МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри,
канд. фіз.-мат. наук, доцент
_____ Я. М. Крайник
« __ » _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

СИСТЕМА ОПОВІЩЕННЯ НА БАЗІ ESP32-CAM, ДАТЧИКУ РУХУ ТА TELEGRAM BOT API

Спеціальність 123 Комп'ютерна інженерія
123 – КР.1 – 405.21810512

Студент _____ В. М. Кім
підпис
«__»_____202__р.

Керівник доцент, канд. фіз.-мат. наук _____ С. В. Пузирьов
підпис
«__»_____202__р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп’ютерних наук
Кафедра комп’ютерної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри,
канд. техн. наук, доцент
_____ Я. М. Крайник
« __ » _____ 2022 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Видано студенту групи 405 факультету комп’ютерних наук

Кіму Віктору Миколайовичу

(прізвище, ім’я, по батькові студента)

1. Тема кваліфікаційної роботи: Система оповіщення на базі ESP32-CAM, датчику руху та Telegram Bot API.

Затверджена наказом по ЧНУ від «__» _____ 2022 р. № _____

2. Строк представлення кваліфікаційної роботи «____» _____ 2022 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Прилад має оперувати інформацією отриманою з датчику руху, приймати зображення з камери та сповіщати користувача через месенджер-бот.

4. Перелік питань, що підлягають розробці

Аналіз апаратних платформ. Опис контролера ESP32-CAM, датчика руху, Telegram Bot API. Аналіз алгоритмів та методів інтеграції. Моделювання прототипу, макетної та принципової схем. Збирання прототипу. Налаштування апаратної платформи. Розробка програмного забезпечення приладу. Спеціальна частина з охорони праці. Висновки. Перелік джерел посилання.

5. Перелік графічних матеріалів

Зображення датчиків руху, контролера

Зображення принципів роботи контролера ESP32-CAM

Зображення роботи програмно-апаратного комплексу.

Макетні та принципові схеми підключення компонентів

6. Завдання до спеціальної частини

Розглянути основні державні будівельні норми України, щодо вентиляції та кондиціонування, забирання зовнішнього та викид витяжного повітря, витрати припливного повітря та організація повітрообміну. Оволодіти навиками розрахунку систем повітрообміну при загально обмінній вентиляції виробничих приміщень.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
ст. викладач А. О. Алексєєва	кафедра екології Медичного інституту ЧНУ ім. Петра Могили	спеціальна частина з охорони праці

Керівник роботи ____канд. фіз.-мат. наук, доцент, С. В. Пузирьов___ _____

(посада, прізвище, ім’я, по батькові) (підпис)

Завдання прийнято до виконання __Кім Віктор Миколайович_____ _____

(прізвище, ім’я, по батькові студента) (підпис)

Дата видачі завдання «____» _____ 20_____ р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: Система оповіщення на базі ESP-32 CAM, датчику руху та Telegram Bot API.

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КР	20.12.2021	05.01.2022	Виконано
2	Огляд літератури за темою роботи	10.01.2022	01.02.2022	Виконано
3	Складання календарного плану КР	03.02.2022	14.02.2022	Виконано
4	Аналіз предметної області	15.02.2022	23.02.2022	Виконано
5	Порівняльний аналіз існуючих технічних рішень	24.02.2022	15.03.2022	Виконано
6	Підбір необхідних для реалізації системи компонентів	16.03.2022	25.03.2022	Виконано
7	Проєктування прототипу системи оповіщення на базі ESP-32 CAM	26.03.2022	11.04.2022	Виконано
8	Реалізація програмної частини системи оповіщення на базі ESP-32 CAM	12.04.2022	26.04.2022	Виконано
9	Тестування та відлагодження отриманого прототипу. Аналіз результатів	27.04.2022	25.05.2022	Виконано
10	Відгук керівника КР	26.05.2021	01.06.2021	Виконано
11	Оформлення результатів дослідження	02.06.2021	09.06.2021	Виконано
12	Попередній захист	10.06.2022	11.06.2022	Виконано
13	Рецензування	14.06.2022	18.06.2022	Виконано
14	Завершення оформлення КР та презентації	18.06.2022	25.06.2022	Виконано
15	Захист кваліфікаційної роботи	27.06.2022	28.06.2022	Виконано

Розробив студент_____Кім Віктор Миколайович_____ _____

(прізвище, ім'я, по батькові) (підпис)

«__» _____ 2022 р.

Керівник роботи ___канд. фіз.-мат. наук, доцент, С. В. Пузирьов____ _____

(посада, прізвище, ім'я, по батькові) (підпис)

«__» _____ 2022 р.

АНОТАЦІЯ

бакалаврської роботи

«Система оповіщення на базі ESP32-CAM, датчику руху та Telegram Bot API»

Студент: Кім Віктор Миколайович

Керівник: канд. фіз.-мат. наук, доцент Пузирьов С. В.

Бакалаврська робота присвячена розробці системи оповіщення на базі ESP32-CAM, датчику руху та Telegram Bot API. Розглянуто наявні в наш час методи та способи проєктування систем з використанням мікроконтролера ESP32-CAM. Практичне значення отриманих результатів полягає у дослідженні особливостей створення та реалізації ефективного алгоритму обробки інформації для технології оповіщення на основі ESP32-CAM, датчика руху та їх інтеграції у спеціалізований бот для Telegram-месенджера.

Пояснювальна записка бакалаврської роботи складається зі вступу, чотирьох розділів, висновків та додатків. У вступі визначається актуальність теми, сформульовані мета, об'єкт, предмет та завдання дослідження та розроблення бакалаврської роботи. У першому розділі досліджуються різноманітні технології використання ESP32-CAM, що є об'єктом дослідження; проводиться аналіз ринку і технічних рішень систем з використанням ESP32-CAM. У другому розділі наведені дані про підходи до проєктування системи та алгоритм її створення. У третьому розділі наведені дані про реалізацію програмно-апаратного комплексу та результати тестування. Четвертий розділ присвячений охороні праці щодо норм роботи з використанням електронних пристроїв. У висновках наведено аналіз виконаної роботи та отриманих результатів дослідження та розроблення.

У додатку А наведено програмний код, що використовувався в проєкті. В цілому, бакалаврська робота без додатків містить 73 сторінки, 51 рисунків, 2 таблиці, 29 джерел посилання.

Ключові слова: автоматизована система оповіщення, ESP32-CAM, датчик руху, програмно-апаратний комплекс, бот, telegram-месенджер.

ABSTRACT

of the Bachelor's Thesis

"Alert system based on ESP32-CAM, motion sensor and Telegram Bot API"

Student: Kim Viktor

Consultant: cand. phys.-math. sciences, associate professor Puzyrev Serhii

The bachelor's thesis is devoted to the development of an alert system based on ESP32-CAM, motion sensor and Telegram Bot API. The methods and methods of designing systems using the ESP32-CAM microcontroller available at present are considered. The practical significance of the obtained results is to study the features of creating and implementing an effective algorithm for information processing for notification technology based on ESP32-CAM, motion sensor and their integration into a specialized bot for Telegram-messenger.

An explanatory note of the bachelor's thesis consists of an introduction, four chapters, conclusions and appendices. The introduction determines the relevance of the topic, formulates the purpose, object, subject and objectives of research and development of the bachelor's thesis. The first section examines the various technologies for using ESP32-CAM that are the subject of the study; the analysis of the market and technical decisions of systems with use of ESP32-CAM is carried out. The second section provides data on approaches to system design and algorithm for its creation. The third section presents data on the implementation of software and hardware and test results. The fourth section is devoted to labor protection in relation to the norms of work with the use of electronic devices. The conclusions provide an analysis of the work performed and the results of research and development.

In Annex A show the program code used in the project. In general, the bachelor's thesis without annexes contains 73 pages, 51 figures, 2 tables, 29 reference sources.

Key words: automated alert system, ESP32-CAM, motion sensor, software and hardware, bot, telegram-messenger.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	–	Application programming interface
AWS	–	Amazon Web Services
BOT	–	Robot
BT	–	Bluetooth
FTDI	–	Future Technology Devices Interglobal
GND	–	GROUND
GPIO	–	General-purpose Input/Output
I2C	–	Inter-Integrated Circuit
IP	–	Internet Protocol
IoT	–	Internet of things
LTP	–	Line Print Terminal
MQTT	–	Message queuing telemetry transport
QoS	–	Quality of service
SDK	–	Software development kit
SPI	–	Serial Peripheral Interface
TCP	–	Transmission Control Protocol
TTL	–	Time to live
TV	–	Television
UART	–	Universal asynchronous receiver/transmitter
USB	–	Universal Serial Bus
VCC	–	Voltage Common Collector
Wi-Fi	–	Wireless Fidelity
МПкс	–	Мегапіксель
ПК	–	Персональний комп’ютер

ВСТУП

Актуальність дослідження. Важливою складовою стрімкого зростання актуальності науки на сучасному етапі розвитку суспільства є комп’ютерні технології, що передбачають математичні проєктування явищ і процесів.

Сучасні месенджери надають користувачам величезну кількість можливостей, окрім їх головної спрямованості, а саме реалізації швидкого обміну повідомлень для комунікації між людьми, компаніями, месенджери включають в себе функціональні можливості, за допомогою яких, маємо змогу інтегруватися з різноманітними сервісами.

Розробка та подальше використання спеціально розроблених застосунків для керування віддаленим відеообладнанням через мобільні месенджери дозволяє швидко обмінюватися даними між пристроями. Інтеграцію значно полегшує добре задокументовані API з відкритим доступом для створення програмного забезпечення на основі ботів.

Не менш важливою є компактність пристрою та швидкість, з якою дані можуть бути отримані з віддаленого обладнання.

В даний час ключову роль в процесах автоматизації займають системи дистанційного керування (СДК). Такими інструментами програмування є мікроконтролери, які входять до складу сучасної побутової та промислової електроніки. Мікроконтролер - це мікросхема для управління електронним обладнанням, незалежно від типу архітектури та складності.

Мета та завдання дослідження.

Мета дослідження – розробити апаратно-програмний модуль системи оповіщення на базі ESP32-CAM з можливістю прийняття зображень з камери за допомогою спеціалізованого боту.

Завдання:

1. Проаналізувати проєкти на базі мікроконтролера ESP32-CAM.

2. Дослідити алгоритм передачі зображень від апаратного модуля до месенджеру.
3. Проаналізувати API створення ботів у месенджерах.
4. Розробити Telegram-бот з відповідною системою команд.
5. Зібрати і налаштувати апаратно-програмний модуль.
6. Протестувати апаратно-програмний модуль.
7. Дослідити вимоги та стандарти з охорони праці.

Об’єкт дослідження – технологія інтеграції мікропроцесорної системи зі спеціалізованими ботами для месенджерів.

Предмет дослідження – особливості розробки системи оповіщення в апаратно-програмних мережах на основі ESP32 камер.

Практичне значення отриманих результатів полягає у дослідженні особливостей створення та реалізації ефективного алгоритму обробки інформації для технології оповіщення на основі ESP32-CAM, датчика руху та їх інтеграції у спеціалізований бот для Telegram-месенджера.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ПАТЕНТНОЇ ІНФОРМАЦІЇ У СФЕРІ АПАРАТНО-ПРОГРАМНИХ МОДУЛІВ

Інтернет речей (англ. Internet of Things, IoT) – це об'єднання будь-яких об'єктів в мережу для покращення та розширення їхньої функціональності. Найбільш поширеним прикладом реалізації можна вважати «розумний будинок», ця система здатна автоматизовано підтримувати комфортабельну температуру, вологість повітря, та інше. Все це можливе завдяки спеціальним датчикам, саме вони дозволяють виміряти поточні показники, а далі система передає певний сигнал до підключених пристроїв, таких як кондиціонер, термостат, зволожувач повітря або інші прилади – з потрібними налаштуваннями.

Однією з важливих складових Інтернету речей є засоби та інструменти інтеграції IoT-пристроїв з мобільними терміналами користувачів, зокрема із месенджерами, найпопулярнішими з яких є Telegram, Discord, Viber, WhatsApp. Функціональні можливості месенджерів дозволяють об'єднувати у мережі не тільки людей, але й технічні пристрої, зокрема IoT-пристрої (рис. 1.1).

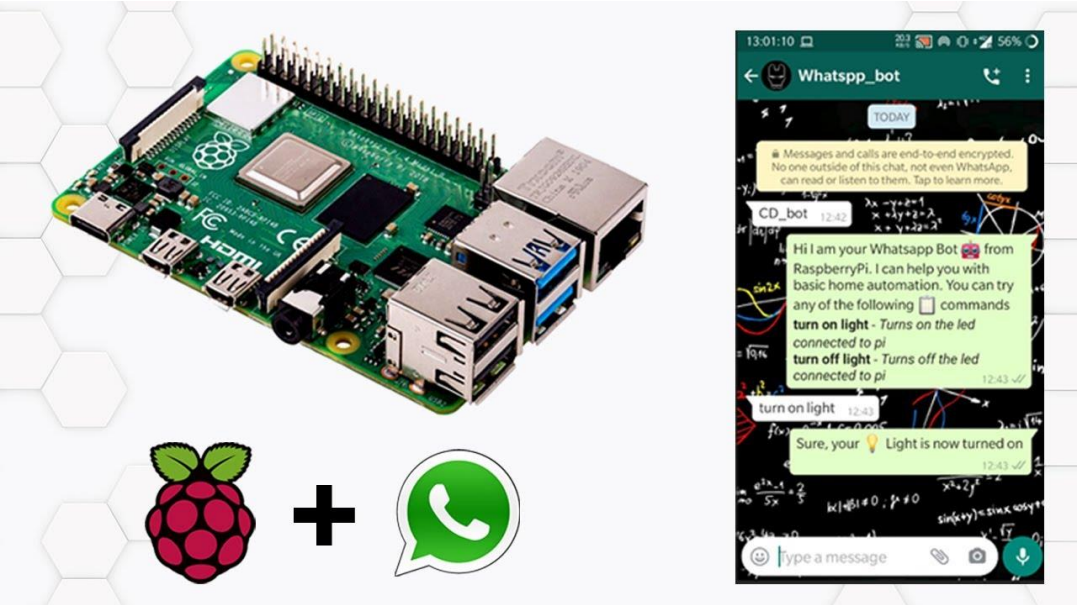

Рисунок 1.1 – Приклад взаємодії мікроконтролера і WhatsApp-бота

Вищезгадана технологія «розумного» будинку заснована на IoT. Більше того, ринок таких систем стрімко зростає (рис. 1.2).

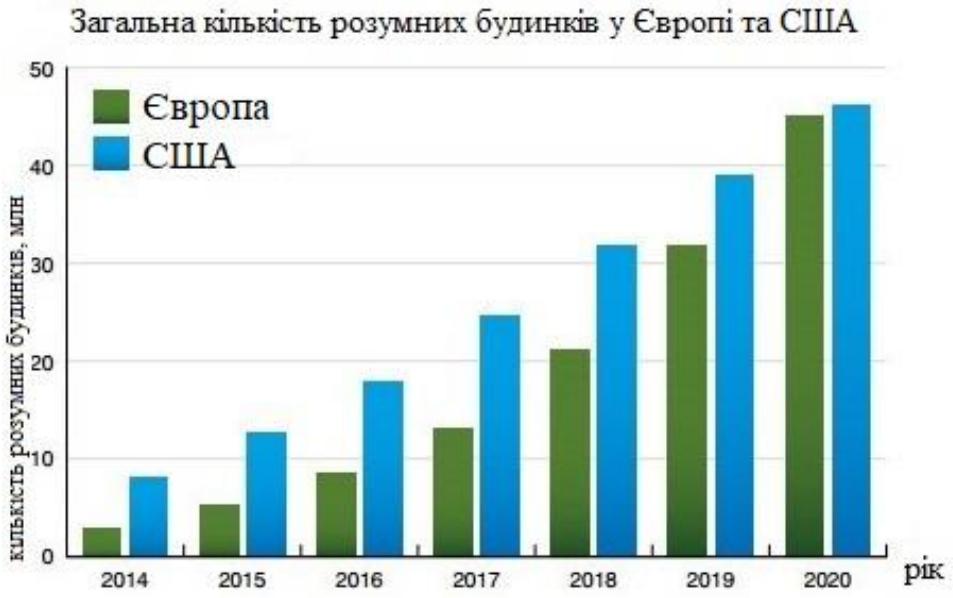

Рисунок 1.2 – Тенденції розвитку розумного будинку

За допомогою смартфона і застосунка можна знімати показання датчиків і керувати системами «розумного» будинку, що знаходиться за сотні кілометрів від дому. Часто таке обладнання для вашого «розумного» будинку коштує недешево.

«Розумний» будинок можна розділити на два основних поняття, які використовуються в повсякденному житті. Перша концепція виконується за допомогою голосового запиту до голосового помічника або за допомогою спеціально запрограмованого додатка для компонентів системи. Додаток можна встановити на смартфон і отримати доступ до «розумних» пристроїв через логічний інтерфейс користувача. Наприклад, ви можете попросити голосового помічника увімкнути кондиціонер, закрити жалюзі, увімкнути опалення або просто включити улюблений фільм. Також використовуються спеціальні пульти, які за замовчуванням можуть підключатися до «розумних» пристроїв і керувати ними. Після обробки запиту користувача система надсилає сигнал «розумним» пристроям на виконання встановленої команди (рис. 1.3).

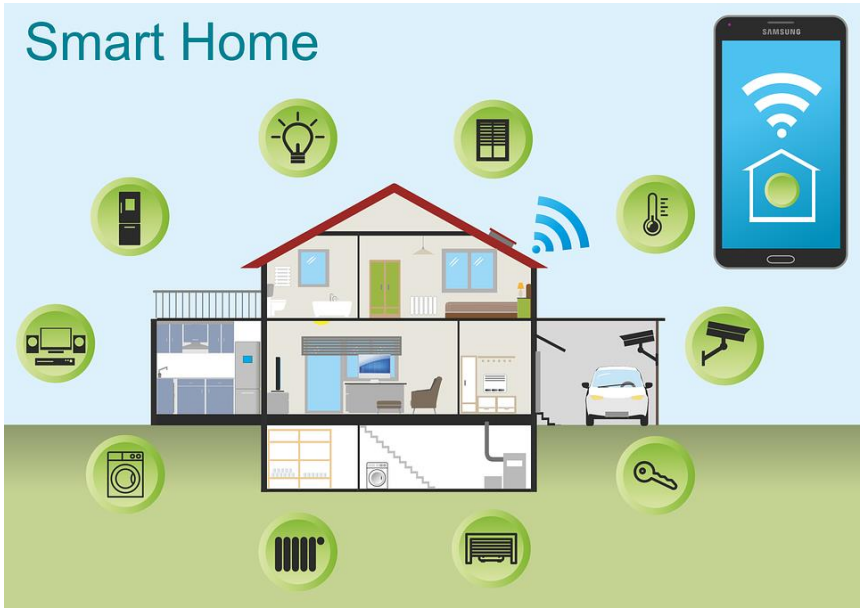

Рисунок 1.3 – Концепція управління розумним будинком

Друга концепція передбачає можливість не перебуваючи в будинку використовувати «розумні» пристрої. Кожен пристрій вже має власне програмування для виконання послідовності дій. Коли ви встановлюєте ці «розумні» пристрої, вони вже запрограмовані на виконання своїх завдань, які починають працювати в певний час. Це може бути проста система включення чайника в потрібний час вранці або наливання води для тварин. Також закладено логіку самостійного прийняття системою рішень у відповідних ситуаціях.

Існує 3 основних елементи, які будують «розумний» будинок. Перш за все, це датчики. Вони можуть отримувати інформацію як із зовнішнього, так і із внутрішнього середовища «розумного» будинку. По-друге, це головний контролер. Він обробляє всі вхідні події, які фіксуються датчиками, і запускає відповідний алгоритм для реалізації запиту. І по-третє, це «розумне» обладнання. Усі технології, які полегшують життя користувачам.

Основний зв'язок між системою і датчиками може підтримуватися як на безпровідному, так і на кабельному з'єднанні. Використання кабельної версії може зменшити кількість проблем із зв’язком та надійністю. виробники такого обладнання пропонують використовувати кабельний зв'язок.

1.1 Аналіз аналогічних проєктів

Багато відкритих проєктів для «розумного» будинку можна знайти на таких платформах, як Kickstarter і ArduinoHub.

1.1.1 Камера спостереження на базі модуля ESP32-CAM

Одним із проєктів є камера спостереження на базі модуля ESP32-CAM, Arduino Nano, датчика руху та програми VoIP Discord [1]. Камера спостереження реагує на рух інфрачервоним датчиком і публікує фотографії на створеному сервері в Discord. Випробувальна схема збирається за допомогою макетної плати та перемикачів (рис. 1.4).

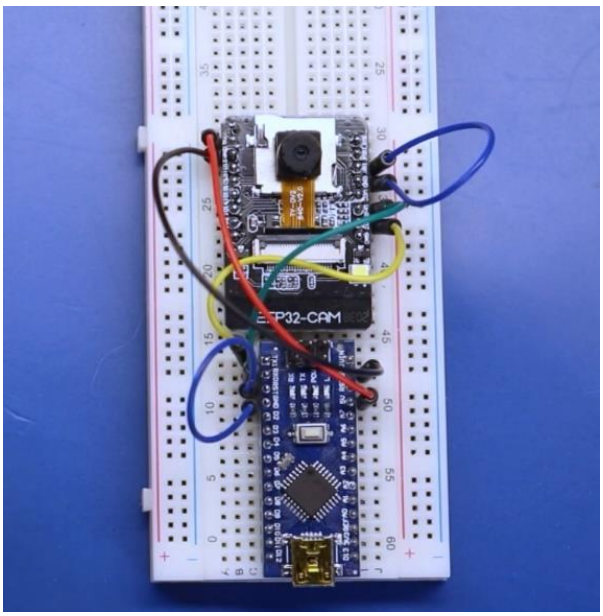

Рисунок 1.4 – Програмування модулю за допомогою Arduino Nano

На рис. 1.5 показано відправлення знімків до програми Discord з вищерозглянутого пристрою.

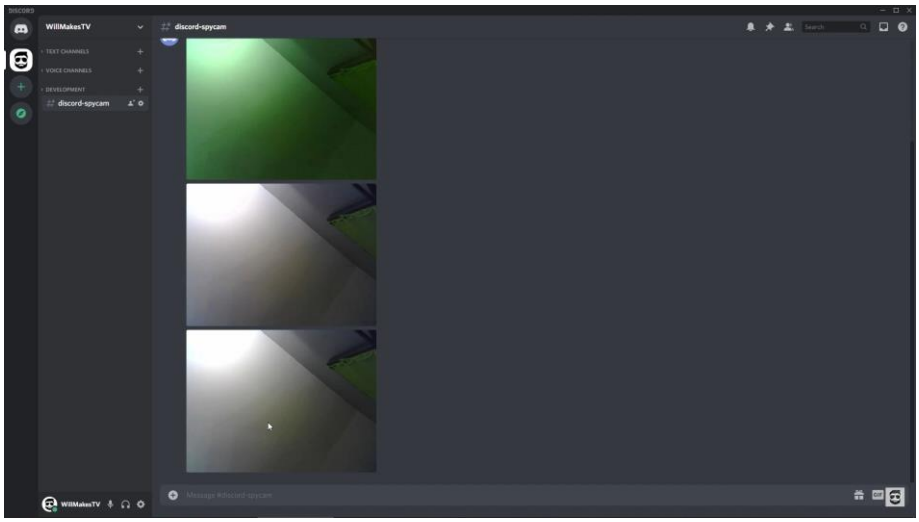

Рисунок 1.5 – Відправка знімків до Discord

Управління системою здійснюється за допомогою скрипту Python, який обробляє запити HTTPS.

Для більш зручного використання приладу можна роздрукувати на 3D-принтері стельові корпуси для інфрачервоного датчика та камери (рис. 1.6).

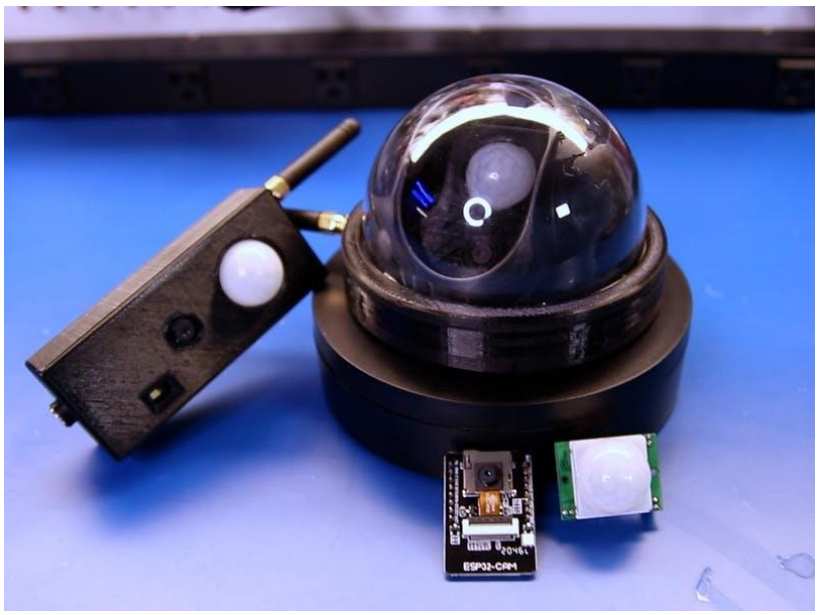

Рисунок 1.6 – Збірка компонентів камери

1.1.2 Робот з дистанційним керуванням по Wi-Fi

Ідея наступного проєкта – робот з дистанційним керуванням по Wi-Fi за допомогою ESP32-CAM [2]. З можливістю керування роботом за допомогою веб-сервера, який відображає потокове відео того, що робот «бачить». Також

є можливість дистанційно керувати роботом, навіть якщо його не видно. ESP32-CAM буде запрограмований за допомогою Arduino IDE.

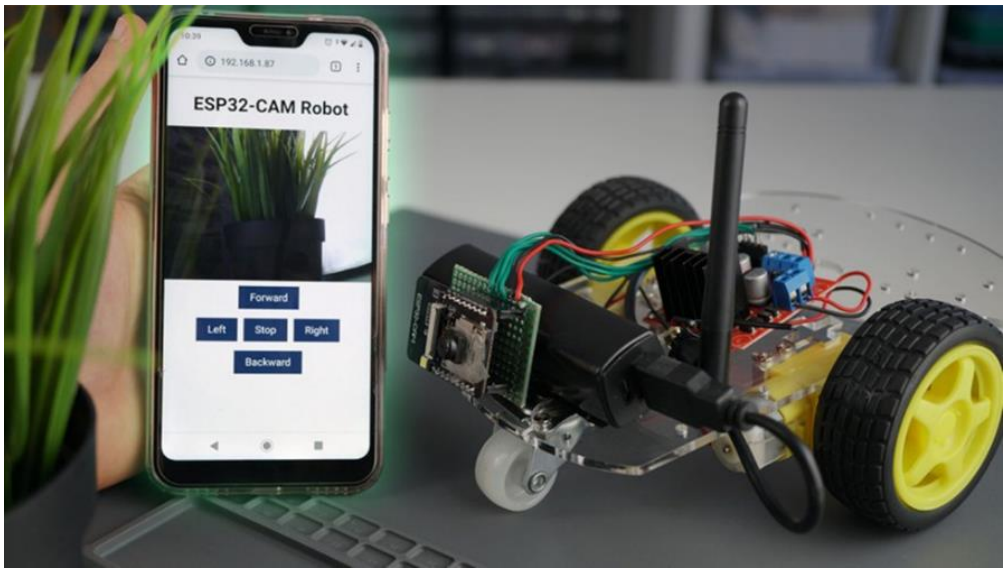

Рисунок 1.7 – Робот на базі ESP32-CAM

Керування роботом здійснюється через Wi-Fi за допомогою ESP32-CAM. Для цього створено веб-інтерфейс для керування роботом, до якого можна отримати доступ з будь-якого пристрою у локальній мережі.

Веб-сервер має 5 елементів керування: Вперед, Назад, Вліво, Вправо та Зупинити. (рис.1.8).

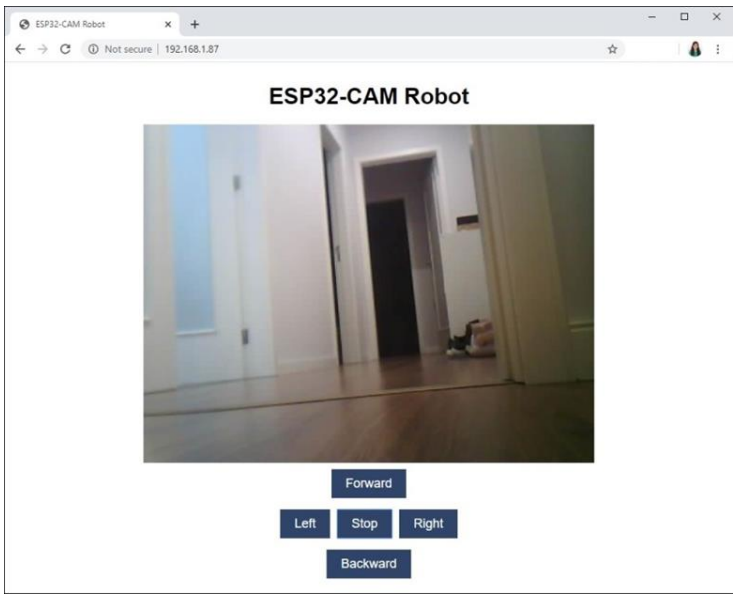

Рисунок 1.8 – Веб-інтерфейс для керування роботом

Схема проєкту представлена на рис. 1.9

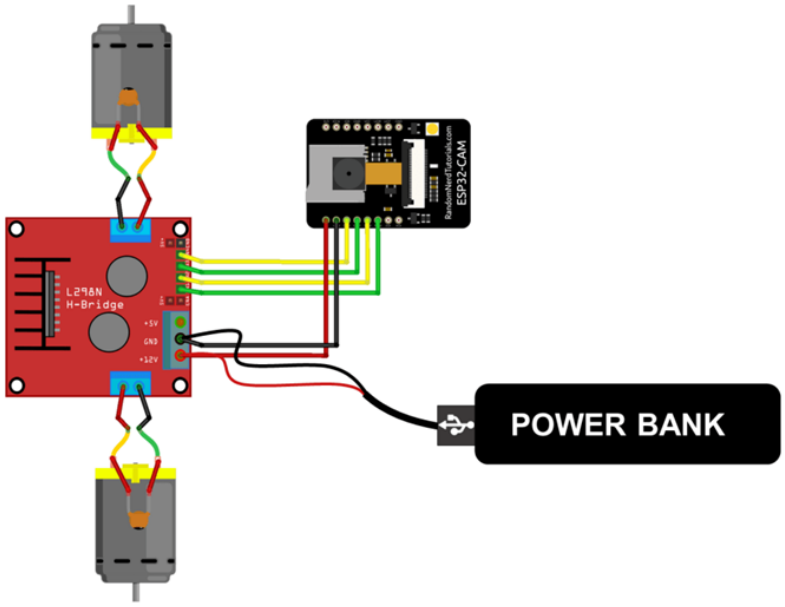

Рисунок 1.9 – Схема підключення компонентів

Для спрощення схеми, живлення робота (двигуни) і ESP32 реалізовано за допомогою одного джерела живлення. В наведеному проєкті використовується павербанк/портативний зарядний пристрій (як ті, що використовуються для зарядки вашого смартфона).

1.1.3 PyroLight

Третій проєкт – PyroLight [3]. Використання невеликої кількості датчиків, таких як пірометр, сервомотор. Основна мета проєкту полягає в тому, щоб автоматизований перемикач на основі серводвигуна можна було модернізувати на поточний розподільний щит. Сучасні рішення для домашньої автоматизації не є портативними і вимагають професіоналів для встановлення проводки та інших налаштувань. Схема проєкту представлена на рис. 2.10.

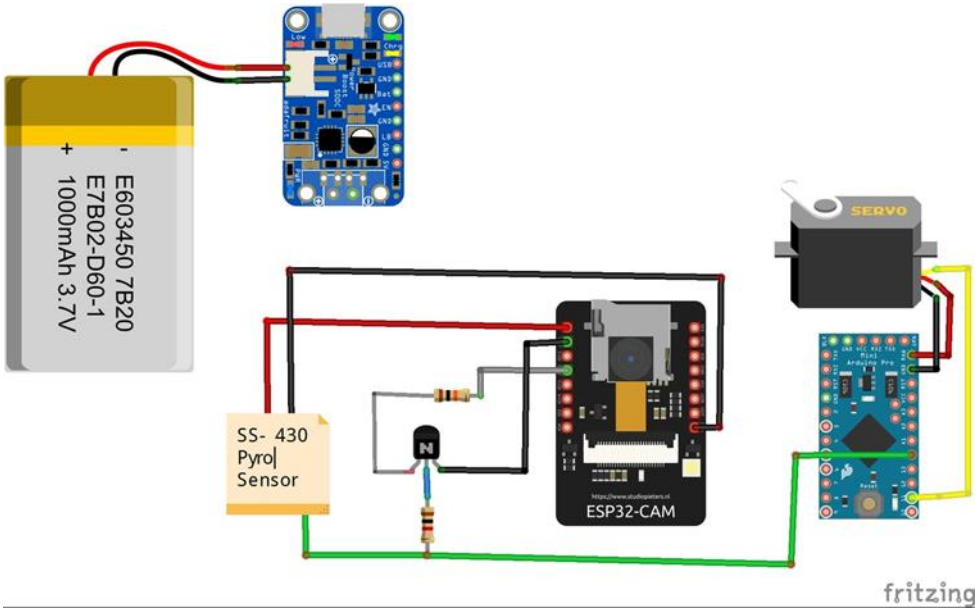

Рисунок 1.10 – Схема підключення компонентів

Сигнал піродатчику передається на транзистор у конфігурації з відкритим колектором, після отримання сигналу спрацьовування транзистора як перемикача. Потім GPIO 13 підключається до землі, і камера ESP32 активується.

Пластиковий корпус допомагає зменшити помилкові спрацювання (рис. 1.11).

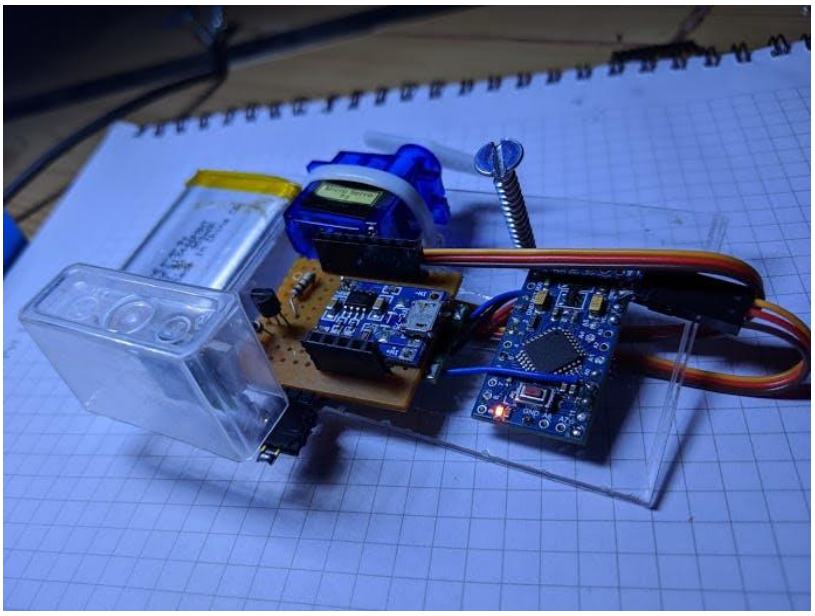

Рисунок 1.11 – Прототип пристрою у пластиковому корпусі

ESP32-CAM переходить у режим сну після відтворення знімку і не може виконувати операції підготовки цифрового сигналу. Для цього передбачений додатковий мікроконтролер для запобігання помилкової роботи тригеру, який також можна використовувати для управління серводвигуном.

Причина використання цієї моделі ESP32 полягає в тому, щоб робити знімок, після чого відбувається перехід до режиму сну, коли у кімнату входить людина, або відбувається незапланована дія у оселі. Зображення зберігається на SD-карті. Крім того, якщо виявлено проникнення злодія, ESP32 надсилає термінові повідомлення на налаштований ідентифікатор пошти. Для цієї мети можна використовувати бібліотеку поштового клієнта ESP32 Mail client.

У зібраному вигляді проєкт представлений на рис. 1.12.

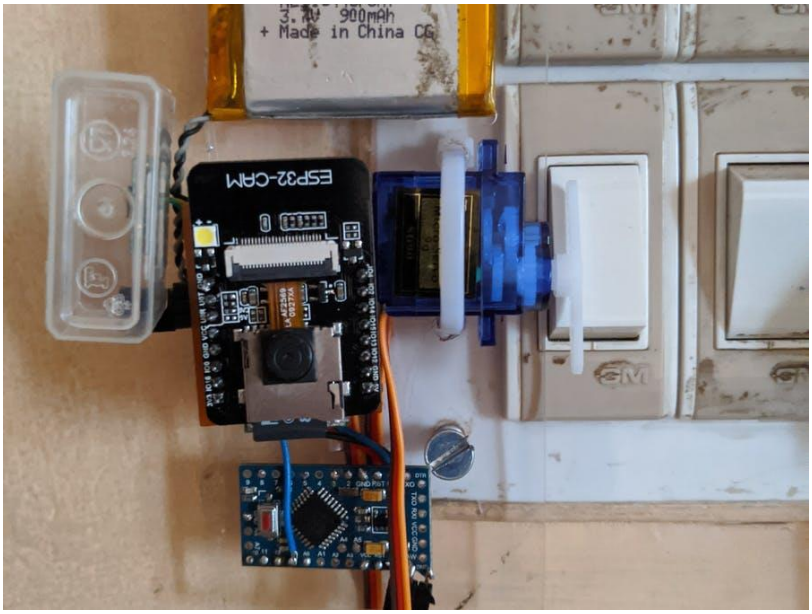

Рисунок 1.12 – PyroLight

1.1.4 Color Detection and Tracking

Наступний проєкт - Color Detection and Tracking - може сподобатися людям, зацікавленим у машинному навчанні, оскільки він заснований на розпізнаванні шаблонів за допомогою фреймворку OpenCV (OpenCV.js) [4].

Ви можете запустити веб-сервер на ESP32. JavaScript та інші мови браузера мають можливість використовувати переваги ESP32 та його камери, які недоступні на інших платах, таких як ESP8266.

Згідно документації docs.opencv.org, «OpenCV.js – це зв'язування JavaScript для підмножини функцій OpenCV на веб-платформах». OpenCV.js використовує Emscripten, компілятор LLVM-JavaScript, який допомагає компілювати функції OpenCV для бібліотеки API.

OpenCV.js працює через браузер, це дозволяє використовувати функції OpenCV для тих людей, які мають невеликий досвід роботи з HTML і JS. Для тих, хто вже працював у програмах ESP32-CAM, вони вже мають цей досвід.

Цей проєкт відтворює веб-сервер, який дозволяє відстежувати кольори рухомого об'єкта. Інтерфейс веб-сервера має набір UX з кількома конфігураціями для правильного відображення кольорів, що відстежуються. Потім браузер виводить координати x і y центру рухомого об'єкта в реальному часі на плату ESP32-CAM (рис. 1.13).

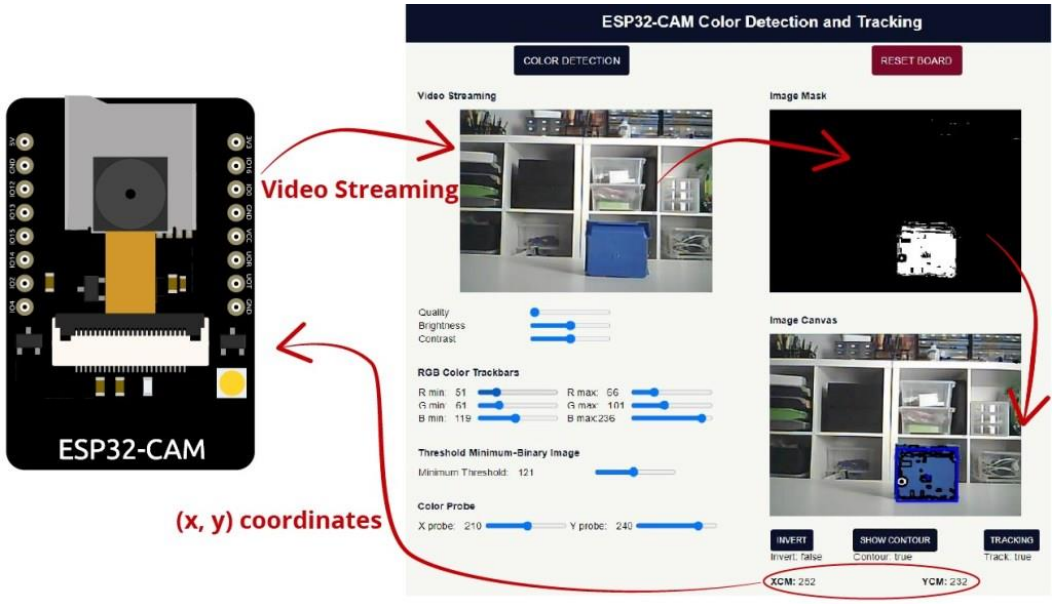

Рисунок 1.13 – Web-сервер OpenCV

На рис. 1.14 показано виявлення кепки червоного кольору в зоні зі звичайною люмінесцентною лампою потужністю 60 Вт. Лампа випромінює 3 спектри: червоний, зелений і синій. Червона кепка відображає всі три спектри, але переважно червоного кольору. Метод дозволяє встановити баланс RGB за допомогою цих мінімальних ефектів.

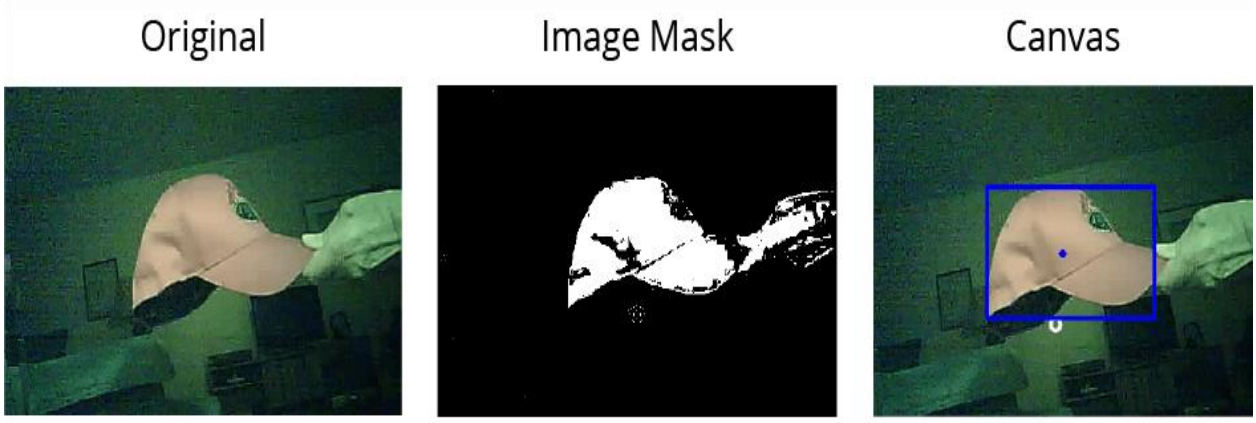

Рисунок 1.14 – Розпізнавання кольору

Ця ж техніка використовується для розпізнавання будь-якого іншого об’єкта, який з’являється в полі зору цього модуля.

1.1.5 Точка доступу на основі ESP32-CAM

Нарешті, подивимося на проєкт точки доступу на основі ESP32-CAM – ESP32-CAM: Access Point (AP) for Web Server [5].

Маршрутизатор здійснює роль точки доступу, а ESP32-CAM працює в режимі робочої станції (рисунок 1.15).

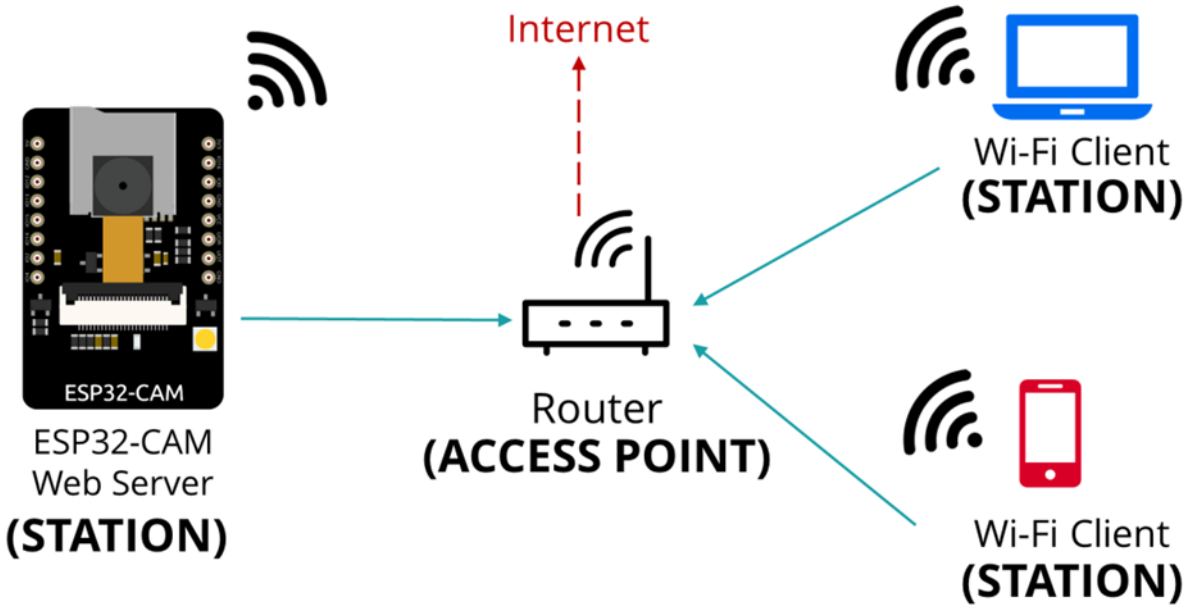

Рисунок 1.15 – Точка доступу ESP32-CAM

У деяких випадках це може бути не найкраща конфігурація (якщо поблизу немає роутера). Якщо карта ESP32-CAM встановлена як точка

доступу (hotspot), можна підключитися за допомогою будь-якого пристрою, який підтримує технологію Wi-Fi, без підключення до маршрутизатора. [16]

По суті, коли ESP32-CAM встановлений як точка доступу, він створює власну мережу Wi-Fi та пристрої, які підтримують Wi-Fi (наприклад, смартфони та комп'ютери).

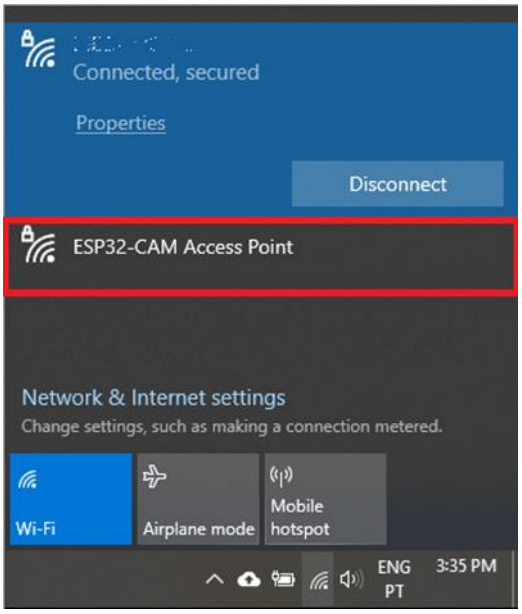

Рисунок 1.16 – Підключення до мережі ESP32-CAM

Щоб підключитися до точки доступу на комп'ютері, вибирається «Точка доступу ESP32» і вводиться пароль, встановлений під час прошивки.

Щоб отримати доступ до веб-сервера ESP32-CAM, просто вводиться IP-адресу 192.168.4.1 у браузері та отримується зображення з камери.

1.2 Системи відеоспостереження на основі IP-камер

Основними перевагами відеоспостереження з цифровими відеокамерами є можливість спостереження за віддаленими об'єктами, виявлення окремих елементів зображення та руху та автоматичного реагування на певні типові ситуації.

Зокрема, поширеними є IP-камери відеоспостереження або «інтернет-протокольні» камери відеоспостереження використовують Інтернет для отримання та зберігання зображень. До переваг IP-камер можна віднести:

– Інтеграція з локальними мережами будь-якого рівня складності;
– IP-камера може працювати в автономному режимі;
– Безпровідні камери не потребують прокладки додаткових проводів;
– Можливість шифрування відеопотоку.

IP-камера може підключатися безпосередньо до вашої мережі або комп’ютера використовуючи один з методів:

– WiFi – безпровідне з’єднання;
– Ethernet – звичайне підключення через кабель витої пари;
– PoE – також кабельне підключення, тільки живлення подається за деякими жилами крученої пари, а не окремо. Потрібен спеціальний інжектор та провід з додатковим виходом.

Загальна схема підключення відеокамер спостереження виглядає наступним чином (рис. 1.17).

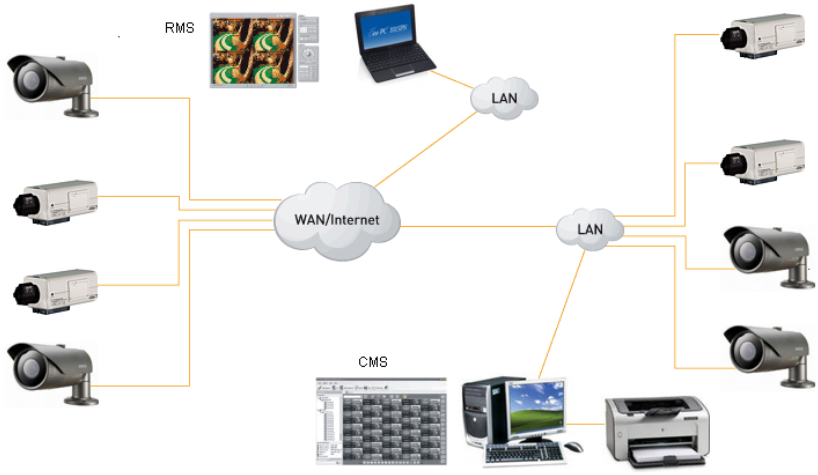

Рисунок 1.17 – Схема підключення системи відеоспостереження

Розподілена система відеоспостереження дозволяє розширити до 500 каналів для IP-камер. Зазвичай використовується спеціальне програмне забезпечення, яке працює на різних ПК:

– центральна система управління CMS (Central Management System).
– система віддаленого управління RMS (Remote Management System).

IP-камери мають вбудований мережевий порт, що дозволяє підключатися до мережі Ethernet, через яку вона підключається до CMS.

Система дозволяє відстежувати зображення приблизно 100 відеоканалів, записувати відео та звичайні зображення з IP-камери на жорсткий диск. Автоматично перевіряється підключення до головного сервера та мережевого сховища, де зберігаються основні відео з камер.

Розподілена система дозволяє виконувати:

– централізований моніторинг камер;

– зйомку з камер, які знаходяться далеко одна від одної (або в різних містах).

IP-камери випускають для експлуатації у приміщенні, на вулиці, у транспорті.

Вуличні IP-камери відповідають певним вимогам:

– Відповідність корпусу міжнародної класифікації із захисту оболонок від пилу та вологи International Protection Marking (IP). Перша цифра – захист від сторонніх предметів (пилу), друга – від проникнення води. Корпус вуличних IP-камер повинен відповідати стандарту (IP54 ~ IP68);

– Діапазон робочих температур (за умови відповідності корпусу стандарту захисту IP54 ~ IP68). При нижньому кордоні мінус 10 ° C - камеру можна встановити в неопалюваному приміщенні, при мінус 20 ° C - на вулиці, при мінус 40 ° C - майже скрізь, а при мінус 60 ° C - на відкритому повітрі навіть в районах Крайньої Півночі (у таких IP-камерах є захист від корозії та зледеніння).

Транспортні IP-камери - спеціалізоване обладнання, захищене від вібрації, укомплектоване спеціальними надійними роз'ємами (як правило, різьбовими M12). Кожна транспортна IP-камера проходить обов'язкову сертифікацію на відповідність.

Незалежно від призначення, для встановлення в місцях, що не потрапляють до зони видимості охоронців, випускають IP-камери, захищені від механічних впливів різного ступеня – корпус відповідає коду міжнародної класифікації IK08 ~ IK10.

Висновки до розділу 1

Підсумовуючи перший розділ, можемо зробити такі висновки:

– Визначено, що мікроконтролер можна розглядати як окрему систему з процесором, пам’яттю та периферійними пристроями та використовувати як вбудовану систему. Більшість мікроконтролерів, які використовуються сьогодні, вбудовані в інше обладнання, таке як автомобілі, телефони, прилади та комп’ютерні периферійні пристрої.

– Розглянуто існуючі проєкти на базі ESP32-CAM.

– Визначено підходи до проєктування систем з використанням месенджер-ботів.

– Виконано порівняння мікроконтролера з традиційними IP-камерами.

РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ

Важливими частинами проєктування є створення алгоритму роботи програмного модуля та вибір необхідних технологій, які зможуть правильним чином вписатися у логіку самого проєкту. Це визначить швидкість програми та її роботу в цілому.

2.1 Підходи до проєктування системи та алгоритм її створення

У сучасних пристроях логіка роботи заснована на використанні керуючих пристроїв - мікроконтролерів, які виступають в якості інтерфейсу між кінцевим користувачем і його керуючим пристроєм, як правило, мобільним телефоном, який у сучасному світі є надійним помічником кожному і має всі необхідні функції користувача для взаємодії з навколишнім світом за допомогою програмних додатків, функціональність яких дозволяє виконувати завдання системи сповіщень і виконує цей процес по радіоканалу, який використовує вбудовані чіпи мобільного телефону, через які здійснюється з'єднання між мобільним телефоном та іншими пристроями за допомогою бездротових дистанційних технологій Wi-fi або Bluetooth.

Рисунок 2.1 – Схема віддаленої взаємодії мобільного пристрою з різними виконавчими кінцевими пристроями

Управління мікроконтролером може здійснюватися в різних режимах роботи:

– імпульсний режим;

– дистанційно-релейний режим.

Імпульсний режим – це коли виходи та реле, вбудовані в мікроконтролер, керуються імпульсним сигналом. Тобто сигнал від пристрою передавача подається на приймач у мікроконтролер у вигляді короткого імпульсу. Режим роботи може бути однонаправленим або двонаправленим. У двонаправленому режимі стан контактів вихідного реле можна контролювати за допомогою вбудованого світлового індикатора [6].

Дистанційний релейний режим - це коли контакти реле керуються безперервним сигналом, тобто передача керуючого сигналу від пристрою керування є постійною в часі, протягом якого реле замикається до тих пір, поки контакт замкнутий відповідний канал пристрою, з якого передається керуючий сигнал, сигнал передається кожні 80 секунд під час роботи. Цей сигнал оновлює стан входів для синхронізації з виходами сигналу прийому пристрою, вбудованого в мікроконтролер. Цей режим має більший захист від зовнішніх впливів або втрати живлення. Моніторинг виходу здійснюється в режимі реального часу. Управління здійснюється в ручному або автоматично запрограмованому режимі із зовнішніх пристроїв керування сигналізацією.

Протоколи віддаленого зв’язку – це набір промислових специфікацій для бездротових персональних мереж, реалізованих як інтерфейс, за допомогою якого пристрої, що підтримують цю технологію, можуть взаємодіяти один з одним. Вони забезпечують обмін інформацією на різних пристроях, за допомогою яких може здійснюватися процес дистанційного керування, наприклад, керування зі спеціального пульта або мобільного телефону за допомогою мобільного додатка.

До таких пристроїв належать [7]:

- персональні комп'ютери;

- мобільні телефони;
- принтери та різноманітна офісна периферія;
- акустичні системи;
- навушники та гарнітури;

Розглянемо основні технології дистанційного обміну даними, які наразі доступні для використання і не потребують спеціального дозволу є безкоштовними та доступними для розробки власних продуктів (рис. 2.2).

Technology	Wi Fi 802.11b	Bluetooth 802.15.1	ZigBee 802.15.4
Target application	Web, Email, Video	Voice, Cable replacement	Monitoring and control
Flash size	1MB+	250KB+	25KB - 70KB
Battery life (days)	0.5 - 5	1 - 7	100 - 1,000+
Basic network	32 / …	7 / …	255 / 65535
Throughput (kbps)	11,000+	1000	20 - 250
Range (m)	1 – 100+	1 - 100+	1 – 75+
Advantage	Speed, flexibility (IP)	Price, convenience	Low power, simple

Рисунок 2.2 – Технології та стандарти бездротової передачі даних

Wi-Fi – це технологія бездротової локальної мережі з пристроями, заснованими на стандартах IEEE 802.11. Логотип Wi-Fi є торговою маркою Wi-Fi Alliance. Під абревіатурою Wi-Fi (від англійської фрази Wireless Fidelity [2], що дослівно можна перекласти як «бездротова точність»), в даний час розробляється сімейство стандартів для передачі цифрових потоків даних по радіоканалах. Основні діапазони Wi-Fi – 2,4 ГГц (2412 МГц-2472 МГц) і 5 ГГц (5160-5825 МГц).

Сигнал Wi-Fi можна передавати на кілометри навіть при низькій потужності передачі, але для отримання сигналу Wi-Fi від звичайного Wi-Fi маршрутизатора на великі відстані потрібна антена з високим коефіцієнтом посилення (наприклад, параболічна антена або Wi-Fi гармата).

Будь-яке обладнання, яке сумісне зі стандартом IEEE 802.11, можна протестувати за допомогою Wi-Fi Alliance, а також сертифікувати та отримати ліцензію на використання логотипу Wi-Fi.

Переваги Wi-Fi.

Дозволяє розгортати мережу без прокладки кабелю, що може зменшити витрати на розгортання та розширення мережі. Зони, де не можна прокладати кабелі, наприклад, на відкритому повітрі та в будівлях, що мають історичну цінність, можуть обслуговуватися бездротовими мережами.

Дозволяє отримати доступ до мережі мобільних пристроїв.

Пристрої Wi-Fi широко поширені на ринку. Сумісність обладнання гарантується обов'язковою сертифікацією обладнання з логотипом Wi-Fi. Мобільність. Ви більше не прив’язані до одного місця і можете користуватися Інтернетом у зручному для вас середовищі. У межах зони Wi-Fi кілька користувачів з різних пристроїв можуть отримати доступ до Інтернету [8].

На сьогоднішній день пропускна швидкість передачі даних по Wi-Fi значно виросла (рис. 2.3).

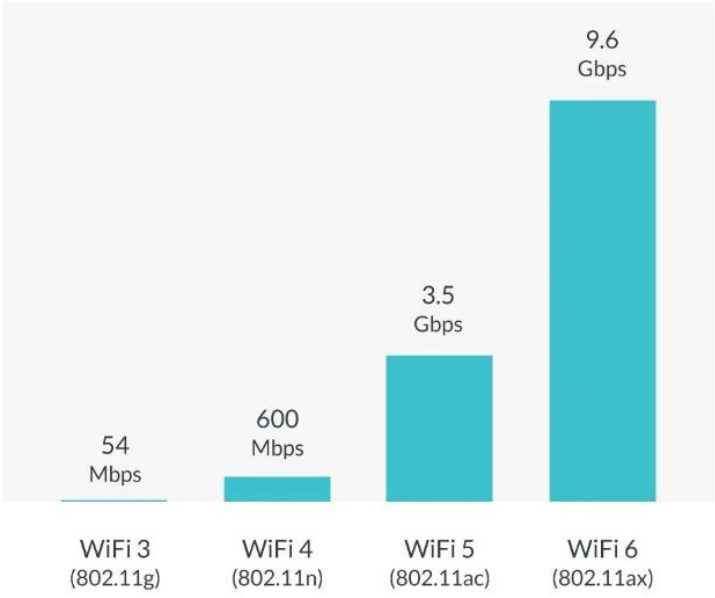

Рисунок 2.3 – Максимальна пропускна швидкість Wi-Fi

2.2 Обґрунтування вибору програмно-апаратних засобів для керування системою оповіщень

Було визначено, що технології «розумного» дому засновані на різноманітних датчиках і пристроях, керування якими здійснюється через глобальну мережу. Основними комунікаційними пристроями для таких систем можуть бути контролери на основі:

- ESP8266
- ESP32
- WINC 1500
- 32u4 FONA

На даний момент одним з найпопулярніших мікроконтролерів з підтримкою Wi-Fi є ESP32. У цьому проєкті планується реалізувати систему, з компонентів, які взаємодіють через об'єкт управління, роль якого виконуватиме мікроконтролер.

1. ESP8266 - мікроконтролер від китайського виробника Espressif Systems з інтерфейсом Wi-Fi. Крім Wi-Fi, у мікроконтролері відсутня флеш-пам’ять в SoC, програми користувача запускаються із зовнішньої флеш-пам’яті з інтерфейсом SPI [17].

Мікроконтролер привернув увагу в 2014 році завдяки випуску перших продуктів на його основі за надзвичайно низькою ціною.

Навесні 2016 року почалося виробництво ESP8285, що поєднує ESP8266 і 1 Мб флеш-пам'яті. Восени 2015 року Espressif презентував розробку лінійки – чіп ESP32 та модулі на його основі.

- мікроконтролер 80 МГц;
- 32-розрядний процесор Tensilica;
- можливий розгін до 160 МГц;
- Wi-Fi IEEE 802.11 b/g/n;
- підтримка WEP і WPA/WPA2;

– 14 портів вводу-виводу (11 з яких можна використовувати), SPI, I²S, UART, 10-розрядний АЦП. I²C можливий лише за допомогою біт-банг;

– блок живлення 2,2 ... 3,6 В;

– споживання - до 215 мА в режимі передачі, 100 мА - в режимі прийому, 70 мА - в режимі очікування;

– підтримка трьох режимів низької потужності, всі без підключення до точки доступу: режим сну модему (15 мА), легкий режим сну (0,4 мА), глибокий сон (15 мА);

– мікроконтролер не має власної енергонезалежної пам'яті на чіпі. Виконання програми здійснюється із зовнішнього SPI ROM шляхом динамічного завантаження необхідних частин програми в кеш інструкцій. Завантаження апаратне, прозоре для програміста;

– підтримка до 16 МБ зовнішньої пам'яті програм. Можливий стандартний, подвійний або чотириразовий інтерфейс SPI;

Електричні параметри, розетки, схеми підключення можна знайти в документах 0A-ESP8266EX__Datasheet і 0B-ESP8266__System_Description з Espressif SDK.

Джерело програми ESP8266 встановлюється станом портів GPIO0, GPIO2 і GPIO15 в кінці сигналу скидання (тобто джерела живлення). Два найбільш цікавих режими: виконання коду з UART (GPIO0 = 0, GPIO2 = 1 і GPIO15 = 0) і із зовнішнього ПЗУ (GPIO0 = 1, GPIO2 = 1 і GPIO15 = 0).

Інструменти розробки.

Інструменти розробки програмного забезпечення (програмний пакет розробника, SDK) складаються з:

Компілятори. Компілятор Xtensa LX106 входить до пакету компілятора GNU Compiler Collection. Компілятор має тексти з відкритим вихідним кодом. Різні пакети SDK можуть містити різні збірки цього компілятора, які дещо відрізняються від підтримуваних параметрів.

Бібліотеки для роботи з периферією контролера, стеками WiFi, TCP/IP.

Засоби завантаження виконуваного файлу в пам'ять програм мікроконтролера.

Додаткова IDE.

Espressif вільно поширює свій набір розробників. Цей набір включає компілятор GCC, бібліотеки Espressif та утиліту завантаження XTCOM. Бібліотеки представлені у вигляді скомпільованих бібліотек, без вихідного коду. Espressif підтримує дві версії SDK: одна на основі ОСРВ, інша на основі зворотного виклику.

Крім офіційного SDK, існує ряд альтернативних проєктів SDK [8]. Ці пакети SDK використовують бібліотеки Espressif або пропонують власний еквівалент бібліотек Espressif, отриманих за допомогою методів зворотної інженерії.

"esp-open-sdk". Покращена версія Expressif SDK. Включає компілятор GCC та деякі бібліотеки Expressif. Тільки Linux [17].

Неофіційний комплект розробки Михайла Григор'єва. Набір включає інсталятор Windows, компілятор GCC власної збірки з інтеграцією з графічною IDE Eclipse, сучасні набори бібліотек та документації Espressif, а також деякі утиліти. Є російськомовний форум.

«Arduino IDE для ESP8266» – це доповнення до Arduino IDE, яке дозволяє вам програмувати ESP8266 так само легко, як і будь-який інший модуль Arduino. Є мережевий функціонал ESP8266. Компілятор GCC, завантажувач мікропрограми ESPTool.

Детальний російськомовний опис процесу встановлення та доступного API тут, приклад роботи тут.

«Набір інструментів GNU для esp8266». Його можна інтегрувати в Visual Studio.

"ESP8266 GCC Toolchain" Макса Філіппова.

«Sming» – це проєкт, який передбачає додавання сумісних з Arduino бібліотек поверх стандартних бібліотек Espressif, але без середовища Arduino.

Мережева інфраструктура.

Типове використання ESP8266 як апаратної основи Інтернету речей часто передбачає встановлення в будинках або офісах. У цьому випадку підключення до домашньої/офісної локальної мережі з доступом до Інтернету здійснюється через роутер. Користувач пристрою може керувати ним за допомогою планшета або комп’ютера через локальну мережу або віддалено через Інтернет.

У свою чергу, ESP32 – це серія недорогих, малопотужних чіпів від Espressif Systems. Це чіп-система з інтегрованими радіоконтролерами Wi-Fi, Bluetooth і Thread. Серії ESP32 і ESP32-S використовують ядра процесора з архітектурою Tensilica, а серії ESP32-C і ESP32-H використовують ядра з відкритим ядром з архітектурою RISC-V. Мікросхема має вбудований радіочастотний тракт: балансувальний трансформатор, вбудовані антенні перемикачі, радіочастотні компоненти, малошумний підсилювач, підсилювач потужності, фільтри та модулі керування потужністю. ESP32 був розроблений і розроблений компанією з Шанхаю і виробляється TSMC за технологією 40 нм і 28 нм. Серія є наступником чіпів ESP8266.

Рисунок 2.4 – Контролер розробки ESP32-CAM

Плата ESP32 – покращена версія попереднього мікроконтролера ESP8266. Незважаючи на недавній випуск оновленої плати, розробники віддають перевагу саме їй, оскільки в ній покращено ядро з швидшим Wi-Fi, Bluetooth 4.2 і різними входами/виходами (аналоговими/цифровими) [8]. Існує чимало відмінностей між цими двома платами (рис. 2.2).

Specifications	ESP8266	ESP32
MCU	Xtensa Single-Core 32-bit L 106	Xtensa Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	N/A	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash up to 16 MBytes	SPI
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	N/A	1
Ethernet MAC Interface	N/A	1
Touch Sensor	N/A	Yes
Temperature Sensor	N/A	Yes
Working Temperature	-40° C – 125° C	-40° C – 125° C

Рисунок 2.5 – Порівняння ESP32 та ESP8266

Одним з основних недоліків плати ESP8266 є мала кількість контактів на платі. [18] У версії ESP32 це було покращено, і самі піни стали набагато більше, і вони також стали багатофункціональними. [9]

Популярні версії ESP32:

1. ESP32-CAM - трохи відрізняється від інших плат і базується на модулі ESP32-S. Він включає в себе такі інтерфейси, як SPI, UART, I2C, PWM. Модуль може завантажувати зображення через Wi-Fi і підтримує тактову частоту до 160 МГц. Він має на борту 10 портів GPIO (рис.2.6). З підтримкою камери OV2640. На самій платі є слот для карти microSD. Відмінний мікроконтролер для моніторингу території. Для підключення доцільно використовувати роз'єми Dupon. [9]

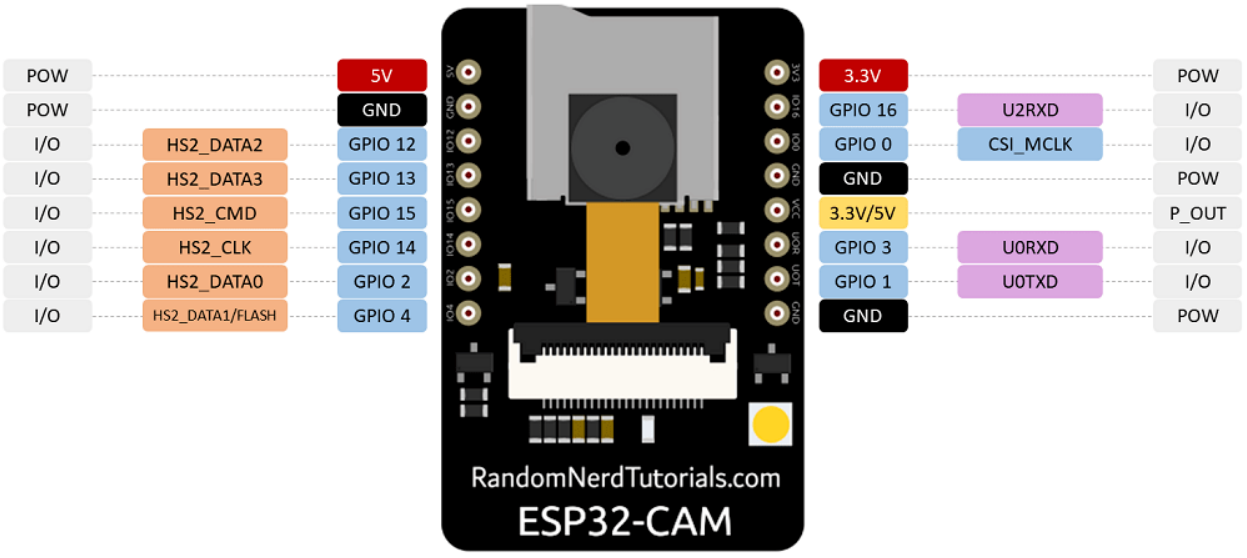

Рисунок 2.6 – Pinout ESP32-CAM

2. Node-MCU-32S – мікроконтролер оснащений підключенням Bluetooth і Wi-Fi, CP2102 і ключами. Головна особливість плати полягає в тому, що піни ESP-WROOM-32 I/O доступні через заголовки розширення (extension headers). Вони також є відкритими і підтримують кілька вихідних кодів. Він добре підходить для великих проєктів і має 32 МБ флеш-пам'яті.

3. ES32-DevKitC - дуже компактна плата. Вхідний і вихідний контакти розділені на контактні заголовки з обох сторін. Це зроблено для полегшення взаємодії. Ви можете підключитися до плати через периферію за допомогою перемичок(jumpers), або макетної плати. Перевага плати не тільки в її розмірах, але і в низькому енергоспоживанні.

4. Adafruit ESP32 Feather - ця плата заснована на ESP32 від Adafruit. Плата оснащена схемою регулятора напруги та інтерфейсом USB-UART. Дуже приємною особливістю цієї плати є те, що на ній є роз'єм для акумулятора Lipoly. Це хороший варіант для портативних проєктів з батарейним живленням. Збірка від Adafruit завжди якісна. Також ця компанія має багато документації для якісної розробки.

Модуль ESP32-CAM OV2640.

На борту цього модуля є спеціальна камера ov2640. За допомогою цієї камери будуть зроблені знімки кімнати, де вона розташована, з роздільною

здатністю 2 МПкс. Цього буде достатньо, щоб чітко відображати об’єкти, які будуть на виду і залишатися в фокусі, без появи артефактів у вигляді пікселації або розмиття.

ESP32-CAM – це невеликий модуль камери з чіпом ESP32-S. На додаток до камери OV2640 і кількох GPIO для підключення периферійних пристроїв, він також має слот для карти MicroSD, який може бути корисним для зберігання зображень, зроблених камерою, або для зберігання файлів для подальшого використання для користувача.

Цей модуль не має стандартного порту USB і запис програмного забезпечення в пам'ять плати буде здійснюватися через спеціальний підключений програматор (USB TO TTL або FTDI) (рис. 2.7).

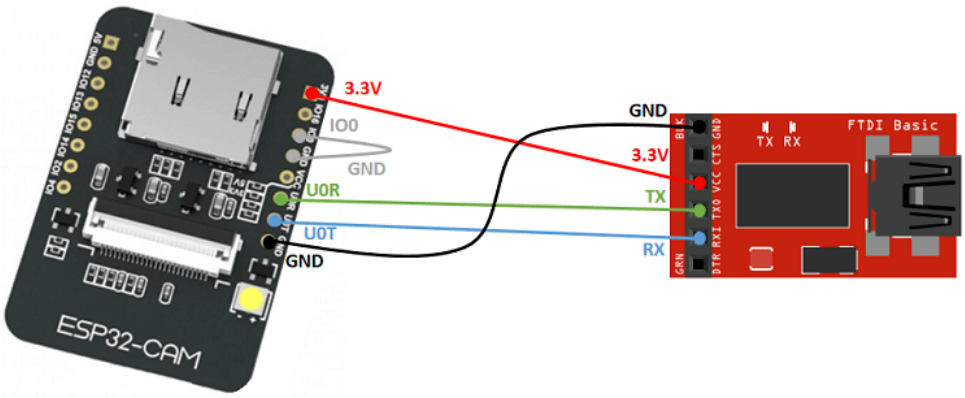

Рисунок 2.7 – Підключення плати до програматору

Характеристики модуля ESP32-CAM наведені у табл. 2.1 [10]

Таблиця 2.1 – Характеристики модуля ESP32-CAM

Найменший модуль	SoC 802.11b / g / n Wi-Fi BT
Центральний процесор	32 біти
Тактова частота	160 МГц
Пам’ять	520 КБ SRAM
Інтерфейси	UART / SPI / I2C / ШІМ / АЦП / ЦАП
Камера	OV2640 та OV7670
Завантаження зображень	Через WiFi

Стандарт карт пам’яті	TF
Підтримка режиму сну	Так!
Підтримка стеків	Lwip та FreeRTOS
Режими робот	STA / AP / STA + AP
Функції підключення до пристрою	Smart Config / AirKiss
Оновлення «по повітрю»	FOTA

Підсистема I2S в ESP32 також забезпечує високошвидкісну шину, підключену безпосередньо до оперативної пам’яті для прямого доступу до пам’яті, тобто це надає можливість налаштувати підсистему I2S ESP32 для надсилання або отримання паралельних даних під апаратним керуванням [11].

Особливості ESP32 [12]:

Серії ESP32 і ESP32-S включають:

– Мікроконтролер і управління.

– Двоядерний (або одноядерний) 32-розрядний процесор Tensilica Xtensa LX6 з тактовою частотою 160 або 240 МГц і до 600 DMIPS (Dhrystone MIPS).

– Надмалопотужний співпроцесор.

– Пам’ять: 520 КБ пам’яті SRAM.

– Бездротові:

– Wi-Fi: 802.11 b / g / N.

– Bluetooth: v4.2 BR / EDR і BLE.

– Периферійні інтерфейси:

– 12-розрядний АЦП до 18 каналів.

– 2 × 8 біт ЦАП.

– 10 × портів для підключення ємнісних датчиків (вимірювання ємності GPIO).

– Немає датчика температури. Інформація про нього вилучена зі специфікації V2.2.

– 4 × головний інтерфейс SPI (провідні пристрої).

– Головний інтерфейс 2 × I²S.

– Головний інтерфейс 2 × I²C.

– Інтерфейс 3 × UART.

– Хост-контролер SD / SDIO / CE-ATA / MMC / eMMC.

– Підпорядковані контролери SDIO / SPI (відомі пристрої).

– Інтерфейс Ethernet MAC із виділеною підтримкою DMA та IEEE 1588 Precision Time Protocol.

– Шина CAN 2.0.

– ІЧ-пульт дистанційного керування (передавач/приймач, до 8 каналів).

– Можливість підключення двигунів і світлодіодів через ШІМ вихід.

– Датчик Холла.

– Аналоговий підсилювач малої потужності.

– Безпека:

– Підтримуються всі функції безпеки IEEE 802.11, включаючи WFA, WPA / WPA2 і WAPI.

– Безпечне завантаження.

– Шифрування флешки.

– 1024-бітний ключ до 768 біт для клієнтів.

– Апаратне криптографічне прискорення: AES, SHA-2, RSA, криптографія з еліптичною кривою (ESS), апаратний генератор випадкових чисел з увімкненим Wi-Fi або Bluetooth, інакше використовується генератор псевдовипадкових чисел.

Управління живленням:

– Лінійний регулятор з низьким падінням напруги.

– Індивідуальне живлення для RTC.

– Споживання 5-2,5 мкА в режимі «глибокий сон».

– Пробудження від переривання GPIO, таймера, вимірювання АЦП, переривання датчика ємнісного датчика.

– Робоча напруга від 2,2 до 3,6 В.

– Робоча температура від -40 °C до +125 °C.

Максимальна швидкість передачі даних становить 150 Мбіт/с при 11n HT40, 72 Мбіт/с при 11n HT20, 54 Мбіт/с при 11g і 11 Мбіт/с при 11b.

Максимальна потужність передачі 19,5 дБм при 11b, 16,5 дБм при 11 g, 15,5 дБм при 11n.

Мінімальна чутливість приймача становить 98 дБм.

Стабільна пропускна здатність UDP 135 Мбіт/с.

ESP32 поставляється в корпусі плану (QFN) з 48 контактами по периметру і одним великим радіатором по центру, який також використовується для сигнального заземлення.

ESP32-D0WDQ6 містить два малопотужних 32-розрядних мікропроцесора Xtensa® LX6. Внутрішня пам'ять включає в себе:

448 КБ ПЗУ для завантаження та основних функцій.

520 КБ (8 КБ швидкої пам’яті RTC включено) вбудована SRAM для даних та інструкцій.

8 КБ SRAM в RTC, який називається швидкою пам'яттю RTC і використовуваною пам'яттю; ЦП має отримати доступ до нього під час завантаження RTC із режиму глибокого сну.

8 КБ SRAM в RTC, який називається вільною пам’яттю RTC і доступний за допомогою ЦП під час глибокого сну.

1 kb eFuse, з яких 256 біт використовуються для системи (MAC-адреса та конфігурація мікросхеми), а ще 768 біт зарезервовано для клієнтських програм, які включають шифрування флеш-пам'яті та ідентифікатор мікросхеми.

ESP32 підтримує до чотирьох зовнішніх банків QSPI і SRAM по 16 МБ з апаратним шифруванням на основі AES із захистом програм і даних користувача.

ESP32 може отримати доступ до зовнішнього Flash QSPI і SRAM через високошвидкісні канали.

До 16 МБ зовнішньої флеш-пам’яті в парі з кодовим простором ЦП, який підтримує 8, 16 і 32-розрядний доступ. Підтримується виконання коду.

Карта пам’яті зовнішньої флеш-пам’яті об’ємом до 8 МБ на ЦП простору даних, підтримка 8, 16 і 3 2-розрядного доступу. Зчитування даних підтримується на флеш-пам'яті та SRAM. Запис даних підтримується SRAM.

ESP32-WROVER інтегрує зовнішній SPI-флеш 4-16 МБ. 4-MB SPI Flash може бути картою пам'яті в центрі процесора, яка підтримує доступ до 8, 16 і 32 біт.

Підтримується виконання коду.

На додаток до 4-16 МБ флеш-пам'яті SPI, ESP32-WROVER також інтегрує 4-8 МБ PSRAM для додаткового місця в пам'яті.

Прошивка ESP32 Wi-Fi / BT може підтримувати лише кварцовий генератор 40 МГц.

[RTC та управління низьким споживанням].

Завдяки сучасним технологіям керування живленням ESP32 ви можете перемикатися між різними режимами живлення (див. таблицю нижче).

Режими живлення.

Активний режим / Active mode: радіочіп увімкнено. Чіп може приймати, передавати або слухати.

Режим сну модему / Modem-sleep mode: ЦП працює, а годинник працює за запитом. Основна смуга Wi-Fi / Bluetooth I радіо вимкнено.

Легкий сплячий режим / Light-sleep mode: ЦП зупинений. Пам’ять RTC і периферійні пристрої RTC, а також процесор ULPC. Усі події створення (MAC, хост, таймер RTC або зворотна передача) будуть передані на чіп.

Гібернація/ Hibernation mode: увімкнено лише пам’ять RTC та периферійні пристрої RTC. З'єднання Wi-Fi і Bluetooth зберігаються в пам'яті RTC. Співпроцесор ULP може працювати.

Режим глибокого сну / Deep-sleep mode: внутрішній генератор 8 МГц і співпроцесор ULP включені. Відновлення пам’яті RTC вимкнено. Активний лише один таймер RTC на вільному годиннику та деякі RTC GPIO. Таймер RTC або RTC GPIO можуть розбудити чіп у сплячому режимі.

[Sleep Patterns / Шаблони сну]

Шаблон асоційованого сну / Association sleep pattern: живлення перемикається між активним режимом, модемом і світловим сном.

Шаблон датчика ULP / ULP sensor-monitored pattern: основний процесор перебуває в режимі сну. Комбінований процесор ULP Вимірювання датчиків і Побудова основної системи на основі даних, зібраних з датчиків.

Модульні плати SMT.

Модулі SMT на базі ESP32 містять SoC ESP32 і призначені для легкої інтеграції в інші плати.

Виміряні конструкції перевернутої F-антени, розроблені для відстеження антени друкованої плати на модулях, перерахованих нижче.

На додаток до флеш-пам'яті деякі модулі включають псевдостатичну пам'ять із довільним доступом (pSRAM).

2.3 Вибір датчика руху

Датчик руху (або детектор руху) – це електронний пристрій, призначений для виявлення та вимірювання руху [20]. Датчики руху використовуються в основному в домашніх і ділових системах безпеки, але їх також можна знайти в телефонах, дозаторах для паперових рушників, ігрових консолях і системах віртуальної реальності. На відміну від багатьох інших типів датчиків (які можуть бути ручними та ізольованими), датчики руху, як правило, є вбудованими системами з трьома основними компонентами:

блоком датчиків , вбудованим комп’ютером та апаратним забезпеченням (або механічним компонентом). Ці три частини відрізняються за розміром і конфігурацією, оскільки датчики руху можна налаштувати для виконання дуже специфічних функцій. Наприклад, датчики руху можна використовувати для активації прожекторів, звукових сигналів, перемикачів і навіть для попередження поліції.

2.3.1 Роль датчиків руху в безпеці будинку

Основною метою виявлення руху є виявлення зловмисника та надсилання сповіщення на панель керування, яку сповіщає центр моніторингу.

Датчики працюють, коли нікого немає вдома або коли користувач повідомляє системі, що його немає. Та навіть можна запрограмувати деякі системи безпеки для запису подій через камеру безпеки під час руху.

Датчики руху стоять на сторожі, готові реагувати на різноманітні ситуації , такі як рух у вітальні, вікна чи двері, що відкриваються чи закриваються, або розбиваються вікна.

Ось деякі поширені способи використання датчиків руху:

– Повідомлення, якщо підліток порушить комендантську годину.

– Ввімкенення дверного дзвінка , коли хтось наближається до вхідних дверей.

– Попередження, коли діти входять у заборонені зони в будинку, як-от підвал, кімната для тренувань або аптечка.

– Економія енергії , використовуючи датчик руху для керування освітленнням в незайнятих приміщеннях.

– Поведомлення, якщо домашні тварини входять у місця, де їх не повинно бути.

2.3.2 Типи датчиків руху

Пасивний інфрачервоний (PIR).

Пасивний інфрачервоний датчик виявляє тепло тіла (інфрачервону енергію), шукаючи зміни температури. Це найбільш широко використовуваний датчик руху в системах безпеки будинку. Коли ви ставите систему на охорону, це активує датчики руху, щоб повідомляти про можливі загрози.

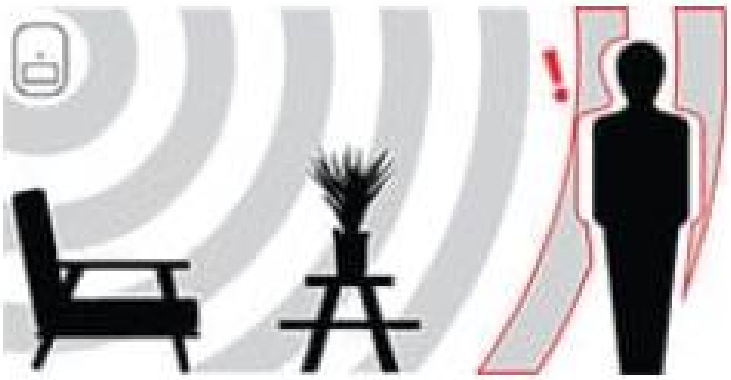

Рисунок 2.8 – PIR датчик руху

Після того, як датчик руху PIR прогріється, він може виявити тепло і рух в навколишніх областях, створюючи захисну «сітку». Якщо рухомий об’єкт блокує занадто багато зон сітки, а рівень інфрачервоної енергії швидко змінюється, інфрачервоний датчик спрацьовує тривогу.

Мікрохвильовий датчик руху.

Цей тип датчиків посилає мікрохвильові імпульси та вимірює відбиття від рухомих об’єктів. Вони охоплюють більшу площу, ніж інфрачервоні датчики, але дорожчі та вразливі до електричних перешкод.

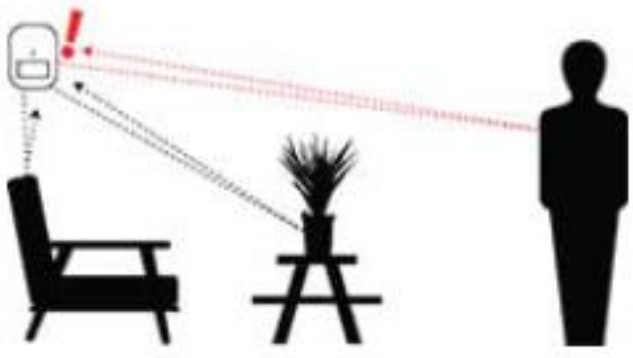

Рисунок 2.9 – Мікрохвильовий датчик руху

Датчики руху з подвійною технологією.

Деякі датчики руху можуть поєднувати кілька методів виявлення, намагаючись зменшити помилкові тривоги. Наприклад, нерідкі випадки, коли датчик з подвійною технологією поєднує пасивний інфрачервоний (PIR) датчик з мікрохвильовим датчиком.

Рисунок 2.10 – Датчик руху з подвійною технологією

Кожен тип датчика працює в різних областях спектра (від пасивного до активного). Датчики руху з подвійною технологією не так імовірно, як інші типи, викликають помилкову тривогу, оскільки обидва датчики повинні спрацювати, щоб подати сигнал тривоги. Однак це не означає, що вони ніколи не викликають помилкових тривог.

Менш поширені типи детекторів руху.

– Датчики, що відбивають місцевість, випромінюють інфрачервоні промені від світлодіода і використовують відображення цих променів для вимірювання відстані до людини або об’єкта, що дозволяє виявити, коли об’єкт рухається в межах визначеної зони.

– Ультразвукові датчики руху вимірюють відбиття від рухомих об'єктів за допомогою імпульсів ультразвукових хвиль.

– Вібраційні датчики руху виявляють невеликі вібрації, які викликають люди, коли вони рухаються по кімнаті. Хоча ви можете купити їх, їх також легко зробити вдома. Саморобний датчик вібрації використовує невелику масу на важелі, яка активує вимикач тривоги, коли він вібрує.

Саморобні датчики руху можуть працювати, але вони також можуть бути ненадійними.

2.3.3 Датчик руху HC-SR501

При вході людини до зони огляду датчика фіксується присутність. Принцип роботи модуля HC-SR501 полягає в реєстрації інфрачервоного випромінювання рухомого об'єкта [21]. Чутливий елемент – піроелектричний датчик 500BP. Він складається із двох елементів ув'язнених в одному корпусі. Чутливий елемент закритий білим куполом – лінзою Френеля. Особливості лінзи Френеля такі, що інфрачервоне випромінювання рухомого об'єкта потрапляє спочатку на один елемент датчика 500BP, потім на інший. Електроніка модуля HC-SR501 реєструє почергове надходження сигналів від двох елементів зі складу 500BP та при фіксації руху вихідний ланцюг модуля формує логічний сигнал.

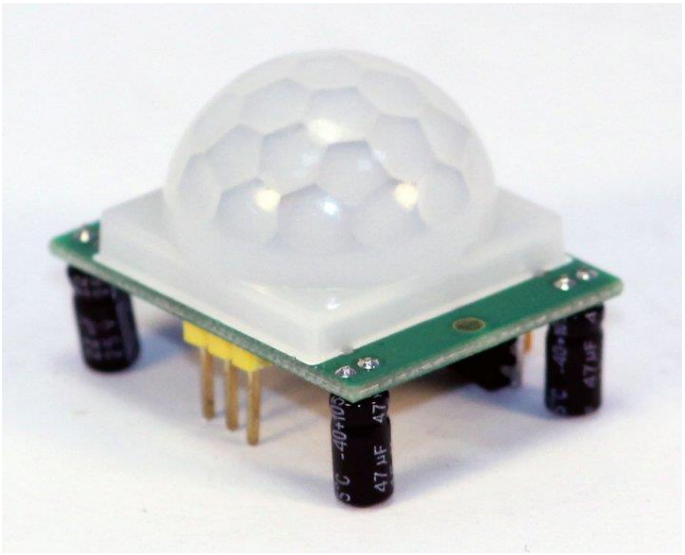

Рисунок 2.11 – Датчик руху HC-SR501

Датчик присутності HC-SR501 застосовується в охоронних системах, увімкненні вентиляції, дозволяє керувати освітленням приміщень без вікон. Спільно з фотореле візьметься в управлінні освітленням дворів та вулиць. Цікаві результати можна отримати за допомогою датчика фотоапаратом або відеокамерою.

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

Режим роботи модуля визначається перемичкою. Є два режими – режим H та режим L.

Режим H - у цьому режимі при спрацьовуванні датчика кілька разів поспіль на виході (на OUT) залишається високий логічний рівень.

Режим L – у цьому режимі на виході при кожному спрацьовуванні датчика з'являється окремий імпульс.

Характеристики модуля HC-SR501 наведені 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

Прилад готовий до роботи за хвилину після подачі живлення. За цей час відбувається автоматичне калібрування. У разі вибору місця встановлення слід уникати орієнтації датчика на відкриті джерела світла. Встановлювати датчик присутності HC-SR501 слід так, щоб рух рухомого об'єкта відбувався вздовж площини плати модуля.

Не варто розташовувати PIR-датчики у місцях, де швидко змінюється температура. Це призведе до того, що датчик не зможе виявляти появу людини в контрольованій зоні, і буде багато помилкових спрацьовувань.

До мікроконтролерів (ну або інших мікросхем) модуль краще (хоч і необов'язково) підключати через транзистор і підтягуючий резистор на 10k.

2.4 Програмне забезпечення

Програмне забезпечення Arduino (IDE) з відкритим кодом дозволяє легко писати код і завантажувати його на плату [18].

Програма IDE підходить для різних операційних систем, таких як Windows, Mac OS X і Linux. Він підтримує мови програмування C і C++. Тут IDE означає інтегроване середовище розробки.

Написання скетчів.

Програми, написані за допомогою програмного забезпечення Arduino (IDE), називаються скетчами. Ці мініатюри зберігаються в текстовому редакторі та з розширенням .ino. Редактор має функції для вирізання/вставки та пошуку/заміни тексту. Область повідомлень забезпечує зворотний зв’язок під час збереження та експорту, а також відображає помилки. Консоль відображає текст, який відображається програмним забезпеченням Arduino (IDE), включаючи повні повідомлення про помилки та іншу інформацію. Налаштована плата та послідовний порт відображаються в нижньому правому куті вікна. Кнопки панелі інструментів дозволяють перевіряти та завантажувати програми, створювати, відкривати та зберігати ескізи, а також відкривати послідовний монітор.

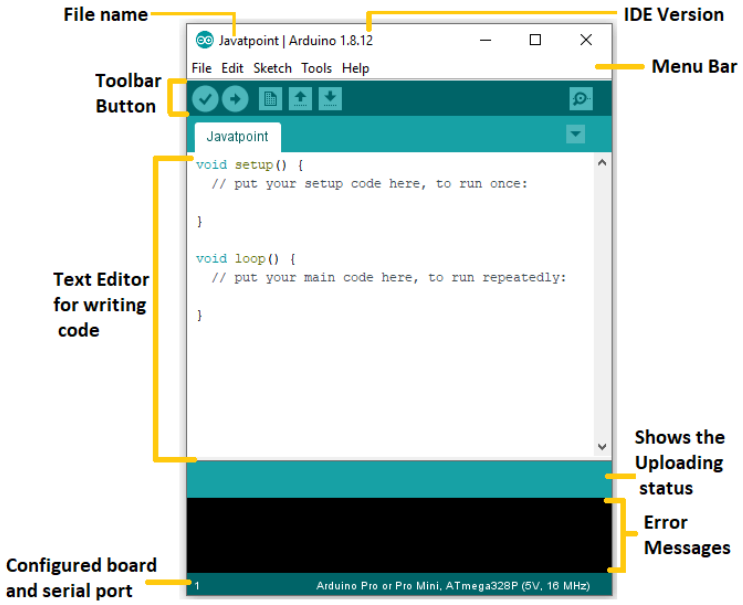

Рисунок 2.12 – Зовнішній вигляд Arduino IDE

Бібліотеки.

Бібліотеки надають додаткову функціональність для використання в ескізах, наприклад, робота з обладнанням або маніпулювання даними. Щоб використовувати бібліотеку в ескізі, треба обрати її в меню Скетч –> Імпорт бібліотеки . Це вставить один або кілька операторів #include у верхній частині ескізу та компілює бібліотеку із скетчем. Оскільки бібліотеки завантажуються на мікроконтролер разом із скетчем, вони збільшують обсяг місця, який він займає. Якщо скетчу більше не потрібна бібліотека, є можливість видаліти його оператори #include у верхній частині коду.

У довідці є список бібліотек . Деякі бібліотеки входять до програмного забезпечення Arduino. Інші можна завантажити з різних джерел або за допомогою Менеджера бібліотеки. Починаючи з версії 1.0.5 IDE, з'явилася можливість імпортувати бібліотеку з zip-файлу та використовувати її у відкритому скетчі.

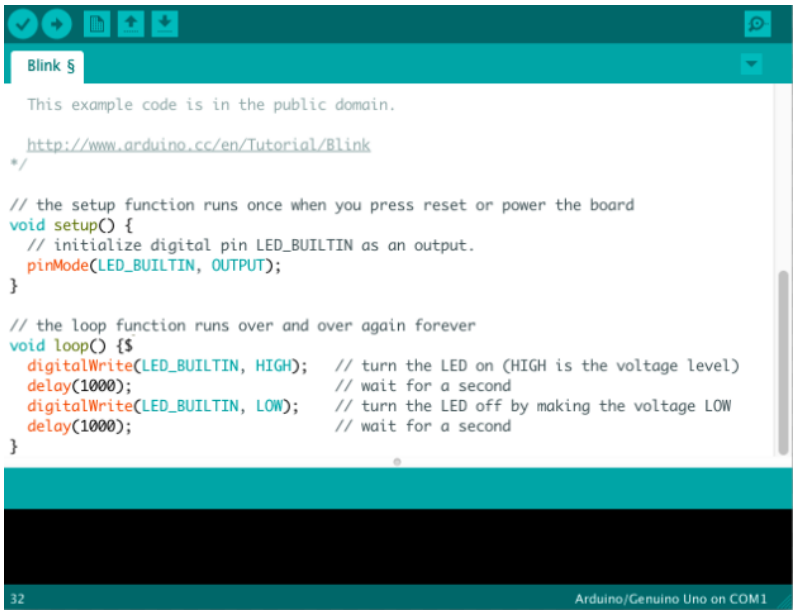

Рисунок 2.13 – Знімок екрану Arduino IDE із простою програмою Blink

Структура складається з двох основних функцій:

– Функція setup() викликається на початку скетчу. Функція використовується для ініціалізації змінних, режимів контактів, початку використання бібліотек тощо. Функція setup() запускатиметься лише один раз, щоразу, коли вмикається живлення або перезавантажується плата.

– Після виклику функції setup (), яка ініціалізує та встановлює початкові значення, функція loop () виконує саме те, що передбачає її назва, і виконує послідовний цикл, що дозволяє програмі змінюватися та надавати зворотній зв'язок. Функція використовується для активного управління платою Arduino.

2.4.1 Мова програмування

Мова програмування – це мова комп’ютера, яка використовується програмістами (розробниками) для спілкування з комп’ютерами. Це набір інструкцій, написаних будь-якою конкретною мовою (C, C++, Java, Python) для виконання певного завдання.

Мова програмування в основному використовується для розробки настільних додатків, веб-сайтів і мобільних додатків.

Мова програмування високого рівня створена для розробки зручних програм та веб-сайтів. Для цієї мови програмування потрібен компілятор або інтерпретатор для перекладу програми на машинну мову (виконання програми).

Основна перевага мови високого рівня полягає в тому, що її легко читати, писати та підтримувати.

Мова програмування на високому рівні включає Python, Java, JavaScript, PHP, C#, C ++, та багато інших.

C++ є розширенням мови C, відомою своїми вбудованими та низькорівневими можливостями проєктів. Мова середнього рівня з об’єктно-орієнтованим і загальними можливостями програмування, що дозволяє маніпулювати пам’яттю низького рівня. Програмне забезпечення, створене за допомогою цієї мови, розроблене для вбудованого програмування, обмежених пристроїв і програмування великих систем, відрізняється швидкістю, продуктивністю та ефективністю.

C++ є скрізь, куди б ми не зверталися. Програмне забезпечення баз даних, операційні системи, медичні програми та ігри – це лише кілька прикладів реальних додатків із використанням C++. Оскільки процесори стали потужнішими, а ландшафт додатків зайняв додаткові складні вимоги, C++ почало використовувати його для рішень IoT. C++ забезпечує більшу гнучкість з меншим споживанням енергії, що робить його ідеальним для невеликих пристроїв, які не можуть підтримувати високий рівень активності через обмежені можливості живлення.

C++ – це «C» з додатковими функціями, такими як абстракція даних, об’єкти та класи, це робить C++ перевагою для коду IoT у операційних системах. Його подібність до C дозволяє використовувати його обчислювальну потужність, що робить його чудовою альтернативою C при роботі над складними завданнями. Такими завданнями можуть бути пристрої, які виконують кілька функцій або мають кілька датчиків, наприклад

термостати або сенсорні мітки , які визначають температуру або вологість у своєму середовищі.

Багато пристроїв Інтернету речей, доступних зараз, насправді запрограмовані на C або C++, оскільки ці пристрої не мають процесорної потужності, необхідної для роботи багатьох мов вищого рівня, а більшість операційних систем, доступних для платформ IoT, вже мають підтримку C або C++.

Мова програмування пристроїв Arduino заснована на C/C++ і скомпонована з бібліотекою AVR Libc і дозволяє використовувати будь-які її функції. Разом з тим він простий у освоєнні, і зараз Arduino – це, мабуть, найзручніший спосіб програмування пристроїв на мікроконтролерах.

2.5 Telegram Bot API

Telegram – це програма для обміну повідомленнями в Інтернеті, яка за своєю суттю працює як популярні програми для обміну повідомленнями WhatsApp або Facebook Messenger [19].

Це означає, що можна використовувати Telegram для надсилання повідомлень контактам при підключенні до Wi-Fi або мобільного Інтернету. Telegram працює в хмарі та надає пріоритет безпеці та швидкості. В результаті месенджер став популярною альтернативою іншим програмам обміну повідомленнями. Сервіс запрацював у 2013 році і наразі налічує понад 500 мільйонів активних користувачів щомісяця.

Боти – це просто облікові записи Telegram, якими керує програмне забезпечення, а не люди, і вони часто мають функції штучного інтелекту. Вони можуть робити все: навчати, грати, шукати, транслювати, нагадувати, підключатися, інтегрувати з іншими службами або навіть передавати команди в Інтернет речей.

Телеграм використовує свій протокол шифрування MTProto. MTProto API (він же Telegram API) - це API, через який програма Telegram зв'язується

з сервером. Telegram API повністю відкритий, тому будь-який розробник може написати свій клієнт месенджера.

MTProto - це допоміжний сервер сам обробляє все шифрування і зв'язок з Telegram API. Користувач з'єднуєтеся з сервером через простий інтерфейс HTTPS, який надає просту версію Telegram API.

Розмова з телеграм-ботом відбувається за допомогою текстових команд, визначених під час створення або програмування, які завжди починаються з «/». Це можуть бути такі команди, як наприклад:

– /start;
– /status.

В принципі, можна використовувати всі мови програмування, які працюють на сервері і можуть відповідати на запити через HTTPS.

– Java / Kotlin
– PHP
– C++
– Python

Висновки до розділу 2

При створенні апаратної частини, проаналізовано декілька мікроконтролерів. Кожен з них мав свої переваги та недоліки, в результаті найбільш оптимальним виявився мікроконтролер ESP32-CAM.

Детально опрацьовано та розглянуто можливі компоненти та програмні засоби для проєктування модулю системи оповіщення, та обрано найоптимальніші, а саме:

– PIR Датчик руху HC-SR501;
– c++ – мова програмування алгоритму для модуля ESP32;
– месенджер Telegram для створення боту;
– arduino IDE – середовище розробки.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ СИСТЕМИ ОПОВІЩЕННЯ НА БАЗІ ESP-32 CAM, ДАТЧИКУ РУХУ ТА TELEGRAM BOT API

Програмно-апаратний комплекс.

Складові апаратного забезпечення (Hardware):

– Мікроконтролер ESP32-CAM;

– Спеціальна камера OV2640;

– PIR датчик руху HC-SR501;

– Кабелі для з’єдання типу Dupon;

– Макетна плата (за бажанням, для зручності підключення та використання);

– Корпус, виготовлений на 3Д-принтері (по можливості).

Складові програмного забезпечення (Software):

– програмне середовище для написання коду, та завантаження його на мікроконтролер – Arduino IDE;

– месенджер Telegram, для приймання та демонстрації медіа даних;

– програмне середовище EasyEDAж

– програмне середовище Fritzing, для візуалізації з’єднання схеми

3.1 Підключення компонентів до плати ESP32-CAM

3.1.1 Програмування ESP32-CAM за допомогою Arduino

Щоб запрограмувати плату ESP32-CAM з Arduino IDE, потрібно встановити Arduino IDE, а також доповнення ESP32.

На платі ESP32-CAM відсутній порт живлення, тому виникає необхідність підключити плату ESP32-CAM до комп’ютера за допомогою модуля послідовного перетворювача FTDI або USB TO TTL. В проєкті був обран модуль послідовного перетворювача CP2102 USB TO TTL (рис. 3.1).

Рисунок 3.1 – CP2102 USB TO TTL

CP2102 - це USB-UART перетворювач (USB to UART Bridge), можна використовувати для програмування Arduino або інших Arduino-подібних контролерів, отримувати інформацію на комп'ютер з усього, що має послідовний інтерфейс з TTL логікою. Також можна використовувати для налагодження одноплатних комп'ютерів, особливо ті, що не мають відеовиходу: NanoPi NEO, Orange Pi Zero, Orange Pi R1 тощо. Інтерфейс дозволяє Windows бачити віртуальний COM-порт (VCP) як звичайний COM-порт.

З одного боку знаходиться роз'єм USB, з іншого 6 pin висновків: +3.3v, GND, +5v, TXD (TX), RXD (RX), DTR, на платі є монтажні отвори з функціями DCD, D3R, RTS, CTS, SUS , SUS, R1, RST. Крім цього на платі є 3 світлодіоди, червоний POWER і два для RX і TX миготливих під час прийому-передачі даних.

Далі відбувається підключення самої плати. GPIO 1 (TX) і GPIO 3 (RX) є послідовними контактами. Оскільки плата не має вбудованого програматора, потрібно використовувати ці контакти для зв’язку з програматором і завантаження коду. Також можна використовувати GPIO 1 і GPIO 3 для

підключення інших периферійних пристроїв. Але якщо вони будуть зайняті, відкрити послідовний монітор буде неможливо.

Плата може живитися від будь-якого джерела живлення, батареї або USB-порту комп’ютера. У цій збірці він буде живитися через порт USB, через блок живлення на USB TO TTL (рис. 3.2).

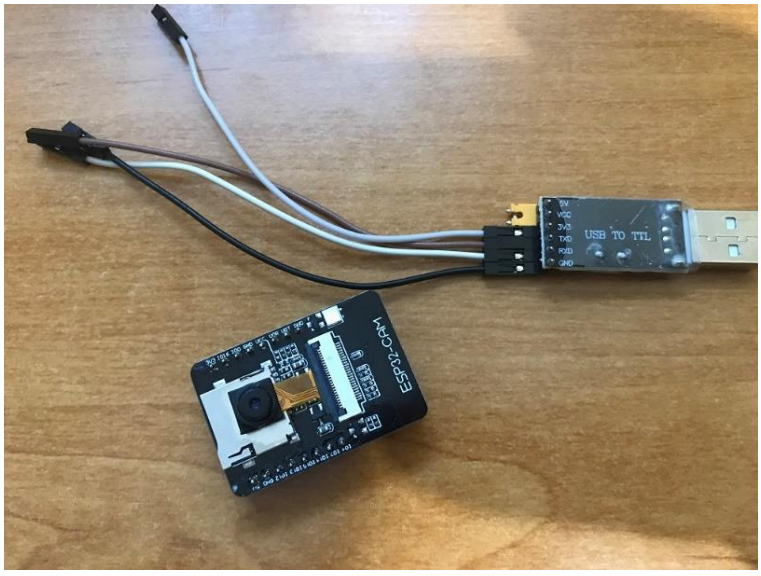

Рисунок 3.2 – Живлення з програматору

Рисунок 3.3 – З’єднання з платою

Підключення плати ESP32-CAM до комп’ютера за допомогою програматора CP2101 за наступною принциповою схемою (рис.3.2).

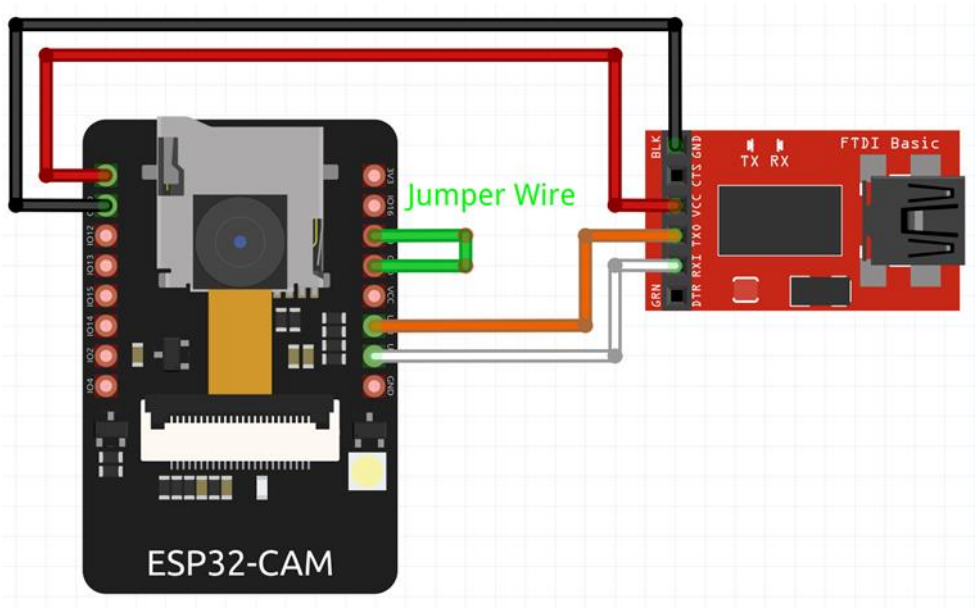

Рисунок 3.4 – Схема підключення модулю ESP32-CAM до програматора у програмі Fritzing

З'єднання контактів між мікроконтролером та програматором відбувається за допомогою кабелів типу Dupon (рис. 3.5).

Рисунок 3.5 – Кабелі з’єднання

Під’єднання пінів програматору і плати показано у табл. 3.1.

Таблиця 3.1 – З’єднання пінів.

ESP32-CAM	FTDI Programmer
GND	GND
5V	VCC (5V)
U0R	TX
U0T	RX
GPIO 0	GND

Важливо: GPIO 0 має бути підключений до GND, щоб завантажувати код.

Нижче наведена функціональна схема з'єднання програматора з модулем ESP-32 CAM (рис. 3.8)

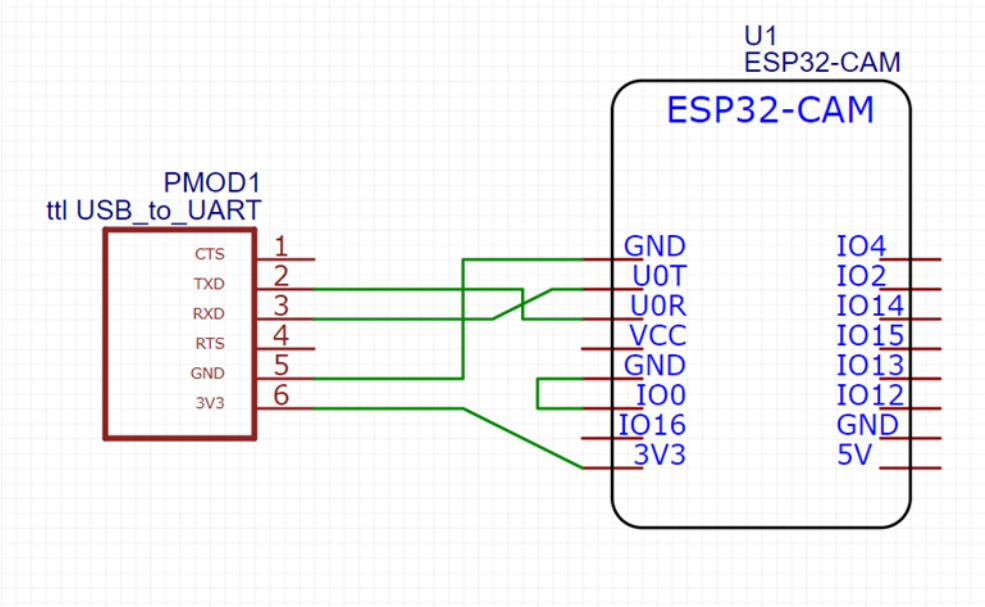

Рисунок 3.6 – Функціональна схема підключення модуля ESP32-CAM для передачі медіа даних до мобільного месенджеру у програмі easyEDA

3.1.2 Підключення датчика PIR до ESP32-CAM

Далі необхідно підключити інфрачервоний датчик руху (PIR) до ESP32–CAM. Для початкового налаштування потрібно підключити живлення 5 В, заземлення(GND), а центральний контакт даних – до GPIO 15 ESP32.

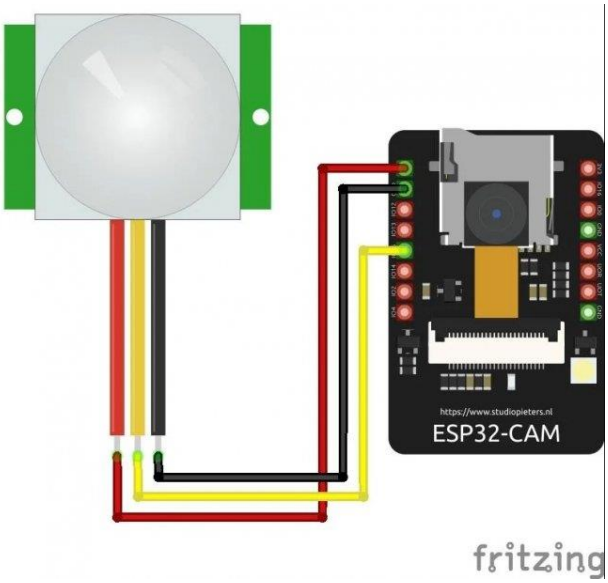

Рисунок 3.7 – Схема підключення датчику PIR до ESP32-CAM у програмі Fritzing

У результаті було спроєктовано схему апаратно-програмного модулю системи оповіщення на базі ESP-32 CAM, датчику руху та Telegram Bot API.

Принципову схему було побудовано у програмному середовищі Fritzing. [22] На рисунку 2.16 зображено принципову схему апаратно-програмного модулю.

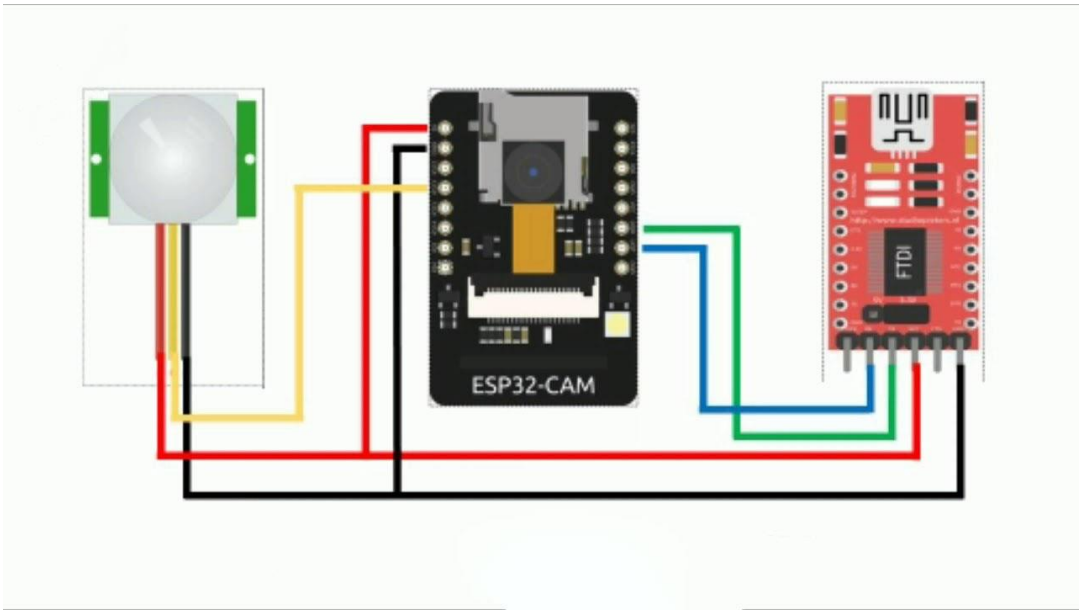

Рисунок 3.8 – Принципова схема пристрою

3.2 Створення бота Telegram

ESP32-CAM буде взаємодіяти з ботом Telegram, щоб отримувати й обробляти повідомлення, а також надсилати відповіді на обліковий запис Telegram (фотографії).

Перш за все необхідно завантажити та встановити Telegram на телефон через Google Play або App Store.

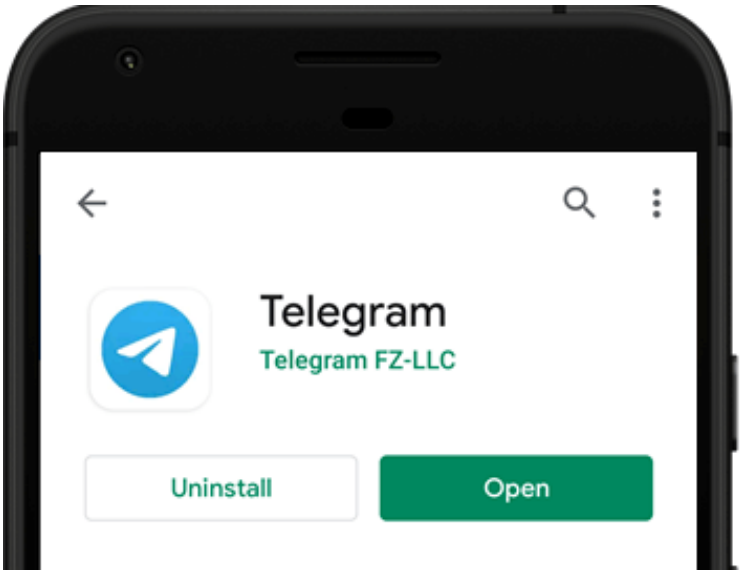

Рисунок 3.9 – Завантаження Telegram

Потім відкрити Telegram і виконати наступні кроки, щоб створити бота Telegram. Спочатку знайдіть «botfather» і клацнути на BotFather, як показано нижче (рис.3.8). Або відкрийти посилання t.me/botfather у своєму смартфоні.

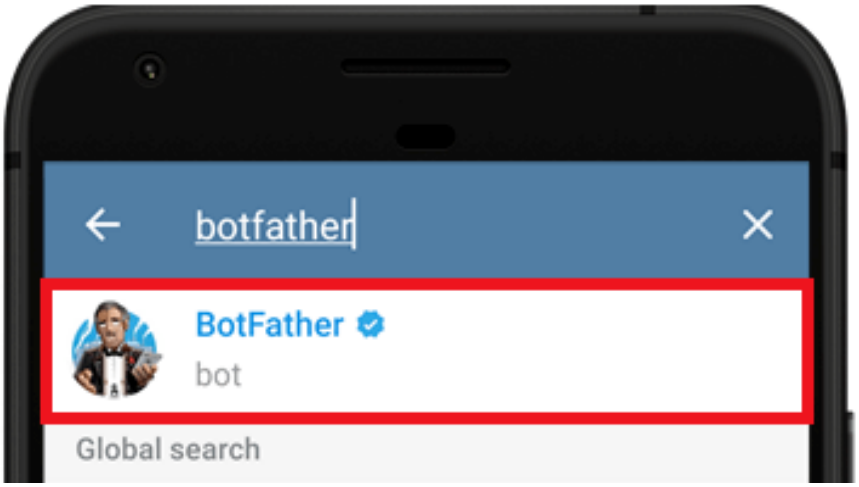

Рисунок 3.10 – BotFather

Відкриється наступне вікно, і буде запропоновано натиснути кнопку «/start».

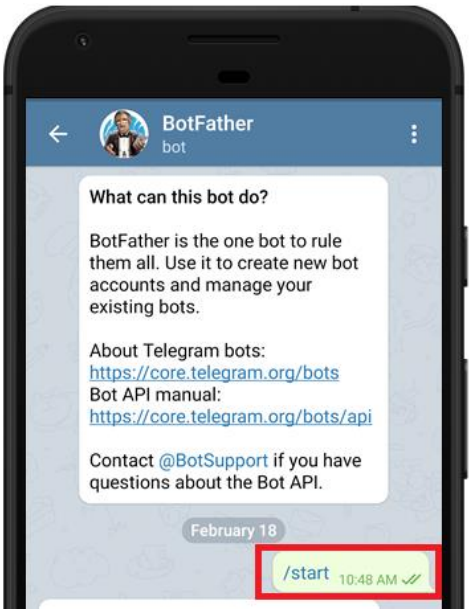

Рисунок 3.11 – Початок роботи з ботом

Ввести команду /newbot і дотримуватись інструкцій, щоб створити свого бота. Дати йому ім'я та ім'я користувача.

Рисунок 3.12 – Команда /newbot

Бот успішно створен, надходить повідомлення з посиланням на доступ до бота та токен бота . Треба зберегти токен бота, оскільки він знадобиться, щоб ESP32/ESP8266 міг взаємодіяти з ботом.

3.3 Отримання ідентифікатора користувача Telegram

Будь-хто, хто знає ваше ім'я користувача бота, може взаємодіяти з ним. Щоб переконатися, що ми ігноруємо повідомлення, які надходять не з нашого облікового запису Telegram (або будь-яких авторизованих користувачів), можна отримати свій ідентифікатор користувача Telegram. Потім, коли бот отримує повідомлення, ESP може перевірити, чи відповідає ідентифікатор відправника вашому ідентифікатору користувача, і обробити повідомлення або проігнорувати його.

У обліковому записі Telegram треба знайти «IDBot» або відкрити це посилання t.me/myidbot у своєму смартфоні.

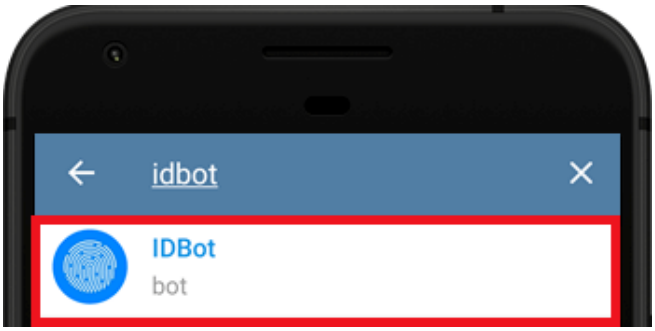

Рисунок 3.13 – IDBot

Необхідно почати розмову з цим ботом і ввести команду /getid . Після його надходить відповідь з унікальним ідентифікатором користувача. Треба також зберегти цей ідентифікатор користувача , оскільки він знадобиться при налаштуванні доступу.

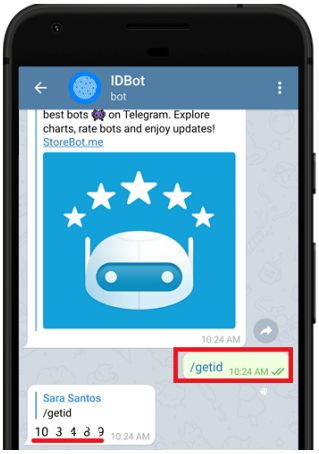

Рисунок 3.14 – Отримання chatid

3.4 Підготовка Arduino IDE

Запрограмування ESP32-CAM за допомогою Arduino IDE, тому треба переконайтися, що у Arduino IDE встановлено доповнення ESP32.

Для взаємодії з ботом Telegram використовується Universal Telegram Bot Library , створену Брайаном Лофом, яка забезпечує простий інтерфейс для API бота Telegram.

Щоб встановити останню версію бібліотеки, потрібно виконати наступні кроки:

– Завантажити бібліотеку Universal Arduino Telegram Bot .

– Перейти до Sketch > Include Library > Add .ZIP Library... .

– Додати бібліотеку.

Також потрібно встановити бібліотеку ArduinoJson. Щоб встановити останню версію бібліотеки, потрібно виконати наступні кроки:

– Перейти до Sketch > Include Library > Manage Libraries .

– Знайти «ArduinoJson».

– Встановити бібліотеку.

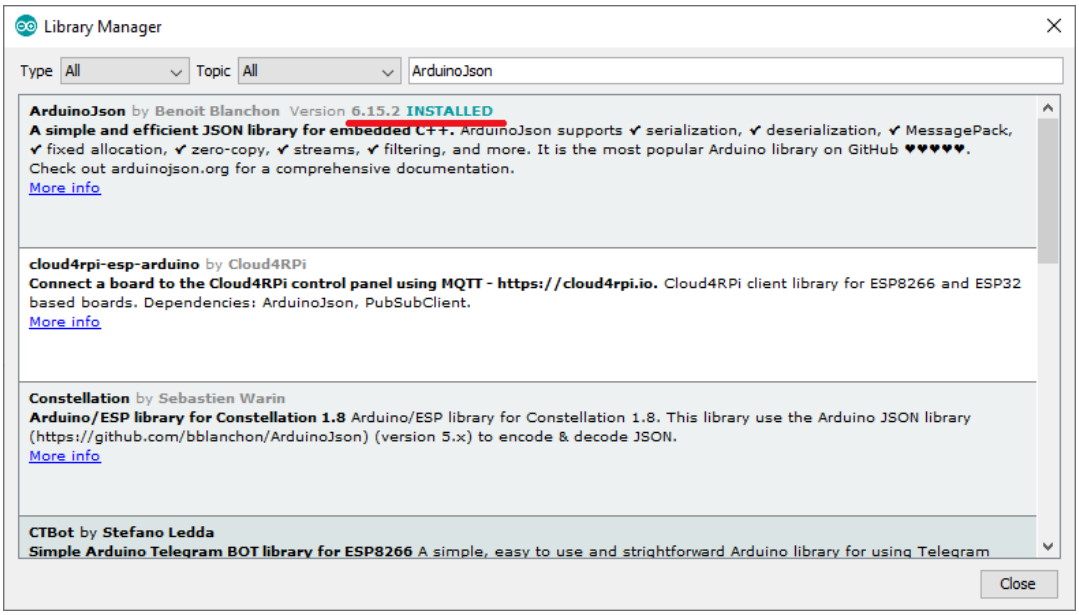

Рисунок 3.15 – Встановлення бібліотеки ArduinoJson

3.5 Завантаження коду на ESP32-CAM

Після внесення необхідних змін потрібно завантажити код на свій ESP32-CAM. Виконати наступні кроки, щоб завантажити код:

– Підключити ESP32-CAM до програматора CP2102

– Перейти до вкладки Інструменти > Плата та обрати AI-Thinker ESP32-CAM .

– 3Перейти до Інструменти > Порт і обрати COM-порт, до якого підключено ESP32-CAM.

– Натиснути кнопку «Завантажити» у Arduino IDE.

Коли у вікні налагодження з’являться крапки, треба натиснути вбудовану кнопку reset(RST) ESP32-CAM. Через кілька секунд код має бути успішно завантажений на плату.

– Коли з’явиться повідомлення «Завантаження завершено», від’єднати GPIO 0 із GND. Потім потрібно відкрити послідовний монітор, натиснути вбудовану кнопку RST і переконайтися, що ESP32-CAM без проблем підключається до вашої мережі.

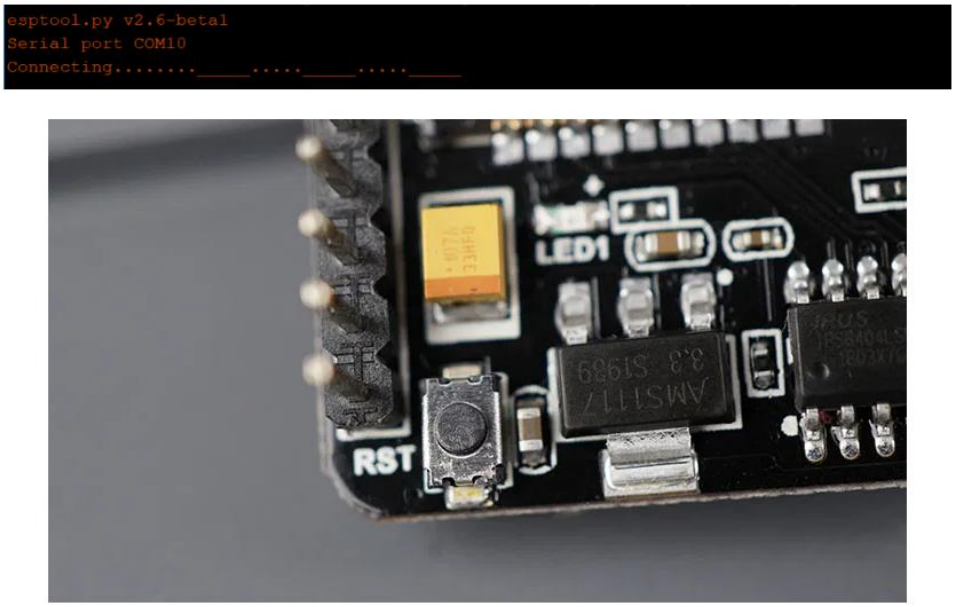

Рисунок 3.16 – ESP32-CAM Reset

3.6 Опис передачі даних до месенджеру

Сам бот створюється в месенджері Telegram, який, у свою чергу, буде виконувати спеціально запрограмовані команди та надсилати відповідні інструкції апаратному та програмному модулю, який вони виконують.

Усі команди містяться в окремих функціях, в яких записана логіка передачі даних до Telegram бота.

Функція sendPhotoTelegram() робить фотографію за допомогою ESP32-CAM.

```
camera_fb_t * fb = NULL;
fb = esp_camera_fb_get();
if(!fb) {
  Serial.println("Camera capture failed");
  delay(1000);
  ESP.restart();
  return "Camera capture failed";
}
```

Потім робить запит HTTP POST, щоб надіслати фотографію телеграм-боту.

```
clientTCP.println("POST /bot"+BOTtoken+"/sendPhoto HTTP/1.1");
clientTCP.println("Host: " + String(myDomain));
clientTCP.println("Content-Length: " + String(totalLen));
clientTCP.println("Content-Type: multipart/form-data; boundary=AlertSystem");
clientTCP.println();
clientTCP.print(head);
```

DetectsMovement() – це функція зворотного виклику, яка викликається при виявленні руху. У цьому випадку встановлюється для змінної

motionDetected значення true. Потім у loop() обробляється те, що відбувається, коли є рух (надсилає фотографію).

```
static void IRAM_ATTR detectsMovement(void * arg){
  //Serial.println("MOTION DETECTED!!!");
  motionDetected = true;
}
```

У loop() перевіряється стан змінної sendPhoto. Якщо це true, викликається функція sendPhotoTelegram(), щоб зробити та надіслати фотографію на обліковий запис Telegram.

```
if (sendPhoto){
  Serial.println("Preparing photo");
  sendPhotoTelegram();
  sendPhoto = false;
}
```

Коли це буде зроблено, встановлюється для змінної sendPhoto значення false.

Коли виявлено рух, надсилається сповіщення на обліковий запис Telegram і викликається функцію senPhototoTelegram(). Потім встановлюється для змінної motionDetected значення false.

```
if(motionDetected){
  bot.sendMessage(chatId, "Motion detected!!", "");
  Serial.println("Motion Detected");
  sendPhotoTelegram();
  motionDetected = false;
}
```

3.7 Демонстрація роботи апаратно-програмного модулю

Для перевірки треба відкрити свій обліковий запис Telegram. Надіслати команду /start для початку роботи з телеграм-ботом.

Рисунок 3.17 – Телеграм-бот

Щоб керувати ESP32-CAM доступні наступні команди:

– /start : надсилає вітальне повідомлення з дійсними командами для керування платою;

– /flash : перемикає світлодіодний спалах ESP32-CAM;

– /photo : робить нове фото та надсилає його на ваш обліковий запис Telegram (рис. 3.18);

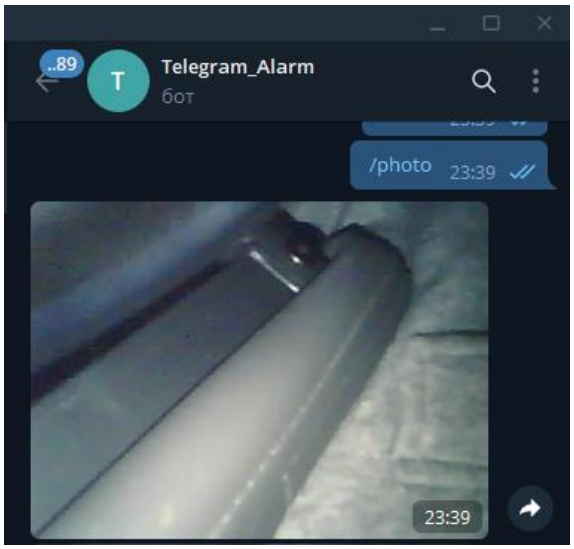

Рисунок 3.18 – Демонстрація роботи команди /photo

Користувач буде отримувати сповіщення з фотографією щоразу, коли буде виявлено рух (рис. 3.19).

Рисунок 3.19 – Сповіщення щодо виявлення духу

Якщо хтось спробує взаємодіяти з ботом з іншого облікового запису, то цей користувач отримає повідомлення «Unauthorized user» (рис. 3.20).

```
String chat_id = String(bot.messages[i].chat_id);
 if (chat_id != chatId){
  bot.sendMessage(chat_id, "Unauthorized user", "");
  continue;
 }
```

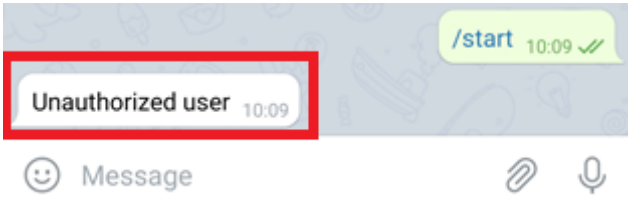

Рисунок 3.20 – Повідомлення щодо неавторизованого користувача

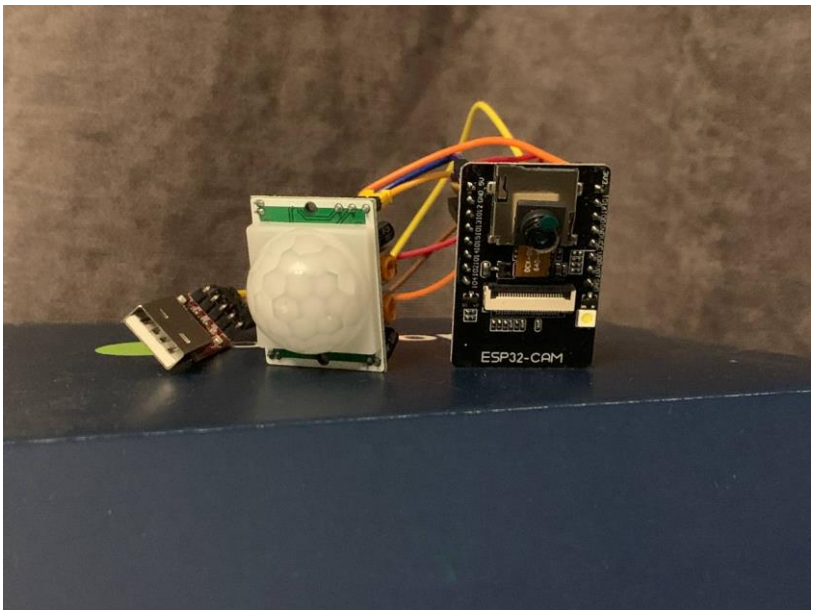

Рисунок 3.21 – Прототип модуля системи сповіщення

Висновки до розділу 3

Підсумовуючи третій розділ, можемо зробити такі висновки:

– Fritzing було використано для побудови принципової схеми пристрою.

– Зібрано програмно-апаратний модуль.

– Реалізовано програмну частину алгоритмів.

– Створено Telegram-бот.

ВИСНОВКИ

Під час написання бакалаврської роботи була проаналізована інформація з різних джерел та патентних робіт, показано, що ця тема актуальна для реалізації програми в гаджетах на основі концепції IoT. Розроблено план поетапного розвитку цього проєкту.

Аналітичний огляд існуючих проєктів показав, що ESP32-CAM з модулем камери OV2604 є найефективнішим для проєкту Telegram Messenger з точки зору його функціональності – швидкості, компактності, камери, карти пам’яті та підтримки різних протоколів передачі даних.

Telegram-месенджер був обраний для проєкту, оскільки він має велику підтримку сторонніх бібліотек, і це пов'язано з простотою використання API, який надає сама компанія месенджером. Також цей месенджер став дуже популярним серед молоді, що може зацікавити їх у розвитку подібних проєктів.

Зібравши такий модуль, можна заощадити на покупці дорогих аналогів для «розумного будинку». Сам зібраний модуль поступається високобюджетним аналогам лише за якістю картинки, але цей фактор вирішується звичайним оновленням камери на модулі.

Після завершення виконаної роботи було отримано програмно-апаратний модуль системи оповіщення на базі ESP32-CAM OV2640 для детектування руху з фотофіксацією, з подальшою відправкою зроблених медіа-даних у Telegram-месенджер, та захисту від стороннього доступу до бота.

Вирішено проблему підключення модуля ESP32-CAM до програматора та завантаження прошивки через COM-порти.

Зібрано прототип програмно-апаратного модуля для детектування руху з фотофіксацією, та подальшою передачі даних через WiFi в Telegram-месенджер.

Знайдено та висвітлено переваги модуля ESP32-CAM перед іншими моделями цієї серії.

Запропоновано ймовірні шляхи розвитку цього програмно-апаратного модуля в різних сферах діяльності.

Апаратно-програмний модуль працює від різних джерел живлення.

Цей проєкт може бути частиною більшого проєкту з використанням додаткових датчиків і модулів в одному комплекті.

Перевага цього повного апаратно-програмного модуля в тому, що його можна використовувати для створення вашого оригінального проєкту, який у майбутньому може стати дуже прибутковим і визнаним.

Завдяки такій розробці можна вивести на новий рівень популярність WiFi-модулів ESP32 і технологій Smart House в цілому. Щоб зацікавити як молоде покоління, так і більш досвідчених розробників.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Посилання на проєкт Arduino Project HUB. Discord Security Camera with an ESP32. URL: https://create.arduino.cc/projecthub/WillMakesTV/discord-security-camera-with-an-esp32-7c4621?ref=search&ref_id=esp32-cam&offset=0. (дата звернення: 21.03.2022).

2. Посилання на проєкт ESP32-CAM Remote Controlled Car Robot Web Server, March 13, 2021.URL: https://randomnerdtutorials.com/esp32-cam-car-robot-web-server/.(дата звернення: 22.03.2022).

3. Посилання на 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. (дата звернення: 23.03.2022).

4. Посилання на проєкт Color Detection and Tracking URL: https://how2electronics.com/color-detection-tracking-with-esp32-cam-opencv/. (дата звернення: 25.03.2022).

5. Посилання на проєкт Точка доступу на основі ESP32-CAM URL: https://randomnerdtutorials.com/esp32-cam-access-point-ap-web-server/. (дата звернення: 29.03.2022).

6. 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.

7. 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.

8. Neil Cameron. Electronics Projects with the ESP8266 and ESP32: Building Web Pages, Applications, and WiFi Enabled Devices. December 18, 2020, p. 723.

9. Craig Hunt. TCP/IP Network Administration (3rd Edition; O'Reilly Networking). April 1, 2002, p. 746.

10. Cuno Pfister. Getting Started with the Internet of Things: Connecting Sensors And Microcontrollers To The Cloud. June 21, 2011, p. 194.

11. Sayak Boral. Article: What is ESP32 and Why Is It Best for IoT Projects? May 25, 2020. URL: https://www.iottechtrends.com/what-is-esp32/.

13. Момот М. В. Мобільні роботи на базі ESP32 у середовищі Arduino IDE. БВХ-Петербург, 2020, 272с.

14. Douglas Comer. Internetworking with TCP/IP Volume One. 2013, 744 p.

15. Ревич Ю. Цікава електроніка. Ревич Ю. –БХВ-Петербург, 2015, 576с.

16. Dogan Ibrahim, Ahmet Ibrahim. The Official ESP32 Book. 2017, 286 p.

17. Marco Schwartz. Internet of Things with ESP8266. 2016, 226p.

18. Офіційний сайт Arduino. URL: https://www.arduino.cc. (дата звернення: 21.03.2022).

19. Інтернет-ресурс. Telegram Bot API. URL: https://core.telegram.org/bots/api (дата звернення: 08.04.2022).

20. Інтернет-ресурс. Motion Sensor. URL: https://www.safewise.com/resources/motion-sensor-guide/ (дата звернення: 20.04.2022).

21. Інтернет-ресурс. PIR Motion Sensor HC-SR501. URL: https://www.chipdip.ru/product/hc-sr501-infrared-pir-motion-sensor-module (дата звернення: 23.04.2022).

22. Fritzing Inc., "Fritzing description," Interaction Design Lab Potsdam, 2 червень 2016. URL: http://fritzing.org/download/. (дата звернення: 1.06.2022).

23. Сташин В.В. Проектирование цифровых устройств на однокристальных микроконтроллерах / Сташин В.В. - М.:Энергоатомиздат, 2010. – 224 с.

24. Усков А. А. Методические указания по выполнению курсового проекта покурсу «Разработка и стандартизация программных средств информационных технологий». Смоленск: СФ АНО ВПО ЦС РФ «РУК»,2007. – С. 15- 44.

25. Уокерлі Дж. Архітектура та програмуваннямікроЕОМ: У 2-х кн./Пер. з англ. М .: Світ, 1984. - Кн. 2. - 359 с.

26. Хосе М. Ангуло. Мікропроцесори: Архітектура, програмування та проєктування систем. Тбілісі: Ганатлеба, 1989. – С. 45-54.

27. Хоровіц П., Хілл У. Мистецтво схемотехніки. М .: Світ, 1983. - Т. 2. - 590 с.

28. Щелкунов Н.Н. Микропроцессорные средства и системы / Щелкунов Н. Н., Дианов А. Н. - М.:Радио и связь, 2009. – 360 с.

29. Koba Natroshvili. Obstacle detection and obstacle detection device and obstacle detection and obstacle detection method. Current Assignee: Intel Corp, 2018, p. 66, URL: https://patents.google.com/patent/DE102019121785A1/en.

Додаток А

Код програми для мікроконтролера

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <Wire.h>

// Network credentials
const char* ssid = "240";
const char* password = "fPlKcvUjM";

// Use @myidbot to find out the chat ID of an individual or a group
String chatId = "468157343";

// Initialize Telegram BOT
String BOTtoken = "5490387788:AAGfV_If8e8dOniANiad3UEjibgekInm6kE";

bool sendPhoto = false;

WiFiClientSecure clientTCP;

UniversalTelegramBot bot(BOTtoken, clientTCP);

//CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
```

```
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

#define FLASH_LED_PIN 4
bool flashState = LOW;

// Motion Sensor
bool motionDetected = false;


int botRequestDelay = 1000;   // mean time between scan messages
long lastTimeBotRan;     // last time messages' scan has been done

void handleNewMessages(int numNewMessages);
String sendPhotoTelegram();

// Indicates when motion is detected
static void IRAM_ATTR detectsMovement(void * arg){
  //Serial.println("MOTION DETECTED!!!");
  motionDetected = true;
}

void setup(){
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);

  pinMode(FLASH_LED_PIN, OUTPUT);
  digitalWrite(FLASH_LED_PIN, flashState);


  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
```

```
        clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for
api.telegram.org
        while (WiFi.status() != WL_CONNECTED) {
          Serial.print(".");
          delay(500);
        }
        Serial.println();
        Serial.print("ESP32-CAM IP Address: ");
        Serial.println(WiFi.localIP());

        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;

        //init with high specs to pre-allocate larger buffers
        if(psramFound()){
          config.frame_size = FRAMESIZE_UXGA;
          config.jpeg_quality = 10;  //0-63 lower number means higher quality
          config.fb_count = 2;
        } else {
          config.frame_size = FRAMESIZE_SVGA;
          config.jpeg_quality = 12;  //0-63 lower number means higher quality
          config.fb_count = 1;
```

```
    }

    // camera init
    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {
      Serial.printf("Camera init failed with error 0x%x", err);
      delay(1000);
      ESP.restart();
    }

    // Drop down frame size for higher initial frame rate
    sensor_t * s = esp_camera_sensor_get();
    s->set_framesize(s,              FRAMESIZE_CIF);                          //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA

    // PIR Motion Sensor mode INPUT_PULLUP
    //err = gpio_install_isr_service(0);
    err = gpio_isr_handler_add(GPIO_NUM_13, &detectsMovement, (void *) 13);
    if (err != ESP_OK){
      Serial.printf("handler add failed with error 0x%x \r\n", err);
    }
    err = gpio_set_intr_type(GPIO_NUM_13, GPIO_INTR_POSEDGE);
    if (err != ESP_OK){
      Serial.printf("set intr type failed with error 0x%x \r\n", err);
    }
  }

  void loop(){
    if (sendPhoto){
      Serial.println("Preparing photo");
      sendPhotoTelegram();
      sendPhoto = false;
    }

    if(motionDetected){
      bot.sendMessage(chatId, "Motion detected!!", "");
      Serial.println("Motion Detected");
      sendPhotoTelegram();
      motionDetected = false;
    }
```

```
    if (millis() > lastTimeBotRan + botRequestDelay){
      int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
      while (numNewMessages){
        Serial.println("got response");
        handleNewMessages(numNewMessages);
        numNewMessages = bot.getUpdates(bot.last_message_received + 1);
      }
      lastTimeBotRan = millis();
    }
  }

  String sendPhotoTelegram(){
    const char* myDomain = "api.telegram.org";
    String getAll = "";
    String getBody = "";

    camera_fb_t * fb = NULL;
    fb = esp_camera_fb_get();
    if(!fb) {
      Serial.println("Camera capture failed");
      delay(1000);
      ESP.restart();
      return "Camera capture failed";
    }

    Serial.println("Connect to " + String(myDomain));

    if (clientTCP.connect(myDomain, 443)) {
      Serial.println("Connection successful");

      String head = "--AlertSystem\r\nContent-Disposition: form-data; name=\"chat_id\"; \r\n\r\n" + chatId + "\r\n--AlertSystem\r\nContent-Disposition: form-data; name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
      String tail = "\r\n--AlertSystem--\r\n";

      uint16_t imageLen = fb->len;
      uint16_t extraLen = head.length() + tail.length();
      uint16_t totalLen = imageLen + extraLen;

      clientTCP.println("POST /bot"+BOTtoken+"/sendPhoto HTTP/1.1");
```

```
          clientTCP.println("Host: " + String(myDomain));
          clientTCP.println("Content-Length: " + String(totalLen));
          clientTCP.println("Content-Type:                        multipart/form-data;
boundary=AlertSystem");
          clientTCP.println();
          clientTCP.print(head);

          uint8_t *fbBuf = fb->buf;
          size_t fbLen = fb->len;
          for (size_t n=0;n<fbLen;n=n+1024) {
            if (n+1024<fbLen) {
              clientTCP.write(fbBuf, 1024);
              fbBuf += 1024;
            }
            else if (fbLen%1024>0) {
              size_t remainder = fbLen%1024;
              clientTCP.write(fbBuf, remainder);
            }
          }

          clientTCP.print(tail);

          esp_camera_fb_return(fb);

          int waitTime = 10000;   // timeout 10 seconds
          long startTimer = millis();
          boolean state = false;

          while ((startTimer + waitTime) > millis()){
            Serial.print(".");
            delay(100);
            while (clientTCP.available()) {
              char c = clientTCP.read();
              if (state==true) getBody += String(c);
              if (c == '\n') {
                if (getAll.length()==0) state=true;
                getAll = "";
              }
              else if (c != '\r')
                getAll += String(c);
              startTimer = millis();
```

```
      }
      if (getBody.length()>0) break;
    }
    clientTCP.stop();
    Serial.println(getBody);
  }
  else {
    getBody="Connected to api.telegram.org failed.";
    Serial.println("Connected to api.telegram.org failed.");
  }
  return getBody;
}

void handleNewMessages(int numNewMessages){
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++){
    // Chat id of the requester
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != chatId){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String fromName = bot.messages[i].from_name;

    if (text == "/flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
    }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("New photo  request");
    }
    if (text == "/start"){
      String welcome = "Welcome to the ESP32-CAM Telegram bot.\n";
```

```
        welcome += "/photo : takes a new photo\n";
        welcome += "/flash : toggle flash LED\n";
        welcome += "You'll receive a photo whenever motion is detected.\n";
        bot.sendMessage(chatId, welcome, "Markdown");
      }
    }
  }
```