Design of smart home security system using object recognition and PIR sensor. *Procedia computer science*. 2018. Pp. 465-472.
33. Gami H. Movement direction and distance classification using a single PIR sensor. *IEEE sensors letters*. 2017. Pp. 1-4.
34. Patnaikuni D. R. P. A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development. *International Journal of Advanced Research in Computer Science*. 2017. Pp. 2350-2352.
35. Jahary M.M, Ichsan M.H.H., Primananda R. Implementasi Protokol Universal Plug and Play (UPnP) Pada Sensor dan Aktuator Untuk Otomasi Lampu. *Universitas Brawijaya*. 2019. Pp. 101-114.
36. Pusparini D., Ichsan M.H.H. Implementasi Low Power System untuk Pengambilan dan Pengiriman Data Berdasarkan Kehadiran Manusia. *Universitas Brawijaya*. 2019. Pp. 32-35.
37. Приціл. URL: <https://pricel.in.ua> (дата звернення 14.05.2022).
38. Kashimoto Y., Fujiwara M., Fujimoto M., Suwa H., Arakawa Y., Yasumoto K. ALPAS: Analog-PIR-sensor-based activity recognition system in smarthome. *In 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. 2017. Pp. 880-885.
39. Verma M., Kaler R. S., Singh M. Sensitivity enhancement of Passive Infrared (PIR) sensor for motion detection. 2021. Vol. 244. P. 167.
40. Kaur B., Pateriya P. K., Rai M. K. An illustration of making a home automation system using raspberry Pi and PIR sensor. *In 2018 International Conference on Intelligent Circuits and Systems (ICICS)*. 2018. Pp. 439-444.
41. Kumar P. Design and implementation of Smart Home control using LabVIEW. *In 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. 2018. Pp. 10-12.

42. Kim S. H., Moon H. J., Yoon Y. R. Improved occupancy detection accuracy using PIR and door sensors for a smart thermostat. *In Proceedings of the 15th IBPSA Conference*. 2017. Pp. 253-258.
43. Aldawira C. R., Putra H. W., Hanafiah N., Surjarwo S., Wibisurya A. Door security system for home monitoring based on ESP32. *Procedia Computer Science*. 2019. Vol. 157. Pp. 673-682.
44. Chandra M. R., Kumar B. V., Babu B. S. IoT enabled home with smart security. *In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. 2017. Pp. 1193-1197.
45. Han J., Chung A. J., Sinha M. K., Harishankar M., Pan S., Noh H. Y., Tague P. Do you feel what I hear? Enabling autonomous IoT device pairing using different sensor types. *In 2018 IEEE Symposium on Security and Privacy (SP)*. 2018. Pp. 836-852.
46. МТА. URL: <https://mta.ua> (дата звернення 14.05.2022).
47. Jolles J. W. Broad-scale applications of the Raspberry Pi: A review and guide for biologists. *Methods in Ecology and Evolution*. 2021. Pp. 1562-1579.
48. Hasan M., Anik M. H., Chowdhury S., Chowdhury S. A., Bilash T. I., Islam S. Low-cost Appliance Switching Circuit for Discarding Technical Issues of Microcontroller Controlled Smart Home. *International Journal of Sensors and Sensor Networks*. 2019. Pp. 16-22.
49. Gota, D. I., Puscasiu, A., Fanca, A., Miclea, L., & Valean, H. Smart home automation system using Arduino microcontrollers. *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*. 2020. Pp. 1-7.
50. Babiuch M., Postulka J. Smart Home Monitoring System Using ESP32 Microcontrollers. *In Internet of Things, IntechOpen*. 2020. Pp. 81-85.
51. Syafa'ah L., Minarno A. E., Sumadi F. D. S., Rahayu D. A. P. ESP8266 for control and monitoring in smart home application. *In 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*. 2019. Pp. 123-128.

52. Alfarsi G., Jabbar J., Tawafak R. M., Malik S. I., Alsidiri A., Alsinani M. Using Cisco Packet Tracer to simulate smart home. *International Journal of Engineering Research & Technology (IJERT)*. 2019. Pp. 670-674.
53. Cuno Pfister. Getting Started with the Internet of Things: Connecting Sensors And Microcontrollers To The Cloud. June 21, 2011, p. 194.
54. Sayak Boral. Article: What is ESP32 and Why Is It Best for IoT Projects? May 25, 2020. URL: <https://www.iottechrends.com/what-is-esp32/>.
55. Douglas Comer. Internetworking with TCP/IP Volume One. 2013, 744.
56. Dogan Ibrahim, Ahmet Ibrahim. The Official ESP32 Book. 2017, 286.
57. Marco Schwartz. Internet of Things with ESP8266. 2016, 226p.
58. Офіційний сайт Arduino. URL: <https://www.arduino.cc>. (дата звернення: 20.03.2023).
59. Alam T., Benaïda M. CICS: Cloud–Internet Communication Security Framework for the Internet of Smart Devices. *International Journal of Interactive Mobile Technologies*, 2018. No. 1. Pp. 74-84.

# Додаток А

## (обов'язково)

### Копія креслення “Схеми з'єднань пристрою”

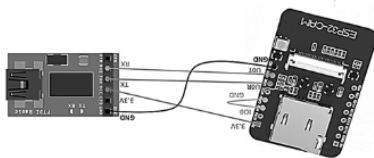
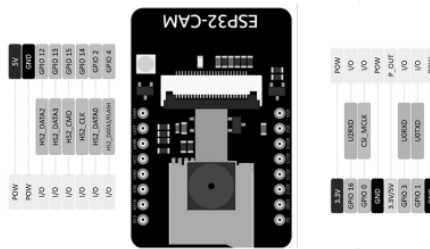


Схема приєднання модуля до програматора



Піни мікроконтролера

КВРКІ 2001141.20.17.01.Е8	
Мікроконтролер системи управління пристроєм ESP32-CAM	ХНУ КІЗ-20-1
Схема	Версія 2
Історія змін	Версія 3
Зміни	Версія 4
Зміни	Версія 5
Зміни	Версія 6
Зміни	Версія 7
Зміни	Версія 8
Зміни	Версія 9
Зміни	Версія 10
Зміни	Версія 11
Зміни	Версія 12
Зміни	Версія 13
Зміни	Версія 14
Зміни	Версія 15
Зміни	Версія 16
Зміни	Версія 17
Зміни	Версія 18
Зміни	Версія 19
Зміни	Версія 20
Зміни	Версія 21
Зміни	Версія 22
Зміни	Версія 23
Зміни	Версія 24
Зміни	Версія 25
Зміни	Версія 26
Зміни	Версія 27
Зміни	Версія 28
Зміни	Версія 29
Зміни	Версія 30
Зміни	Версія 31
Зміни	Версія 32
Зміни	Версія 33
Зміни	Версія 34
Зміни	Версія 35
Зміни	Версія 36
Зміни	Версія 37
Зміни	Версія 38
Зміни	Версія 39
Зміни	Версія 40
Зміни	Версія 41
Зміни	Версія 42
Зміни	Версія 43
Зміни	Версія 44
Зміни	Версія 45
Зміни	Версія 46
Зміни	Версія 47
Зміни	Версія 48
Зміни	Версія 49
Зміни	Версія 50

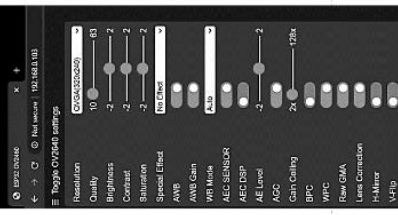

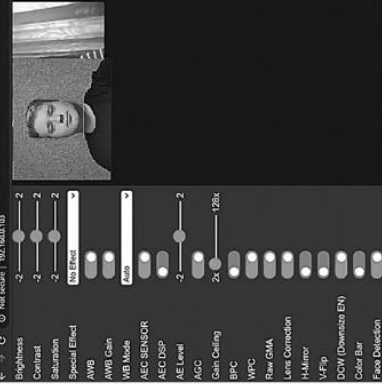
КВРКІ 2001141.20.17.01.Е8



# Додаток В

## (обов'язково)

Копія креслення “Інтерфейси програмно-апаратного засобу”

<p>КвРКІ 2001141.20.17.01.Е8</p>	<p><b>Запущений сервер</b></p> 	<p><b>Налаштувань функцій виявлення та розпізнавання обличчя в системі</b></p> 	<p><b>Отриманий відеопотік</b></p> 
----------------------------------	--	---	--

КвРКІ 2001141.20.17.01.Е8			
ЛП	Мова	Матриця	
Н			
Мікропроцесорна система управління ЕРП/СДМ			
Інтерфейс програмно-апаратного засобу			
Апарат 2			Апарат 2
ХНУ №С-20-1			

## Додаток Г

Лістинг коду для моделювання в Packet Tracer

```
from gpio import *
from time import *
from ioeclient import *

def main():

    pinMode(0, IN)

    pinMode(1, OUT)

    pinMode(2, OUT)

    pinMode(3, OUT)

    pinMode(4, OUT)

    pinMode(5, IN)

    # Convert from the old range to the new range.

    IoEClient.setup({

        "type": "Sensors",

        "states": [{

            "name" : "Temperature",

            "type" : "number",

            "unit": "C",

            "imperialUnit": "F",

            "toImperialConversion": "x*1.8+32",

            "toMetricConversion": "(x-32)/1.8",

            "decimalDigits": 1 ,}],
```

```

{"name" : "Input motion sensor",
  "type" : "bool",
},
{"name" : "Output motion sensor",
  "type" : "bool",
}]
});

```

while True:

```

value = analogRead(A0)

temperature = (((value - 0) * (100 - -100)) / (1023 - 0)) + -100

motion = digitalRead(0);

motionout = digitalRead(5);

IoEClient.reportStates((temperature, motion, motionout))

if (digitalRead(0) == HIGH)and (digitalRead(5) == LOW):

    digitalWrite(2, HIGH)

    customWrite(1, '2')

    digitalWrite(0, HIGH)

    digitalWrite(5, LOW)

    if (temperature > 20):

        digitalWrite(3, HIGH)

        customWrite(3, '1')

        digitalWrite(4, LOW)

        customWrite(4, '0')

```

```
elif(temperature < 20):
```

```
    digitalWrite(3, LOW)
```

```
    digitalWrite(3, '0')
```

```
    digitalWrite(4, HIGH)
```

```
    digitalWrite(4, '1')
```

```
elif digitalRead(5) == HIGH and (digitalRead(0) == LOW):
```

```
    digitalWrite(2, LOW)
```

```
    digitalWrite(1, '0')
```

```
    digitalWrite(0, LOW)
```

```
    digitalWrite(5, HIGH)
```

```
    digitalWrite(3, LOW)
```

```
    digitalWrite(3, '0')
```

```
    digitalWrite(4, LOW)
```

```
    digitalWrite(4, '0')
```

```
    delay(500)
```

```
if __name__ == "__main__":
```

```
    main()
```

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Самойлюк Максим Ігорович

Тема: Мікроконтролерна система аварійного керування замком з використанням мікроконтролера ESP32-CAM

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 55

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення мікроконтролерної системи аварійного керування замком з використанням мікроконтролера 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 результаті кваліфікаційної роботи було розроблено мікроконтролерну систему аварійного керування замком з використанням мікроконтролера ESP32-CAM.

В першому розділі здійснено аналіз предметної області і виявлення наявних проблем і завдань, проведено порівняльний аналіз переваг та недоліків існуючих рішень, описано підходи до вирішення задачі за темою дослідження. Також в розділі зроблено висновки та зроблено постановку задачі.

В другому розділі описано апаратне середовище реалізації та опис ESP32-CAM. Також принцип роботи контактного магнітного замка.

Вибір методів і середовища програмування програмного забезпечення та середовища проектування.

В третьому розділі описано програмна та апаратна реалізація системи аварійного керування замком з використанням мікроконтролера ESP32-CAM.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага моделюванню схем в середовищі Fritzing.

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

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

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: задовільно, 3.5/D

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

доцент кафедри Інженерії програмного забезпечення, к.т.н., доц. Гурман І.В.

---

“30” травня 2023 р.

І.В. Гурман (підпис)

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

Самоїлюка Максима Ігоровича

ШБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-20-1

### 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 квітня 2023 року



Ім'я користувача:  
Кафедра КІ

ID перевірки:  
1015342760

Дата перевірки:  
31.05.2023 12:31:01 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
31.05.2023 12:32:57 EEST

ID користувача:  
100005591

Назва документа: **Самойлюк\_Мікроконтролерна система аварійного керування замком з використанням мік-р...**  
Кількість сторінок: 66 Кількість слів: 8875 Кількість символів: 72929 Розмір файлу: 3.64 MB ID файлу: 1015011293

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

## 25% Схожість

Найбільша схожість: 15.5% з Інтернет-джерелом (<http://elar.khmnu.edu.ua/jspui/bitstream/123456789/12003/1/%d0%91>).

22% Джерела з Інтернету

885

Сторінка 68

9.1% Джерела з Бібліотеки

81

Сторінка 69

## 0.55% Цитат

Цитати

3

Сторінка 70

Не знайдено жодних посилань

## 0% Вилучень

Немає вилучених джерел

## Модифікації

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

Підозріле форматування

23  
сторінки

# Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилко в документах: 12%

ID: 114367 Назва: БКР Мікроконтролерна система аварійного керування замком з використанням мік-роконтролера ESP32-SAM Додано в БД: 2023-05-31 Автора: М. І. Самойлюк Керівники: О. С. Засорнов Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	44602	429	1757 (4%)	19 (4%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

**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 аварійного керування замком з використанням мікроконтролера ESP32-CAM

Автор: Самойлюк Максим Ігорович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

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

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, які викладені в розділах аналізу наявних аналогів та прототипів, не пов'язані безпосередньо з моїм дослідженням і не стосуються отриманих результатів;
- 2) всі цитовані джерела є фрагментарними або мають належним чином оформлені посилання;
- 3) окремі співпадіння, виявлені у тексті, є загальноприйнятими фразами або виразами, оскільки система посилається на збіг з певними джерелами лише на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності програмного коду, які є вхідними даними до програмно-технічного засобу і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) усі виявлені системою ознаки модифікації тексту включають комбінації латинських символів з україномовними скороченнями індексів у формулах. Однак ці модифікації не впливають на зміст тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 22% і адресується до 885 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

О. С. Засорнов

С. М. Лисенко

Т. О. Говорущенко