

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ
УПРАВЛІННЯ ІР КАМЕРОЮ ЗА
ДОПОМОГОЮ ESP32»

Виконав: студент 2 курсу, групи 8.1212-іпз-1
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення
(назва освітньої програми)

О.О. Грицюк

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
доцент, к.т.н., Мухін В.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри комп'ютерних наук,
доцент, к.т.н., Матвіїшина Н.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.

(підпис)

“ _____ ” _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Грицюку Олександр Олександровичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка мобільного додатку управління IP камерою
за допомогою ESP32

керівник роботи Мухін Віталій Вікторович, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 01 » травня 2023 року № 642-с

2. Строк подання студентом роботи 27.11.2023 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Реалізація системи.

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

презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 05.05.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану виконання кваліфікаційної роботи магістра.	15.05.2023	
2.	Збір вихідних даних та аналіз предметної області.	20.06.2023	
3.	Обробка методичних та теоретичних джерел.	14.07.2023	
4.	Специфікація вимог до системи. Робота над першим розділом.	25.08.2023	
5.	Моделювання та проєктування системи. Робота над другим розділом.	29.09.2023	
6.	Реалізація та тестування системи. Робота над третім розділом.	27.10.2023	
7.	Оформлення та нормоконтроль кваліфікаційної роботи магістра.	20.11.2023	
8.	Захист кваліфікаційної роботи.	14.12.2023	

Студент _____
(підпис)

О.О. Грицюк _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

В.В. Мухін _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка мобільного додатку управління IP камерою за допомогою ESP32»: 45 с., 21 рис., 2 табл., 18 джерел, 1 додаток.

МОБІЛЬНА РОЗРОБКА, ESP, ESP32-CAM, FLASK, IP КАМЕРА, JAVASCRIPT, NODEMCU ESP32, REACT NATIVE.

Об'єкт дослідження – плата ESP32-CAM.

Мета роботи: Розробка мобільного додатку управління IP камерою за допомогою ESP32.

Метод дослідження – методи збору та аналізу вимог до програмного забезпечення, методи моделювання, проектування, конструювання та тестування програмного забезпечення.

У даному дослідженні буде проаналізовано плату ESP32 та її програмування за допомогою Arduino. Для допоміжних завдань використовуватиметься плата NodeMcu Esp32. Створений сервер з використанням Flask буде відповідальний за отримання та передачу інформації з датчика руху, а також мобільний додаток, розроблений з використанням React Native, відобразить статус датчика руху та за бажанням транслюватиме відео з камери. Усі ці компоненти є зменшеною модифікацією ключових елементів сучасних модулів систем розумного будинку, галузь, яка швидко розвивається та проникає в різні галузі бізнесу. Таким чином, тема є дуже актуальною та цікавою.

SUMMARY

Master's qualifying paper «Development of a Mobile Application for IP Camera Controlling Using ESP32»: 45 pages, 21 figures, 2 tables, 18 references, 1 supplement.

MOBILE DEVELOPMENT, ESP, ESP32-CAM, FLASK, IP KAMEPA, JAVASCRIPT, NODEMCU ESP32, REACT NATIVE.

The subject of investigation is the ESP32-CAM board. The objective of this project is to develop a mobile application for managing an IP camera through the use of the ESP32. The research methodology includes techniques for gathering and analyzing software requirements, as well as methods for modeling, designing, and testing the software.

This study aims to analyze the ESP32 board and its programming capabilities through the use of Arduino. The NodeMcu Esp32 board will be utilized for auxiliary tasks. The server, developed with Flask, will be responsible for receiving and transmitting data from the motion sensor. Additionally, the mobile application, created with React Native, will display the status of the motion sensor and potentially stream video from the camera. These components are reduced versions of essential components found in contemporary smart home systems, a rapidly growing industry that is making its way into various business sectors. Therefore, this topic is highly relevant and intriguing.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	7
1 Огляд мобільних додатків управління IP камерою за допомогою ESP32	9
2 Моделювання та проєктування апаратно-програмного модулю.....	15
2.1 Системи відеоспостереження на основі IP-камер	15
2.2 Обґрунтування вибору програмно-апаратних засобів	17
2.3 Вибір датчика руху	22
2.3.1 Датчик руху HC-SR501	25
2.4 Підключення компонентів та програмування плати.....	28
2.5 Програмування серверу	30
3 Розробка мобільного додатку управління IP камерою.....	32
3.1 Розробка мобільного додатку	32
3.2 Тестування мобільного додатку	35
Висновки	37
Перелік посилань.....	39
Додаток А Код плати ESP32-CAM.....	41

ВСТУП

В сучасному світі, де технологічні інновації руйнують звичні рамки та змінюють наше повсякденне життя, розробка інтелектуальних систем стає необхідністю. Однією з галузей, яка вже давно перетворюється під впливом передових технологій, є системи безпеки та відеоспостереження. У цьому контексті, IP-камери виявилися ключовим інструментом для забезпечення безпеки та віддаленого спостереження за навколишнім середовищем. Проте, для максимальної ефективності та зручності користувача, необхідно розвивати нові шляхи управління цими технологічними чудесами.

Кваліфікаційна робота фокусується на розробці мобільного додатку для управління IP-камерою, з використанням мікроконтролера ESP32. Цей проєкт вирізняється не лише технічною складністю, але і відкриттям нових можливостей для кінцевого користувача в сфері віддаленого моніторингу та контролю. Віддаючи перевагу ESP32 як основному елементу системи, ми маємо намір не лише забезпечити функціональність, але й врахувати аспекти енергоефективності та кібербезпеки, що стають важливими чинниками у сучасних технологічних рішеннях.

Розвиток сучасних систем відеоспостереження породжує виклики, що вимагають нових підходів та інтеграції передових технологій. Зокрема, недостатня зручність управління IP-камерами та обмежена можливість взаємодії з ними в режимі реального часу залишають питання ефективності відкритими. Наш дослідницький проєкт виходить із визначення цих проблем та має на меті їх вирішення через розробку мобільного додатку.

Актуальність роботи полягає в поєднанні потужності ESP32 та можливостей мобільних платформ для створення єдиної системи, що об'єднує зручність та функціональність. Застосовуючи інноваційні підходи до управління IP-камерою, ми прагнемо визначити нові стандарти в області віддаленого моніторингу та забезпечення безпеки.

Метою нашої кваліфікаційної роботи є розробка мобільного додатку для ефективного та зручного управління IP-камерою з використанням мікроконтролера ESP32.

Таким чином для досягнення мети в роботі поставлені наступні задачі:

- провести огляд мобільних додатків управління IP камерою з використанням ESP32;
- розробити інтуїтивно зрозумілий та функціональний мобільний додаток для управління IP-камерою;
- забезпечити можливість віддаленого з'єднання та управління через мобільний пристрій;
- реалізувати двосторонню взаємодію між додатком та ESP32 для отримання та відправлення даних в режимі реального часу;
- провести систематичне тестування мобільного додатку та його взаємодії з ESP32.

Ці задачі визначають шлях досягнення мети нашого дослідження, яке має на увазі не лише розробку, але і вдосконалення інфраструктури відеоспостереження з використанням передових технологій.

1 ОГЛЯД МОБІЛЬНИХ ДОДАТКІВ УПРАВЛІННЯ ІР КАМЕРОЮ ЗА ДОПОМОГОЮ ESP32

Розглянемо деякі додатки для управління ІР-камерою за допомогою ESP32. Існує кілька додатків, які надають зручний інтерфейс для керування камерою через цей мікроконтролер. Ось кілька з таких додатків.

TinyCam Monitor (див. рис. 1.1) [1].

Це додаток, який дозволяє використовувати ESP32 для керування ІР-камерою та перегляду відеопотоку в реальному часі.

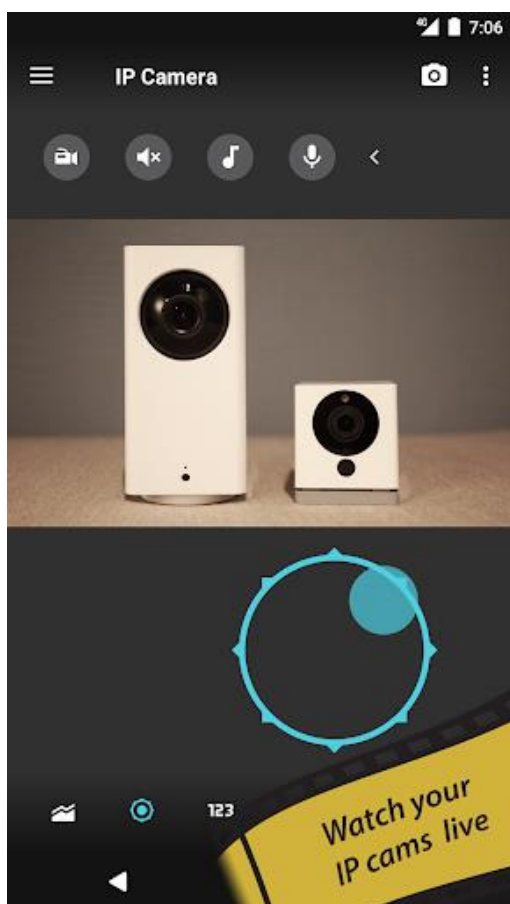


Рисунок 1.1 – TinyCam Monitor

Інтерфейс та навігація:

- *привабливий та інтуїтивний інтерфейс*: TinyCam Monitor має чистий та привабливий інтерфейс, який спрощує процес налаштування та

користування додатком;

- *зручна навігація*: відмінна організація меню та інтуїтивно зрозумілі значки дозволяють легко навігувати та знаходити необхідні функції.

Підтримка камер:

- *широкий вибір моделей*: TinyCam Monitor підтримує багато різних моделей IP-камер, роблячи його універсальним для великої кількості користувачів;
- *легке додавання камер*: процес додавання нової камери в додаток є простим та зручним.

Функціональність:

- *двосторонній аудіозв'язок*: функція двостороннього аудіозв'язку дозволяє вам не лише спостерігати, але і взаємодіяти зі своєю областю спостереження;
- *детектор руху та сповіщення*: вбудований детектор руху виводить додаток на новий рівень, дозволяючи вам отримувати сповіщення при виявленні руху;
- *запис відео та зберігання*: можливість запису відео прямо на мобільний пристрій або в хмарні сховища забезпечує збереження важливих моментів.

Додаткові опції:

- *підтримка облачних сервісів*: TinyCam Monitor взаємодіє з різними облачними сервісами для зберігання та обміну відеоматеріалами;
- *плагіни та розширення*: можливість встановлення різних плагінів та розширень розширює функціонал додатку.

Доступ із зовнішніх мереж:

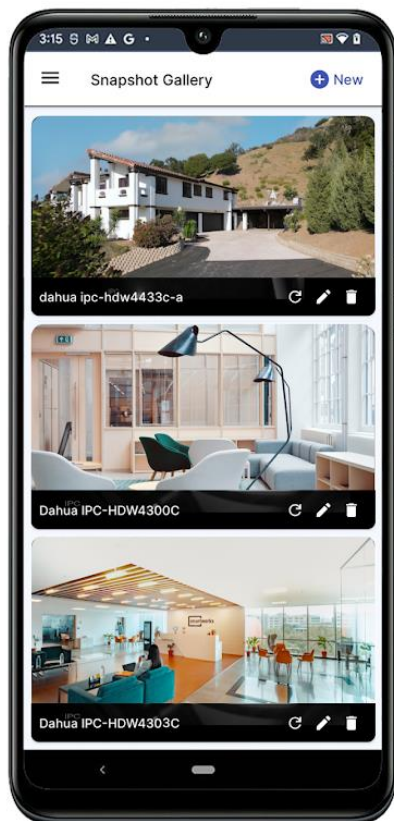
- *робота через інтернет*: TinyCam Monitor дозволяє здійснювати віддалене керування камерою через Інтернет, що дуже корисно для віддаленого моніторингу;
- *безпека та шифрування*: забезпечення захисту та шифрування забезпечують конфіденційність ваших відеоданих.

TinyCam Monitor є потужним та функціональним додатком для управління IP-камерою через ESP32. З його допомогою ви отримаєте широкий спектр можливостей для моніторингу та безпеки вашого приміщення. Зручний інтерфейс та ряд корисних функцій роблять його привабливим вибором для тих, хто шукає ефективний спосіб контролю за своєю IP-камерою.

IP Camera Monitor

Це універсальне додаток, сумісне з ESP32 для керування та моніторингу IP-камери (див. рис. 1.2) [2].

Add IP cameras
to view and control.



Add your camera manually by entering
the IP address and login details.

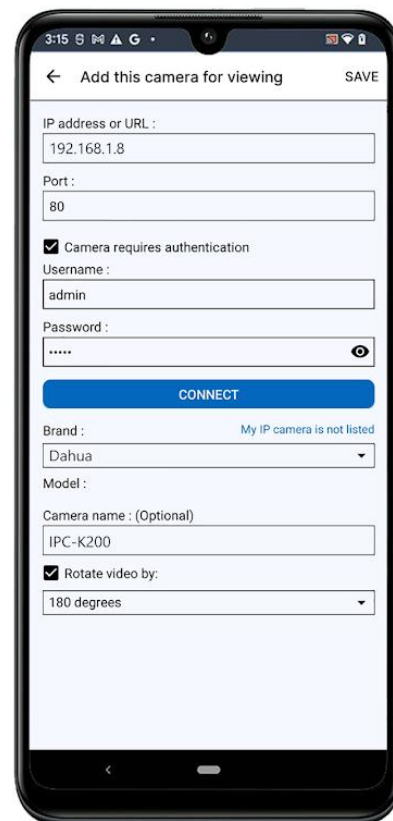


Рисунок 1.2 – IP Camera Monitor

Інтерфейс та Навігація:

- *зручний та легкий у використанні інтерфейс*: IP Camera Monitor пропонує користувачам зручний та легкий у використанні інтерфейс, що спрощує взаємодію з додатком;

- *інтуїтивна навігація*: меню та налаштування логічно організовані, що робить навігацію легкою та зрозумілою.

Підтримка камер:

- *різноманітність підтримуваних моделей*: додаток підтримує широкий вибір різних брендів та моделей IP-камер, що дозволяє вам використовувати його з різними пристроями;
- *простота додавання камер*: процес додавання нової камери у додаток є швидким та безпроблемним.

Функціональність:

- *відображення кількох камер одночасно*: можливість переглядати відеопотік з кількох камер одночасно робить додаток більш універсальним;
- *запис відео та фотографій*: можливість записувати відео та робити фотографії прямо з додатку.

Додаткові опції:

- *підтримка облачних сервісів*: IP Camera Monitor інтегрується з різними облачними сервісами для збереження та обміну відеоматеріалами;
- *спільний доступ*: можливість надавати спільний доступ до відеопотоку іншим користувачам.

Доступ із зовнішніх мереж:

- *віддалене управління*: додаток дозволяє вам віддалено керувати камерою через Інтернет, що важливо для віддаленого моніторингу;
- *безпека та захист*: заходи безпеки, такі як шифрування, забезпечують конфіденційність та захист ваших відеоданих.

Недоліки та можливі покращення:

- *можливі проблеми сумісності*: деякі користувачі можуть зіткнутися з проблемами сумісності з певними моделями камер.

IP Camera Monitor є зручним та функціональним додатком для керування IP-камерою. Він вражає своєю простотою використання та широким спектром функцій для ефективного моніторингу. Зручний доступ із зовнішніх мереж,

підтримка облачних сервісів та інші функції роблять його відмінним вибором для користувачів, які цінують якість та зручність.

IP Cam Viewer

Ця програма забезпечує керування IP-камерою через ESP32 і має велику базу підтримуваних пристроїв (див. рис. 1.3) [3].



Рисунок 1.3 – IP Cam Viewer

Інтерфейс та навігація:

- *продуманий та зручний інтерфейс*: IP Cam Viewer має естетичний та логічно організований інтерфейс, що полегшує користування;
- *ефективна навігація*: зручне розташування меню та налаштувань дозволяє швидко знаходити необхідні функції.

Підтримка камер:

- *широкий спектр підтримуваних моделей*: IP Cam Viewer вражає широким вибором підтримуваних IP-камер різних брендів та моделей;
- *спрощене додавання камер*: процес додавання нової камери у додаток є інтуїтивно зрозумілим та швидким.

Функціональність:

- *многократний монтаж камер*: додаток дозволяє вам монтувати відеопотік з декількох камер одночасно на екрані, що є зручним для великих систем моніторингу;
- *детектор руху та звукові сповіщення*: вбудовані функції детектора

руху та звукових сповіщень дозволяють вам оперативно реагувати на події;

- *запис відео та фотографій*: IP Cam Viewer дозволяє записувати відео та робити знімки безпосередньо з додатку.

Додаткові опції:

- *інтеграція з хмарними сервісами*: додаток підтримує інтеграцію з різними хмарними сервісами для зручного зберігання та обміну відеоданими;
- *розширені налаштування камер*: доступ до різноманітних налаштувань камер для забезпечення оптимальної роботи.

Доступ із зовнішніх мереж:

- *можливість віддаленого доступу*: IP Cam Viewer дозволяє вам віддалено керувати камерою через Інтернет, що важливо для віддаленого моніторингу;
- *безпека та шифрування*: заходи безпеки, такі як шифрування, гарантують конфіденційність ваших відеоданих.

Недоліки та можливі покращення:

- *можливі затримки в роботі*: деякі користувачі вказують на можливі затримки в роботі додатку під час віддаленого моніторингу.

IP Cam Viewer є потужним та розширеним додатком для управління IP-камерами. З його допомогою ви можете насолоджуватися різноманітними функціями, що полегшують віддалений моніторинг та забезпечують безпеку вашого приміщення чи об'єкта. Незважаючи на деякі можливі затримки, додаток вражає своєю функціональністю та розширеними можливостями для задоволення потреб великої кількості користувачів.

Кожен з додатків має свої переваги та особливості. TinyCam Monitor вражає двостороннім аудіозв'язком, IP Camera Monitor – простим інтерфейсом, а IP Cam Viewer – розширеними можливостями та інтеграцією з хмарними сервісами. Вибір між ними залежить від конкретних потреб користувача та його вподобань.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ

Необхідно створити ефективний алгоритм для роботи програмного модуля та вибрати відповідні технології, щоб вони гармонійно взаємодіяли із загальною логікою проєкту. Це визначить продуктивність програми та її загальну ефективність.

2.1 Системи відеоспостереження на основі IP-камер

Системи відеоспостереження на базі IP-камер володіють рядом переваг. Основною перевагою є можливість віддаленого спостереження за об'єктами, виявлення конкретних деталей на зображенні та виявлення руху з подальшою автоматичною реакцією на типові ситуації. Зокрема, широко використовуються IP-камери, які використовують Інтернет для передачі та зберігання зображень [4]. До переваг таких камер можна віднести:

- інтеграція з локальними мережами будь-якого рівня складності;
- IP-камера може працювати в автономному режимі;
- безпроводні камери не потребують прокладки додаткових проводів;
- можливість шифрування відеопотоку.

Підключення IP-камери до вашої мережі або комп'ютера може відбуватися за допомогою різних методів:

- використання WiFi – безпроводний зв'язок;
- використання Ethernet – звичайне підключення за допомогою кабелю витой пари;
- використання PoE – кабельне підключення, при якому живлення передається через певні жили кабелю витой пари, а не окремо.

Для цього потрібен спеціальний інжектор та кабель із додатковим виходом. Загальна схема підключення відеокамер спостереження може бути представлена так, як показано на рисунку 2.1.

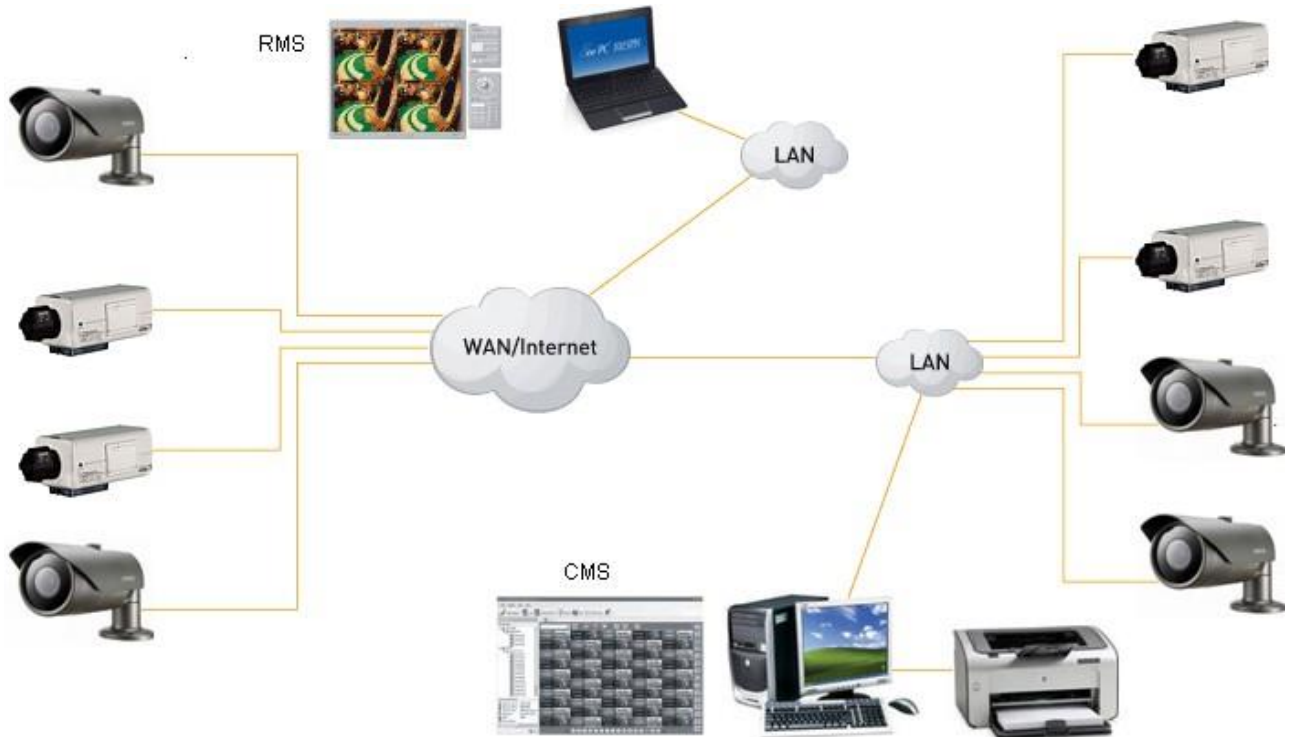


Рисунок 2.1 – Схема підключення системи відеоспостереження

Система відеоспостереження може бути розподіленою, що дозволяє розширити кількість каналів до 500 для IP-камер. Зазвичай для цього використовується спеціальне програмне забезпечення, яке функціонує на різних ПК, включаючи:

- центральну систему управління CMS (Central Management System);
- систему віддаленого управління RMS (Remote Management System).

IP-камери обладнані вбудованим мережевим портом, який дозволяє їм підключатися до мережі Ethernet, а через неї взаємодіяти з CMS.

Система здатна відслідковувати близько 100 відеоканалів, записувати відео та стандартні зображення з IP-камери на жорсткий диск. Вона автоматично перевіряє підключення до головного сервера та мережевого сховища, де зберігаються основні відеозаписи з камер. Розподілена система дозволяє

здійснювати централізований моніторинг камер і виконувати зйомку з камер, розташованих на великій відстані одна від одної або у різних містах. IP-камери призначені для застосування у приміщенні, на вулиці та в транспорті.

Вуличні IP-камери відповідають конкретним вимогам, зокрема стандартам захисту оболонок від пилу та вологи, визначеним міжнародною класифікацією International Protection Marking (IP). Перша цифра вказує на захист від сторонніх предметів (пилу), друга – від проникнення води. Корпус вуличних IP-камер повинен відповідати стандарту (IP54 ~ IP68). Також важливий діапазон робочих температур, при якому камера може працювати, і цей діапазон повинен відповідати стандарту захисту IP54 ~ IP68.

Транспортні IP-камери – це спеціалізоване обладнання, яке захищене від вібрації та обладнане надійними роз'ємами, зазвичай різьбовими M12. Кожна транспортна IP-камера обов'язково сертифікується на відповідність. Незалежно від призначення, існують IP-камери, які призначені для встановлення в місцях, недоступних для зору охоронців, і вони захищені від механічних впливів різних рівнів – корпус відповідає коду міжнародної класифікації IK08 ~ IK10.

2.2 Обґрунтування вибору програмно-апаратних засобів

ESP32 – це лінійка мікроконтролерів, що входять у категорію «система на кристалі». Вона об'єднує в собі інтегровані контролери Wi-Fi та Bluetooth (дворежимний), низьке енергоспоживання та доступну ціну. Мікропроцесор Tensilica Xtensa LX6 використовується в двоядерних і одноядерних варіаціях серії ESP32, і включає вбудовані антенні перемикачі, підсилювачі потужності, приймачі з низьким рівнем шумів, фільтри та модулі керування живленням. Espressif Systems, китайська компанія зі штаб-квартирою у Шанхаї, розробила і створила ESP32, а виробництво відбувається компанією TSMC. Цей мікроконтролер є наступником ESP8266 [5].

Основні особливості ESP32 включають [6]:

- процесор: Xtensa двоядерний (або одноядерний) 32-розрядний LX6 мікропроцесор, що працює на 160 або 240 МГц і виконує до 600 DMIPS;
- ультра низька потужність (ОТП) співпроцесор;
- пам'ять: 520 Кб пам'яті SRAM;
- бездротовий зв'язок: Wi-Fi 802.11 b/g/N, Bluetooth V4.2 БР/EDR і BLE;
- периферійні інтерфейси: 12-розрядний АЦП до 18 каналів, 2 × 8-біт ЦАПи, 10 × сенсорних датчиків, датчик температури, 4 × СВО, 2 × i2s для інтерфейсів, 2 × з I2C інтерфейси, 3 × UART, SD/SDIO/CE-ATA/MMC/eMMC хост-контролер, SDIO/SPI підпорядкований контролер, Ethernet Mac інтерфейс з виділеними DMA і стандартами IEEE 1588 точного часу, CAN bus 2.0, Інфрачервоний пульт дистанційного управління (передавач/приймач, до 8 каналів), можливість підключення двигунів та світлодіодів через ШІМ-вихід, ультра низька потужність аналоговий передпідсилювач;
- безпека: стандарт IEEE 802.11 підтримує всі функції безпеки, у тому числі АБФ, захист WPA/WPA2 і ВАПІ, безпечне завантаження, шифрування флеш, 1024-бітний ключ, до 768 біт для клієнтів, криптографічне апаратне прискорення: AES, SHA-2, RSA, криптографія на основі еліптичних кривих (ECC), генератор випадкових чисел (ГВЧ);
- управління живленням: внутрішній низький регулятор відключення, індивідуальний енергетичний домен для RTC, 5 мкА струм режиму «глибокий сон», прокидання з переривання від GPIO, таймера, вимірювання АЦП, переривання ємнісного сенсорного датчика.

ESP32 можна програмувати за допомогою різних мов програмування, платформ та середовищ, таких як Arduino IDE з ESP32 Arduino Core, Espressif IoT Development Framework, Espruino, Lua RTOS, Mongoose OS, PlatformIO Ecosystem і IDE, Pymakr IDE, Simba Embedded Programming Platform, Whitecat Ecosystem Blockly, MicroPython, Zerynth.

У даній роботі використовується Arduino IDE з ESP32 Arduino Core через його популярність, документацію та широку базу прикладів в Інтернеті, а також підтверджений часом шлях. ESP32-CAM, яка базується на ESP32, обрана для дослідження розпізнавання облич за її низькою ціною та можливістю розширення через слот для SD карт (див. рис. 2.2, 2.3).

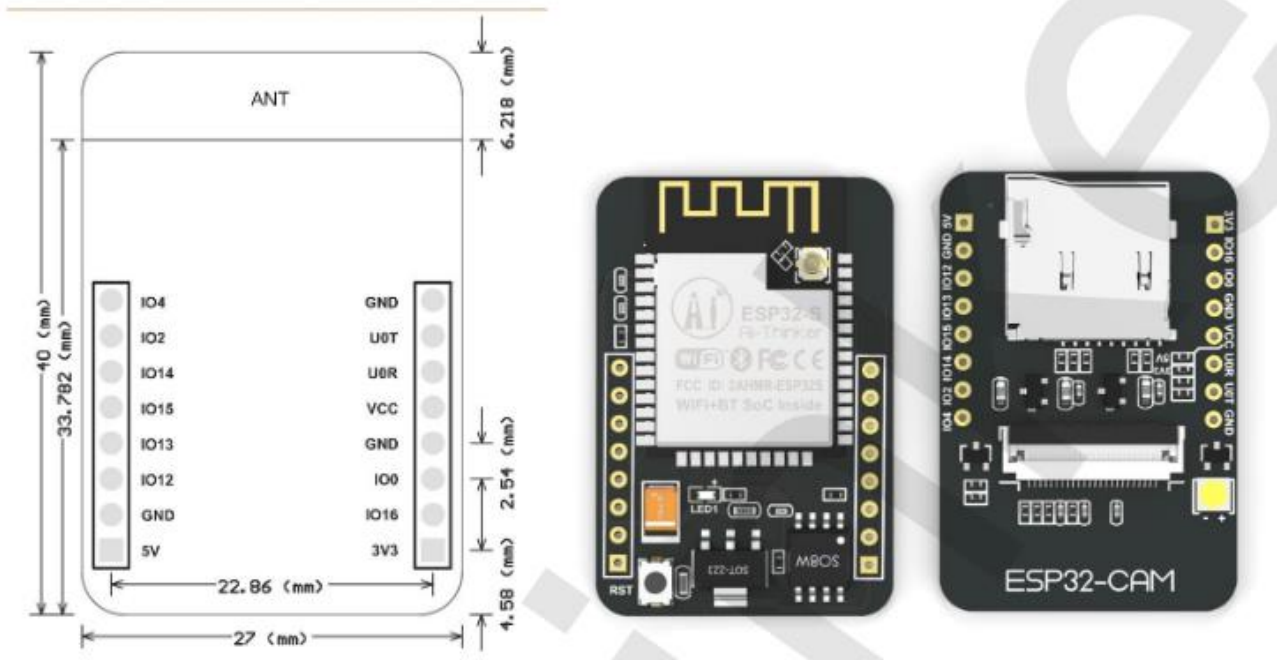


Рисунок 2.2 – Схема плати ESP32-CAM

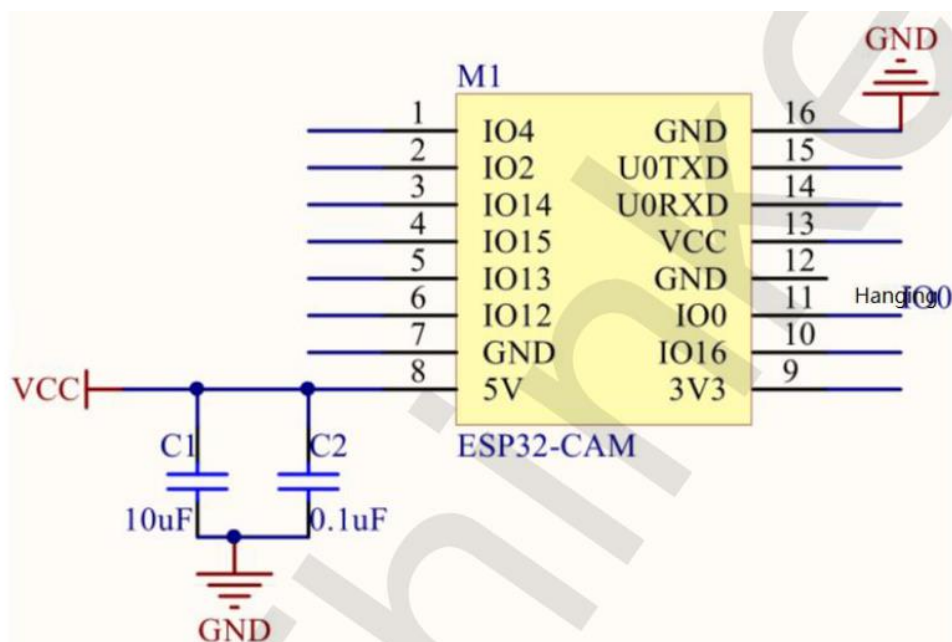


Рисунок 2.3 – Схема виводів плати ESP32-CAM

Основна інформація про модуль виглядає наступним чином (див. табл. 2.1).

Таблиця 2.1 – Таблиця пінів плати ESP32-CAM

Cam	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1/Flash lamp	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		

Розглянемо детальніше:

- найменший модуль SoC Wi-Fi BT 802.11b/g/n;
- 32-розрядний процесор низької потужності, який може також використовуватися як процесор додатків;
- тактова частота до 160 МГц, що призводить до підвищеної обчислювальної потужності до 600 DMIPS;
- вбудований обсяг 520 KB SRAM, а також можливість зовнішньої пам'яті 4MPSRAM;

- підтримка UART/SPI/I2C/PWM/ADC/DAC;
- здатність працювати з камерами OV2640 і OV7670, а також вбудована лампа Flash;
- можливість завантаження зображення через Wi-Fi;
- підтримка карт TF;
- підтримка декількох режимів сну;
- вбудовані Lwip і FreeRTOS;
- підтримка режимів роботи STA/AP/STA+ точка доступу;
- підтримка технологій Smart Config/AirKiss;
- здатність до локального та віддаленого оновлення мікропрограми через послідовний порт (FOTA).

Окрім ESP32 CAM буде використана плата NodeMcu ESP32, тому що ESP32 CAM не має USB інтерфейсу чи іншого гарного інтерфейсу для програмування за замовчуванням, інша популярна опція працюючи з ESP32 CAM є FTDI програматор (див. рис. 2.4).

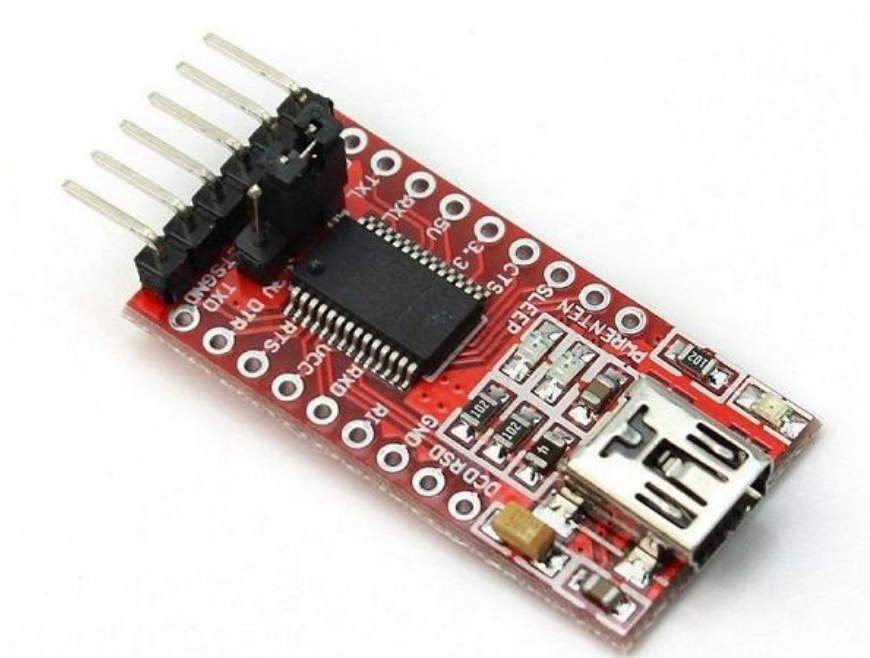


Рисунок 2.4 – FTDI програматор

На рисунку 2.5 показано схему програмування ESP32-CAM з використанням FTDI програматору.



Рисунок 2.5 – Схема з'єднання ESP32-CAM та FTDI

2.3 Вибір датчика руху

Датчик руху, також відомий як детектор руху, це електронний пристрій, спроектований для виявлення та вимірювання руху [7]. Ці пристрої в основному використовуються у домашніх та комерційних системах безпеки, а також можуть бути включені до телефонів, дозаторів для паперових рушників, ігрових консолей і систем віртуальної реальності.

На відміну від багатьох інших видів датчиків, які можуть працювати в ручному або ізольованому режимі, датчики руху, як правило, є вбудованими системами, складаються з трьох основних компонентів: блоку датчиків, вбудованого комп'ютера та апаратного забезпечення (або механічного компонента).

Ці компоненти відрізняються за розміром і конфігурацією, оскільки датчики руху можна налаштовувати для виконання конкретних функцій. Наприклад, їх можна використовувати для активації прожекторів, відтворення звукових сигналів, управління перемикачами та навіть відсилення сигналу поліції.

Основною метою використання датчиків руху є виявлення зловмисника та надсилання сповіщення на панель керування, яка в свою чергу інформує центр моніторингу. Ці датчики активуються, коли немає осіб вдома або коли користувач повідомляє систему про свою відсутність.

Деякі системи безпеки можуть бути налаштовані для запису подій за допомогою камер безпеки при виявленні руху.

Датчики руху виступають у ролі сторожа, готового реагувати на різноманітні ситуації, такі як рух у вітальні, відкривання або закривання вікон і дверей, або навіть розбиття вікон.

Ось кілька типових застосувань датчиків руху:

- повідомлення у випадку порушення комендантської години;
- активація дверного дзвінка, коли хтось наближається до вхідних дверей;
- попередження, якщо діти потрапляють у заборонені зони в будинку, такі як підвал, кімната для тренувань;
- збереження енергії за допомогою датчика руху для управління освітленням у незайнятих приміщеннях.

Пасивний інфрачервоний (PIR) датчик руху (див. рис. 2.6) із пасивним інфрачервоним ефектом виявляє теплове випромінювання (інфрачервону енергію), реагуючи на зміни температури. Цей вид датчика руху є найбільш розповсюдженим у системах безпеки для будинку.



Рисунок 2.6 – PIR датчик руху

При активації системи охорони, пасивні інфрачервоні датчики руху надсилають повідомлення про можливі загрози.

Після прогрівання датчик руху PIR може виявляти теплове випромінювання і рух в навколишніх областях, утворюючи захисну «сітку». Якщо рухливий об'єкт блокує занадто багато зон сітки, і рівень інфрачервоної енергії швидко змінюється, датчик руху спрацьовує тривогу.

Мікрохвильовий датчик руху відсилає мікрохвильові імпульси та вимірює їх відбиття від рухомих об'єктів (див. рис. 2.7). Вони охоплюють більшу площу, порівняно з інфрачервоними датчиками, але вони дорожчі та більш вразливі до електричних перешкод.

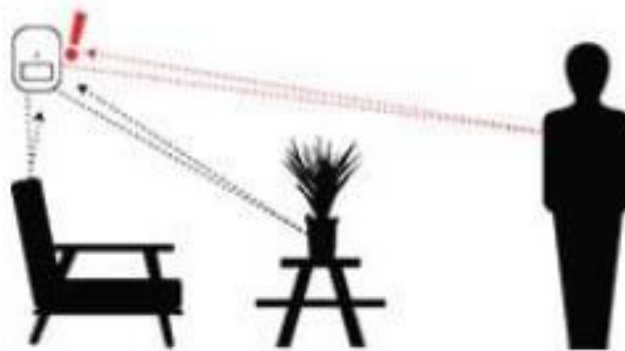


Рисунок 2.7 – Мікрохвильовий датчик руху

Датчики руху з подвійною технологією можуть використовувати кілька методів виявлення з метою зменшення помилкових тривог (див. рис. 2.8).



Рисунок 2.8 – Датчик руху з подвійною технологією

Наприклад, у випадках, коли датчик із подвійною технологією комбінує пасивний інфрачервоний (PIR) датчик із мікрохвильовим датчиком.

Кожен вид датчика операційний у різних частинах спектра, від пасивного до активного. Датчики руху з подвійною технологією, порівняно з іншими видами, менш схильні до виклику помилкових тривог, оскільки обидва типи датчиків повинні спрацювати для ініціації сигналу тривоги. Проте це не означає, що вони абсолютно виключені від помилкових сигналів.

Інші менш поширені види датчиків руху включають:

- датчики, які використовують відбиття місцевості, випромінюючи інфрачервоні промені від світлодіода і використовуючи відбиття для вимірювання відстані до об'єкта або людини, що дозволяє виявити рух в конкретній зоні;
- ультразвукові датчики руху, які вимірюють відбиття від рухомих об'єктів за допомогою імпульсів ультразвукових хвиль;
- вібраційні датчики руху, які реагують на невеликі вібрації, спричинені рухом людей по кімнаті. Їх можна придбати готовими, але також легко зробити вдома, використовуючи невелику масу на важелі, яка активує вимикач тривоги при вібраціях.

Саморобні датчики руху можуть бути функціональними, але їх також можна характеризувати як мало надійні.

2.3.1 Датчик руху HC-SR501

При вході людини до зони огляду датчика реєструється наявність. Робочий принцип модуля HC-SR501 базується на виявленні інфрачервоного випромінювання рухомого об'єкта [8].

Чутливий елемент цього модуля – піроелектричний датчик 500BP, що складається з двох ув'язнених елементів в одному корпусі. Цей чутливий елемент знаходиться під білим куполом, який виступає лінзою Френеля.

Особливість лінзи Френеля полягає в тому, що інфрачервоне випромінювання рухомого об'єкта спочатку потрапляє на один елемент датчика 500BP, а потім на інший.

Електроніка модуля HC-SR501 реєструє послідовне надходження сигналів від обох елементів 500BP і при виявленні руху формує логічний сигнал у вихідному ланцюзі модуля (див. рис. 2.9).



Рисунок 2.9 – Датчик руху HC-SR501

Датчик присутності HC-SR501 застосовується у системах безпеки, в управлінні вентиляцією та освітленням в приміщеннях, де відсутні вікна. Спільно з фотореле він може бути використаний для керування освітленням на дворах і вулицях. Зацікавлені результати можуть бути досягнуті, використовуючи даний датчик у поєднанні з фотокамерою або відеоканерою.

PIR-датчик перекладається з англійської як Pyroelectric (Passive) InfraRed sensor – піроелектричний (пасивний) інфрачервоний сенсор.

Режим роботи модуля визначається положенням перемикача. Існують два режими: режим H та режим L.

У режимі H, при спрацьовуванні датчика, вихід (OUT) протягом кількох послідовних подій залишається на високому логічному рівні.

У режимі L, при кожному спрацьовуванні датчика на виході з'являється окремий імпульс.

Характеристики модуля HC-SR501 подані в таблиці 2.2.

Таблиця 2.2 – Характеристики модуля HC-SR501

Напруга живлення	DC 4.5-20V
Струм на OUT	<60uA
Напруга на виході	Високі та низькі рівні у 3.3V TTL логіці
Дистанція виявлення	3 – 7м (налаштовується), за замовчуванням 7 м
Кут виявлення	до 110 °, на відстані 7 м 120 °
Тривалість імпульсу при виявленні	5 – 200сек. (налаштовується)
Час блокування до наступного вимірювання	2.5сек. (але можна змінити заміною SMD-резисторів)
Робоча температура	-20 +80 ° C
Режим роботи	L – одиночний захоплення, H - повторювані виміри
Розміри	3.2см x 2.4см x 2.8см

Пристрій готовий до роботи всього за хвилину після підключення живлення, під час якого автоматично проводиться калібрування.

При виборі місця для встановлення важливо уникати орієнтації датчика на відкриті джерела світла.

Монтаж датчика присутності HC-SR501 рекомендується проводити так, щоб рух об'єкта відбувався вздовж площини плати модуля.

Розташовувати PIR-датчики не рекомендується в областях зі стрімко змінюючоюся температурою, оскільки це може призвести до недостатньої здатності датчика виявляти присутність людини та викликати багато помилкових спрацьовувань.

Для підключення модуля до мікроконтролерів (або інших мікросхем) краще (хоча й необов'язково) використовувати транзистор та підтягуючий резистор на 10k.

2.4 Підключення компонентів та програмування плати

Для проєкту було використано плати ESP32-CAM, NodeMcu ESP32 та датчик руху HC-SR501. Плата NodeMcu ESP32 використовувалась насамперед для програмування ESP32-CAM, датчик руху під'єднувався до ESP32-CAM. Підключення відбувалося за схемою (див. рис. 2.10).

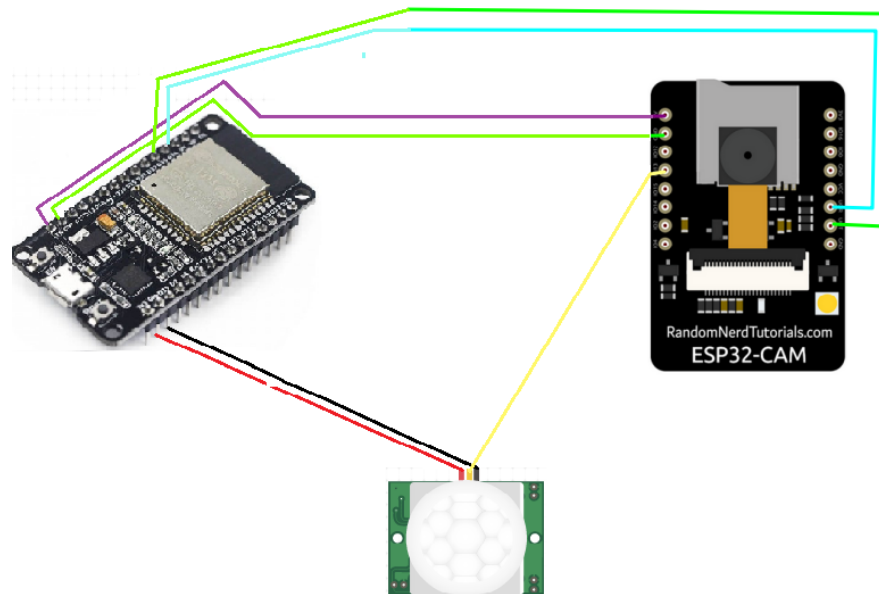


Рисунок 2.10 – Схема підключення плат NodeMcu ESP32, ESP32-CAM і датчика HC-SR501

Програмування було здійснене лише для плати ESP32-CAM. Для цього було використано Arduino IDE [9] та стандартні бібліотеки з пакету ESP32 Camera Web Server, які були мінімально відредаговані для з'єднання з локальною мережею та відправлення даних з датчика на сервер.

У функції `setup` відбувається налаштування сервера для трансляції відео та конфігурування деяких параметрів сенсора камери. Для серверу камери використовується стандартний код з бібліотеки Camera Web Server. Повний код буде доступний у додатку А.

Нижче наведено частину коду для датчика руху, а саме головна функція `loop` (див. рис. 2.11).

```

void loop() {
int level;
level = digitalRead(13);
if(level == HIGH){
Serial.print("Я бачу тебе ");
if ((WiFi.status() == WL_CONNECTED)) { // Перевірте поточний стан
підключення
HTTPClient http;
http.begin(host); // Вкажіть URL
int httpCode = http.GET(); // Зробіть запит
if (httpCode > 0) { // Перевірте код повернення
String payload = http.getString();
Serial.println(httpCode);
Serial.println(payload);
Serial.print("Звіт успішно надіслано!\n");
}
else {
Serial.println("Помилка HTTP-запиту ");
Serial.println(httpCode);
}
http.end(); // Звільніть ресурси

} else {
Serial.print("Не вдається надіслати звіт, немає підключення до Wi-Fi
\n");
}
}
delay(3000);
}

```

Рисунок 2.11 – Код для датчика руху (функція loop)

Сервер має дві логічні адреси для обробки запитів.

Перша адреса повертає статус датчика руху, за замовчуванням встановлено «Руху не виявлено», і слугує точкою входу для запитів з мобільного додатку.

Друга адреса використовується для обробки запитів з плати ESP32-CAM і оновлює значення змінної у випадку виявлення руху датчиком. Після спрацювання датчика сервер протягом 10 секунд повертає статус «Виявлено рух».

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ УПРАВЛІННЯ ІР КАМЕРОЮ

3.1 Розробка мобільного додатку

Для розробки мобільного додатку було використано фреймворк React Native [15-16] та мову програмування JavaScript [17-18]. Цей вибір дозволив нам використовувати загальний код для обох платформ – iOS та Android. Вибір також обумовлений простотою програми, оскільки вона не залежить від нативних компонентів операційних систем.

Наш мобільний додаток в реальному часі відображає статус датчика руху (див. рис. 3.1), отримуючи дані з сервера, і має кнопку для перегляду відео з камери. Повний вихідний код буде доступний у додатку, а ось основні функції програми можна побачити як приклад нижче.

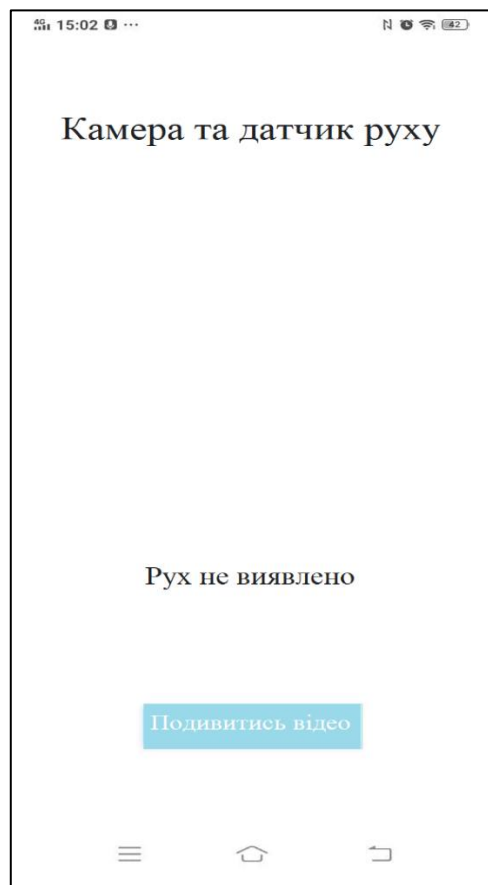


Рисунок 3.1 – Інтерфейс мобільного додатку

Функція `render` (див. рис. 3.2) відтворює основні складові програми: виводить текст, що відображає статус з датчика, створює кнопку для перегляду відео та відображає назву додатку (див. рис. 3.3).

```
render() {
return (
  <View style={styles.container}>
  <Text style={{marginBottom: 400, fontSize: 20}}> Камера та датчик руху
  </Text>
  <Text style={{marginBottom: 100, fontSize: 14}}>{this.state.move_msg}
  </Text>
  <Button
    onPress={
      () => this.videoRequest()
    }
    title = " Подивитись відео"
  />
  </View>
  )
}
```

Рисунок 3.2 – Функція `render`

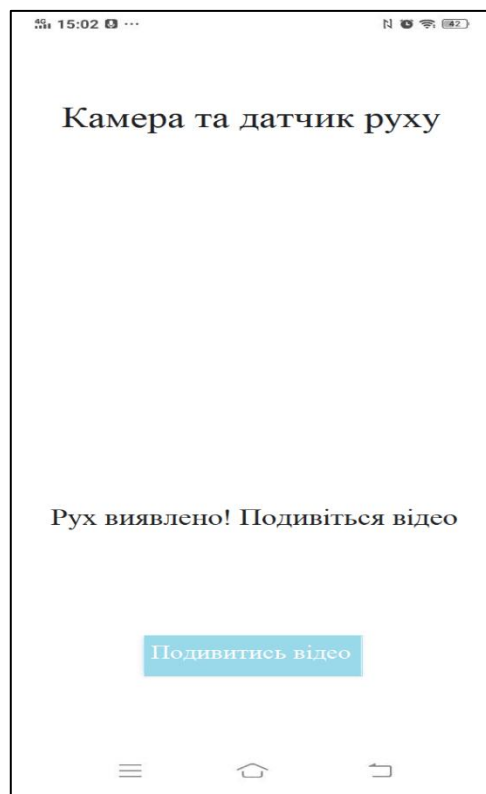


Рисунок 3.3 – Повідомлення про знаходження руху

Функція `videoRequest` (див. рис. 3.4) відкриває вебсторінку у вбудованому браузері додатку, де користувач може переглядати відео з камери (див. рис. 3.5).

```
videoRequest() {  
  // При натисканні кнопки – показ відео  
  console.log("Кнопка натиснута ");  
  Linking.openURL("http://192.168.0.105")  
}
```

Рисунок 3.4 – Функція `videoRequest`

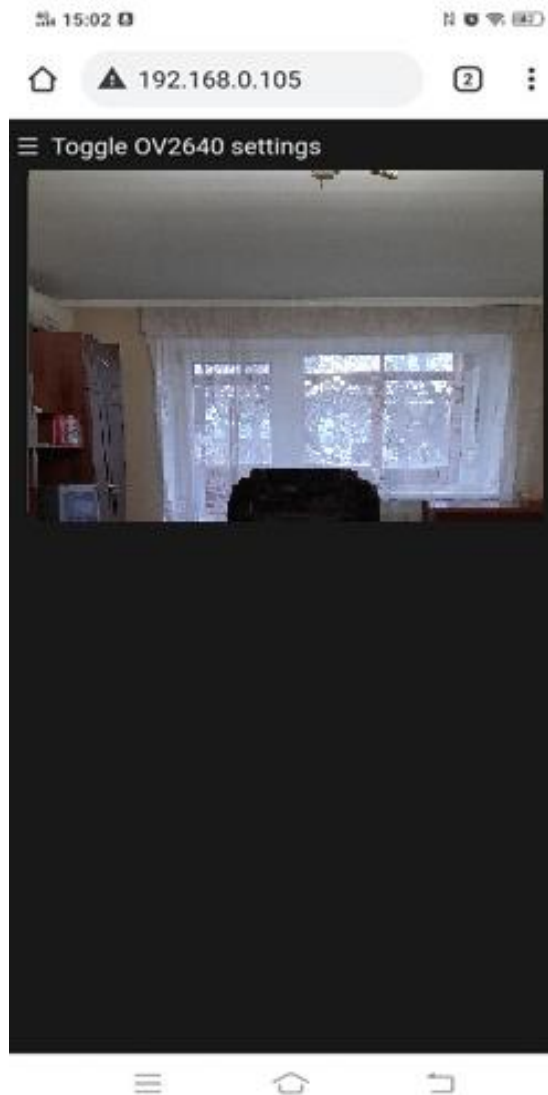


Рисунок 3.5 – Перегляд відео в мобільному додатку

Функція `checkMove` (див. рис. 3.6) ініціює запит на сервер для перевірки стану датчика руху.

```
checkMove() {  
  resp = fetch('http://192.168.0.103:5000/ask')  
    .then((response) => response.json())  
    .then((responseJson) => {  
      console.log(responseJson)  
      return responseJson;  
    }).then((rObj) => {  
      if(rObj.move_detected) {  
        this.setState({move_msg : " Рух виявлено! Подивіться відео"})  
      } else {  
        this.setState({move_msg : " Рух не виявлено  
      "})  
    })  
  })  
}
```

Рисунок 3.6 – Функція `checkMove`

У випадку виявлення руху, функція змінює текст елемента додатку та повідомляє користувача про виявлений рух.

3.2 Тестування мобільного додатку

Для проведення тестування була використана збірна схема плати та датчика. Був встановлений мобільний додаток, і сервер був запущений у локальній мережі. Після цього була проведена серія випробувань додатку в умовах, що наближені до реального використання. Датчик достатньо точно

реагував на рух на відстані від 0,5 до 3 метрів і передавав коректні дані на сервер. Мобільний додаток миттєво відображав ці дані в інтерфейсі користувача.

Загалом результати тестування були задовільними. Як логічний наступний крок, систему можна легко перенести на віддалений хостинг із доступом в Інтернет, забезпечивши доступ до неї на великій відстані від камери. Для поліпшення системи також необхідно перевести сервер на зашифрований канал зв'язку HTTPS і встановити плати та датчик в якийсь каркас.

ВИСНОВКИ

У сучасному світі, де технологічні новації неухильно перетворюють наше повсякденне життя, важливість створення інтелектуальних систем стає нагальною. Системи безпеки та відеоспостереження, які вже перетворюються під впливом передових технологій, потребують нових методів управління. В цьому контексті розробка мобільного додатка для управління IP-камерою за допомогою мікроконтролера ESP32 виступає як ключовий етап у вдосконаленні та зручності використання таких систем.

Ця кваліфікаційна робота зосереджується на розробці мобільного додатка, який відкриває нові можливості для кінцевого користувача в галузі віддаленого моніторингу та контролю. Обираючи ESP32 як центральний елемент системи, мета полягає не тільки в наданні функціоналу, але й у врахуванні аспектів енергоефективності та кібербезпеки, які набувають важливості в сучасних технологічних рішеннях.

Розробка нових систем відеоспостереження породжує виклики, що вимагають нових підходів та інтеграції передових технологій. У нашому випадку, недостатня зручність управління IP-камерами та обмежена можливість взаємодії в режимі реального часу залишають багато питань стосовно ефективності. Дослідницький проєкт не лише визначає ці проблеми, але й ставить за мету їх вирішення через створення мобільного додатка.

Актуальність роботи полягає в злитті потужності ESP32 та можливостей мобільних платформ для створення єдиної системи, яка комбінує зручність та функціональність. Розробка та вдосконалення інфраструктури відеоспостереження з використанням передових технологій стає важливим кроком у розвитку цього напрямку.

Мета нашої кваліфікаційної роботи – розробка ефективного та зручного мобільного додатка для управління IP-камерою, використовуючи мікроконтролер ESP32. Задачі, які вирішуються, включають огляд існуючих

додатків, розробку інтуїтивно зрозумілого і функціонального додатка, забезпечення віддаленого з'єднання та впровадження двосторонньої взаємодії між додатком та ESP32. Ці завдання визначають стратегію для досягнення нашої основної мети – вдосконалення інтегрованих систем відеоспостереження та управління.

ПЕРЕЛІК ПОСИЛАНЬ

1. TinyCam Monitor. URL: <https://play.google.com/store/apps/details?id=com.alexvas.dvr&hl=en> (дата звернення: 05.07.2023).
2. IP Camera Monitor. URL: https://play.google.com/store/apps/details?id=com.ip_camera_monitor&hl=en (дата звернення: 07.07.2023).
3. IP Cam Viewer. URL: <https://play.google.com/store/apps/details?id=com.rcreations.ipcamviewer&hl=uk&gl=US> (дата звернення: 09.07.2023).
4. Сучасні камери відеоспостереження: основні критерії вибору. URL: <https://asisvok.com.ua/blog/item/suchasni-ip-kamery-videosposterezhennia-osnovni-kryterii-vyboru> (дата звернення: 10.07.2023).
5. Wikipedia. ESP32. URL: <https://uk.wikipedia.org/wiki/ESP32> (дата звернення: 16.07.2023).
6. Компанія Espressif. ESP32 Series datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (дата звернення: 16.07.2023).
7. The Beginner's Guide to Motion Sensors. URL: <https://www.safewise.com/resources/motion-sensor-guide/> (дата звернення: 30.07.2023).
8. Radio Store. URL: <https://radiostore.ua/products/datchik-dvizheniya-pir-motion-sensor-hc-sr501-2> (дата звернення: 05.08.2023).
9. Офіційний сайт Arduino. URL: <https://www.arduino.cc> (дата звернення: 09.08.2023).
10. Flask. URL: <https://uk.wikipedia.org/wiki/Flask> (дата звернення: 02.09.2023).

11. Shalabh Aggarwal Flask Framework Cookbook: Enhance your Flask skills with advanced techniques and build dynamic, responsive web applications 3rd Edition, Kindle Edition. Packt Publishing, 2023.
12. Васильєв О. Програмування мовою Python. Навчальна книга. Богдан, 2019, 504 с.
13. Eric Matthes Python Crash Course: A Hands-On, Project-Based Introduction to Programming. 2015. 560 p.
14. Python. URL: <https://www.python.org/> (дата звернення: 12.09.2023).
15. React Native. URL: <https://reactnative.dev/> (дата звернення: 12.09.2023).
16. Bonnie Eisenman Learning React Native: Building Native Mobile Apps with JavaScript 2nd Edition, Kindle Edition. O'Reilly Media, 2017.
17. David Flanagan JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language 7th Edition, Kindle Edition. O'Reilly Media. 2020.
18. JavaScript Довідка. URL: <http://xn--80adth0aefm3i.xn--j1amh/> (дата звернення: 20.09.2023).

ДОДАТОК А

Код плати ESP32-CAM

```
#include "esp_camera.h"
#include <WiFi.h>
#include <HTTPClient.h>
// Виберіть модель камери

//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_ESP_EYE
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"
const char* ssid = "*****";
const char* password = "*****";
const char* host = "http://192.168.0.103:5000/"; // Моя локальна IP-АДРЕСА

void startCameraServer();
void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
  pinMode(13, INPUT);
  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;
// ініціалізація з високими специфікаціями для попереднього виділення більших //
буферів
if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

#ifdef CAMERA_MODEL_ESP_EYE
pinMode(13, INPUT_PULLUP);
pinMode(14, INPUT_PULLUP);
```

```

#endif

// ініціалізація камери
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Помилка запуску камери з помилкою
0x%x", err);
    return;
}

sensor_t * s = esp_camera_sensor_get();
//початкові датчики перевернуті вертикально, а кольори трохи насичені
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1);//flip it back
    s->set_brightness(s, 1);// лише трохи зменшити блідість
    s->set_saturation(s, -2);// знизити насиченість
}
//випадаючий розмір кадру для вищої початкової частоти кадрів
s->set_framesize(s, FRAMESIZE_QVGA);
#ifdef CAMERA_MODEL_M5STACK_WIDE
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi підключено");

```

```

startCameraServer();

Serial.print("Камера готова! Використовуйте "http://");
Serial.print(WiFi.localIP());
Serial.println("“ для підключення");
}

void loop() {
// помістіть тут свій основний код для повторного запуску:
int level;
level = digitalRead(13);
if(level == HIGH){
  Serial.print("I see you");
  if ((WiFi.status() == WL_CONNECTED)) { // Перевірте поточний стан
                                          // підключення

    HTTPClient http;
    http.begin(host); // вкажіть URL
    int httpCode = http.GET(); // Зробіть запит
    if (httpCode > 0) { // Перевірте код повернення
      String payload = http.getString();
      Serial.println(httpCode);
      Serial.println(payload);
      Serial.print("Звіт успішно надіслано!\n");
    }
    else {
      Serial.println("Помилка HTTP-запиту");
      Serial.println(httpCode);
    }
    http.end(); // звільніть ресурси
  } else {
    Serial.print("Не вдається надіслати звіт, немає підключення до Wi-Fi\n");
  }
}

```

```
}  
delay(3000);  
}
```