МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю. М. Потебні

Кафедра електроніки, інформаційних систем
та програмного забезпечення
(повна назва кафедри)

Кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на *тему «Дослідження та розробка системи віртуальної приватної мережі (VPN) на базі одноплатного комп'ютера»*

Виконав: студент (ка) *II* курсу, групи *8.1532*
спеціальності *176 Мікро- та наносистемна техніка*
(код і назва спеціальності)
освітньої програми *176 Мікроелектронні інформаційні системи*
(код і назва освітньої програми)
спеціалізації _____
(код і назва спеціалізації)
Небеснюк Владислав Олександрович
(ініціали та прізвище)

Керівник *професор кафедри ЕІСПЗ, доцент, к.т.н., Ніконова Зоя Андріївна*
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент *інженер ТОВ «НВП Імпульс» Кузько А.О.*
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя
2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні

Кафедра електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти другий (магістерський)
(перший (бакалаврський) рівень
Спеціальність 176 Мікро- та наносистемна техніка
(назва)
Освітня програма Мікроелектронні інформаційні системи
(шифр)

ЗАТВЕРДЖУЮ
Завідувач кафедри Т. В. Критська
«30» листопада 2023 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ (СТУДЕНТЦІ)

Небеснюку Владиславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи (проєкту) *Дослідження та розробка системи віртуальної приватної мережі (VPN) на базі одноплатного комп'ютера*
керівник роботи *Ніконова З.А., к.т.н.,доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від "01" травня 2023 року № 639-с
2.Строк подання студентом роботи 30.11 2023 р.
3. Вихідні дані до роботи: *віртуальна приватна мережа: Raspberry Pi 3B: 64-бітний чотириядерний процесором ARM Cortex-A53 з тактовою частотою 1,2 ГГц на однокристальному чіпі Broadcom BCM2837; вбудований Wi-Fi 802.11n та Bluetooth 4.1.*
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) *1 Порівняльний аналіз та класифікація віртуальних приватних мереж 2 Дослідження та розробка апаратно-програмного комплексу для віддаленого доступу до заблокованих мережна основі RASPBERRY P3 3Техніко-економічне обґрунтування 4 Охорона праці та техногенна безпека*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) *А4 Порівняння технологій проксі і VPN. Типи проксі-серверів. Структурна схема віртуальної приватної мережі (VPN) на базі одноплатного комп'ютера RASPBERRY PI 3. Розробка панелі управління. Протокол передачі даних HTT. Структура та основні характеристики Raspberry PI 3.Схема електрична*

принципова. Схема розміщення елементів на платі.

6. Консультанти розділів кваліфікаційної роботи бакалавра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
I	Ніконова З.А., професор каф. ЕІСПЗ	24.10.22	17.02.23
II	Ніконова З.А., професор каф. ЕІСПЗ	17.02.23	20.10.23
III	Ніконова З.А., професор каф. ЕІСПЗ	23.10.23	30.10.23
IV	Ніконова З.А., професор каф. ЕІСПЗ	01.11.23	16.11.23

7. Дата видачі завдання *24.10.2022р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів кваліфікаційної роботи бакалавра	Примітка
1	*Аналіз матеріалу за темою кваліфікаційної роботи*	*24.10. -28.11.22*	
2	*Порівняльний аналіз та класифікація віртуальних приватних мереж*	*28.11.22 - 17.02.23*	
3	*Розробка структурної схеми віртуальної мережи*	*17.02.23 - 04.04.23*	
4	*Схемотехнічний аналіз та написання програмного забезпечення*	*04.04-30.08.23*	
5	*Оформлення другого розділу*	*01.09 -20.10.23*	
6	*Розділ «Техніко-економічне обґрунтування»*	*23.10-30.10.23*	
7	*Розділ «Охорона праці та техногенна безпека»*	*01.11. -16.11.23*	
8	*Оформлення пояснювальної записки, виконання креслень*	*20.11. -27.11.23*	

Студент _____ *Небеснюк В.О.*
(підпис) (прізвище та ініціали)

Керівник роботи (проєкту) _____ *Ніконова З.А..*
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ *Верьовкін Л. Л.*
(підпис) (прізвище та

Реферат

Кваліфікаційна робота містить 99 сторінок, 10 рисунків, 7 таблиць, 30 джерел літератури.

Об'єкт дослідження – віртуальні приватні мережі.

Мета роботи – створення апаратно-програмного комплексу для віддаленого доступу до заблокованих мереж на основі Raspberry PI.

Задачі роботи - провести порівняльний аналіз технологій створення віртуальних приватних мереж; дослідити Raspberry PI 3 Model B; розробити апаратно-програмний комплекс на базі Raspberry PI 3 Model B; розробити програмне рішення для реалізації VPN; реалізувати допоміжні компоненти проксі серверів.

Методика досліджень – розробка програмного забезпечення на мові програмування Python; моделювання приладу в програмному середовищі Splan.

Короткий виклад результатів досліджень – розроблена віртуальна приватна система повного циклу, що дозволяє отримувати доступ до заблокованих ресурсів та бути максимально анонімним користувачем для систем аналізу трафіку.

Результаті впроваджень – комплекс пройшов випробування на кафедрі ЕІСПЗ.

Прогнозні пропозиції – рекомендується для надання віддаленого доступу до заблокованих мереж як поодиноким користувачам, так і великим кампаніям.

VPN, RASPBERRY PI, PYTHON, ОДНОПЛАТНИЙ КОМП’ЮТЕР, СЕРВЕР, ЕЛЕКТРИЧНА СХЕМА

Кваліфікаційну роботу виконано в Інженерному навчально-науковому інституті ім. Ю.М. Потебні на кафедрі електроніки інформаційних систем та програмного забезпечення в період з 24.10.22 р. по 30.11.23р.

ЗМІСТ

ВСТУП

Війна в Україні примусила багатьох людей тимчасово виїхати за кордон, а частина населення опинилася на тимчасово окупованих територіях. Ці категорії не мають можливості отримувати повноцінний доступ до контенту, який обмежений певним регіоном, в даному випадку -Україна.

Окрім того, у всесвітній мережі є безліч небажаних небезпек, зокрема, зловмисники можуть збирати персональні дані, відстежувати активність та перехоплювати з’єднання. Також деякі мережі встановлюють обмеження у вигляді блокування певних веб-сайтів, що може значно обмежувати свободу в Інтернет.

VPN (англ. Virtual Private Network - віртуальна приватна мережа) - узагальнена назва технологій, що дозволяють забезпечити одне або кілька мережних з'єднань (логічну мережу) поверх іншої мережі (наприклад, Інтернет).

Віртуальні приватні мережі можуть використовуватися для віддаленого доступу співробітників до робочої мережі, для безпечного підключення різних відділів організації, для обходу блокування сайтів, для забезпечення анонімності. Одним з можливих способів застосування VPN є обхід наявних мережевих обмежень. Це знадобиться, наприклад, щоб отримати доступ до сайтів і ресурсів, які заблоковані на певній території [1]

Тож застосування віртуальної приватної мережі (VPN) дозволить забезпечити захист та конфіденційність особи під час користування Інтернетом**.**

Таким чином, дослідження та розробка системи віртуальної приватної мережі (VPN) на базі одноплатного комп'ютера є достатньо актуальним питанням.

1 ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА КЛАСИФІКАЦІЯ ВІРТУАЛЬНИХ ПРИВАТНИХ МЕРЕЖ

1.1 Види підключення віртуальної приватної мережі

VPN (англ. Virtual Private Network - віртуальна приватна мережа) - узагальнена назва технологій, що дозволяють забезпечити одне або кілька мережних з'єднань (логічну мережу) поверх іншої мережі (наприклад, Інтернет) (рис1.1).

При підключенні до Інтернет через VPN-додаток на пристрої користувача (його ще називають VPN-клієнтом) між його пристроєм і VPN-сервером встановлюється безпечне з’єднання. Трафік проходить через провайдера клієнта, але він не може його прочитати або побачити кінцевий пункт призначення. Веб-сайти, які відвідує користувач, більше не бачать його вихідну IP-адресу, а бачать тільки IP-адресу VPN-сервера, яка спільно використовується багатьма іншими користувачами і регулярно змінюється.

Рисунок 1.1- Схематичне зображення віртуальної приватної мережі [2]

Залежно від застосовуваних протоколів і призначення, VPN може забезпечувати з'єднання трьох видів: вузол-вузол, вузол-мережу та мережу-мережу [3].

Сучасні види VPN підключення :

OpenVPN — протокол з відкритим вихідним кодом, який відрізняється своєю надійністю та безпечністю. Відкритий вихідний код означає, що його програмний код є загальнодоступним, і тому будь-хто може перевірити його на міцність та надати власні рекомендації, які допоможуть зробити протокол ще безпечнішим. Він дуже популярний серед користувачів і шифрує трафік з обох боків, тобто лише відправник і одержувач мають ключ шифрування. Також він регулярно оновлюється та вдосконалюється, що додатково підвищує рівень його безпеки.

WireGuard — ще один протокол з відкритим кодом, швидший, ніж OpenVPN, і в той самий час не менш безпечний. Рекомендується використовувати його для потокової передачі, онлайн-ігор та відео дзвінків. Проте він може мати невиявлені вразливості, оскільки є відносно новим.

IKEv2 — надійний протокол зі швидкістю на рівні OpenVPN. Цей протокол дуже стабільний в роботі, тому він зможе захистити користувача навіть під час переходу з мобільної мережі на Wi-Fi. Однак він менш безпечний, ніж OpenVPN та WireGuard, тому його краще використовувати як резервний варіант.

SSTP — досить старий протокол для Windows, схожий за принципом роботи на OpenVPN, оскільки лише одержувач і відправник мають ключі для розшифрування з’єднання. Це дозволяє ефективно долати системи контролю трафіку, проте швидкість його роботи дещо занизька.

L2TP/IPSec — застарілий протокол, який в основному використовується на смартфонах. Він не шифрує дані та працює дуже повільно, тому його краще уникати.

1.2 Класифікація VPN

VPN класифікують за кількома основними параметрами [4]:

1. За ступенем захищеності використовуваного середовища

1.1. Захищені

Найбільш поширений варіант віртуальних приватних мереж. З його допомогою можливо створити надійну і захищену мережу на основі ненадійної мережі, як правило, Інтернету. Прикладом захищених VPN є: IPSec, OpenVPN і PPTP.

1.2. Довірчі

Використовуються у випадках, коли передавальну середу можна вважати надійною і необхідно вирішити лише завдання створення віртуальної підмережі в рамках більшої мережі. Проблеми безпеки стають неактуальними. Прикладами подібних рішень VPN є: Multi-protocol label switching (MPLS) і L2TP (Layer 2 Tunnelling Protocol) (точніше буде сказати, що ці протоколи перекладають завдання забезпечення безпеки на інші, наприклад L2TP, як правило, використовується в парі з IPSec).

2. За способом реалізації

2.1. У вигляді спеціального програмно-апаратного забезпечення

Реалізація мережі VPN здійснюється за допомогою спеціального комплексу програмно-апаратних засобів. Така реалізація забезпечує високу продуктивність і, як правило, високий ступінь захищеності.

2.2. У вигляді програмного рішення

Використовують персональний комп'ютер зі спеціальним програмним забезпеченням, що забезпечує функціональність VPN.

2.3. Інтегроване рішення

Функціональність VPN забезпечує комплекс, вирішальний також завдання фільтрації мережевого трафіку, організації мережевого екрану і забезпечення якості обслуговування.

3. По призначенню

3.1. Intranet VPN

Використовують для об'єднання в єдину захищену мережу декількох розподілених філій однієї організації, які обмінюються даними по відкритих каналах зв'язку.

3.2. Remote Access VPN

Використовують для створення захищеного каналу між сегментом корпоративної мережі (центральним офісом або філією) і одиночним користувачем, який, працюючи вдома, підключається до корпоративних ресурсів з домашнього комп'ютера, корпоративного ноутбука, смартфона або інтернет-кіоску.

3.3. Extranet VPN

Використовують для мереж, до яких підключаються «зовнішні» користувачі (наприклад, замовники або клієнти). Рівень довіри до них набагато нижче, ніж до співробітників компанії, тому потрібне забезпечення спеціальних «кордонів» захисту, що запобігають або обмежують доступ останніх до особливо цінної, конфіденційної інформації.

3.4. Internet VPN

Використовується для надання доступу до інтернету провайдерами, зазвичай якщо по одному фізичному каналу підключаються кілька користувачів. Протокол PPPoE став стандартом в ADSL-підключення.

3.5. L2TP

Був широко поширений в середині 2000-х років в будинкових мережах: в ті часи внутрішньо трафік не оплачувалася, а зовнішній коштував дорого. Це давало можливість контролювати витрати: коли VPN-з'єднання вимкнено,

користувач нічого не платить. В даний час провідний інтернет дешевий або безлімітний, а на стороні користувача часто є маршрутизатор, на якому вмикати-вимикати інтернет не так зручно, як на комп'ютері. Тому L2TP-доступ відходить в минуле.

3.6. Client / Server VPN

Він забезпечує захист переданих даних між двома вузлами (не мережами) корпоративної мережі. Особливість даного варіанту в тому, що VPN будується між вузлами, що перебувають, як правило, в одному сегменті мережі, наприклад, між робочою станцією і сервером. Така необхідність дуже часто виникає в тих випадках, коли в одній фізичній мережі необхідно створити кілька логічних мереж. Наприклад, коли треба розділити трафік між фінансовим департаментом та відділом кадрів, які звертаються до серверів, що знаходяться в одному фізичному сегменті. Цей варіант схожий на технологію VLAN, але замість поділу трафіку використовується його шифрування.

4. За типом протоколу

Існують реалізації віртуальних приватних мереж під TCP / IP, IPX і AppleTalk. Але на сьогоднішній день спостерігається тенденція до загального переходу на протокол TCP / IP, і абсолютна більшість рішень VPN підтримує саме його. Адресація в ньому найчастіше вибирається відповідно до стандарту RFC5735, з діапазону Приватних мереж TCP / IP.

5. За рівнем мережевого протоколу

За рівнем мережевого протоколу на основі зіставлення з рівнями еталонної мережевої моделі ISO / OSI.

1.3 Основні функції віртуальної приватної мережі

Можна виділити такі основні функції VPN [5]:

- Конфіденційність (анонімність в мережі). За допомогою VPN користувач може приховати свій реальний IP-адрес і таким чином уникнути ідентифікації особистості в мережі і забезпечити захист персональних даних. Інтернет-провайдер буде бачити тільки факт захищеного з'єднання з одним з віддалених серверів VPN-сервісу. Таким чином, користувач захищає свої особисті дані, історію відвідин сайтів і зберігаєте анонімність**.** Більшість VPN використовують 256-бітне шифрування за стандартом AES, що є найнадійнішим серед доступних рівнів шифрування. 256-бітний ключ є найдовшим серед всіх ключів шифрування, а чим він довший, тим більше часу буде потрібно для його розшифрування. Саме ним користуються провідні служби безпеки та уряди для захисту найбільш конфіденційних даних.
- Розблокування сайтів і сервісів . При використанні VPN особа зможе отримувати доступ до будь-яких заблокованих сайтів або додатків, завантажувати потрібну інформацію без обмежень провайдерів. В цьому випадку VPN виконує пряму функцію «тунелю», дозволяючи обходити заборони або обмеження за гео-ознакою.
- Wi-Fi безпеку . Якщо користувач використовує публічний Wi-Fi, який є найнебезпечнішим з погляду безпеки, то VPN зашифрує і убезпечить його інформацію - це можуть бути паролі від пошти, соціальних мереж, ключі доступу до гаманців та іншу важливу особисту інформацію. Сигнал Wi-Fi йде по звичайній радіохвилі і перебуваючи в зоні досяжності цих хвиль, шахраї легко отримують інформацію з Вашого девайса. Найчастіше, хакери, перебуваючи поблизу - створюють і настроюють мережу близнюк Wi-Fi - один в один з мережею, яку користувачеві представляють офіційно, і перехоплюють його інформацію.

1.4 Недоліки застосування VPN

Не зважаючи на велику кількість переваг застосування віртуальної приватної мережі слід відзначити наступні недоліки:

Використання VPN може дещо знизити швидкість з’єднання. Деякі VPN-сервіси значно знижують швидкість Інтернету, інші — лише незначним чином, а топові сервіси роблять цю затримку майже непомітною. Сповільнення з'єднання обумовлюється шифруванням даних та їх відправленням на сервер VPN, що вимагає часу.

Деякі VPN-сервіси можуть поставити під загрозу конфіденційність користувача. Краще обирати VPN-сервіс, який дотримується суворої політики відсутності журналів входу, перевіреної незалежними компаніями із забезпечення кібербезпеки.

Користування VPN не є безкоштовним. VPN є сервісами на основі підписки, які регулярно стягують плату за свої послуги. Проте користування більшістю VPN-сервісів коштує лише кілька доларів на місяць, тож насправді вони досить доступні. Крім того, вони, як правило, пропонують гарантію повернення коштів, що дозволяє спочатку випробувати їхні послуги, а потім, протягом встановленого терміну, отримати відшкодування коштів. Таким чином, користувачу не доведеться оформлювати підписку, доки він не пересвідчитеся, що сервіс йому підходить.

Погані VPN-сервіси мають обмежену мережу серверів та кількість IP-адрес. Деякі з них використовують маленькі серверні мережі або застарілу інфраструктуру. Якщо VPN не здатен оновлювати свої IP-адреси та сервери, це, ймовірно, буде негативно впливати на швидкість та значно обмежувати доступ користувача до потокового контенту.

Деякі країни обмежують або забороняють використання VPN-сервісів. Такі країни як Китай, Росія та Іран частково обмежують або навіть

забороняють користування VPN. В певних країнах, наприклад в Китаї, вебсайти деяких VPN-сервісів заборонені, тому перш ніж користуватися VPN в регіонах з подібними обмеженнями, краще ознайомитися з останніми місцевими законами щодо його використання [6].

Ще однією технологією, що виконує функцію посередника між Інтернетом і користувачем є використання проксі-сервера.

1.5 Проксі-сервер

Проксі-сервер — це інший сервер, який представляє пристрій користувача в Інтернеті. Він виконує роль посередника між пристроєм і веб-сайтами чи онлайн-службами. У разі підключення до нього всі вихідні та вхідні дані користувача проходять через проксі-сервер. Ваша IP-адреса замінюється IP-адресою сервера, при цьому всі онлайн-системи «думають», що ви підключаєтеся з сервера, а не зі свого пристрою [7].

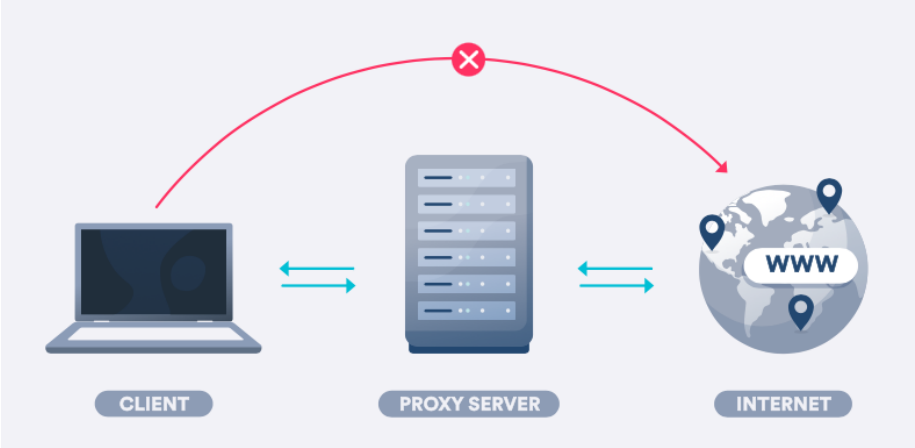

Рисунок 1.2 -Принцип застосування проксі-сервера

Алгоритм роботи проксі-сервера:

- користувач вводить адресу веб-сайту у своєму браузері;
- проксі-сервер отримує цей запит;
- проксі-сервер спрямовує запит на веб сервер, до якого користувач намагається підключитися;
- веб сервер надсилає відповідь (дані веб-сайту) назад на проксі-сервер;
- проксі-сервер пересилає відповідь користувачу.

1.5.1 Призначення проксі-сервера

Завдання, які можуть виконувати проксі-сервера:

- Обхід обмежень.

Якщо компанія або навчальний заклад блокує IP-адреси певних сайтів, користувачі можуть заходити на них через проксі-сервер. При цьому брандмауер підприємства або навчального закладу бачитиме, що користувачі під'єднуються до проксі-сервера, а не до сайту. Якщо проксі не заблокований, запит буде успішно виконано.

- Анонімність потокового передавання.

Якщо користувач хоче переглядати потоковий вміст конфіденційно (наскільки це можливо для передплатника потокової служби), то може використовувати для цього проксі-сервер. Головна умова — цей проксі має бути зашифрованим. Безкоштовний проксі-сервер не забезпечить захист і конфіденційність.

- Посилення безпеки в мережі.

Проксі-сервер може виконувати функцію брандмауера, що захищає окремих користувачів та компанії від шкідливих атак в Інтернеті. Він може посилювати безпеку на базовому рівні, приховуючи IP-адресу користувача — адресу пристрою в Інтернеті. Якщо користувач при цьому сам себе не ідентифікуєте, ніхто не дізнається, які сайти він відвідував.

- Контроль користування Інтернетом.

  Ця функція потрібна в основному компаніям або іншим особам, які мають власні проксі-сервери. Налаштувавши проксі, можна заблокувати деякі веб-адреси, щоб користувачі пристроїв, підключених до вашого проксі-сервера, не мали до них доступу.
- Збільшення швидкості з економією пропускної спроможності.

  Іще одна функція, розрахована більше на компанії, ніж на окремих користувачів, — це кешування (збереження) проксі-серверами копій веб сторінок, що часто відвідуються. Окрім того, якщо п'ять користувачів мережі відвідують той самий сайт, сервер може пропінгувати його (перевірити зв'язок) один раз, а потім роздати інформацію на п'ять пристроїв, що зменшить навантаження на мережу [8].

1.5.2 Типи проксі-серверів і протоколи, які вони використовують

Розглянемо типи проксі-серверів:

- Прямий (звичайний) проксі-сервер (рис.1.3) : найпоширеніший тип проксі — це посередник, що пересилає дані користувача від його імені. Простіше кажучи, цей проксі представляє користувача у мережі. Окрім того, цей проксі забезпечує певний рівень захисту, оскільки не перенаправляє трафік доти, доки дані не будуть перевірені та визначені безпечними.
- Зворотний проксі-сервер (рис.1.4)**:** якщо прямий проксі представляє користувача, то зворотний проксі використовується веб сервером (тому і називається зворотним). Веб сервери (постачальники послуг) використовують зворотні проксі для кешування та отримання необхідних даних. Завдяки цим діям вони забезпечують безперебійність роботи для

користувачів та зменшують навантаження на свої служби. Користувач не знає, що підключаєтеся до нього, але він може збирати необхідні дані з декількох сайтів і потім передавати їх користувачу (за принципом «об'єднаної доставки» з онлайн-магазинів).

- Веб проксі**:** грає роль балансувача навантаження — розподіляє запити між кількома серверами для ефективної роботи сервісу та запобігання перебоїв в разі значного збільшення числа запитів.
- Анонімний проксі-сервер**:** іноді його називають анонімайзером або викривляючим проксі: він приховує вашу вихідну IP-адресу і надає вам нову, але не приховує самого факту використання вами проксі-сервера. ідентифікують себе як проксі-сервери у заголовку відповіді, але на запит видають помилкову IP-адресу клієнта.

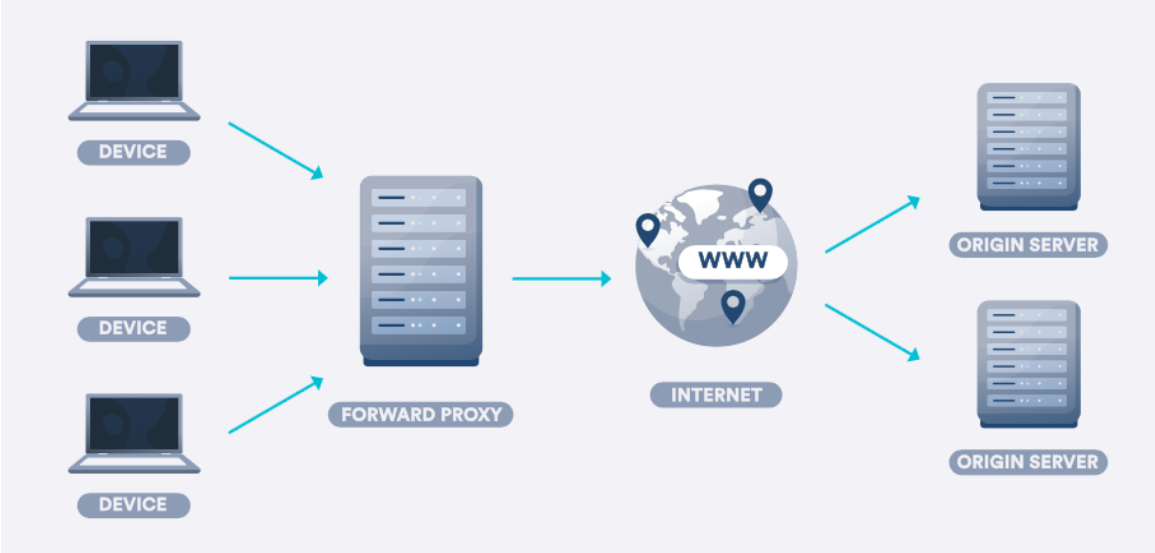

Рисунок 1.3 – Реалізація прямого проксі-сервера [9]

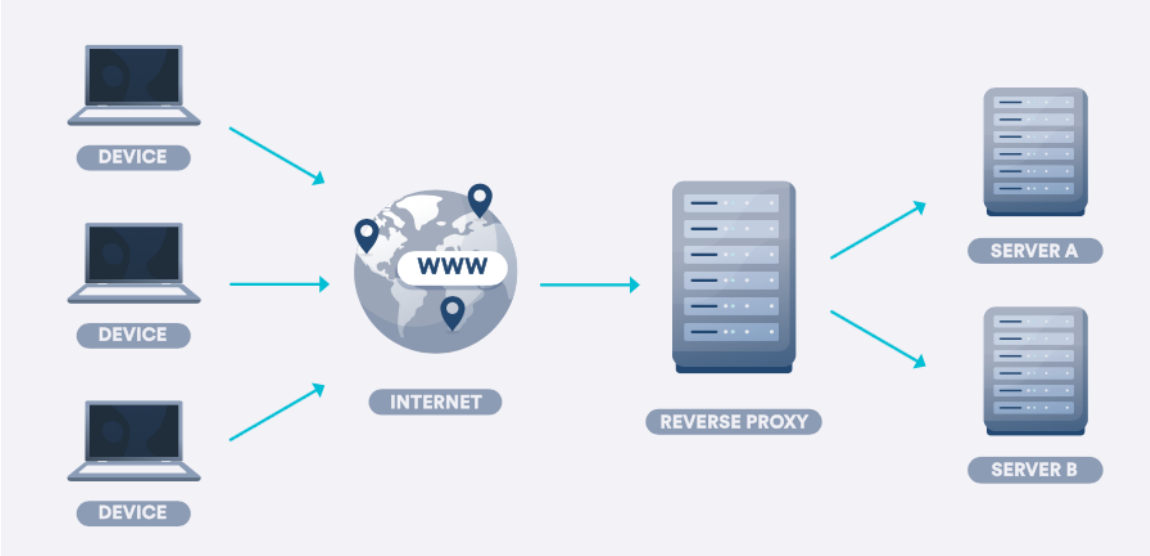

Рисунок 1.4 – Реалізація зворотного проксі-сервера [9]

- Проксі з посиленою анонімністю: приховує як вихідну IP-адресу, так і факт використання проксі-сервера, регулярно змінюючи IP-адреси і не маючи в заголовку даних, що його розкривають.
- Прозорий проксі-сервер: називається так через свою непомітність для користувача. Фактично він не змінює онлайн-запити та використовується для відстеження використання Інтернету й обмеження доступу. Їх часто використовують роботодавці, щоб контролювати своїх співробітників та не давати їм «гуляти по мережі» замість виконання робочих завдань. Можуть використовуватися також публічними бібліотеками.
- CGI-проксі (загальний шлюзовий інтерфейс)**:** ця технологія підключається до проксі-сервера через веб-сайт. Користувач заходить на проксі-сайт CGI, вводить адресу потрібного сайту у веб форму, і він відображається на сторінці проксі-сайту — виходить схоже на браузер у браузері. Якщо у користувача немає доступу до налаштувань проксі або пристрій не підтримує таку функцію, то CGI-проксі — підходяще рішення.
- Суфікс-проксі**:** додає закінчення (суфікс) до адреси сайту задля обходу фільтрів брандмауера (проте сучасні фільтри можуть блокувати такі запити).

- DNS-проксі (Domain Name System — «система доменних імен»)**:** комп'ютери використовують DNS для перекладу домашньої адреси веб сторінки з людської мови на цифрову — наприклад, з surfshark.com на 104.18.120.34 (IP-адреса). Проксі-сервер DNS обробляє, дозволяє або блокує всі DNS-запити. Наприклад, в разі введення Surfshark.com DNS вибере, який із серверів Surfshark буде виконувати цей запит.

Проксі-протоколи — набори правил цифрової взаємодії, що визначають спосіб їх налаштування.

Існують різні проксі-протоколи [10]:

- SSL**:** Secure Sockets Layer (іноді його називають проксі-сервером HTTPS) – це проксі-протокол, який використовується для захисту даних у процесі їх передавання (наприклад, у транзакції під час онлайн-покупки). При цьому зашифровується і веб сторінка, до якої звертається користувач, і трафік даних, що передаються на неї і виходить від неї.
- FTP**:** File Transfer Protocol використовується для завантаження даних на сервер (наприклад, коли ви вивантажуєте свої зображення в хмару або додаєте файли музики в музичні онлайн-сервіси). FTP-проксі пропонує посилену безпеку файлів, що завантажуються користувачами.
- HTTP: (hypertext transfer protocol) використовується для кешування (себто зберігання) веб сторінок і файлів, щоб прискорити доступ користувачів до них на сайтах, які вони часто відвідують. При цьому слід регулярно очищувати кеш на своєму пристрої, щоб він не сповільнював роботу вашого браузера.
- SOCKS**:** SOCKets Secure зв'язується зі стороннім проксі і спрямовує дані трафіку через їхні сервери на рівні, нижчому за HTTP, для обходу брандмауерів. Протокол Ergo SOCKS має додаткові заходи безпеки, які ускладнюють його виявлення (відомий як частина протоколу Shadowsocks).

1.5.3 Порівняння технологій проксі і VPN

Результати порівняння цих технологій представлені в таблиці 1.1

Таблиця 1.1-Порявняння технологій проксі і VPN

Проксі-сервер	VPN
Скеровує трафік користувача у браузері	Скеровує весь трафік пристрою користувача
Може приховувати IP-адресу користувача	Приховує IP-адресу користувача
Може мати шифрування	Шифрує усі дані користувача

Порівняння технологій VPN та проксі-сервера показує, що вони схожі тільки за принципом своєї роботи на базовому рівні — тим, що перенаправляють трафік даних через зовнішній сервер і змінюють IP-адресу користувача.

Відрізняються ці технології двома ключовими компонентами-протоколами, які використовують для забезпечення конфіденційності дій користувача у мережі, та алгоритмами шифрування даних.

Аналіз VPN та проксі-серверів показав, що найкращим підходом є використання змішаної моделі, яка комбінує переваги обох технологій. Це дозволить користувачам забезпечити високий рівень приватності, шифрування та анонімності завдяки VPN, а також використовувати проксі-сервери для додаткової прихованості їхньої реальної IP-адреси. Такий підхід буде особливо ефективним для тих випадків, коли важливо забезпечити якісний захист особистої інформації та одночасно мати можливість обходити географічні обмеження.

Мета роботи – створення апаратно-програмного комплексу для віддаленого доступу до заблокованих мереж на основі Raspberry Pi.

Завдання:

- провести порівняльний аналіз технологій створення віртуальних приватних мереж;
- дослідити Raspberry PI 3 Model B;
- розробити апаратно-програмний комплекс на базі Raspberry PI 3 Model B;
- розробити програмне рішення для реалізації VPN;
- реалізувати допоміжні компоненти проксі серверів.

2 ДОСЛІДЖЕННЯ ТА РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ВІДДАЛЕНОГО ДОСТУПУ ДО ЗАБЛОКОВАНИХ МЕРЕЖ НА ОСНОВІ RASPBERRY PI 3

2.1 Розробка структурної схеми роботи VPN

Запропонована віртуальна приватна мережа (VPN) на базі одноплатного комп'ютера RASPBERRY PI 3 працює наступним чином. Користувач за допомогою персонального комп'ютера по мережі зв’язується з “Панеллю управління”. Це звичайний веб-сайт, за допомогою якого можна гнучко керувати усіма частинами розробленої системи. За допомогою API інтерфейсу “Панель управління” пов’язана з “Головним сервером”. “Головний сервер” контролює усі мережеві процеси та за допомогою асинхронної комунікації вміє спілкуватись з системами Raspberry Pi через маршрутизатор. В свою чергу Raspberry Pi 3 має інтерфейси (USB) для спілкування з модемами та має змогу виконувати певні команди, які відносяться до фінальних пристроїв.

Розроблено структурну схему (рис.2.1) запропонованої мережі, що наочно демонструє логічний зв’язок між елементами системи та дозволяє пояснити принцип їх роботи та функціональне призначення.

Проведемо дослідження та розробку апаратної та програмної частини системи. Для системи віртуальної приватної мережі (VPN) на базі одноплатного комп'ютера RASPBERRY PI 3 в якості панелі управління пропонується застосувати систему Typical Proxy.

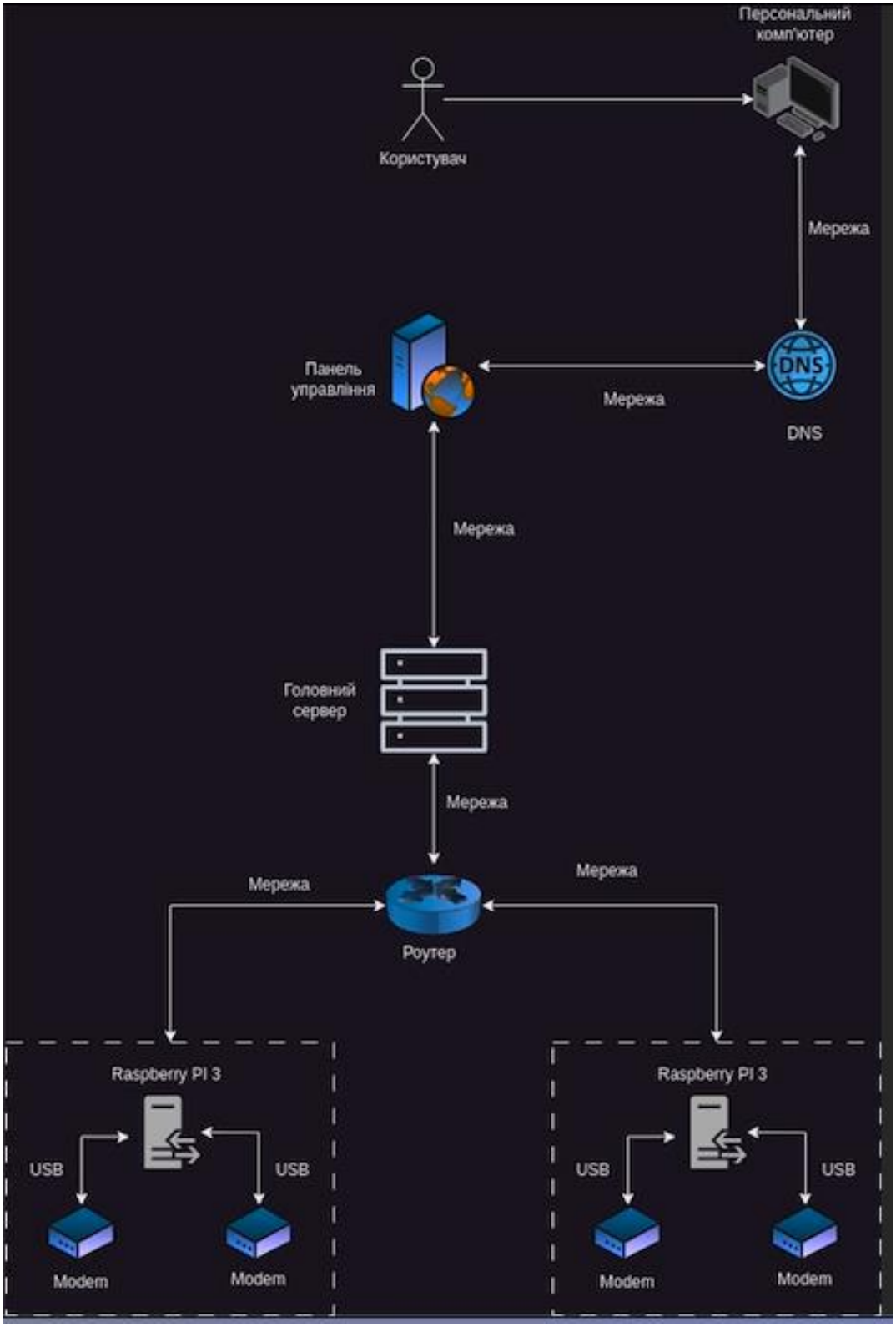

Рисунок 2.1- Структурна схема віртуальної приватної мережі (VPN) на базі одноплатного комп'ютера RASPBERRY PI 3

2.2 Розробка графічного інтерфейсу

2.2.1 Панель управління

В якості панелі управління використовується Typical Proxy - швидкий і простий проксі-сервіс. Його можна використовувати як проксі, VPN та отримувати смс-повідомлення в месенджері Telegram. [11]

Для входу необхідно мати логін і пароль (рис.2.2). Потім можливо перейти на сторінку панелі і ввести їх, щоб отримати доступ до панелі керування. Під час входу користувач бачить головну сторінку, що має автоматичне оновлення або інформаційну панель з основною інформацією.

У лівій колонці відображаються активні підключення в представленій системі (рис.2.23):

0 хвилин - час сесії

213.108.199.134 - IP-адреса користувача, який створив цей сеанс

46.133.193.125 - публічна IP-адреса, яку користувач отримує, коли будете використовувати цей сеанс. Червона іконка - додавання загальнодоступного IP до чорного списку

46.175.249.195:50005 - адреса проксі лише для інформації

Активне з'єднання може бути створено, тільки якщо ip-адреса користувача є у списку дозволу.

В центральній колонці головної сторінки представлена панель керування IP, де можливо додати нові IP-адреси до списку дозволу (тоді користувач матимете доступ для використання проксі-сервера) або чорного списку (лише загальнодоступні IP-адреси, які використовуються для реєстрації).

У правій колонці головного вікна представлені деталі проксі-сервера для підключення. Тут можливо використовувати ip та порт проксі-сервера для

підключення до запропонованої системи. Також перевірити, чи активна система чи ні та як довго.

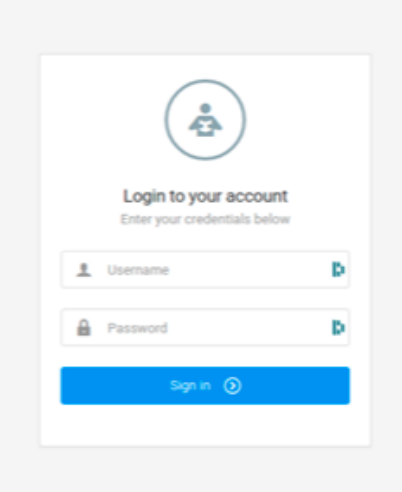

Рисунок 2.2 -Вигляд вікна для аутентифікації

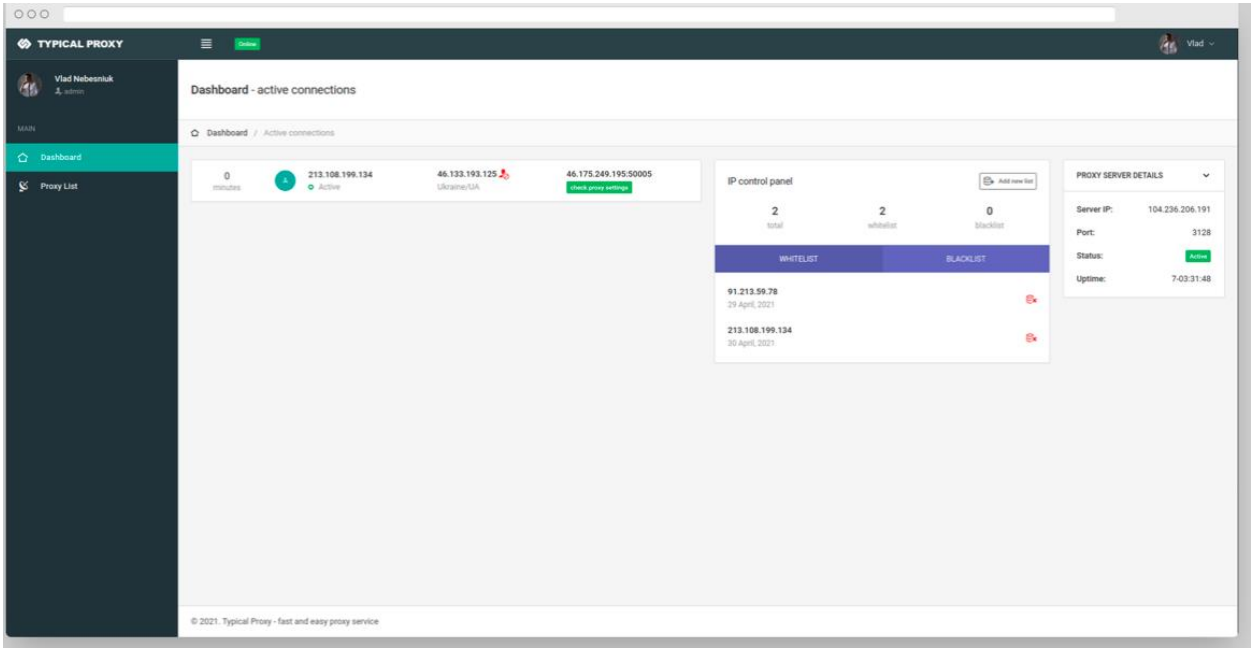

Рисунок 2.3- Головне вікно програмного застосунку

На другій сторінці головного вікна (рис.2.4) користувач може перевірити всі доступні модеми для використання та, за необхідності, ввести відповідні

зміни (оновити номер телефону, отримати нові sms, редагувати або видалити проксі).

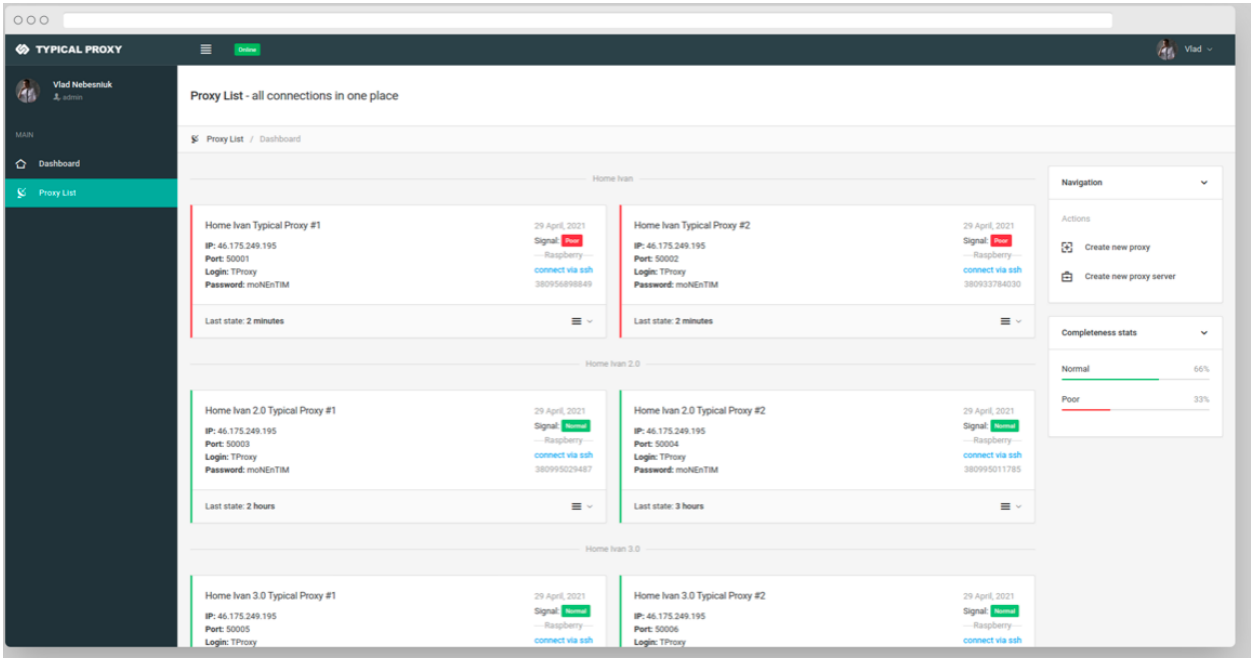

Рисунок 2.4- Модеми, доступні до використання

В якості проксі-серверу використовується міні-комп'ютер Raspberry PI 3, який керує двома модемами. В якості проксі – модему обрано Huawei e3372h. Щоб використовувати проксі необхідно додати свій IP до списку дозволу. Для цього:

Firefox

- запустіть Firefox
- перейдіть до Налаштування --> Налаштування мережі
- натисніть Налаштування проксі вручну
- введіть HTTP-проксі з IP 104.236.206.191 і портом 3128
- встановити прапорець Використовувати цей проксі для FTP і HTTPS
- натисніть OK
- закрийте Firefox

Chrome

- запустіть Google Chrome
- перейдіть до Параметри --> Додатково --> Система --> Відкрити налаштування проксі
- вимкнути автоматичне визначення параметрів
- включити Використовувати проксі-сервер
- введіть IP 104.236.206.191 і порт 3128
- натисніть Зберегти
- закрийте Google Chrome
- вимкніть Використовувати проксі-сервер

Для налаштування проксі-серверу необхідно виконати кілька кроків:

- придбати rasbperry, модеми, ethernet кабель, кабель живлення
- підключення модемів і rasbperry через USB-порт
- підключіть домашній маршрутизатор за допомогою кабелю Ethernet
- підключіть rasbperry до кабелю живлення

Тепер необхідно зробити деякі порти публічними, щоб отримати доступ ззовні:

- підключитися до домашньої мережі WIFI
- перейти на сторінку http://192.168.0.1 і ввести логін admin і пароль admin
- перейти до DHCP -> Список клієнтів DHCP і знайти малину та її IP-адресу

Потім перейти до розділу NAT Redirect -> Virtual Servers і створити кілька нових рядків:

- сервісний порт 22, IP-адреса -> rasbperry IP-адреса, внутрішній порт 22, протокол TCP
- сервісний порт 8333, ip-адреса -> IP-адреса rasbperry, внутрішній порт 8333, протокол TCP

- порт служби 50001, ip-адреса -> IP-адреса rasbperry, внутрішній порт 50001, протокол TCP
- порт служби 50002, ip-адреса -> IP-адреса rasbperry, внутрішній порт 50002, протокол TCP

Наступним кроком необхідно повторно ініціалізувати систему:

- Витягнути всі модеми з HUB-ів
- Увімкнути проксі-сервер (Raidmax)
- Зачекати завантаження
- Підключити модем по одному з тайм-аутом 2 хвилини
- Перевірити адміністративну панель
- Якщо деякі з модемів не працюють - перевірити, чи є у них активний план.

Всі запити в системі відображаються за допомогою протоколу HTTP.

2.2.2 Протокол передачі даних HTTP

HTTP – дуже широко розповсюджений протокол передачі даних, який попередньо був призначений для передачі гіпертекстових документів. Власне абревіатура розшифровується як протокол передачі гіпертексту.

Відповідно до специфікації OSI, HTTP відноситься до протоколів прикладного (верхнього, 7-го) рівня. Даний протокол передбачає використання клієнт-серверної структури передачі даних (рис.2.5).

На стороні клієнта формується запит і відправляється на сервер, після того як сервер приймає запит і успішно його опрацьовує, він одномоментно формує відповідь і повертає її зворотно, на сторону клієнта. Після цього клієнтський застосунок може надіслати інший запит, і процес “спілкування” відбудеться по аналогічному шляху.

На сьогоднішній день саме завдяки протоколу HTTP відбувається взаємодія всесвітньої павутини. Варто зазначити, що даний протокол часто застосовується при передачі даних іншими протоколами, а саме протоколами прикладного рівня, таких як SOAP, XML-RPC та WebDAV. [12]

В даному випадку прийнято говорити, що протокол HTTP використовується як “ транспорт ”. API, а також багато інших програмних продуктів передбачає використання HTTP, для передачі інформації. Дані в такому випадку можуть мати будь – який формат, наприклад, XML або JSON.

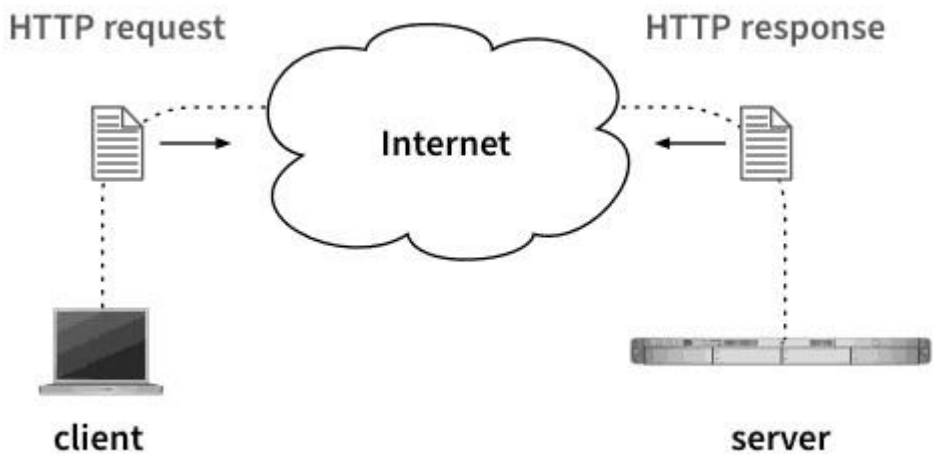

Рисунок 2.5 - Клієнт-серверна структура передачі даних [12]

В основному, передача даних здійснюється через TCP/IP з’єднання. Дане програмне забезпечення використовує TCP-порт 80, вказаний порт зазвичай використовується зі сторони клієнта по замовчуванню, проте при необхідності він може бути змінений на будь-який інший.

Для відправлення HTTP – запиту, необхідно сформувати пошукову строку якій в свою чергу як мінімум потрібно задати один заголовок, а саме Host. Даний заголовок є обов’язковим і повинен бути присутнім в кожному запиті. Варто зазначити, що перевизначення доменного імені відбувається на стороні

клієнта, і відповідно коли ми відкриваємо TCP-з'єднання, то завантажений сервер не має ніякої інформації про цей, який саме адреса використовується для з'єднання.

Приклад HTTP запиту

Розглядаючи HTTP запит можна виокремити такі його частини:

- Метод: Дану складову запиту можна інтерпретувати як послідовність символів, окрім розділювачів і службових знаків, і визначає операцію, яку необхідно виконати. Розглядаючи специфікацію HTTP 1.1 можна зазначити, що існує необмежена кількість методів, які можна використати, проте переважно використовуються стандартні методи такі як:

1. GET – отримання даних
2. POST – надсилання даних
3. PUT – вставка, обновлення даних 4. DELETE – видалення даних

- URI (Uniform Resource Identifier) - шлях до конкретного ресурсу, над яким необхідно здійснити певну операцію, наприклад використання метода GET. Деякі запити можуть не відноситися до конкретного ресурсу, в такому випадку на місці URL, може знаходитися службовий символ “*”. Прикладом цьому може послугувати запит, який безпосередньо адресується серверу, а не якомусь конкретному ресурсу
- Заголовки – це набір пар, ім’я та значення. В заголовках передається різноманітна інформація про службу, яка може містити тип кодування повідомлень, назву, а також версію браузера.
- Тіло повідомлення – власне, самі дані, які ми хочемо передати. Дані можуть варіюватися від html – сторінки, яку повертає сервер, до фотографій, які користувач, власне загружає в зворотному напрямку.

Відповідь сервера має наступну структуру:

```
HTTP/1.1 302 Moved Temporarily
Server: nginx
Date: Sat, 08 Mar 2014 22:29:53 GMT
Content-Type: text/html
Content-Length: 154
Connection: keep-alive
Keep-Alive: timeout=25
```

Код статусу – три цифри котрі визначають статус і результат поверненого запиту. Наприклад, коли ми використали запит GET і сервер успішно опрацював наш запит і повернув очікувану інформацію, код статусу в такому випадку – 200. В разі, коли сервер повідомляє, що такого ресурсу не існує – це код 404. В разі, якщо у клієнта недостатньо привілей для отримання того чи іншого ресурсу, код статусу – 403. Специфікація HTTP 1.1 визначає 40 різних кодів HTTP, а також допускається розширення протоколу і використання додаткових кодів станів.

Пояснення до коду стану (Reason Phrase) – пояснення до коду відповіді призначене для спрощеного розуміння людиною можливістю краще зрозуміти природу помилки. Дане пояснення може бути стандартним, або визначеним самими розробниками при створенні ресурсу.

Тіло відповіді: Для визначення закінчення використовується значення заголовка Content-Length (в даному випадку відповідь містить 7 виїмкових байтів: слово «Wisdom» і символ розриву рядків).

Сам по собі протокол HTTP не передбачає використання шифрування для передачі інформації. Проте для даного протоколу є досить популярним розширення, яке реалізує передачу інформації, за допомогою криптографічного протоколу SSL або TLS.

Дане розширення - HTTPS (HyperText Transfer Protocol Secure). Для конкретних з’єднань здебільшого прийнято використовувати TCP-порт 443. HTTPS передбачає захист даних від перехоплення, а також гарантує захист від

можливих атак типу - man-in-the-middle, в ситуації, коли сертифікат верифікується на стороні клієнта, і при цьому приватний ключ сертифіката, не є скомпрометованим, користувач не використання непідписаного сертифіката, і на комп'ютері користувача не були впроваджені сертифікати центру сертифікації зловмисника. [13]

Графічна частина системи виконується за допомогою HTML.

2.2.3 Мова для розмітки гіпертексту (HTML)

Мова для розмітки гіпертексту (HTML) – мова розмітки призначена для розробки веб-сайтів та різноманітних додатків. HTML є нащадком SGML, яка в свою чергу була надто складна для пересічних людей. Виокремивши також каскадні таблиці стилів(CSS) та мову програмування JavaScript, можна зазначити, що дане тріо формує фундамент всесвітньої павутини.

Одними з вагомих переваг HTML можна відмітити простоту, яка була досягнута за рахунок використання структурних елементів, так званих дескрипторів, або ж тегів. Також це можливість форматування документа без посилання на засоби відображення. Веб-браузери отримують HTML – документи з веб – сервера, або ж локального сховища вашого комп’ютера. HTML описує структуру веб-сторінки семантично та спочатку містить підказки для зовнішнього вигляду документа.

Елементи HTML - це складові HTML-сторінок. За допомогою HTML-конструкцій, зображення та інші об'єкти, такі як інтерактивні форми, можуть вбудовуватися у візуалізовану сторінку. HTML забезпечує засіб для створення структурованих документів, позначаючи структурну семантику для тексту, таких як заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML розмежовані тегами, написаними за допомогою кутових

дужок. Такі теги, як <img /> та <input /> безпосередньо вводять вміст на сторінку. Інші теги, такі як <p> оточують і надають інформацію про текст документа, і можуть включати інші теги як під-елементи. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки. [14]

Для розгортання панелі управління необхідно застосовувати web-сервери.

2.2.4 Види серверів

В залежності від складності і затратності того, які сервер виконує функції, верифікації клієнтів, або баз даних товарів, деякі реалізації використовують один сервер для кількох цілей. Мережа, орієнтована на середню компанію, може використовувати декілька типів серверів, наприклад:

- Веб – сервер: Відповідає за відображення сторінок та запуск програм через веб-браузери. Сервер, котрий взаємодіє з вашим браузером, рендерить текст сторінки, зображення та можливі її інтерактивні компоненти. В даному випадку клієнтською програмою є веб-переглядач, такий як Internet Explorer, Chrome, Firefox, Opera або Safari. До функціоналу веб-сервера можна віднести виконання багатьох задач, таких як завантаження, а також резервне копіювання файлів в інтернеті через хмарні служби зберігання даних, або ж сервіси резервного копіювання, на випадок непередбачуваних ситуацій.
- Сервер електронної пошти: Дані сервери надсилають та отримують повідомлення електронної пошти. Програмне забезпечення відповідає за підключення до таких серверів, як IMAP, або POP для завантаження повідомлення на пристрій користувача. Сервер SMTP, в свою чергу відповідає за надсилання електронних листів.

- FTP-сервер: Завдання даного сервера полягає в переміщенні файлів, використовуючи інструменти протоколу передачі файлів. Доступ до FTP – сервера можна здійснити віддалено за допомогою клієнтських програм, які мають спільний доступ до файлів, розташованих на сервері, або ж за використанням спеціального програмного забезпечення сервера FTP.
- Identity сервер: Реалізує верифікацію користувачів, перевірку їхніх даних, а також відповідає за рівні безпеки відповідні до повноважень користувачів.

На сьогоднішній день існують сотні спеціалізованих серверів, котрі підтримують комп'ютерні мережі. Окрім корпоративних випадків, пересічні користувачі досить часто взаємодіють з ігровими серверами, серверами, які реалізують онлайн чати, а також різноманітними стрімінговими серверами. Деякі сервери зосередженні у вузькоспеціалізованих напрямках, прикладом цьому можуть послужити DNS, або proxy сервери.

В переважній більшості у всесвітній павутині розповсюджена мережева модель клієнт – сервер, котра інтегрує веб-сайти та комунікаційні послуги. Існує альтернативна модель, котра називається одноранговою мережею, і принцип її дії полягає в можливості всім пристроям функціонувати як сервер, або клієнт за потребою. Peer networks здатні реалізувати більшу конфіденційність, оскільки з'єднання між комп'ютерами є більш вузько націленими. Однак внаслідок, обмеження пропускної здатності, домінуюча більшість однорангових систем недостатньо надійні, щоб витримувати стрибки трафіку.

Кластер як термін використовується для реалізації спільних обчислювальних ресурсів. Здебільшого кластер відповідає за інтеграцію та взаємодію, двох, або більше спільних, обчислювальних ресурсів. Зазвичай кластер інтегрує ресурси двох, або більше обчислювальних пристроїв, які при необхідності можуть функціонувати окремо для реалізації якоїсь конкретної мети(часто робочої станції, або серверного пристрою).

Ферма веб-серверів – це сукупність мережевих веб – серверів, котрі є взаємо пов’язані, та мають доступ до певного веб – сайту. Варто зазначити, що дані сервери функціонують як кластери концептуально. Деякі спеціалісти вважають, що класифікація даних ферм залежить від конфігурацій апаратних та програмних засобів. [15]

Для розробки soft- частини застосовували мову програмування Python.

2.3 Розробка soft- частини системи із застосуванням мови програмування Python

Становлення мови програмування python розпочалося в кінці 1980-років. Дана мова була задумана в кінці 1980-х і її реалізація була розпочата в грудні 1989 року, нідерландським дослідником Гвідо ван Россумом. Python є нащадком мови програмування ABC, з можливістю обробки виключень і помилок, а також взаємодія з операційною системою Amoeba. Напрямок розвитку Python, власне можна відслідкувати з його назви, яка була придумана громадою розробників. Дана мова, на сьогоднішній день широко використовується в багатьох галузях і являється мовою програмування високого рівня. Його дизайн та філософію підкреслює читабельність коду, а синтаксис дає змогу лаконічніше описувати поставлені задачі, чим це було б можливо використовуючи C++, або Java. Мова надає змогу реалізовувати конструкції для побудови чітких і зрозумілих прогам, як малого так і великого маштабів.

Python реалізує декілька парадигм програмування, включаючи об'єктно-орієнтовану, імперативну та функціональну. Він має динамічну систему типів та прибиральника сміття. Однією із значних переваг Python є велика кількість відкритих і доступних бібліотек, практично в усіх галузях, котрі постійно доповнюються і розвиваються, за рахунок величезної спільноти.

Інтерпретатори Python доступні для установки на багатьох операційних системах, що дозволяє використовувати написані на ньому програми, в широкому спектрі систем. В Python можна використовувати і інші парадигми, таких як design by contract та логічне програмування, за допомогою зовнішніх розширень. Python використовує динамічне введення тексту та циклічний пошук прибиральника сміття для управління пам'яттю. Також однією з особливостей Python є пізнє зв'язування, котре пов'язує імена методів та змінних під час виконання програми. Дизайн Python надає змогу розробляти програми прогами в функціональному стилі, дотримуючись традицій Lisp. [16]

Мова має вбудовані функції фільтрації, роботи з кортежами, списками, генераторами випадкових значень. Стандартна бібліотека має два модулі itertools та functools. Також Python має значні переваги перед різними мовами програмування:

- чистий синтаксис, дозволяє розбивати програму на окремі блоки та модулі;
- довільний стиль написання прогами (що характерно для більшості інтерпретованих мов);
- принцип роздільного створення модулів передбачає змогу використання тільки необхідних елементів і мінімальну кількість написаного коду;
- використання Python в інтерактивному режимі (дуже корисно для експериментів та вирішення простих проблем);
- наявність великої кількості бібліотек для візуалізації гафічного інтерфейсу і даних;
- підходить для розв'язання математичних задач.

Однак у Python все ж є деякі недоліки. Python, як і в багатьох інших інтерпретованих мовах програмування, де не застосовуються, JIT-компілятори мають загальний недолік - відносно низьку швидкість

виконування програми. Крім того, відсутність статичного набору тексту та деякі інші фактори, на жаль, не дозволяють, під час компіляції, реалізувати механізм перевантаження функцій.

Окрім того, що Python це добре продумана і збалансована мова програмування, його активно використовують для реалізації в найрізноманітніших областях. Це може бути створення сценаріїв різних компонентів та реалізації автономних програм. Як мова загального призначення, Python розвивається в багатьох сферах від розробки веб-сайтів та ігор, до робототехніки та управління космічними кораблями. Програми Python можуть шукати файли та дерева каталогів, запускати інші програми, робити паралельну обробку процесів та ниток. Стандартна бібліотека Python оснащена прив'язками POSIX, розширення імен файлів, утиліти zip-файлів, аналізатори XML та JSON, обробники файлів CSV та інше. Крім того, основна частина системних інтерфейсів Python адаптовані для інтеграції; наприклад, сценарій, який зазвичай копіює дерева каталогів працює без змін на всіх основних платформах Python

2.4 Розробка додатків з використанням фреймворку Flask

Flask це ліцензований BSD мікрофреймворк на основі Werkzeug та Jinja2. Особливість мікрофреймворків, полягає в тому, що вони намагаються надати розробнику тільки необхідні компоненти для реалізації поставленої задачі. Мікрофрейморки можуть бути спеціально розроблені для створення APIs для певного сервісу, або сайту. Flask доволі простий, але одночасно і дуже гнучкий, що дає можливість розробникам використовувати тільки необхідні конфігурації, що полегшує розробку програм, або плагінів.

Два головні компоненти Flask це Werkzeug і Jinja2. Попри те, що Werkzeug несе відповідальність за надання маршрутизації, налагодження та інтерфейс шлюзу веб-сервера (WSGI), двигуном шаблону являється Jinja2.

Сам по собі, Flask не підтримує доступ до бази даних, автентифікацію користувачів чи будь-яку іншу утиліту високого рівня, але він реалізує підтримку великої кількості розширень, котрі реалізують вище перерахований функціонал. Простий додаток можна реалізувати навіть в одному файлі, проте при реалізації великого застосунку, краще розподілити програму на модулі. Модульна структура також один з переваг Flask. Сама ідея даного фреймворку полягає в реалізації надійної основи додатку, а функціонал верхнього рівня втілюють розширення. [17]

Flask, як і всі інші бібліотеки Python, можна встановити, використовуючи індекси пакетів Python (PPI), і його дуже просто налаштувати і почати розробляти.

Даний код імпортує бібліотеку Flask, ініціює додаток, створивши екземпляр класу Flask, оголошує маршрут, а потім визначає функцію для виконання при виклику маршруту.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, From Flask!'

if __name__ == '__main__':
      app.run()
```

Цього коду достатньо для запуску першої програми Flask. Цей код запускає дуже простий вбудований сервер, котрий чудово підходить для тестування, але недостатньо потужний для введення додатку в експлуатацію. Flask не здійснює підтримку доступу до бази даних, і для здійснення взаємодії з БД, здебільшого використовують розширення Flask під назвою Flask-SQLAlchemy, що надає підтримку бібліотеки SQLAlchemy. По суті, SQLAlchemy - це набір інструментів Python SQL та Object Relational Mapper, що забезпечує розробникам повну потужність і гнучкість SQL.

SQLAlchemy здійснює повну підтримку парадигм дизайну на рівні підприємства та розроблена для високоефективного доступу до бази даних, зберігаючи ефективність та простоту використання. Хорошим тоном при розробці застосунку вважається, реалізація модуля аутентифікації користувача, CRUD (створення, читання, оновлення та видалення даних), API REST для створення, пошуку, маніпуляцій та видалення об'єктів. Також Flask надає змогу інтеграції утиліти Swagger для створення документації API, написання тестів та їхньої інтеграції. Для вузьконаправленого тестування функцій, прийнято використовувати pytest, який є повнофункціональним інструментом для тестування Python – застосунків. Pytest дозволяє легко розробляти тести, і все ж ця бібліотека достатньо масштабована для підтримки складних випадків використання. Postman, являється повноцінною платформою API REST, і надає інструменти інтеграції для кожного етапу життєвого циклу API, що робить розробку API простішою та надійнішою.

2.5 Застосування графічної бібліотеки Folium для розробки додатків

Folium - це потужна бібліотека Python, яка реалізовує декілька типів карт Leaflet. Той факт, що результати Folium є інтерактивними, робить цю бібліотеку дуже ефективною при побудові інформаційної панелі. Вона використовує механізм шаблонів Jinja2 Python для візуалізації кінцевих результатів, а Pandas – відповідає за прив'язку статистичних даних CSV. Реалізація починається з імпорту, а потім відбувається визначення даних джерела. Folium реалізує міст між можливостями обробки даних Python та можливостями візуалізації інтерфейсу, які пропонує JavaScript. Зокрема, це дозволяє розробникам Python інтегрувати дані GeoJSON і TopoJSON з бібліотекою Leaflet, однією з найбагатших бібліотек, котру використовують на фронтенді, для створення інтерактивних карт. Перевага використання такої бібліотеки, як Folium, полягає в тому, що вона

безперебійно обробляє переклад між структурами даних Python та компонентами JavaScript, HTML та CSS. [18]

До мінусів цієї бібліотеки можна віднести проблеми з відображенням карт, у випадку комбінації маркерів та спливаючих вікон, візуалізуючи велику кількість елементів. При рендерину карти Folium, необхідно створити об'єкт самої карти, встановивши поряд координати центру карти, рівень масштабування карти, базової плитки для нашого фону.

Розглядаючи бібліотеку Folium, варто зазначити декілька речей:

- Карта створена за допомогою даної бібліотеки визначаються як folium. Map object. Ми можемо додати інші об'єкти, поверх першого, таким чином реалізуючи більш детальна відображення і можливість додавати нові об'єкти.
- Бібліотека дозволяє власноруч створювати, або вибирати шаблони карт, наприклад, з OpenStreetMap, MapBox.
- Folium надає змогу вибирати різні проекції на карті. Доволі часто використовують сферичну проекцію Меркатора, особливо при візуалізації площі, порівняно невеликих розмірів.

Розглядаючи атрибути бібліотеки варто відмітити такий параметр як location, котрий задає точку фокусу на карті. Атрибут zoom_start дозволяє змінювати маштаб карти. Параметр control_scale, вимикає масштаб карти при певному, заданому рівні збільшення. Це те, що іноді може бути корисним для користувача, щоб отримати уявлення про масштаби географічної області, котру він переглядає.

2.6 Міні -комп'ютер Raspberry PI

Основним елементом запропонованої віртуальної приватної мережі (VPN) є міні- комп'ютер Raspberry PI. Незважаючи на незначні розміри, плата має високу продуктивність, що дозволяє їй вийти на один рівень із стаціонарними

ПК. Спочатку Raspberry PI була розроблена, як навчальний посібник по інформатиці. Але сама ідея виявилася настільки вдалою, що за декілька років міні-комп'ютер став популярний в дуже широких кругах. З часом Raspberry PI пережила декілька модифікацій, кожна з яких відрізнялася від попередника яким-небудь параметром. Такий підхід дозволив регулювати вартість виробу залежно від потреб користувача, що також позитивно позначилося на популярності пристрою. Уся лінійка Raspberry PI застосовує процесори з Арм-архітектурою. Існує декілька модифікацій Raspberry PI. Останні версії оснащені безпровідними WiFi і Bluetoot модулями, що розширюють межі застосування мини-пк в області Ethernet -технологій. Основною відмінною рисою Raspberry PI від звичайних комп'ютерів, являється наявність програмованих портів введення-виведення GPIO. За допомогою їх можна управляти різними пристроями і приймати телеметрію з різного роду датчиків. [19]

2.7 Операційні системи для Raspberry PI

Raspbian - ця операційна система в 2015 році була представлена як основна для Raspberry PI. Вона по максимуму оптимізована для процесорів з Арм-архітектурою і досить активно продовжує розвиватися. Основою операційної системи є Debian GNU/Linux. Середовище робочого столу складається з LXDE (середовище для UNIX і інших POSIX –сумісних систем типу Linux і BSD), а також менеджера вікон Openbox (безкоштовний менеджер для X Window System). До складу дистрибутива входять: програма комп'ютерної алгебри Mathematica; модифікована версія Minecraft PI; урізана версія Chrome.

Debian - операційна система з відкритим початковим кодом. До складу Debian входить більше 59000 пакетів вже скомпільованого ПО. Система використовує ядро Linux або FreeBSD. У стандартний дистрибутив включені: середовище

робочого столу GNOME з набором найбільш популярних програм, таких як Firefox, LibreOffice, Evolution, і інший набір для роботи з мультимедіа. Також є можливість установки образів з використовуваними середовищами робочих столів KDE, Xfce, LXDE, MATE і Cinnamon.

Ubuntu - система заснована на Debian GNU/Linux. За популярністю Ubuntu займає перше місце серед дистрибутивів Linux, призначених для web - серверов. До складу дистрибутива входять: програма для перегляду Інтернет; офісний пакет, програми для комунікації і так далі

Fedora - ця операційна система заснована на дистрибутиві Linux від відомої фірми Red Hat. До складу дистрибутива входять LibreOffice, Mozilla Firefox, а також інше ПО, яке можна додатково встановити через Цент Додатків GNOME.

Arch Linux - це вільно поширюваний дистрибутив GNU/Linux загального призначення. Особливістю цієї системи є відсутність графічного установника, що може неабияк потренувати навички затятих дослідників Linux. Gentoo Linux - один з популярних дистрибутивів GNU/Linux з гнучкою технологією управління пакетами. У системі передбачена можливість максимальної оптимізації під конкретне апаратне рішення. Алгоритм управління пакетами дає можливість легко реалізувати як робочу станцію, так і сервер.

RISC OS - операційна система спеціально розроблялася для процесорів з архітектурою ARM. Особливості ядра RISC OS дозволяють системі робити прискорений запуск за рахунок зберігання даних в ПЗП. Такий підхід також допомагає захистити дані різного роду збоях і впливи шкідливого ПО.

OpenELEC, OSMC - це програмні и для організації домашнього кінотеатру під управлінням GNU/Linux. [20]

2.8 Raspberry PI 3 Model B

Міні-комп’ютер складається з процесора, оперативної пам'яті роз'єма HDMI, композитного виходу, USB, Ethernet, Wi-Fi та Bluetooth (рис.2.6).
Головна перевага Raspberry Pi - 40 контактів введення/виведення загального призначення (GPIO). До них ви зможете підключати периферію для взаємодії із зовнішнім світом: виконавчі пристрої, будь-які сенсори та все, що працює від електрики. Штатною операційною системою для Raspberry Pi є Linux. Вона встановлюється на microSD карту, а та – у спеціальний слот на платі.
Raspberry Pi 3 Model B є прямим спадкоємцем Raspberry Pi 2 Model B. Плата повністю сумісна з попередником, але має більшу продуктивність і нові засоби комунікації:

- 64-бітним чотириядерним процесором ARM Cortex-A53 з тактовою частотою 1,2 ГГц на однокристальному чіпі Broadcom BCM2837;
- вбудованими Wi-Fi 802.11n та Bluetooth 4.1.

Крім того, процесор має архітектуру ARMv53, а значить, ви зможете використовувати улюблену операційну систему: Debian Wheezy, Ubuntu Mate, Fedora Remix і навіть MS Windows 10 IoT. На Raspberry Pi 3 встановлений 64-бітний чотириядерний процесор ARM Cortex-A53 з тактовою частотою 1,2 ГГц на ядро в складі однокристальної платформи Broadcom BCM2837. Цей чіп забезпечує приріст продуктивності на 50-60% порівняно з Raspberry Pi 2 і майже десятикратну перевагу перед першим Raspberry Pi. Завдяки цьому комп'ютер відкриває ще більше можливостей для «інтернету речей» та проектів, що вбудовуються. Raspberry Pi 3 Model B має 1 ГБ оперативної пам'яті, але це пам'ять ділиться з графічної підсистемою. Графічний двоядерний процесор VideoCore IV® підтримує стандарти OpenGL ES 2.0, OpenVG, MPEG-2, VC-1 і здатний кодувати, декодувати та виводити Full HD-відео (1080p, 60 FPS, H.264 High-Profile).

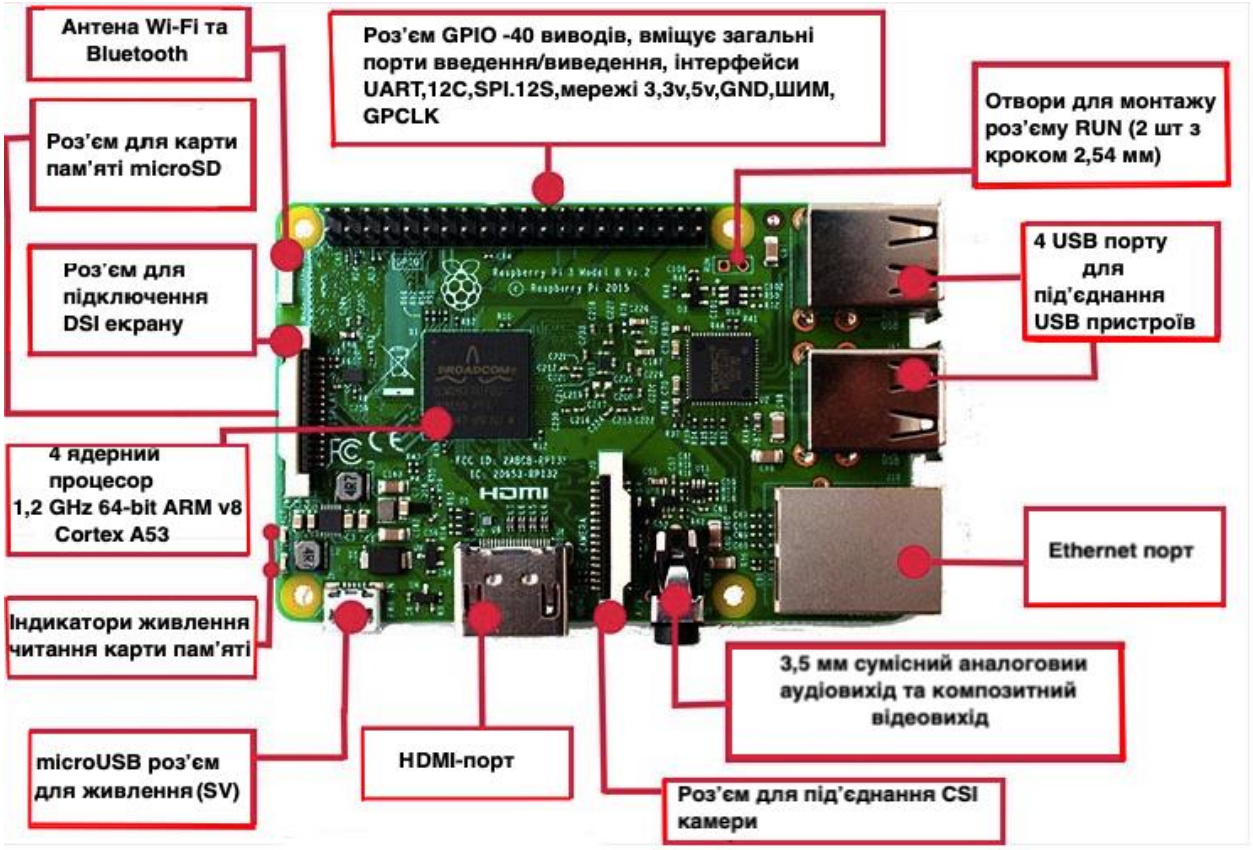

Рисунок 2.6- Конструкція Raspberry Pi 3 Model B

Для підключення монітора або телевізора використовуйте композитний відеовихід або роз'єм HDMI. Роздільна здатність варіюється від 640×350 (EGA) до 1920×1200 (WUXGA) для HDMI. Композитний вихід працює у форматах PAL та NTSC. Колонки чи навушники підключаються через стандартне гніздо 3,5 мм. Також звук може передаватися HDMI. Raspberry Pi 3 Model B надає 4 USB-порти, об'єднаних внутрішнім хабом. До них, крім іншого, можна підключити клавіатуру та мишу. Для економії ресурсів центрального процесора Raspberry Pi пропонує підключення штатних модулів через 15-пінові слоти:

- CSI-2 - для підключення камери за інтерфейсом MIPI
- DSI – для підключення штатного дисплея.

Як низькорівневі інтерфейси доступні:

- 40 портів введення-виведення загального призначення;
- UART (Serial);
- I²C/TWI;
- SPI із селектором між двома пристроями;
- піни живлення: 3,3 В, 5 В та земля.

Для комунікації на Raspberry Pi 3 Model B доступні інтерфейси:

- Ethernet на 10/100 Мбіт із виходом на стандартне гніздо 8P8C (RJ45);
- Wi-Fi 802.11n та Bluetooth 4.1, що забезпечуються мікросхемою Broadcom BCM43438.

Живлення Raspberry Pi 3 здійснюється від 5-вольтового адаптера через роз'єм micro-USB або піни живлення. Рекомендуємо використовувати джерело живлення з силою струму не менше 2 А, щоб мати можливість підключати до USB-портів більш енергоємні пристрої. Апаратний вимикач живлення на платі відсутній. Для увімкнення комп'ютера достатньо підключити кабель живлення.

Замість традиційного для звичайних комп'ютерів жорсткого диска Raspberry Pi використовує microSD флеш-карту. Вона має бути попередньо підготовлена - на неї слід встановити операційну систему. Підтримуються картки розміром від 4 ГБ. Об'єм, що рекомендується, — не менше 8 ГБ.

Таким чином, основні характеристики [21]:

- Процесор: 64-бітний чотириядерний ARM Cortex-A53 з тактовою частотою 1,2 ГГц на однокристальному чіпі Broadcom BCM2837;
- оперативна пам'ять: 1ГБ LPDDR2 SDRAM;
- цифровий відеовихід: HDMI;
- композитний вихід: 3,5 мм (4 pin);
- USB порти: USB 2.0 4;

- мережа: WiFi 802.11n, 10/100 Мб RJ45 Ethernet;
- Bluetooth: Bluetooth 4.1, Bluetooth Low Energy;
- роз'єм дисплея: Display Serial Interface (DSI);
- роз'єм відеокамери: MIPI Camera Serial Interface (CSI-2);
- картка пам'яті: MicroSD;
- порти введення-виведення: 40;
- габарити: 85x56x17 мм.

2.9 Аналіз схемотехнічних рішень плати Raspberry PI

2.9.1 Структура портів GPIO

Для підключення до зовнішніх периферійних елементів використовується порт обміну GPIO (рис. 2.8), що має порти з функцією SDA/SCL – обмін з пристроями за протоколом I2C.Також наявні виводи з функціями MOSI – Master Out Slave In, MISO – Master In Slave Out, SCLK – Serial Clock. Ці виводи забезпечують обмін з пристроями, що включено ланцюгом – один за одним. Порт також забезпечує живлення +3,3 В та +5 В. Проте сила струму обмежена можливостями центрального чипу.

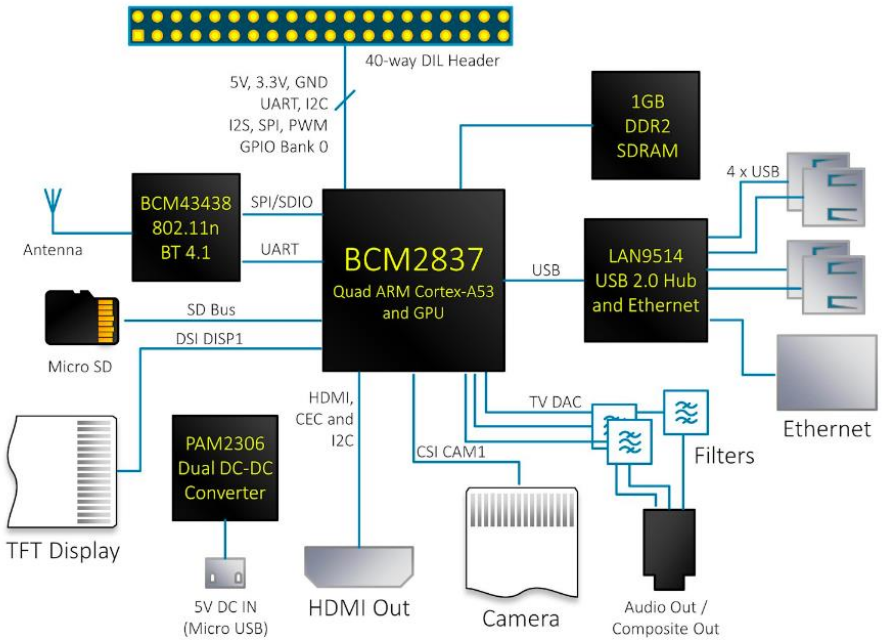

Рисунок 2.7-Структура Raspberry PI 3

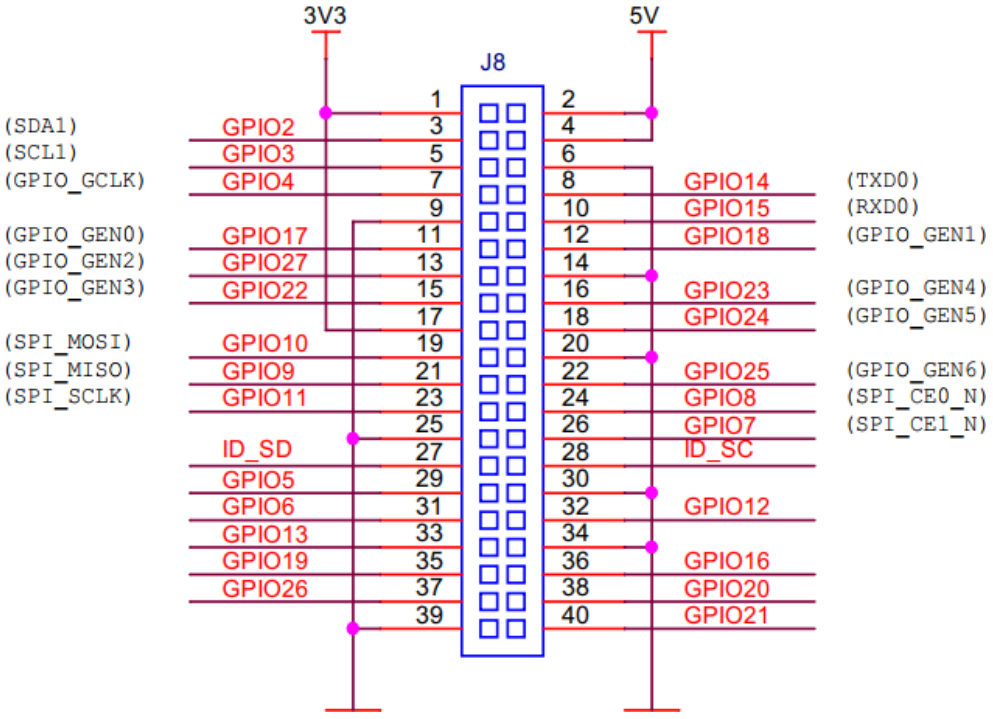

Рисунок 2.8- Схема порту обміну GPIO

2.9.2 Порт USB 3.0

Невід'ємною частиною сучасного пристрою є порт USB (рис.2.9). В платі сімейства Raspberry PI спроектовано порт USB 3.0. Це високошвидкісний порт. Конструктивно, порт реалізовано в форматі TYPE-C, малогабаритний порт.

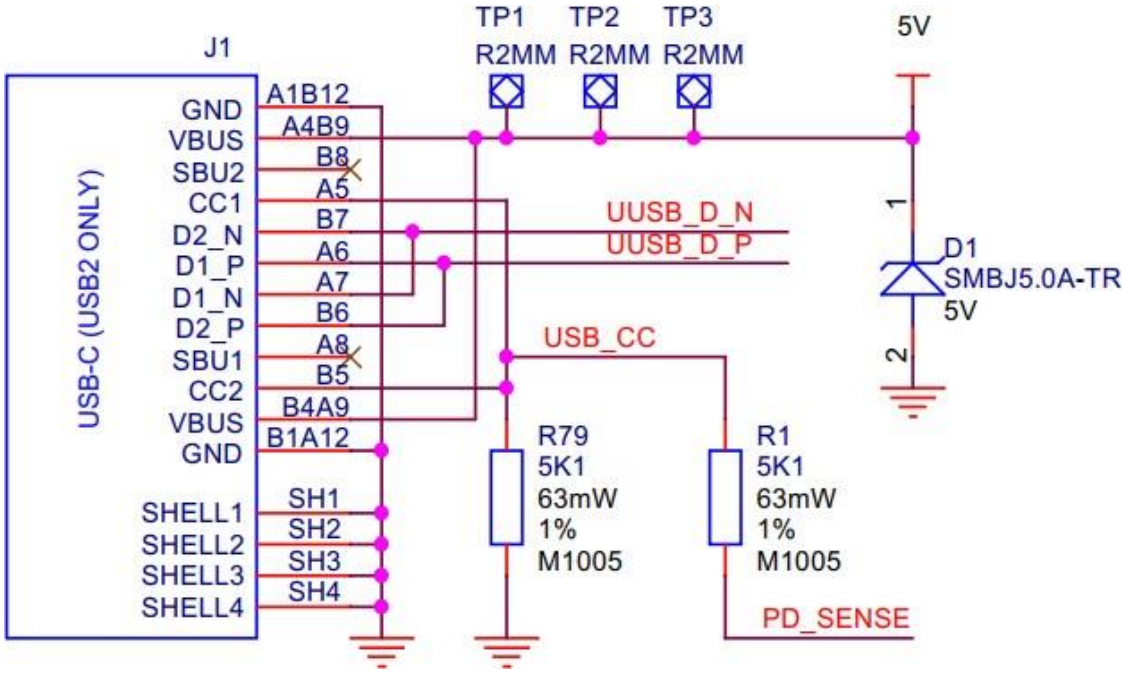

Рисунок 2.9 – Порт USB та живлення

2.9.3 Периферійне обладнання

Плата Raspberry PI дозволяє розширити можливості мікросистеми шляхом підключення такого звичного елементу як дисплей та камера. Використовуються стандартні цифрові інтерфейси підключення, тому плата надає типову схемотехніку.

Паралельний RGB-інтерфейс до 24 біт доступний на всіх платах Raspberry PI

із 40-смуговим заголовком та обчислювальними модулями. Цей інтерфейс дозволяє приєднувати паралельні RGB-дисплеї до Raspberry PI GPIO у форматі RGB24 (8 біт для червоного, зеленого та синього) або RGB666 (6 біт для кожного кольору) або RGB565 (5 біт червоного, 6 зелених та 5 синіх) (рис.2.10).

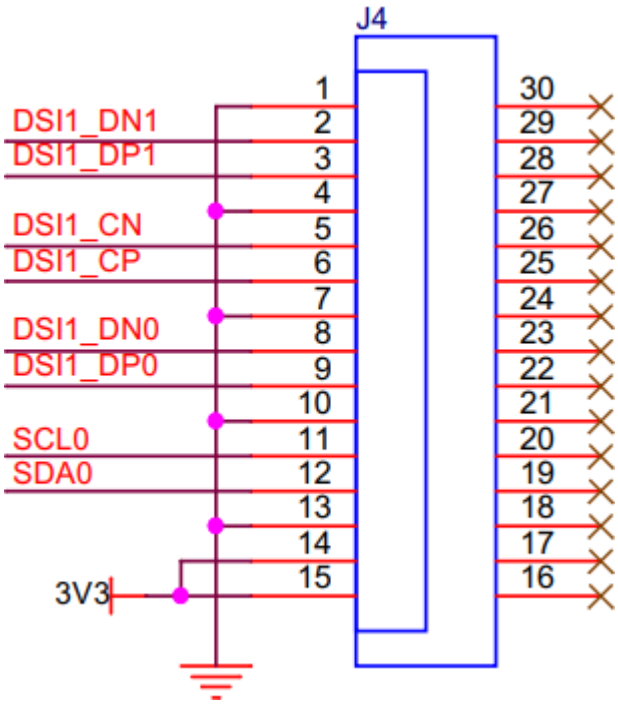

Рисунок 2.10 – Дисплей

Цей інтерфейс контролюється прошивкою графічного процесора і може бути запрограмований користувачем за допомогою спеціальних параметрів config.txt та ввімкнення правильного накладання дерева пристроїв Linux. Однією з альтернативних функцій, яку можна вибрати на банку 0 Raspberry PI GPIO, є DPI (Display Parallel Interface), який є простим тактовим паралельним інтерфейсом (до 8 бітів R, G і B; годинник, увімкнути, hsync та vsync). Цей інтерфейс доступний як альтернативна функція 2 (ALT2) на GPIO- банку 0 (рис.2.11).

Всі SoC, що використовуються в лінійці Raspberry PI, мають два інтерфейси камер, які підтримують джерела CSI-2 D-PHY 1.1 або CCP2 (Compact Camera

Port 2). Цей інтерфейс відомий під кодовою назвою "Unicam". Перший екземпляр Unicam підтримує 2 смуги передачі даних CSI-2, тоді як другий підтримує 4. Кожна смуга може працювати зі швидкістю до 1 Гбіт / с (DDR, тому максимальна частота зв'язку становить 500 МГц).

GPIO	ALT Func2
GPIO0	PCLK
GPIO1	DE
GPIO2	LCD_VSYNC
GPIO3	LCD_HSYNC
GPIO4	DPI_D0
GPIO5	DPI_D1
GPIO6	DPI_D2
GPIO7	DPI_D3
GPIO8	DPI_D4
GPIO9	DPI_D5
GPIO10	DPI_D6
GPIO11	DPI_D7
GPIO12	DPI_D8
GPIO13	DPI_D9
GPIO14	DPI_D10
GPIO15	DPI_D11
GPIO16	DPI_D12
GPIO17	DPI_D13
GPIO18	DPI_D14
GPIO19	DPI_D15
GPIO20	DPI_D16
GPIO21	DPI_D17
GPIO22	DPI_D18
GPIO23	DPI_D19
GPIO24	DPI_D20
GPIO25	DPI_D21
GPIO26	DPI_D22
GPIO27	DPI_D23

Рисунок 2.11– Опис функцій на GPIO порту

Для спілкування з периферією Unicam доступні 3 незалежні програмні інтерфейси (рис.2.12).

Прошивка GPU із закритим вихідним кодом має драйвери для Unicam, три датчики камери та мостовий чіп – це Камера Raspberry PI v1.3 (Omnivision OV5647), Камера Raspberry PI v2.1 (Sony IMX219), Камера Raspberry PI HQ (Sony IMX477) та непідтримуваний драйвер для мостового чипа Toshiba TC358743 HDMI-> CSI2.

Цей драйвер інтегрує вихідний драйвер, Unicam, провайдера та управління тюнером у повний стек камери, забезпечуючи оброблені вихідні зображення. Його можна використовувати через MMAL, OpenMAX IL та V4L2 за допомогою модуля ядра bcm2835-v4l2. Через цей інтерфейс підтримуються

лише камери Raspberry PI.

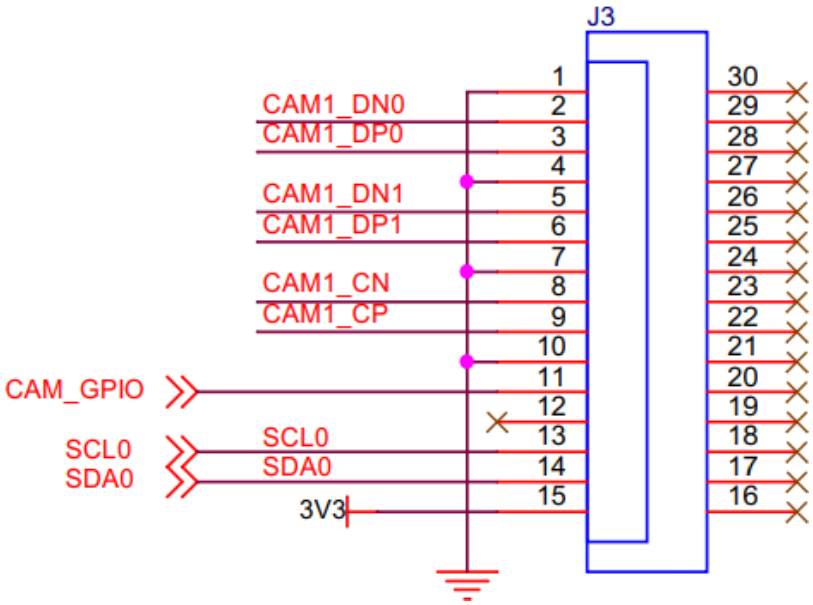

Рисунок 2.12 – Камера плат

Це було тимчасовим варіантом до того, як був доступний драйвер V4L2. Компонент MMAL vc.ril.rawcam дозволяє отримувати необроблені дані CSI2 так само, як і драйвер V4L2, але вся конфігурація джерела повинна виконуватися користувачем через будь-який інтерфейс, який вимагає джерело. Додаток raspiraw доступний на github. Він використовує цей компонент та стандартні набори регістрів I2C для OV5647, IMX219 та ADV7282M для підтримки потокової передачі.

Для блоку Unicam доступний повністю відкритий драйвер ядра; це модуль ядра під назвою bcm2835-unicam. Це інтерфейси до драйверів підпристроїв V4L2 для джерела для доставки необроблених кадрів. Цей драйвер bcm2835-unicam керує датчиком і налаштовує приймач CSI-2 таким чином, що периферійний пристрій запише необроблені кадри (після Debayer) в SDRAM для V4L2 для доставки в додатки. За винятком цієї можливості розпакувати формати CSI-2 Bayer до 16 біт / піксель, немає обробки зображень між

джерелом зображення (наприклад, датчиком камери) та bcm2835-unicam, розміщуючи дані зображення в SDRAM.

Також плата містить HDMI-вихід (рис.2.13).

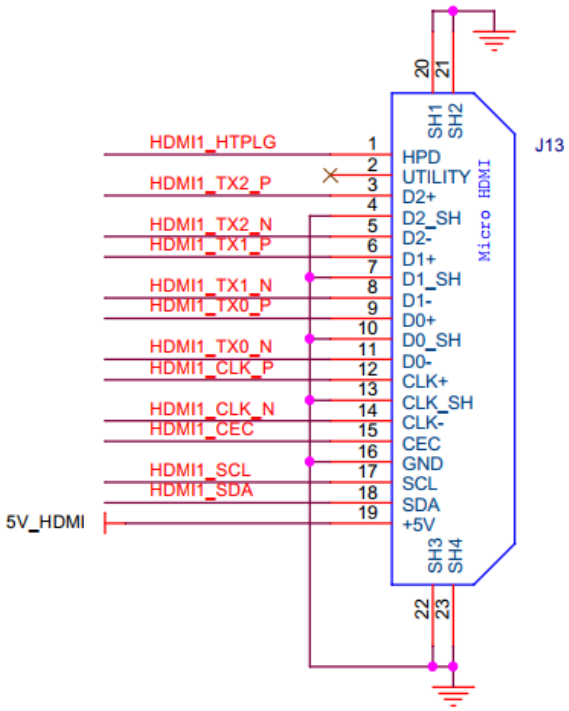

Рисунок 2.13 – HDMI плата

2.9.4 Аудіо канали

Для роботи зі звуком, для його створення, Raspberry PI не використовує цифро-аналоговий перетворювач. Його не має. Проте функцію ЦАП виконує звичайний НЧ-фільтр, через який пропускається цифровий сигнал з широтно-імпульсною модуляцією (рис. 2.14, 2.15,2.16)

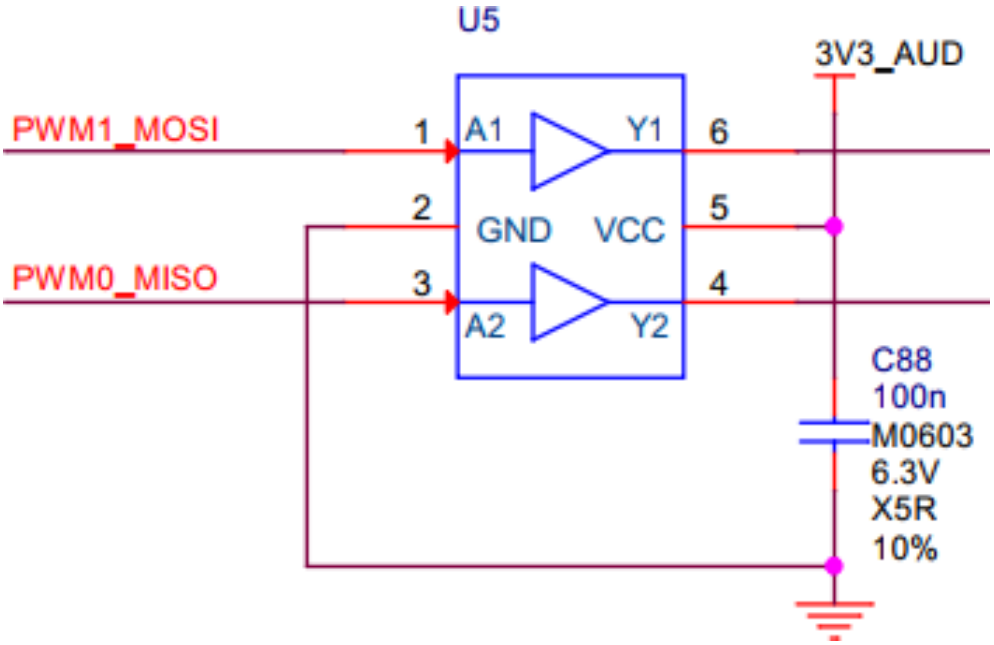

Рисунок 2.14 – Буферний підсилювач

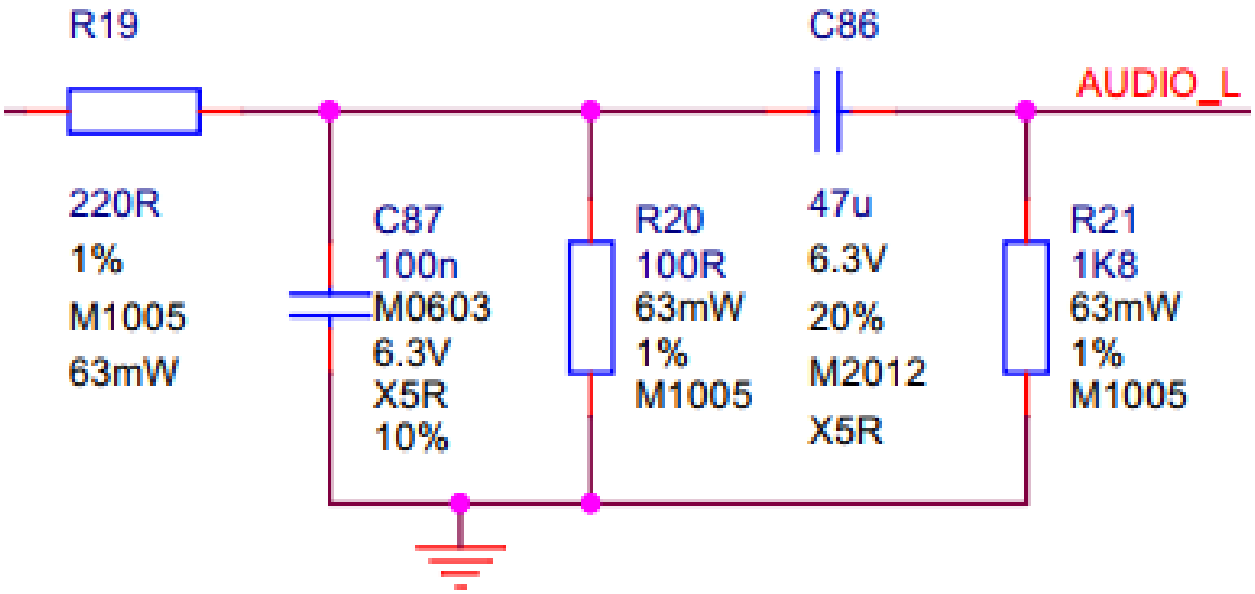

Рисунок 2.15– Фільтр НЧ (Лівий канал)

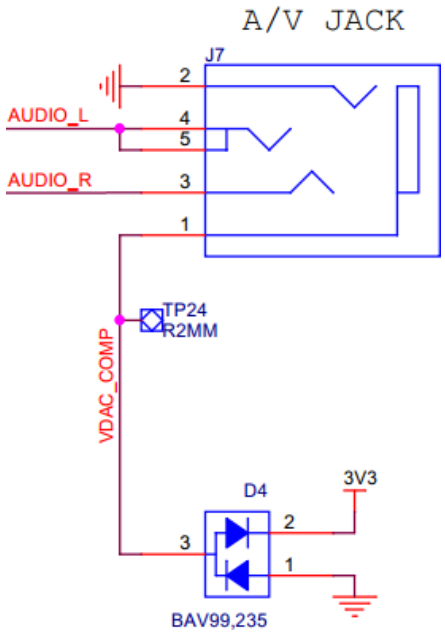

Рисунок 2.16 – Вихідний роз’єм

2.9.5 Джерело живлення 1.0 В

Живлення 1.0 В забезпечує живлення ядра системи. Для побудови джерела живлення використано одночипний перетворювач U3, який містить необхідні елементи для побудови такого джерела живлення (рис.2.17).

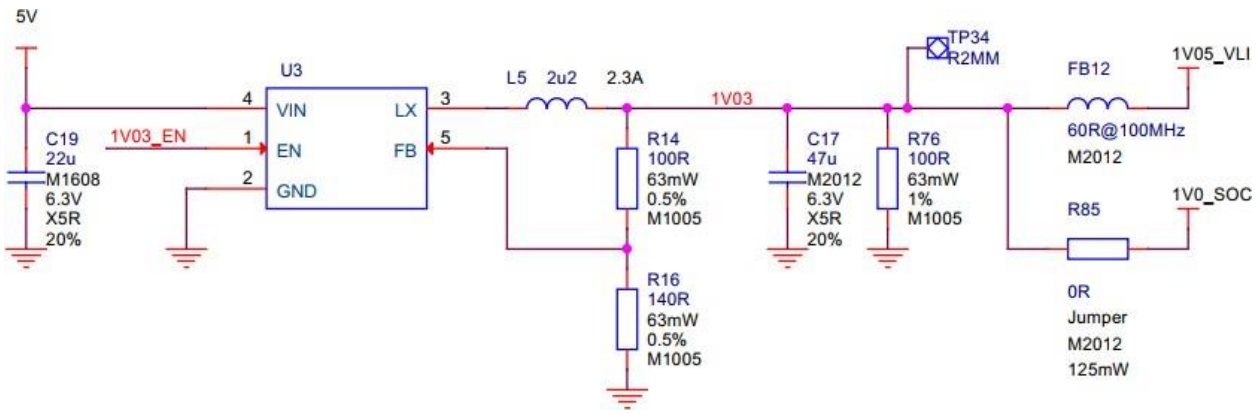

Рисунок 2.17 – Джерело живлення 1.0 В

2.9.6 Ethernet канал

Ethernet це стандарт, який відноситься тільки до побудови локальних мереж LAN (Local Area Network). Локальна мережа мала, на відміну від старшого брата WAN (Wide Area Network), яку ще називають глобальною мережею. Саме локальна мережа - один з основних ідентифікаторів наявності Ethernet (рис.2.18).

У термінах семирівневої моделі OSI, стандарт Ethernet живе на першому і на другому рівнях. На першому рівні описані способи передачі електричних, оптичних і бездротових (радіо, наприклад) сигналів, а на другому формування кадрів (фреймів). Ethernet - це набір описів способів фізичної передачі сигналів (електрика) на першому рівні моделі OSI і формування кадрів (фреймів) на другому рівні моделі OSI всередині локальних мереж LAN.

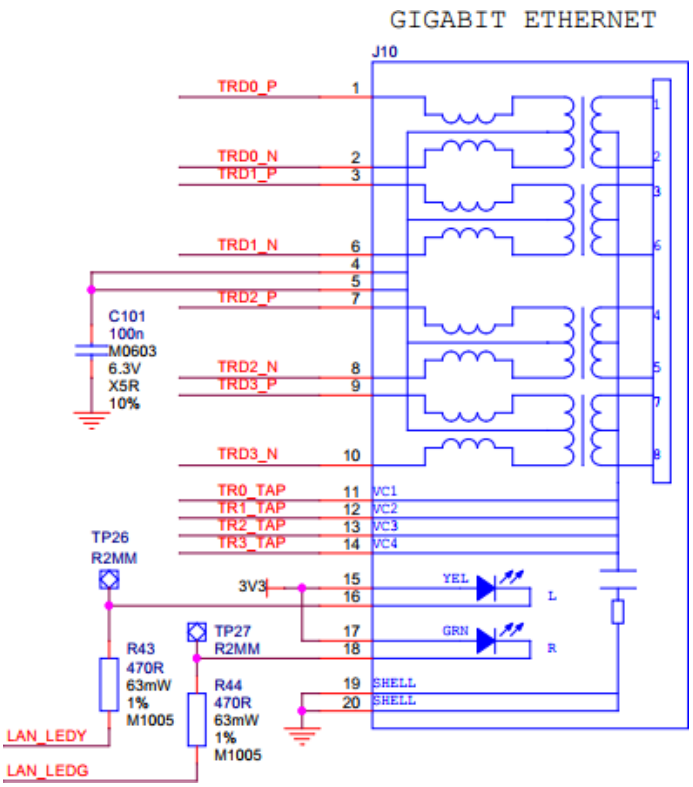

Рисунок 2.18– Ethernet канал Raspberry PI

2.9.7 Система терморегуляції

Усі моделі Raspberry PI працюють у певному температурному режимі, щоб уникнути перегріву під великим навантаженням. SoC мають внутрішній датчик температури, програмне забезпечення на опитуваннях графічного процесора забезпечує перевищення температури заздалегідь визначеною межею – це 85 ° C на всіх моделях. Можна встановити це значення на більш низьке, але не на більш високе. Коли пристрій наближається до межі, різні частоти, а іноді і напруги, що використовуються на мікросхемі (ARM, GPU), зменшуються. Це зменшує кількість виробленого тепла, підтримуючи температуру під контролем.

Коли температура в ядрі знаходиться в діапазоні від 80 ° C до 85 ° C, відображатиметься піктограма попередження, що показує червоний наполовину заповнений термометр, а ядра ARM будуть поступово повертатися назад. Якщо температура досягає 85 ° C, відображатиметься значок із повністю заповненим термометром, а ядра ARM і графічний процесор будуть зупинятись.

Для Raspberry PI 3 Model B технологія друкованої плати була змінена для забезпечення кращого відведення тепла та збільшення теплової маси. Крім того, запроваджено м’яку температурну межу, метою якої є максимізація часу, протягом якого пристрій може «спринтувати», перш ніж досягати жорсткого обмеження при 85 ° C. При досягненні м’якого обмеження тактова частота зменшується з 1,4 ГГц до 1,2 ГГц, робоча напруга також трохи зменшується. За замовчуванням обмежене обмеження становить 60 ° C, і це можна змінити за допомогою налаштування temp_soft_limit у config.txt.

Загальна схема електрична принципова Raspberry Pi 3 знаходиться в Додатку А. Схема розміщення елементів представлена в Додатку В.

2.10 Розробка програмного коду керування Raspberry Pi 3

1.Підготовка Raspberry

• Необхідно знайти та встановити наступний образ raspberry pi Raspberry Pi OS Lite (32-bit)

• Підключитися до raspberry за допомогою імені pi та паролю raspberry

• Зробити root доступ

sudo su

• Активувати ssh

sudo systemctl enable ssh

sudo systemctl start ssh

• Видалити авторизацію за допомогою пароля

sudo nano /etc/ssh/sshd_config

 # change PasswordAuthentication yes to no$ PasswordAuthentication no

 # restart ssh$ sudo service ssh restart

• Додати авторизацію по ключу

install -d -m 700 ~/.ssh

nano ~/.ssh/authorized_keys

 # add your keys$ sudo chmod 644 ~/.ssh/authorized_keys

sudo chown root:root ~/.ssh/authorized_keys

sudo service ssh restart

2. Налаштування Raspberry

• Встановити vim текстовий редактор

sudo apt install vim

• Активувати ipv4 переадресацію

sudo vim /etc/sysctl.conf

• Знайти і розкоментувати рядок

 # net.ipv4.ip_forward=1

• Застосувати зміни

sudo sysctl -p

• Встановити допоміжні бібліотеки

sudo apt-get -y install git fail2ban software-properties-common build-essential libevent-dev libssl-dev supervisor

Для налаштування Raspberry Pi 3 застосуємо бібліотеку 3proxy.

3proxy - вільна проксі-серверна програма, яка надає можливість використовувати проксі-сервери для роботи з різними протоколами, включаючи HTTP, HTTPS, SOCKS4 та SOCKS5. Вона працює як на Windows, так і на Linux і підтримує безліч функцій, таких як блокування певних сайтів та логування активності користувача.

Переваги 3proxy:

• Економія ресурсів: використовує мінімальні ресурси системи, а також оптимізований код, який дозволяє швидко обробляти велику кількість запитів.

• Швидкість: працює швидше, ніж більшість інших проксі-серверів, та забезпечує високу швидкість обробки даних та мінімальні затримки.

• Гнучкість налаштування: може бути налаштований та адаптований під різні системи, мережі та цілі використання. Він має докладну документацію та широкий набір параметрів, що забезпечує гнучкість його налаштування.

• Великий функціонал: підтримує різні протоколи та аутентифікацію, включаючи Socks5, Socks4, HTTP, HTTPS, FTP-проксі, а також бази даних користувачів (MySQL, LDAP та інші).

- Надійність: забезпечує високу надійність та захист даних завдяки використанню шифрування та аутентифікації. Крім того, він оновлюється регулярно, що забезпечує високу безпеку та стабільність його роботи.

Особливістю 3proxy є його невеликий розмір та висока продуктивність. Розмір 3прокси значно менше, ніж у більшості інших проксі-серверів, які надають такий же функціонал. При цьому 3proxy має високу продуктивність і дозволяє обробляти велику кількість запитів за короткий час.

3proxy має гнучке налаштування, що дозволяє адаптувати його під різні системи, мережі та цілі використання. Багатий набір опцій та налаштувань, докладна документація, підтримка різних протоколів та аутентифікації – все це робить 3proxy зручним та потужним інструментом для роботи з проксі-серверами.

Docker — це програмне забезпечення, яке дає можливість на певній ділянці пам'яті ізольовано встановити необхідну ОС (операційну систему), версію Java, налаштувати змінні оточення, встановити різні залежності і дати доступ тільки за певних умов. При цьому це відбувається абсолютно автономно.

Переваги технології Docker — це незалежність платформи. Можливо описати запуск програми, працюючи на Windows, а потім без проблем запустити на MacOS. Це дає можливість дуже швидко переносити і налаштовувати програму на різних серверах.

Встановлюємо Docker на комп'ютер, щоб можна було створювати, налаштовувати і запускати контейнери.

Схема створення контейнера виглядає наступним чином:

- Створюємо 'Dockerfile' — файл, в якому необхідно описати, як буде створюватися образ.
- Image — це образ, на підставі якого в подальшому буде запущений контейнер.

• Container — це запущений образ, в якому працює Ваша програма з описаними залежностями відповідно до інструкції.

Netplan – це нова утиліта мережевих налаштувань за допомогою командного рядка, встановлений починаючи з Ubuntu 17.10 для легкого керування та мережевих налаштувань у системах Ubuntu. Вона дозволяє налаштувати мережевий інтерфейс із використанням абстракції YAML. Він працює спільно з мережевими демонами NetworkManager і systemd-networkd (званими рендерерами, ви можете вибрати, який з них використовувати) як інтерфейси до ядра.

3. Налаштування 3proxy

• Встановлюємо 3proxy з github сховища

git clone https://github.com/z3apa3a/3proxy

• Додаємо налаштування для socks5 з'єднань

cd 3proxy/

sudo vim src/proxy.h

• Додаємо поперед *#define MAXUSERNAME 128 рядок #define ANONYMOUS 1*

• Робимо нову збірку 3proxy

sudo ln -s Makefile.Linux Makefile

sudo make

sudo make installsudo systemctl is-enabled 3proxy.service

• Додаємо маршрутизацію для 3proxy

sudo vim /etc/iproute2/rt_tables

• Додаємо нові рядки

1 gw1 2 gw2

uDEV rules. Scripts for start / reboot modem proxy

• Створюємо файл */usr/local/bin/proxy-udev-add-{modem_id}.sh*

#!/bin/bash

INTERFACE_ID={modem_id}

```
CONFIG_LOCATION=/root/3proxy
# create 3proxy config
FILE_NAME=3proxy_$INTERFACE_ID.cfg
FILE_LOCATION="${CONFIG_LOCATION}/${FILE_NAME}"
tee $FILE_LOCATION <<- EOF >/dev/null
  #! /usr/local/bin/3proxy
  daemon
  nserver 8.8.8.8
  nscache 65536
  timeouts 1 5 30 60 180 1800 15 60
  users TProxy:CL:moNEnTIM
  log /var/log/3proxy_${INTERFACE_ID}.log
  auth strong
  allow TProxy
  proxy -n -a -p5000${INTERFACE_ID} -i192.168.0.100 -
e192.168.${INTERFACE_ID}.100
  flush
EOF
# create ip route settings
ifconfig eth$INTERFACE_ID 192.168.$INTERFACE_ID.100
sleep 2

ip route add 192.168.$INTERFACE_ID.0/24 dev eth$INTERFACE_ID src
192.168.$INTERFACE_ID.100 table gw$INTERFACE_ID
ip route add default via 192.168.$INTERFACE_ID.1 dev eth$INTERFACE_ID
table gw$INTERFACE_ID
ip rule add from 192.168.$INTERFACE_ID.100/32 table gw$INTERFACE_ID
ip rule add to 192.168.$INTERFACE_ID.100/32 table gw$INTERFACE_ID
```

sleep 5

3proxy $FILE_LOCATION

sleep 2

• Створюємо файл */usr/local/bin/proxy-udev-remove-{modem_id}.sh*

#!/bin/sh

kill -9 $(ps aux | grep 3proxy_{modem_id} | grep Ssl | tr -s ' ' | cut -d ' ' -f2 | head -n 1)

• Створюємо udev правило */etc/udev/rules.d/70-huawei_e33372.rules*

SUBSYSTEM=="net", ACTION=="remove", DRIVERS=="?",*

KERNEL=="eth1", RUN+="/usr/local/bin/proxy-udev-remove-1.sh"

SUBSYSTEM=="net", ACTION=="remove", DRIVERS=="?",*

KERNEL=="eth2", RUN+="/usr/local/bin/proxy-udev-remove-2.sh"

• Перезапускаємо службу udev

sudo service udev restart

4. Монітор сценаріїв підключення модему

• Створити файл *monitor_{modem_id}.sh*

#!/bin/bash

while true

do

FIRST_INTERFACE=$(ip a | grep 192.168.{modem_id}.100 | wc -l)

RUN_CONFIG=$(ps aux | grep 3proxy_{modem_id} | grep Ssl | wc -l)

if [$FIRST_INTERFACE == 1] && [$RUN_CONFIG != 1]

then

echo "Run script"

source /usr/local/bin/proxy-udev-add-{modem_id}.sh

fi

sleep 30

done

• Додаємо конфіг до supervisor

[program:monitor_1]

command=/root/monitor_1.sh

autostart=true

autorestart=true

stopasgroup=true

stopsignal=QUIT

stderr_logfile=/var/log/%(program_name)s.log

stdout_logfile=/var/log/%(program_name)s.log

• Перезапускаємо supervisor

supervisorctl reread

supervisorctl update

5. Налаштування API для модемів

• Зробити копію git сховища

git clone git@git.idkfa.club:TT/typical-proxy.git

• Встановити docker

sudo apt install docker.io

docker build -t proxy-api .

• Запустити контейнер

docker run --restart=always -d --name proxy-api --network host proxy-api

2.11 Головний сервер

Для контролю Raspberry Pi 3 використовується головний сервер. Необхідно провести налаштування серверу.

1. Зміна конфігурації netplan

```
eth0:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
eth1:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
eth2:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
eth3:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
eth4:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
eth5:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
```

```
eth6:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
eth7:
  dhcp4: true
  dhcp4-overrides:
    route-metric: 100
```

2. Створення файлу /usr/local/bin/setup.sh

```
#!/bin/bash
#
#

set -e
trap "exit 255" ERR
#

declare KABOOM=255
declare ONE=1
declare ZERO=0

declare NAME=`basename $0`
declare ACTION=$1
declare DEVICE=$2

declare INDEX=0
```

```
declare INTERFACE=${DEVICE##*/}
declare BASE=${INTERFACE//[!0-9]/}
declare INTERFACE_ID=$((BASE+1))
#
Kernel() {

        echo "${1}" > /dev/kmsg
}

#
Echo() {

        logger "${1}"
        Kernel "${1}"

        echo "${1}" >> /var/log/udev.log;
}

#
Echo "[${INTERFACE}] ----------------------------"
Echo "[${INTERFACE}] ${NAME}"
Echo "[${INTERFACE}] Action '${ACTION}'"
Echo "[${INTERFACE}] Base '${BASE}'"
Echo "[${INTERFACE}] Device '${DEVICE}'"

#
Add()
{
```

```
# ----------------------------------------
Echo "[${INTERFACE}] Add ${INTERFACE}..."

# ----------------------------------------
Echo "[${INTERFACE}] Wait interface is up..."
declare INTERFACE_FOUND=false

# ----------------------------------------
while [ 1 ]; do
    if [ $(ip addr show | grep ${INTERFACE} | grep inet | wc -l) -ne 0 ];
then
        INTERFACE_FOUND=true
        break
    fi

    Echo "[${INTERFACE}] ...sleep 10 seconds"
    sleep 10

    ((INDEX=INDEX+1))

    if [ $INDEX -gt 5 ]; then
        break
    fi
done
# ----------------------------------------
if $INTERFACE_FOUND; then
    Echo "[${INTERFACE}] Interface ready to setup 3proxy"
```

```
# ----------------------------------------
Echo "[${INTERFACE}] Start to setup 3proxy"
CONFIG_LOCATION=/root/3proxy

# ----------------------------------------
Echo "[${INTERFACE}] Create 3proxy config"
FILE_NAME=3proxy_${INTERFACE_ID}.cfg
FILE_LOCATION="${CONFIG_LOCATION}/${FILE_NAME}"

# ----------------------------------------
Echo "[${INTERFACE}] Config location ${FILE_LOCATION}"

tee $FILE_LOCATION <<- EOF >/dev/null
    #! /usr/local/bin/3proxy
    daemon
    nserver 8.8.8.8
    nscache 65536
    timeouts 1 5 30 60 180 1800 15 60
    users TProxy:CL:moNEnTIM
    log /var/log/3proxy_${INTERFACE_ID}.log
    auth strong
    allow TProxy
    proxy -n -a -p5000${INTERFACE_ID} -i192.168.0.103 -
e192.168.${INTERFACE_ID}.100
    flush
EOF

# ----------------------------------------
```

```
        Echo "[${INTERFACE}] Create ip route config"
        ifconfig ${INTERFACE} 192.168.${INTERFACE_ID}.100
        sleep 2

        # ----------------------------------------
        Echo "[${INTERFACE}] Create routing"
        ip route del 192.168.${INTERFACE_ID}.1

        ip route add 192.168.${INTERFACE_ID}.0/24 dev ${INTERFACE} src
192.168.${INTERFACE_ID}.100 table gw${INTERFACE_ID}
        ip route add default via 192.168.${INTERFACE_ID}.1 dev
${INTERFACE} table gw${INTERFACE_ID} metric 100
        ip rule add from 192.168.${INTERFACE_ID}.100/32 table
gw${INTERFACE_ID}
        ip rule add to 192.168.${INTERFACE_ID}.100/32 table
gw${INTERFACE_ID}
        sleep 5

        # ----------------------------------------
        Echo "[${INTERFACE}] Start 3proxy"
        3proxy $FILE_LOCATION
        sleep 2
    fi

    # ----------------------------------------
    Echo "[${INTERFACE}] ...Add ${INTERFACE}."
}
```

```
Remove()
{
        # ----------------------------------------
        Echo "[${INTERFACE}] Remove ${INTERFACE}..."

        # ----------------------------------------
        Echo "[${INTERFACE}] Remove 3proxy process
3proxy_${INTERFACE_ID}"
        kill -9 $(ps aux | grep 3proxy_${INTERFACE_ID} | grep S[Ns]sl | tr -s ' ' |
cut -d ' ' -f2 | head -n 1)

        # ----------------------------------------
        Echo "[${INTERFACE}] Remove ip rules"
        ip rule del from all to 192.168.${INTERFACE_ID}.100
        ip rule del from 192.168.${INTERFACE_ID}.100 lookup
gw${INTERFACE_ID}

        # ----------------------------------------
        Echo "[${INTERFACE}] ...Remove ${INTERFACE}."
}
#
case "${ACTION}" in

        add)
                Add
                ;;
        remove)
                Remove
```

;;

esac

3. Створитиюємо файл з uDEV правилами /etc/udev/rules.d/99-modems.rules

SUBSYSTEM=="net", KERNELS=="3-3.4",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth0",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"
SUBSYSTEM=="net", KERNELS=="3-3.2.4",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth1",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"
SUBSYSTEM=="net", KERNELS=="3-3.2.3",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth2",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"
SUBSYSTEM=="net", KERNELS=="3-3.2.2",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth3",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"
SUBSYSTEM=="net", KERNELS=="3-3.1",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth4",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"
SUBSYSTEM=="net", KERNELS=="3-4.1.1",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth5",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"
SUBSYSTEM=="net", KERNELS=="3-4.1.3",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth6",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"

SUBSYSTEM=="net", KERNELS=="3-4.2",
ATTR{address}=="0c:5b:8f:27:9a:64", ACTION=="add", NAME="eth7",
RUN+="/usr/local/bin/auto-usb-network.sh add $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth0", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth1", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth2", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth3", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth4", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth5", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth6", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"
SUBSYSTEM=="net", ENV{INTERFACE}=="eth7", ACTION=="remove",
RUN+="/usr/local/bin/auto-usb-network.sh remove $name"

4. KERNELS - параметри для валідації модемів по USB
udevadm monitor # check connections via usb
udevadm info --attribute-walk --name=/dev/sdd # here we can find kernel

Таким чином була розроблена система повного циклу, що складається з наступних частин:

- Панель управління: розроблена за допомогою мови програмування Python та web-framework Flask. У якості веб - серверу була обрана модель Nginx +

Werkzeug. Клієнтський інтерфейс був реалізований за допомогою шаблонізатора Jinja2. База данних - PostgreSQL, так як вона має гарну підтримку ORM у Python за допомогою бібліотеки SQLAlchemy.

- Головний сервер: містить в собі змішаний тип розробки. API частина була розроблена за допомогою Flask-RESTfull та мови програмування Python. Також були додані bash скрипти для контролю за підключеннями та управління Raspberry PI міні-комп'ютерами.
- Проведено детальний схемотехнічний аналіз міні-комп'ютеру Raspberry PI, встановлена операційна система (Raspbian). За допомогою bash скриптів та мови програмування Python було створено взаємозв'язок з модемами (Huawei e3372h). Реалізація маршрутизації пакетів була виконана за допомогою бібліотеки 3proxy. Реалізація VPN з'єднань мала змогу завдяки стандартній бібліотеці OpenVPN.

3 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

Чим вищий рівень розвитку інформаційних технологій та Інтернету, тим вищий і попит на VPN, які вже стали невід'ємною частиною нашого життя та нашої безпеки. Якщо ви читаєте важливі листи, переглядаєте соціальні мережі або транслюєте, завантажуєте відео та інший контент, VPN захистить вас від більшості загроз при використанні загальнодоступних та приватних мереж Wi-Fi. Кваліфікаційна робота присвячена розробці віртуальної приватної мережі на базі одноплатного комп’ютера RASPBERRY PI 3, що дозволяє отримувати доступ до заблокованих ресурсів та бути максимально анонімним користувачем для систем аналізу трафіку.

3.1 Огляд проксі-серверів

Bright Data є одним із найбільших у галузі та найнадійніших провайдерів послуг проксі, яким користуються тисячі користувачів Інтернету по всьому світу. Він широко відомий своїми надійними та якісними проксі-серверами та інструментами для розблокування веб-сайтів для обходу блокувань веб-сайтів і отримання загальнодоступних веб-даних. Використовуючи мільйони змінних IP-адрес із різних країн світу, цей інструмент дає змогу виконувати швидкий і стабільний збір загальнодоступних веб-даних у великих масштабах.

– Ключові особливості рішення від Bright Data [22]:

- Понад 120 країн (150 локацій), у яких можна вибрати VPN-сервер.

- Масштабна мережа серверів – понад 1500 одиниць.
- Можливість парсингу сайтів і web-сервісів з боку Bright Data явно обумовлена в угоді користувача. Цією опцією можна навіть керувати за допомогою розкладу доступу.
- Для підключення не потрібна email-адреса або будь-які інші персональні дані.
- До 10 одночасних підключень одного користувача.
- Надійне шифрування та підтримка популярного протоколу IKEv2.
- Вбудована функція Kill Switch (для екстреного розриву з'єднання, якщо відсутнє підключення VPN).
- Є розширення для браузерів (без програми для ПК не працюють).
- Торрент-трафік заборонений та блокується (завантажувати та роздавати торенти через BrightVPN не вийде).
- Трафік без будь-яких суворих лімітів.
- Поки що клієнти реалізовані тільки для ОС Windows (додатки для Mac і для мобільних ОС у процесі розробки).

Proxy-Seller — це постачальник проксі-серверів центру обробки даних, який прагне балансувати між доступними проксі-серверами та робочими проксі-серверами. Їхні проксі-сервери є елітними проксі-серверами і можуть уникнути виявлення з боку багатьох служб виявлення проксі-серверів і веб-сайтів.

− Переваги Proxy-Seller [23]:

- Стабільне високошвидкісне з'єднання до 1 Gb/s;
- Підтримка протоколів HTTP(s) та Socks5;
- Підтримка клієнтів цілодобово;
- Допомога в налаштуванні проксі, при необхідності заходимо на пк користувача через тимчасовий доступ по TeamViewer;
- Велика кількість мереж 400 і підмереж 800;

- Більше 12 країн проксі, і навіть можливість вибрати певне місто;
- Заміна проксі або повернення протягом 24 годин після отримання персональних даних проксі.

3.2 Порівняльний аналіз серверів за критеріями

Представлені сервери мають свої переваги та недоліки. Тому проведемо порівняння представлених серверів (табл.3.1) враховуючи шкалу відносної важливості (табл.3.2)

Solution	Uptime (Стабільність)	Price (Вартість)	Speed (Швидкість)	Target (Локації)	Services (Послуги)
Typical Proxy	99.999%	6$ / місяць	100 Mb / s	Україна	CMC, VPN, Proxy
Bright Data	99.99%	4000$ / місяць	95 Mb / s	Немає	VPN, Proxy
Proxy Seller	99.99%	3700$ / місяць	80 Mb / s	Немає	Proxy

Таблиця 3.1 – Варіанти серверів

Технологія	Короткий опис (ФПП)
A	Bright Data
B	Proxy-Seller
C	Typical Proxy (запропонований сервер)

Таблиця 3.2 – Шкала відносної важливості

Інтенсивність відносної важливості	Визначення
1	рівна важливість
3	помірна перевага
5	сильна перевага
7	значна перевага
9	дуже сильна перевага
2,4,6,8	проміжні судження

Вибір робимо за критеріями, наведеними в таблиці 3.3, встановлюємо відносну вагу кожного критерію на основі матриці попарних порівнянь для обраних критеріїв.

Таблиця 3.3 – Попарне порівняння критеріїв

Критерій	1	2	3	4	5	$\sqrt[5]{\prod_{i=1}^{5}\omega_i}$	X_i
1	2	3	4	5	6	7	8
1. Стабільність	1	5	7	5	1/3	2,25	0,33
2. Швидкість	1/5	1	5	7	1/5	0,67	0,1
3.Локації	1/7	1/5	1	1/3	1/7	0,27	0,04
4. Послуги	1/5	1/7	3	1	1/3	0,49	0,07
5. Вартість	3	5	7	3	1	3,15	0,46
∑						6,83	1

Далі аналогічно складаємо 6 матриць попарних порівнянь альтернатив стосовно кожного критерію. Оскільки тепер порівнюються 3 технології по одному критерію, то i = 1, 2, 3;

$$X_i = \frac{\sqrt[3]{\prod_{i=1}^{3} \omega_i}}{\sum_{i=1}^{3} \sqrt[3]{\prod_{i=1}^{3} \omega_i}}, \qquad (3.1)$$

де $\sum$ - сума по стовпцю $\sqrt[3]{\prod_{i=1}^{3} \omega_i}$.

Таблиця 3.4 – Порівняння альтернатив стосовно критерію «стабільність»

Технологія	A	B	C	$\sqrt[3]{\prod_{i=1}^{3} \omega_i}$	X_i
A	1	5	1/3	1,18	0,29
B	1/5	1	1/5	0,34	0,09
C	3	5	1	2,46	0,62
∑				3,98	1

Таблиця 3.5 – Порівняння альтернатив стосовно критерію «швидкість»

Технологія	A	B	C	$\sqrt[4]{\prod_{i=1}^{4} \omega_i}$	X_i
A	1	1/3	1/3	0,47	0,14
B	3	1	1/3	1	0,28
C	3	3	1	2,08	0,58
∑				3,55	1

Таблиця 3.6 – Порівняння альтернатив стосовно критерію «локації»

Технологія	A	B	C	$\sqrt[4]{\prod_{i=1}^{4}\omega_i}$	X_i
A	1	3	1/3	1	0,26
B	1/3	1	1/5	0,4	0,11
C	3	5	1	2,46	0,63
∑				3,86	1

Таблиця 3.7 – Порівняння альтернатив стосовно критерію «послуги»

Технологія	A	B	C	$\sqrt[4]{\prod_{i=1}^{4}\omega_i}$	X_i
A	1	3	1	1,44	0,43
B	1/3	1	1/3	0,48	0,14
C	1	3	1	1,44	0,43
∑				3,36	1

Таблиця 3.8 – Порівняння альтернатив стосовно критерію «вартість»

Технологія	A	B	C	$\sqrt[4]{\prod_{i=1}^{4}\omega_i}$	X_i
A	1	1/2	1/3	1,22	0,26
B	2	1	1/3	1,5	0,32
C	3	3	1	1,91	0,41
∑				4,63	1

Глобальний пріоритет для кожної альтернативи обчислюється як сума добутків кожного локального пріоритету на його ваговий коефіцієнт.

Таблиця 3.9 – Глобальний пріоритет для кожної альтернативи

Пріоритети	№1	№2	№3	№4	№5	Глобальний
Вага	0,33	0,1	0,04	0,07	0,46	
Bright Data	0,29	0,4	0,26	0,43	0,26	0,2958
Proxy-Seller	0,09	0,28	0,11	0,14	0,32	0,2191
Typical Proxy (запропонований сервер)	0,62	0,58	0,63	0,43	0,41	0,5065

З порівняння глобальних пріоритетів різних серверів (табл.3.9) видно, що найбільшим є пріоритет у Typical Proxy.

Висновки: За допомогою методу аналізу ієрархій проведене порівняння трьох проксі-серверів за критеріями:1) стабільність; 2) швидкість; 3) локації; 4) послуги; 5) вартість. За даними таблиць глобальний пріоритет за багатьма критеріями є найвищим для проксі-серверу Typical Proxy.

4 ОХОРОНА ПРАЦІ ТА ТЕХНОГЕННА БЕЗПЕКА

4.1 Характеристика потенційних небезпечних та шкідливих виробничих факторів при монтажі радіо-електронних компонентів

Згідно з «ГОСТ 12.0.003-74 Небезпечні та шкідливі виробничі фактори. Класифікація» небезпечні та шкідливі виробничі фактори поділяються за своєю природою дії на наступні групи:

- фізичні;
- хімічні;
- біологічні;
- психофізіологічні.

Підвиди фізичних небезпечних та шкідливих виробничі чинників, які зустрічаються в монтажному цеху:

- рухомі машини і механізми; рухомі частини виробничого обладнання;рухомі вироби, заготовки, матеріали; руйнуються конструкції;
- підвищена запиленість та загазованість повітря робочої зони;
- підвищена або знижена температура поверхонь обладнання, матеріалів;
- підвищена або знижена температура повітря робочої зони;
- підвищений рівень шуму на робочому місці;
- підвищена або знижена вологість повітря;
- відсутність або нестача природного світла;
- підвищена яскравість світла;
- пряма і відбита близькість;
- гострі кромки, задирки і шорсткість на поверхнях заготовок, інструментів та обладнання;

За характером впливу на організм людини хімічні небезпечні і шкідливівиробничі фактори поділяються на:

- токсичні;
- дратівливі;
- сенсибілізуючі. [24]

В даний час майже всі електромонтажні з'єднання радіоелектронної апаратури (РЕА) здійснюються пайкою. Технологічний процес пайки включає в себе випал ізоляції і лудіння.

При виконанні пайки на працюючих можуть впливати наступні небезпечні і шкідливі виробничі фактори:

- запиленість і загазованість повітря робочої зони;
- наявність інфрачервоних випромінювань від розплавленого припою уванні або від паяльника;
- наявність електромагнітного випромінювання високої частоти;
- дію ультразвуку на організм монтажника при пайці хвилею, яка утворюється за рахунок дії ультразвуку на розплавлений припій;
- вплив електростатичного заряду;
- незадовільна освітленість робочих місць або підвищена яскравість;
- незадовільні метеорологічні умови в робочій зоні;
- вплив бризок і крапель розплавленого припою;
- ураження електричним струмом;
- група психофізіологічних шкідливих виробничих факторів: фізичні перевантаження (статичні і динамічні) і нервово-психічні (монотонність праці, емоційні перевантаження).

Операції пайки, залуження та випалу ізоляції супроводжуються забрудненням повітряного середовища в приміщеннях:

- парами свинцю, олова, сурми та інших елементів, що входять до

складу припою;

- парами каніфолі і різних рідин, що застосовуються для флюсу, змивки і розчинення різних лаків, які застосовуються для покриття друкованих плат;
- парами соляної кислоти; газами (окис вуглецю, вуглеводню) і т. інш.

Пари, потрапляючи в атмосферу цеху, конденсуються і перетворюються в аерозоль такої конденсації, частки якої за своєю дисперсності наближаються до диму.

Перебуваючи в запиленій атмосфері, робітники піддаються впливу пилу і парів; шкідливі речовини осідають на поверхні шкірного покриву, потрапляють на слизову оболонку порожнини рота, очей, верхніх дихальних шляхів, зі слиною заковтуються в травний тракт, вдихаються в легені. Поряд із за- брудненням повітряного середовища забруднюються робочі поверхні, одяг і шкіряні покриви працюючих.

Особливо шкідливі при пайці олов'яно-свинцевими припоями пари свинцю. Свинець і його сполуки отруйні. Частина свинцю, що надійшов в організм, виводиться з нього через кишечник і нирки, а частина затримується в кістковій речовині, м'язах, мозку, печінки. При несприятливих умовах свинець починає циркулювати в крові, викликаючи явища свинцевого отруєння. Свинець викликає зміни в складі крові, вражає нервову систему, нирки і печінку.

Властивість свинцю—накопичуватися в організмі призводить до хронічного отруєння при систематичному надходженні в організм навіть його малих кількостей. Для запобігання гострих і професійних захворювань вміст свинцю в повітряному середовищі не повинен перевищувати гранично допустимої концентрації 0,1 мг/м^3.

У виробництві радіоелектронної апаратури крім олов'яно-свинцевих припоїв знаходять застосування припої, до складу яких входять мідь, літій, срібло, кадмій та інші метали. У деяких випадках пайка здійснюється

шляхомзанурення в розплавлені хлористі солі кадмію, натрію, бору, літію з додаванням активних добавок – фтористих солей. Пари більшості з перерахованих речовин, що утворюються при пайці, можуть мати шкідливий вплив на організм працюючих.

Найбільш небезпечні пари окису кадмію, міді і фтористі з'єднання. Не байдужі для організму також літій і хлористий цинк, які надають подразнюючу дію на шкіру і дихальні шляхи. Пайка в атмосфері звичайними припоями проводиться із застосуванням флюсів.

Біологічна дія флюсів на організм людини залежить від компонентів, що входять до складу паяльних флюсів. Одні компоненти (каніфоль соснова, етил ацетат, олеїнова кислота та ін.) мають подразнюючу дію; інші (спирт етиловий) – наркотичну, треті (семікарбазід гідрохлорид, етиленгліколь) – високою токсичністю; дію четвертих (кремнійорганічна рідина) на організм ще недостатньо вивчено.

Деякі марки флюсів (ФГСп, ФДФс, ФСкСп і ін.) через високу токсичність рекомендується не застосовувати або обмежувати їх застосування. У всіх флюсах слід етиленгліколь замінювати гліцерином, так як він здатний проникати в організм навіть через неушкоджену шкіру.

Для видалення залишків флюсів після пайки в залежності від марки флюсу застосовуються різні мийні середовища, які володіють токсичними властивостями. [25]

4.2 Розрахунок необхідного повітрообміну приміщення з виділенням шкідливих речовин

Вихідні данні:

- Об'єм монтажного цеху V = 20000 м^3;
- Паяння та лудіння проводиться м'яким припоєм ПОС-40 (в його

складвходить t = 40% свинцю);

- За 1 годину роботи витрачається m = 0,4 кг припою;
- Кількість припою, що випаровується q = 0,3%;
- Число працюючих n = 25 чоловік;
- Вміст парів свинцю в зовнішньому повітрі $C_{прит}$ = 0;
- Гранично-допустима концентрацію свинцю в повітрі робочої зониПДК$_{Pb}$ = 0,01 мг/м^3.

Визначити: необхідний повітрообмін.

Рішення:

Визначаємо кількість свинцю, яке випарується за 1 годину роботи:

$$W = t \cdot m \cdot q \cdot 10^6 = 0{,}4 \cdot 0{,}4 \cdot 0{,}3\% \cdot 10^6 = 480 \text{ мг/год.} \quad (4.1)$$

де 10^6 – коефіцієнт для перекладу з кг/год в мг/год.

Визначаємо кількість повітря, яке потрібно подати в робочу зону для того, щоб концентрація свинцю в робочому обсязі не перевищувала значень ГДК:

$$G = \frac{W}{C_{гдк} - C_{прит}} = \frac{480}{0{,}01 - 0} = 4{,}8 \cdot 10^4 \ \frac{м^3}{год} \quad (4.2)$$

Визначаємо кількість повітря, яке потрібно подати в робочу зону для того, щоб забезпечити необхідну кількість повітря на працюючого:

$$G_1 = n \cdot G_{люд} = 25 \cdot 60 = 1500 \frac{м^3}{год} \quad (4.3)$$

де $G_{люд}$ – норма подачі припливного повітря на 1 людину, яка дорівнює60 м^3/год.

Порівнюючи норми подачі G і G_1 для подальших розрахунків приймаємо більше значення, тобто значення G. [24]

Знаходимо кратність повітрообміну:

$$k = \frac{G}{v} = \frac{4{,}8 \cdot 10^4}{2 \cdot 10^4} = 2{,}4 \frac{1}{год} \qquad (4.4)$$

4.3 Заходи з поліпшення умов праці та виробнича санітарія

Метеорологічні умови в приміщенні – температура повітря, відносна вологість повітря й швидкість його переміщення відповідають встановленим санітарно-гігієнічним вимогам ДСН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень» і ГОСТ 12.1.005-88 (1991) «ССБТ. Загальні санітарно-гігієнічні вимоги до повітря робочої зони». [24]

З огляду на шкідливість вихідних компонентів, що входять до складу припоїв, флюсів, миючих середовищ, і забруднення атмосфери виробничих приміщень пилом, парами і газами, для досягнення сприятливих умов праці необхідно провести комплекс наступних заходів:

1) Ділянки, на яких зосереджені операції пайки, виділяють в окремі приміщення. Якщо пайки проводяться на потоковій лінії при чергуванні їх з іншими технологічними операціями, виробничі приміщення в цьому випадку розглядають як приміщення, призначені для пайки.

2) Стіни, віконні рами, опалювальні прилади, повітроводи повинні бути гладкими і покриваються олійною фарбою світлих тонів

(панелі на рівні 1,5 -2 м від підлоги краще облицювати плиткою). Підлоги повинні бути водонепроникними, мати підвищену міцність і стійкістю до стирання і займання, без щілин і мати ухили до трапів каналізації. На ділянках пайки їх миють після кожної зміни. Не рідше одного разу на тиждень роблять вологе прибирання всього приміщення.

3) При ручній пайці і випалюванні ізоляції з метою захисту від ураження електричним струмом електропаяльник і електровідпал повинні працювативід електромережі напругою не вище 42 В.

4) Прибирання обладнання проводиться із застосуванням пневмозбиральної системи. Робочі поверхні столів, ящиків для зберігання інструментів та тара в кінці зміни очищаються і обмиваються гарячим мильним розчином.

5) Використані серветки і ганчір'я після зміни повинні спалюватися, повторне їх використання не допускається.

6) Шафи для зберігання робочого одягу та особистих речей щотижня всередині і зовні обмиваються гарячою водою з милом.

7) Експлуатація ділянок пайки, не обладнаних витяжною вентиляцією, забороняється. Вентиляційні установки повинні включатися до початка робіті вимикатися після їх закінчення.

8) Приміщення, в яких розміщуються ділянки пайки, обладнуються відокремленою припливно-витяжною вентиляцією. Приплив повітря повинен складати 95% обсягу витяжки. Відсутні 5%, припливного повітря надходятьіз суміжних, більш чистих приміщень.

9) Особи, які не досягли 18-річного віку, до постійної роботи з припоями, що містять свинець і кадмій, не допускаються.

10) Жінки, зайняті паянням, в період вагітності і годування дітей пере- водяться на роботу, не пов'язану з пайкою.

11) Працівники повинні бути проінструктовані про запобіжні

заходи при поводженні з припоями та флюсами. Особлива увага при інструктажі слід приділяти питанням особистої гігієни.

Місця, відведені для куріння, а також кімнати для прийому їжі і виробничі ділянки обладнуються умивальниками, до яких безперебійно повинна подаватися гаряча і холодна вода. Біля вмивальників передбачаються банки з 1%-ним розчином оцтової кислоти або змивочні пасти на основі ОП-7 для попереднього обмивання рук з подальшим миттям їх теплою водою з милом. Перед прийомом їжі і курінням обов'язково мити руки і полоскати порожнину рота. Для обтирання рук застосовуються разові серветки. Застосування рушників загального користування не дозволяється.

Для захисту шкіри рук від впливу сенсибілізуючих речовин, що входять до складу флюсів, застосовують захисні мазі і пасти, казеїнову пасту і біологічні рукавички, які наносять на шкіру перед початком роботи і після обідньої перерви. Після роботи для шкіри рук необхідно застосовувати жир- ні живильні креми.

Питну воду для працюючих на ділянках пайки слід подавати через фонтанчики, які встановлюються поза паяльних ділянок, але поблизу їх.

Паяльні роботи повинні виконуватися робітниками в передбаченому для цієї мети спецодязі (бавовняний халат, гумові рукавички, тапочки на шкіряній підошві, захисні окуляри), який забороняється забирати додому. У приміщеннях, де проводиться паяння, забороняється зберігати спецодяг, особисті речі, приймати і зберігати їжу, питну воду, а також палити. Перебувати в приміщеннях для прийому їжі, їдальнях та буфетах в робочому одязі забороняється. [26]

4.4 Електробезпека

Оскільки в приміщенні відділу знаходиться електроустаткування, основні заходи щодо техніки безпеки повинні здійснюватися відповідно до НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів».

Порядок навчання і перевірки знань працівників має бути відповідним до галузевого положення про навчання, інструктаж і перевірку знань працівників з питань охорони праці узгодженого з Держнаглядохоронпраці, а такождо вимог до електротехнічної обслуги, які містяться в ПТЕ.

Первинний (під час прийняття на роботу) та періодичний (протягом трудової діяльності) медичний огляд працівників провадиться згідно з Положенням про медичний огляд працівників певних категорій.

Працівники, що обслуговують електроустановки, зобов`язані знати ці Правила відповідно до займаної посади чи роботи, яку вони виконують, і мати відповідну групу з електробезпеки.

Забороняється допускати до роботи в електроустановках осіб, які не пройшли навчання і перевірку знань цих Правил.

Ті працівники, зайняті виконанням спеціальних видів робіт, до яких висуваються додаткові вимоги безпеки, мають бути навчені безпечному виконанню таких робіт і мати відповідний запис про це у посвідченні з перевірки знань з питань охорони праці.

Працівник допускається до роботи в електроустановках до 1000 В або до і вище 1000 В. Кожний працівник особисто відповідає за свої дії в частині дотриманнявимог цих Правил.

У випадку, якщо працівник самостійно не спроможний вжити дійових заходів з усунення виявлених ним порушень Правил, він зобов’язаний

негайно повідомити про це безпосереднього керівника, а у випадку його відсутності – керівника вищого рівня.

В разі нещасних випадків з людьми зняття напруги для звільнення потерпілого від дії електричного струму має бути виконано негайно, без попереднього дозволу.[26]

Електричні мережі і установки у приміщенні відділу виконані так, що струмопровідні частини їх недоступні для випадкового дотику, по периметру приміщення проведено заземлюючий контур, підлога дерев'яна для зниження величини виникаючих зарядів статичної електрики, також передбачена система аварійного відключення електрики в разі поломки або аварійної ситуації.

4.5 Пожежна безпека

Приміщення відділу згідно з ДСТУ Б В.1.1-36: 2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою» за пожежною небезпекою відноситься до категорії В і класу пожежі А (пожежі твердих горючих речовин, в основному органічно- го походження, горіння яких супроводжується тлінням, таких як, пластик, текстиль, папір, дерево) та (Е) (пожежі, пов'язані з горінням електроустановок). [27]

Для забезпечення захисту використовується пожежна сигналізація на базі автоматичних комбінованих (димових і теплових) пожежних систем сповіщення, таких, як ІДФ- 1, Ді- 1, АТІМ- 1, АТІМ- 3, ДТЛ, ПТІМ- 1, ПТІМ- 2, ДПС- 038 та ін. У разі займання передбачається використання переносних ручних вогнегасників: вуглекислотних марок ОУ- 2, ОУ- 5, ОУ- 8, порошкових, - ОП- 1, ОП- 2, ОП- 10, заряджений порошком ПСБ, який застосовується для гасіння електроустановка під напругою і ЕОМ. Для

порятунку людей у разі займання застосовуються проти димні маски з фільтрами респіраторного типу. Вогнегасники і захисні маски розташовані в легкодоступних і помітних місцях, де виключено попадання на них прямих сонячних променів і безпосередньо вплив опалювальних і нагрівальних приладів.

Ручні вогнегасники розміщені методом навішування на вертикальні конструкції на висоті 1,5 м. При спрацьовуванні автоматичних установок пожежної сигналізації автоматично відключається система вентиляції і кондиціонування повітря. Локалізація вогнища займання і відвертання поширення вогню забезпечено застосуванням вогне- перешкоджальних облаштувань оснащення і протипожежних перешкод.

Шляхи евакуації визначені виходячи з об'ємного планування і технічного виконання будов і споруд з урахуванням їх вогнестійкості (збереження функцій, що несуть, при пожежі упродовж розрахункового часу евакуації). Організовано голосове сповіщення і світлові індикатори для управління рухом по евакуаційних шляхах. Система проти димного захисту забезпечує не задимлення, пониження температури і видалення продуктів горіння на шляхах евакуації упродовж часу, який буде достатнім для евакуації людей.

Виходячи з норм пожежної безпеки в залі обчислювального центру (площею $60м^2$) є первинні засоби пожежогасінні :

- один вуглекислотний вогнегасник типу ОУ- 5 або ОУ- 8, за допомогою яких можна гасити займання різних матеріалів і установок напругою до 1000 В;
- один хімічно-пінний (ОХП- 10) або легко-пінний вогнегасник (ОВП- 5 або ОВП- 10). За допомогою якого можна гасити тверді матеріали і горючі рідини (окрім установок під напругою);
- повсть або повстяний азбест (1х1; 2х1, 5; 2х2 м).

Приміщення обчислювального центру має бути обладнане пожежними оповісниками, які дозволяють оповістити черговий персонал про пожежу. В якості оповісників встановлені димні фотоелектричні оповісники типу ІДФ-1 або ДіП- 1[28].

4.6 Заходи безпеки в надзвичайних ситуаціях

Серед захисних заходів щодо безпеки в умовах надзвичайної ситуації, особливо важливе місце займає організація сповіщення органів цивільної оборони, формувань населення про загрозу нападу ворога і про застосування ним ядерної зброї, хімічної або бактеріологічної зброї та інших сучасних засобів масової поразки. Тому захист населення від зброї масового ураження залежить від добре організованої системи сповіщення. Усі сигнали цивільної оборони в підрозділи підприємства передаються по каналах зв'язку і радіотрансляційним мережам, а також через місцеві радіостанції.

Для попередження працівників підприємства встановлені наступні сигнали:

- "увага всім";
- "повітряна тривога";
- "відбій повітряної тривоги";
- "радіаційна небезпека";
- "хімічна тривога".

Сигнал "увага усім" подається для усього населення свідомо до подання кожного з сигналів. Це необхідно для того, щоб підготувати людей до сприйняття наступних сигналів, повідомлень або вказівок. По радіотрансляційних системах передається текст: "Увага всім"!.

Сигнал "повітряна тривога" подається для усього населення. По радіотрансляційних системах передається текст: "Увага! Увага! Громадяни! Повітряна тривога"!. Одночасно з цим сигнал дублюється звуком сирен, гудками заводів і транспортних засобів. Тривалість сигналів складає 2-3 хвилини.

В цьому випадку працівники припиняють роботу і виконують усі заходи, передбачені спеціальною інструкцією підприємства і, дотримуючись встановленого порядку, йдуть в укриття, закріплене за обчислювальним центром на цьому підприємстві.

Сигнал "відбій повітряної тривоги" передається органами цивільної оборони по радіотрансляційних мережах, через місцеві радіо і телевізійну станцію і іншими способами, які можна використати в конкретній обстановці (телефон, гучномовці). Передається текст: "Увага! Увага! Відбій Повітряної тривоги"!. З цим сигналом працівники обчислювального центру повертаються на робочі місця з укриття і приступають до роботи.

Сигнал "радіаційна небезпека" подається в населених пунктах, у напрямі до яких рухається радіоактивна хмара, яка виникла при вибуху ядерних боєприпасів. Цей сигнал подається за допомогою усіх місцевих технічних засобів зв'язку і сповіщення, а на місцях дублюються звуковими і світловими засобами. При цьому сигналі працівники надівають респіратори, проти-запорошену тканинну маску, ватно-марлеву пов'язку або противогаз, комплекти яких передбачені на цьому підприємстві, а також беруть індивідуальні засоби медичного захисту, предмети першої необхідності і йдуть в укриття.

Сигнал "хімічна тривога" подається при прогнозі або безпосередньому виявленні хімічного або бактеріологічного зараження, за допомогою технічних засобів зв'язку. На місцях він дублюється звуковими і світловими сигналами. Далі працівники виконують усі заходи, передбачені спеціальною

інструкцією підприємства. Про те, що небезпека хімічного або бактеріологічного зараження пройшла і про порядок подальших дій розпорядження прийдуть по тих же каналах зв'язку, що і сигнал сповіщення.
Ядерний вибух супроводжується електромагнітним випромінюванням у вигляді потужного короткого імпульсу, який вражає головним чином електричну і електронну апаратуру. На утворення електромагнітного імпульсу (EMI) витрачається невелика частина ядерної енергії, проте він здатний викликати потужні імпульси струмів і напруги в дротах і кабелях повітряних і підземних ліній зв'язку, сигналізації, управління, електропередачі, в антенах радіостанцій і так далі [29].
Вплив EMI може викликати займання чутливих електронних і електричних елементів, які пов'язані з великими антенами або відкритими дротами, а також викликати серйозні порушення в цифрових і контрольних пристроях, зазвичай без безповоротних змін. Тому вплив EMI необхідно враховувати для усіх електричних і електронних систем.
Особливо EMI впливає на радіоелектронну апаратуру, яка виконана на напівпровідникових і інтегральних схемах, які працюють на малих струмах і напрузі і, отже, чутливих до впливу зовнішніх електричних і магнітних полів. EMI пробиває ізоляцію, випалює елементи інтегральних схем комп'ютерів, викликає коротке замикання, іонізацію діелектриків, або повністю стирає магнітний запис на носіях інформації.
У кожному конкретному випадку мають бути знайдені найбільш ефективні і економічно доцільні методи захисту електронної апаратури. Серед таких методів, які застосовані в обчислювальному центрі, найбільш поширено екранування, оптимальне розміщення і заземлення окремих частин системи, використання приладів, які перешкоджають перенапруженню в найбільш критичних місцях і так далі.

Сполучні кабелю для захисту прокладені в земляних траншеях під цементною або бетонною підлогою будівлі або укладені в металеві коробки, які заземляють.

Усі заходи, розроблені в цій главі, відповідають законодавчим і нормативним актам України і забезпечують необхідні умови для роботи персоналу обчислювального центру[30].

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

– Аналіз VPN та проксі-серверів показав, що найкращим підходом є використання змішаної моделі, яка комбінує переваги обох технологій. Це дозволить користувачам забезпечити високий рівень приватності, шифрування та анонімності завдяки VPN, а також використовувати проксі-сервери для додаткової прихованості їхньої реальної IP-адреси.

– Розроблена віртуальна приватна система повного циклу, що складається з наступних частин:

- Панель управління: розроблена за допомогою мови програмування Python та web-framework Flask. У якості веб - серверу була обрана модель Nginx + Werkzeug. Клієнтський інтерфейс був реалізований за допомогою шаблонізатора Jinja2. База данних - PostgreSQL, так як вона має гарну підтримку ORM у Python за допомогою бібліотеки SQLAlchemy.
- Головний сервер: містить в собі змішаний тип розробки. API частина була розроблена за допомогою Flask-RESTfull та мови програмування Python. Також були додані bash скрипти для контролю за підключеннями та управління Raspberry PI міні-комп’ютерами.
- На міні-комп’ютері Raspberry PI була встановлена операційна система (Raspbian). За допомогою bash скриптів та мови програмування Python було створено взаємозв’язок з модемами (Huawei e3372h). Реалізація маршрутизації пакетів була виконана за допомогою бібліотеки 3proxy. Реалізація VPN з’єднань мала змогу завдяки стандартній бібліотеці OpenVPN.

– Запропонована SaaS (Software-as-a-Service) система має змогу надавати послуги як поодинокім користувачам, так і великим кампаніям. Продукт дозволяє отримувати доступ до заблокованих ресурсів та бути максимально анонімним користувачем для систем аналізу трафіку. Також увесь трафік має шифрування, що дозволяє підвищити безпеку використання системи.

ЛІТЕРАТУРА

1. Що таке VPN, і навіщо він вам у 2023. URL: https://is.gd/rhFjgR (дата звернення: червень 2023).

2. Що таке VPN-підключення і як працює VPN? URL: https://samoosvita.in.ua/scho-take-vpn-pidklyuchennya-i-yak-pratsyue-vpn (дата звернення: червень 2023).

3. Посібник з VPN: Що таке VPN-з'єднання та як воно працює? URL: https://is.gd/mygQrg (дата звернення: червень 2023).

4. Технології забезпечення безпеки мережевої інфраструктури: підручник / В. Л. Бурячок та інш. Київ.: КУБГ, 2019. 218 с

5. Yee C. K., Zolkipli M. F. Review on Confidentiality, Integrity and Availability in Information Security. *Journal of ICT in Education.* 2021. Vol. 8, P.34-42.

6. Проксі-сервер: що це таке і чи потрібен він вам? URL: https://surfshark.com/uk/blog/proxy-server (дата звернення: жовтень 2023).

7. Coleman J. CCPA Clarity in California. *ACA International.* 2020. Vol.1, P.24-26.

8. Garg A., Mittal N. A security and confidentiality survey in wireless internet of things (iot). In Internet of Things and Big Data Applications. 2020. P. 65-88.

9. Grimmelmann James. Saving Facebook. Iowa Law Review. 2019. P. 1137–1206.

10. Typical Proxy. URL: https://is.gd/8IbjQR (дата звернення: жовтень 2023).

11. Організація та принципи роботи веб-сервісу. Протокол HTTP. URL: https://is.gd/65EmvM (дата звернення: жовтень 2023).

12. Захищений протокол HTTPS: що це таке, чим відрізняється від HTTP, як на нього перекласти сайт. URL: https://is.gd/i9IPJT (дата звернення: жовтень 2023).

13. HTML — мова розмітки гіпертексту. URL: http://www.znannya.org/?view=html (дата звернення: жовтень 2023).

14. Що таке веб-сервер? URL: https://is.gd/1gNb6W (дата звернення: жовтень 2023).

15. Підручник з Python. URL: https://docs.python.org/uk/3/tutorial/index.html, (дата звернення: жовтень 2023).

16. Flask -посібник користувача. URL: https://flask.palletsprojects.com/en/2.3.x/ (дата звернення: жовтень 2023).

17. Folium-бібліотека Python. URL: https://python-visualization.github.io/folium/ (дата звернення: жовтень 2023).

18. Міні комп'ютер Raspberry. URL: https://www.raspberrypi.com/ (дата звернення: жовтень 2023).

19. Все про Raspberry P3. URL: https://botland.com.pl/399-raspberry-pi (дата звернення: жовтень 2023).

20. Bright Data. URL: https://brightdata.com/ (дата звернення: жовтень 2023).

23. Огляд Proxy Seller URL: https://www.affiliatebay.net/uk/proxyseller-review/ (дата звернення: жовтень 2023).

24. Основи охорони праці: навчальний посібник для студентів вищих навчальних закладів. / Р.М. Івах та ін. Київ: Кодар, 2010. 462 с.

25.., Дерев'янко О.А., Христич В.В., Антошкін О.А. Системи пожежної та охоронної сигналізації. / С.М. Бондаренко та інш. Харків: УЦЗУ, 2008. 144 с.

26. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. Чинний від 1999-12-01. Вид. офіц. Київ: Державні санітарні норми, 2011. 105 с.

27. ДСТУ 7238:2011 Система стандартів безпеки праці. Засоби колективного захисту працюючих. Загальні вимоги та класифікація. Чинний від 2011-12-01. Вид. офіц. Київ: УкрНДНЦ, 2011. 135 с.

28. ДСТУ 7238:2011 Система стандартів безпеки праці. Засоби колективного захисту працюючих. Чинний від 14.03.2011. Вид. офіц. Київ: УкрНДНЦ, 2011. 78 с.
29. ДСТУ 12.1.018-93 Система стандартів безпеки праці. Пожежовибухобезпека статичної електрики. Загальні вимоги. Чинний від 01.01.1998. Вид. офіц. Київ: УкрНДНЦ, 2001. 97 с.
30. ДСТУ 7237:2011 Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту. Чинний від 01.08.2011. Вид. офіц. Київ: УкрНДНЦ, 2011. 201 с.

MODERN RESEARCH IN WORLD SCIENCE

Proceedings of XI International Scientific and Practical Conference

Lviv, Ukraine

29-31 January 2023

Lviv, Ukraine

2023

РОЗПІЗНАВАННЯ МАТЕМАТИЧНИХ ВИРАЗІВ В ГРАФІЧНОМУ ПРЕДСТАВЛЕННІ ДАНИХ

Небеснюк Владислав Олександрович,
студент кафедри електроніки,
інформаційних системи та програмного забезпечення
Ніконова Зоя Андріївна,
к.т.н., доцент, професор кафедри
електроніки, інформаційних системи та програмного забезпечення
Інженерний навчально-науковий інститут
ім. Ю. М. Потебні
Запорізький національний університет
м. Запоріжжя, Україна

Актуальність. Проблема розпізнавання тексту дуже актуальна у наш час. Існує безліч рукописних матеріалів, які потребують переводу у електронний варіант. Також в останній час дуже популярні різноманітні тач-скрін прилади, такі як: планшети, фаблети, смартфони та комп'ютери. Значно зручніше написати текст від руки, ніж друкувати на клавіатурі, яка може бути маленькою та незручною. Проаналізувавши ринок програмного забезпечення, можна зробити висновок, що існує велика кількість програм з розпізнавання тексту. В основі лежать два підходи до розпізнавання: онлайн та офлайн. Найкращі результати з розпізнавання в обох підходах показують такі застосунки, що використовують нейронні мережі [1]. Тому було обрано офлайн підхід з використанням нейронних мереж. Але програм, що розпізнають саме математичні вирази дуже мало і вони дорого коштують, їх використовують для складних обчислень. Розробка комп'ютерної системи, яка б автоматично розпізнавала математичні вирази є актуальною науково - технічною задачею.

Мета. Метою дослідження є підвищення ефективності розпізнавання математичних виразів за рахунок класифікації та кластеризації зображень.

Матеріали та методи./Materials and methods. Аналіз конкуруючих рішень дозволив виділити три головні сучасні проекти для розпізнавання

математичних виразів:

- Wolfram – проект комплексного рішення різного роду задач, в тому числі зв'язаних з математичними виразами.

Wolfram | Alpha - база знань і набір обчислювальних алгоритмів (англ. Computational knowledge engine), питально-відповідна система.

Wolfram | Alpha не повертає перелік посилань, заснований на результатах запиту, а обчислює відповідь, ґрунтуючись на власній базі знань, яка містить дані з математики, фізики, астрономії, хімії, біології, медицини, історії, географії, політики, музики, кінематографії, а також інформацію про відомих людей та інтернет-сайти. Він здатний переводити дані між різними одиницями виміру, системами числення, підбирати загальну формулу послідовності, знаходити можливі замкнуті форми для наближених дробових чисел, обчислювати суми, межі, інтеграли, розв'язувати рівняння і системи рівнянь, проводити операції з матрицями, визначати властивості чисел і геометричних фігур. Однак, розрахунок на підставі власної бази має і свої недоліки, в тому числі - вразливість до помилок даних.

Движок Wolfram | Alpha заснований на обробці природної мови, великій бібліотеці алгоритмів і NKS-підході для відповідей на запити. Він написаний на мові Mathematica і становить близько 5 мільйонів рядків, в даний час виконується приблизно на 10000 процесорах.

- MathPix – проект розпізнання математичних виразів з власним API для розробників.

Програмний додаток дозволяє вирішувати і візуалізувати рішення розпізнаючи рукописний текст, включаючи складні формули. Використовувати програмне забезпечення (ПЗ) можна для побудови графіків, вирішення квадратних рівнянь, математичних виразів, що містять корені та дробові числа. При цьому на екран видається не тільки результат, але і проміжні етапи рішення.

Також слід відзначити можливість побудови графіків функцій завдяки інтеграції з передовим графічним калькулятором Desmos. Користувачу

доступні інструменти для роботи з завданнями в режимі графіка: редагування вступних даних, додавання таблиць, заміток і додаткових функцій для кількох графіків.

До недоліків слід віднести неможливість вирішення складних задач, що включають тригонометричні і логарифмічні рівняння, нерівності, а також рівняння з модулем. Додаток просто ігнорує їх рішення.

- PhotoMath – проект комплексного рішення математичних виразів з системою розпізнання в реальному часі.

Photomath - мобільний додаток, описаний як «камера-калькулятор», який використовує камеру телефону для розпізнання математичних рівнянь і відображення покрокового рішення на екрані. Додаток в безкоштовному доступі на Android і iOS. На даний момент PhotoMath підтримує такі математичні дії як: додавання, віднімання, множення, ділення, вилучення коренів, а також прості лінійні рівняння. Нові функції знаходяться в розробці і будуть додаватися в PhotoMath як оновлення. До недоліків слід віднести не можливість розпізнавання рукописних символів, а лише друкованого тексту.

Результати та обговорення./Results and discussion. Розроблена архітектура комп'ютерної системи розпізнавання зображень складається з серверу навчання моделей, веб-серверу та клієнтського застосунку (рис. 1).

Рис.1. Архітектура системи

Глибоке порівняння мов програмування: C++, C#, Matlab та Python дозволило обрати C++ в якості мови розробки. Цей вибір обумовлюється тим, що ця мова має ряд значних переваг:

- C++ кросплатформена мова;

C++ дуже швидка мова. Одним з критеріїв успіху можливо відзначити швидкодію застосунку. Швидкість необхідна і при навчанні моделі, і при роботі

сервера [2]. В програмному застосунку використано потужні інструменти такі, як CUDA та OpenCV. Вони реалізовані на мові C++.

Для реалізації серверної частини та огортки навколо системи розпізнавання використовуємо сервер Nginx з технологією proxy pass для відстеження прямих запитів та Flask framework з використанням технології REST API.

В запропонованому проекті математичні вирази використовуються в якості вхідних даних. Бібліотека FreeImage дозволяє використовувати зображення з будь-яким розширенням. Ці данні будуть отримуватися від користувача. Користувач отримує відповідь в LaTeX форматі.

Висновки./Conclusions. Авторами розроблено програмний застосунок, що підвищує ефективність розпізнавання математичних виразів за рахунок класифікації та кластеризації зображень. Дослідження ефективності роботи системи показало, що запропоноване програмне рішення має точність розпізнавання математичних виразів близько 98 %, що дає широкі можливості для розвитку і великі шанси зайняти своє місце на ринку. Реалізований програмний продукт може бути використано для розробки програмного забезпечення з застосуванням розпізнавання математичних виразів та для веб-сайтів, що використовують математичну модель представлення даних.

ЛІТЕРАТУРА./ REFERENCES

1. Що таке нейронні мережі? URL: https://evergreens.com.ua/ua/development-services/neural-network.html (дата звернення: 10.01.2023).

2. Найновіша довідка по C++ URL: https://cplusplus.com/ (дата звернення: 25.12.2022).


CERTIFICATE
is awarded to
Nebesniuk Vladyslav
for being an active participant in
XI International Scientific and Practical Conference
"MODERN RESEARCH
IN WORLD SCIENCE"
24 Hours of Participation
(0,8 ECTS credits)
LVIV
29-31 January 2023
INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE
MODERN RESEARCH IN WORLD SCIENCE
SCI-CONF.COM.UA
• LVIV •
29-31 JANUARY 2023
sci-conf.com.ua