

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Миколишин Павло Петрович

**Пристрій двостороннього зв'язку згідно протоколу SIP/
Two-way communication device according to the SIP protocol**
спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-41
Миколишин П.П.

Науковий керівник
Романець І.Є.

Кваліфікаційну роботу
Допущено до захисту
«___» _____ 20 ___ р.

Завідувач кафедри
_____ О.М. Березький

ТЕРНОПІЛЬ - 2021

РЕЗЮМЕ

Кваліфікаційна робота на тему «Пристрій двостороннього зв'язку згідно протоколу SIP» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 56 сторінки пояснючої записки, 9 рисунків, 8 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка пристрою для двостороннього зв'язку у мережі Інтернет згідно протоколу SIP.

Методи дослідження включають модель та засоби віртуалізації і комунікацій мережі VoIP-телефонії.

1. У проекті надані необхідні розрахунки й креслення, специфікація роботи з приладом і його налаштування, необхідного для здійснення двостороннього зв'язку по протоколу SIP.

2. Реалізовано побудова моделі VoIP-телефонії та проектування корпоративної мережі на її основі з використанням засобів віртуалізації та комунікаційного додатку з відкритим кодом Asterisk. У проекті надані необхідні розрахунки й креслення, специфікація встаткування й матеріалів, необхідних для побудови локальної обчислювальної мережі.

3. Головним елементом комп'ютерної системи є сервер, до якого під'єднується плата Olimex-ADF. Розроблено для налаштування сервера.

4. Розроблено структурну схему мережі і план розташування її елементів в будівлі відповідно до вимоги до розширюваності мережі і можливостям її подальшого удосконалення.

Ключові слова: КОРПОРАТИВНА КОМП'ЮТЕРНА МЕРЕЖА, НАЛАШТУВАННЯ ESP32 ДЛЯ SIP ТЕЛЕФОНІЇ.

RESUME

The qualification work on the topic "Two-way communication device according to the SIP protocol" in the specialty 123 "Computer Engineering" with a bachelor's degree contains 72 pages of explanatory note, 9 figures, 8 tables, 3 appendices. Volume of graphic material 2 sheets of A3 format.

The purpose of the qualification work is to develop a device for two-way communication on the Internet according to the SIP protocol.

Research methods include a model and means of virtualization and communications of the VoIP-telephony network.

1. The project provides the necessary calculations and drawings, specifications for operation of the device and its settings required for two-way communication via SIP.

2. The construction of the model of VoIP-telephony and the design of a corporate network based on it using the means of virtualization and communication application with open source Asterisk. The project provides the necessary calculations and drawings, specifications of equipment and materials needed to build a local area network.

3. The main element of the computer system is the server to which the Olimex-ADF board is connected. Designed to configure the server.

4. The structural scheme of a network and the plan of an arrangement of its elements in the building according to the requirement to expandability of a network and possibilities of its further improvement is developed.

Keywords: CORPORATE COMPUTER NETWORK, ESP32 SETUP FOR SIP.

ЗМІСТ

Перелік умовних скорочень.....	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Поняття IP-телефонії.....	11
1.2 Телекомунікаційні послуги через IP-мережі.....	14
1.3 Аналіз технічного завдання та постановка цілей проектування.....	15
2 Пристрій двостороннього зв'язку згідно протоколу SIP.....	16
2.1 Структура пристрою VoIP.....	16
2.2 Алгоритм роботи приладу двостороннього зв'язку.....	17
2.3 Обґрунтування вибору апаратного забезпечення.....	19
3 Реалізація і налаштування пристрою двостороннього зв'язку.....	25
3.1 Опис та налаштування пристрою на базі ESP32.....	25
3.2 Розробка основних функцій.....	28
3.2 Підбір і налаштування сервера.....	33
3.3 Тестування пристрою двостороннього зв'язку.....	39
4 Техніко-економічний розділ.....	46
4.1 Розрахунок витрат на реалізацію пристрою.....	46
4.2 Розрахунок ціни проекту.....	54
4.3 Визначення показників економічної ефективності.....	56
Висновки.....	61
Список використаних джерел.....	62
Додаток А Лістинг налаштування файлу voip.c.....	67
Додаток Б Світлокопія виданої публікації.....	80

					КР.КІ.07177/17.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Миколишин П			ПРИСТРІЙ ДВОСТОРОННЬОГО ЗВ'ЯЗКУ ЗГІДНО ПРОТОКОЛУ SIP	Літ.	Арк.	Акрушів
Перевір.		Мельник Г.М.					8	83
Консульт.		Савка Н.Я.				ЗУНУ.ФКІТ. КІ-41		
Н. Контр.		Мельник Г.М.						
Затвердив		Березький О.М.						

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IP	–	Internet Protocol
SIP	–	Session Initiation Protocol
LAN	–	Local area network
VLAN	–	Virtual Local Area Network
MGCP	–	Media Gateway Control Protocol
RTSP	–	Real Time Streaming Protocol
RSVP	–	Reservation Protocol
RTCP	–	Real Time Transport Control Protocol
TCP	–	Transmission Control Protocol
HTTP	–	HyperText Transfer Protocol
GSM	–	Groupe Spécial Mobile
Wi-Fi	–	Wireless Fidelity
USB	–	Universal Serial Bus
ADF	–	Audio Development Framework
IDF	–	IoT Development Framework

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

За останні роки 20 століття розвиток Інтернету приніс багато нових можливостей.

Основною рушійною силою технологічного розвитку є економіка, яка ставить все більше вимог для ІТ-спеціалістів. Однією із сфер, яка відіграє важливу роль є телекомунікація. На даний момент більшість компаній володіють комп'ютеризованими робочими місцями, які пов'язані між собою локальною мережею. Кожна філія, як правило, має широкосмуговий доступ до Інтернету. Спілкування в компаніях відбувається в електронному вигляді, але також традиційним способом - через телефонну мережу PSTN (телефонна комунікаційна мережа загального користування).

Сучасні рішення полягають у заміні аналогового зв'язку на IP-телефонію, яка підключена до пакетної мережі. Відбувається це з метою мінімізації витрат. Традиційна телефонна лінія вимагає власної проводки та телефонної станції для управління мережею. Проте IP-телефонія використовує комп'ютерну мережу і в ролі комутатора може виступати безкоштовний сервер програмного забезпечення, встановлений на існуючому обладнанні.

21 століття пропонує нам вже перевірені сучасні рішення для телефонії, які підтримує Інтернет-мережа. Все більше корпорацій з ІТ-галузі бачать необхідність розвитку комп'ютерних наук. В даний час можна також спостерігати значний розвиток програмування та можливості інвестиції в телефонне обладнання, що співпрацює з пакетними мережами. Є це позитивним явищем не лише для компаній, але й для індивідуальних клієнтів. Компанії, що мають доступ до глобальної мережі, все частіше вирішують використовувати IP-телефонію. Використання такого методу спілкування приносить значні фінансові вигоди. Філії компаній, робота яких полягає в частому спілкуванні з клієнтами (наприклад, call-center), зменшують витрати за телефонні рахунки від постачальників телефонної мережі PSTN. Індивідуальні клієнти також мають змогу зменшити витрати на телефонні дзвінки. Таке

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

рішення є особливо сприятливим для дзвінків на великі відстані, які в традиційній телефонії є значно дорожчі.

В сучасному світі дуже важко уявити своє життя без телефону. Оскільки він є основою для швидкого вирішення будь-яких питань. Тому в IP-телефонії попереду ще багато перспектив.

Єдиною перешкодою для швидкого розвитку є відсутність уніфікованих стандартів. IP-телефонія, як і будь-яка нова галузь, розвивалася в декількох середовищах розробки. Проте зараз ситуація швидко стабілізується, що призводить до досягнення головної мети - інтеграції передачі голосу та даних, створення універсальної мережі.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття IP-телефонії

IP-телефонія (Інтернет-телефонія), яку також називають ІРТ, - це передача телефонних дзвінків через Інтернет. Не має значення, чи передаються дзвінки повністю або частково по пакетній мережі, і які пристрої для цього використовуються (традиційні, ПК, спеціальні пристрої) [3].

IP-телефонія складається з багатьох компонентів. Поєднання їх разом дає позитивні результати у спілкуванні через Інтернет. Використовується весь стек протоколів, кожен з яких відіграє важливу роль. На рисунку 1.1 показаний стек протоколів, що використовується в пакетній мережі телекомунікацій.

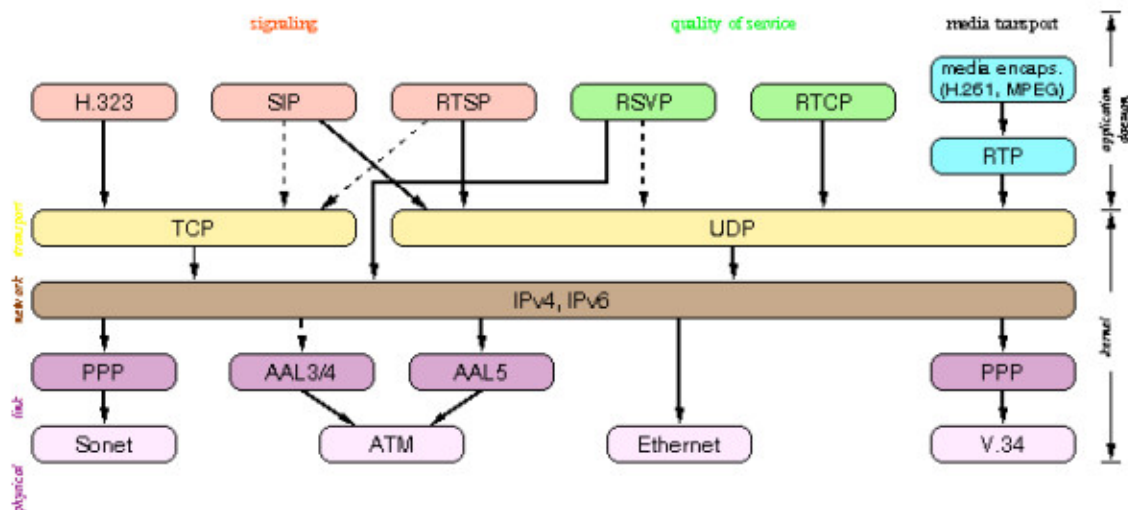


Рисунок 1.1 - Стек протоколів, необхідних для IP-телефонії [7]

Ми розрізняємо протоколи, що належать до кількох груп, кожна з яких виконує певні функції. А саме:

1) Сигналізація

- SIP (Session Initiation Protocol),
- H.323,
- MGCP (Media Gateway Control Protocol),

- H.248 / Megaco,

- RTSP (Real Time Streaming Protocol),

2) Якість обслуговування (Quality of Service)

- RSVP (Reservation Protocol),

- RTCP (Real Time Transport Control Protocol),

3) Зв'язок в реальному часі

- RTP,

- Kodeki,

4) Пакутий зв'язок

- TCP, UDP,

5) Підключення до інтернету

- мережевий рівень моделі ISO / OSI: IPv4, IPv6,

- рівень каналу передачі даних моделі ISO / OSI: PPP, ATM, Ethernet тощо.

6) Інші

- DNS.

Завдання протоколів сигналізації

Протоколи сигналів є найвищими в ієрархії серед стеку протоколів. Вони є логічним елементом мережі, який відповідає за контроль ходу сеансу у зв'язку. До конкретних завдань реалізованих протоколами сигналізації належить:

- визначення IP-адрес та номерів портів, що використовуються для обміну даними

- встановлення зв'язку

Якщо користувач А (абонент) хоче встановити зв'язок з користувачем В, протокол зобов'язаний сигналізувати про вхідний дзвінок користувачеві В, це робиться шляхом надсилання так званого "запрошення". Запрошення є іншою формою для окремих протоколів. Після прийняття запрошення користувачем В, параметри передачі встановлюються і відкриваються відповідні порти для транспортних протоколів.

- Доступність користувача

Протоколи сигналізації повинні мати повідомлення, при їх використанні,

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

визначаючи доступність користувача на даний момент. Це важливо, оскільки недоступний користувач не зможе отримати запрошення на дзвінок.

- Можливості користувача

Дуже важливо встановити параметри, яких можна досягти для двох користувачів, які хочуть спілкуватися. Протокол сигналізації визначає найкращі можливі параметри, які слід здійснити протягом транспортування пакетів з даними.

- Закінчення сесії

Завершення розмови між користувачами також має деякі певні зобов'язання, які повинен виконувати протокол сигналізації. До них відносять закриття портів транспортних протоколів та відправлення повідомлень про припинення розмови.

Огляд існуючих рішень для протоколів сигналізації.

В даний час розробляється багато стандартів сигналізації, і для виробників пристроїв IP-телефонії постає питання, який стандарт підтримувати та впроваджувати у власні продукти. Ситуація стане зрозумілою, коли переважатиме один стандарт, який буде впроваджено на всіх пристроях. Це призведе до розповсюдження пристроїв, які тоді будуть мати не велику ціну і без ризику неадекватних інвестицій.

До поширених протоколів сигналізації відносять RTSP, SIP, H.323.

RTSP (Real-Time Streaming Protocol) - це допоміжний протокол сигналізації, який не виконує критичних дій для IP-телефонії [4]. RTSP контролює ініціювання та передачу мультимедійного потоку з медіасерверів. Це протокол прикладного рівня для управління мультимедійними даними в потокових трансляціях, що запускаються в реальний час. Забезпечує синхронізацію передачі декількох потоків даних і для кожного потоку вибирає відповідний транспортний протокол UDP, TCP або RTP [5]. Основними командами RTSP є:

- SETUP - налаштування з'єднання,
- PLAY - відправка даних,
- RECORD - отримання мультимедійних даних,

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

- PAUSE - перерва в передачі без вивільнення ресурсів,
- TEARDOWN - закінчення передачі та вивільнення ресурсів.

1.2 Телекомунікаційні послуги через IP-мережі

SIP - це стандарт, створений для багатостороннього керування мультимедійними сеансами (MMUSIC) [8]. У ньому багато загальних функцій протоколу HTTP (повідомлення зворотного зв'язку). Протокол SIP все ще існує і вдосконалюється та отримує все більшу підтримку завдяки реалізаціям на пристроях. Архітектура протоколу SIP складається з двох інших протоколів сигналізації: протокол опису сеансу (SDP) та протокол оголошення сеансів (SAP). SDP використовується для опису можливостей та типу переданого носія інформації, реалізованого в терміналі SIP, при цьому SAP застосовується для інформування про багатоадресні сесії, такі як публічні конференції або Інтернет-радіо та телебачення.

H.323 - це стандарт, старший за SIP. Працює над низкою протоколів, що належать до стандарту H.323. [7] Донедавна він мав досить хорошу позицію на ринку телекомунікацій. Але на даний час організація MCE-T все частіше і частіше замінює його протоколом SIP. H.323 є частиною серії, яка включає протоколи для різних типів мереж. Це:

- H.320 для вузькосмугових мереж ISDN,
- H.321 для широкосмугових цифрових мереж ISDN та банкоматів,
- H.322 для пакетних мереж із гарантованою пропускнуою здатністю,
- H.323 для пакетних мереж з негарантованою пропускнуою здатністю,
- H.324 для аналогових мереж POTS (Public Old Telephone Service),

Стандарт H.323 досить складний у впровадженні, бо він використовує багато інших стандартів MCE. Це включає послуги, що стосуються процедур контролю, типу середовища та сигналізації повідомлень. Він базується на протоколах TCP та UDP. TCP використовується для сигналізації та обміну

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

керуючими даними, тоді як інший призначений для передачі даних аудіо, відео та RAS (статус прийому на реєстрацію).

H.248/Megaco

Подвійна назва протоколу не випадкова. Розробку протоколу підтримали організації MCE-T та IETF. Протокол Megaco є наступником MGCP. Він поєднує в собі функції MGCP та H.323. Підтримує різні системи сигналізації мереж з комутацією каналів, включаючи тонову сигналізацію, ISDN, ISUP, QSIG і GSM. Незважаючи на підтримку двох найбільших організацій стандартизація протоколів не набула популярності [10].

1.3 Аналіз технічного завдання та постановка цілей проектування

Для реалізації цілей дипломного проектування та технічного завдання необхідно провести аналіз предметної області, дослідити і налаштувати пристрій для двостороннього зв'язку згідно протоколу SIP. Для досягнення поставленої мети в роботі необхідно вирішити наступні задачі:

- вибрати технології для побудови IP-телефонії;
- провести аналіз побудови мереж інтернет зв'язку;
- розрахувати основні параметри мережі та визначити вимоги до продуктивності мультисервісного вузла доступу;
- вибрати активне мережеве обладнання, сервери та робочі станції та відповідне програмне забезпечення;
- провести імітаційне моделювання пакетної передачі та проаналізувати його результати.

Метою роботи є визначення меж ефективної передачі мовної інформації при реалізації технології в IP-телефонії.

Об'єктом дослідження є пристрій для здійснення двостороннього зв'язку з використанням технології IP-телефонії. Предметом дослідження є телефонна мережа, виконана на основі протоколу SIP.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

2 ПРИСТРІЙ ДВОСТОРОННЬОГО ЗВ'ЯЗКУ ЗГІДНО ПРОТОКОЛУ SIP.

2.1 Структура пристрою VoIP

Система двостороннього зв'язку складається з базової станції (БС), яка, як правило, має постійне живлення та високошвидкісного каналу зв'язку, мікрофона, динаміка та Wi-Fi модуля (рисунок 2.1). В якості БС планується використати одноплатний комп'ютер.

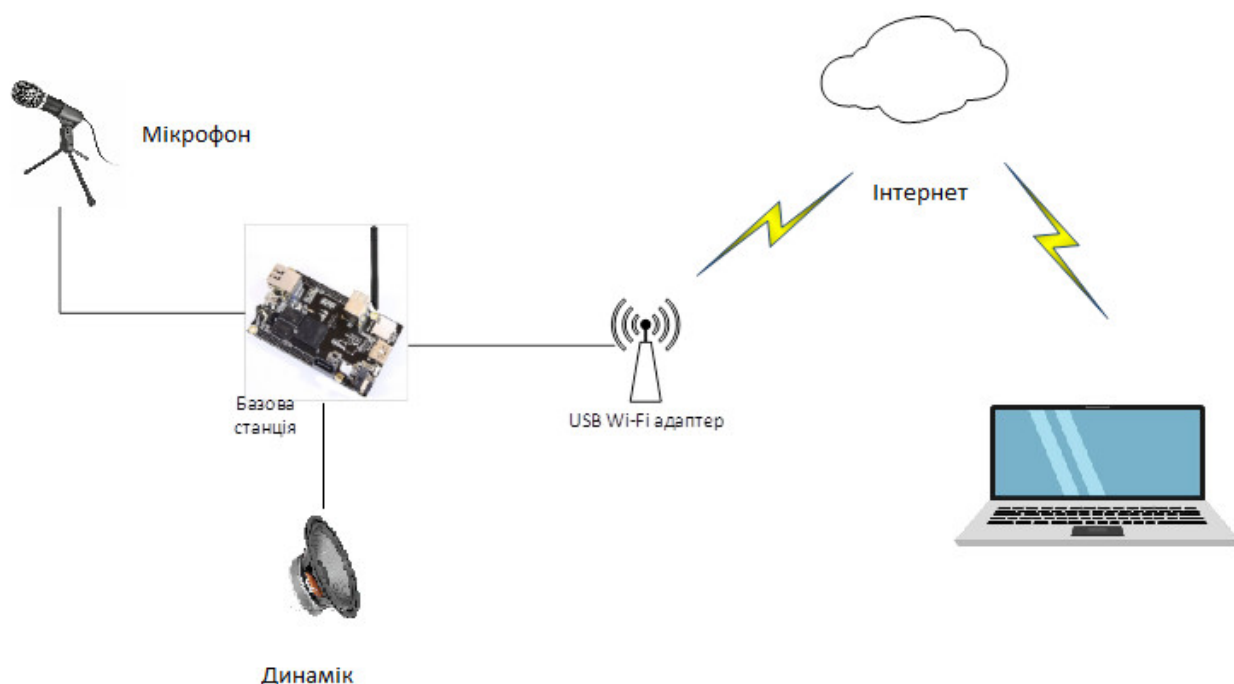


Рисунок 2.1 – Структура системи двостороннього зв'язку

Мікрофон призначений для передачі сигналу на БС з подальшою його обробкою. Мікрофон може розміщуватися як вмонтований в платі, так і під'єднаний до неї. Для окремого живлення плати, можемо використати акумулятор або зарядний пристрій з відповідною напругою.

Базова станція дозволяє проводити попередню обробку мультимедійних даних на місці, зокрема, цифрову обробку, стиснення даних, агрегацію даних, розділення на пакети та відправку на сервер через високошвидкісну безпроводну мережу.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Мікрофон — прилад, що перетворює звукові коливання на коливання сили електричного струму [11]. Мікрофони використовуються у багатьох пристроях, таких як телефони і магнітофони, у звукозаписі та відеозаписі, на радіо і телебаченні, для радіозв'язку, а також для ультразвукового контролю та вимірювання.

Wi-Fi - адаптер призначений для підключення БС до високошвидкісної мережі Internet, для завантаження даних на сервер [8].

2.2 Алгоритм роботи приладу двостороннього зв'язку

Система двостороннього зв'язку складається з базової станції, мікрофона, динаміка та Wi-Fi модуля, які з'єднані між собою за допомогою інтерфейсів.

Інтерфейс - сукупність можливостей, способів і методів взаємодії двох систем, пристроїв або програм для обміну інформацією між ними, певними їх характеристиками, характеристиками з'єднань, сигналів обміну і т. п. Інтерфейси є основою взаємодії всіх сучасних інформаційних систем. Якщо інтерфейс якого-небудь об'єкта (персонального комп'ютера, програми, функції) не змінюється (стабільний, стандартизований), це дає можливість модифікувати сам об'єкт, не перебудовуючи принципи його взаємодії з іншими об'єктами.

У обчислювальній системі взаємодія може здійснюватися на призначеному для користувача, програмному та апаратному рівнях.

В даному модулі пристрої взаємодіють за допомогою такого інтерфейсу, як USB. Розглянемо його більш детально.

USB - послідовний інтерфейс передачі даних для середньо швидкісних і низько швидкісних периферійних пристроїв в обчислювальній техніці [11].

Відповідно, існують поняття: «головний пристрій» (хост, який керує обміном даними через інтерфейс, виступає ініціатором обміну), і «периферійний пристрій» (клієнт, в процесі обміну даними «підпорядковується» хосту) [10].

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Логіка роботи хоста і клієнта принципово відрізняється , відповідно не можна безпосередньо з'єднувати пристрої « хост - хост » і «клієнт - клієнт».

Є спеціальні пристрої - хаби , які підключаються в якості клієнта до одного хосту і , в той же час , виступають хостом для інших периферійних пристроїв. Хаби використовують для «розгалуження» шини USB.

USB дозволяє проводити обмін інформацією з периферійними пристроями на трьох швидкостях (специфікація USB 2.0):

- Низька швидкість (Low Speed - LS) - 1,5 Мбіт / с;
- Повна швидкість (Full Speed - FS) - 12 Мбіт / с;
- Висока швидкість (High Speed - HS) - 480 Мбіт / с.

Для підключення периферійних пристроїв використовується 4-жильний кабель: живлення +5 В, сигнальні проводи D +, D- і GND.

Алгоритм роботи двосторонньої системи зв'язку такий: першим етапом роботи є те, що голос оцифровується. Після цього оцифровані дані аналізуються та обробляються для зменшення фізичного обсягу даних, що надсилаються одержувачу. Як правило, на цьому етапі зникають непотрібні перебої та фоновий шум.

На наступному етапі отримана послідовність даних ділиться на пакети і до неї додається інформація про протокол - адреса одержувача, серійний номер пакета, якщо вони не доставляються послідовно, та додаткові дані для виправлення помилок. Це тимчасове накопичення необхідного обсягу даних, щоб сформувати пакет, перш ніж він буде відправлений безпосередньо в мережу.

Вилучення переданої голосової інформації з отриманих пакетів також відбувається в кілька етапів. Коли голосові пакети доходять до терміналу одержувача, спочатку перевіряється їх порядок. Оскільки IP-мережі не гарантують часу доставки, пакети з вищими порядковими номерами можуть надходити раніше, а часовий інтервал також може змінюватися.

Щоб відновити початкову послідовність та терміни, пакети тимчасово накопичуються. Однак деякі пакети можуть бути повністю загублені під час доставки або їх затримка в доставці може перевищити норму.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

За звичайних обставин приймаючий термінал вимагає повторної передачі помилкових або втрачених даних. Однак передача голосу дуже важлива для часу доставки, тому в цьому випадку ввімкнено алгоритм наближення, який дозволяє приблизно відновити втрачене на основі отриманих пакетів. Або ж ці втрати просто ігноруються, а прогалини заповнюються випадковим чином.

Отримана послідовність даних декомпресується і буде перетворена безпосередньо в звуковий сигнал, що передає голосову інформацію одержувачу.

Таким чином, з великим ступенем ймовірності, отримана інформація не відповідає вихідній (перекручена) і затриманій. Однак в деяких межах надмірність голосової інформації дозволяє миритися з такими втратами.

2.3 Обґрунтування вибору апаратного забезпечення

В якості базової станції може виступати одноплатний комп'ютер. Оскільки всі основні компоненти розміщуються на одній платі, тому він є компактним та може використовуватися в системі двостороннього зв'язку.

Розглянемо декілька сучасних мікрокомп'ютерів.

Плата Olimex ESP32-ADF.

ESP32-ADF - це плата з відкритим кодом для розробки апаратного забезпечення Espressif Audio Development Framework [13]. Це материнська плата, що продається в якості одноплатного комп'ютера, виготовляється в Європі.

Особливості:

- Модуль бездротової мережі - модуль Espressif Systems ESP32-WROVER-B із 8 МБ PSRAM, 4 МБ пам'яті
- Стерео мікрофони
- Стерео динаміки 2x3W з підсилювачем
- Вбудований зарядний пристрій та Li-Po акумулятор

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

- Роз'єм UEXT для 2,8-дюймового РК-дисплея
- Вихідний аудіо вхід
- Мікророз'єм USB для живлення та програмування
- ІЧ-приймач
- 4 сенсорні кнопки
- 3 тактильні кнопки
- Джек живлення
- Розміри: (83 x 52) мм

Живлення:

- 3.3 і 5V на GPIO роз'ємі;
- 5V через мікро USB порт ;
- додатковий вхід 5V через 2х контактні отвори на платі з кроком 2.54

мм;

- 2х 3.3 V регулятора для ESP32 і GPIO

Дана плата зображена на рисунку 2.1.

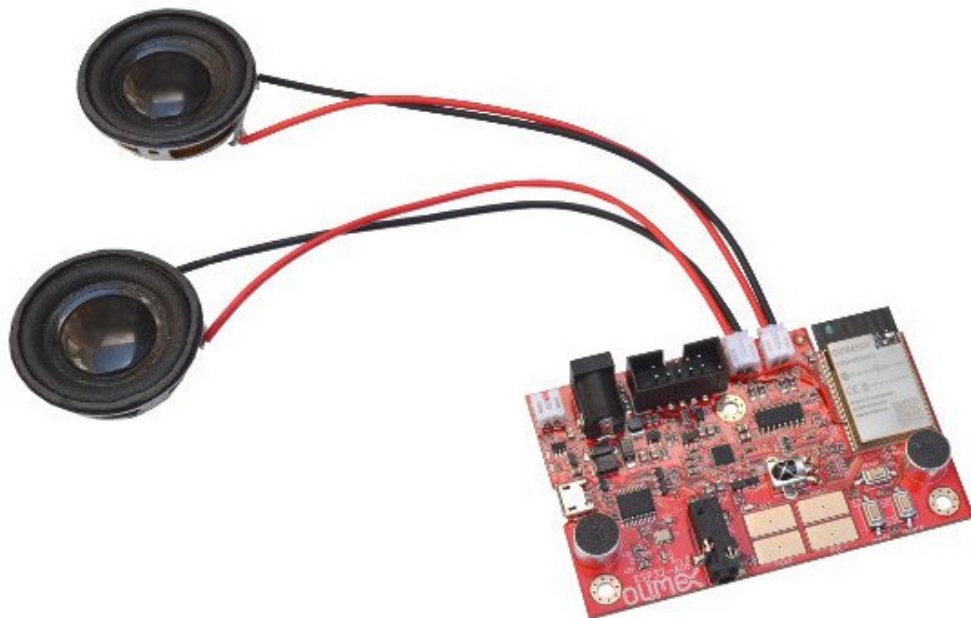


Рисунок 2.2 – Olimex ESP32-ADF [16]

Плата Obsidian ESP32.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Obsidian ESP32 підтримує фреймворк ESP-IDF, ESP32 Arduino Core і MicroPython, в той час як плата була розроблена з використанням KiCad [17].

Технічні характеристики Obsidian ESP32:

- Модуль бездротової мережі - модуль Espressif Systems ESP32-WROVER-B 802.11 b / g / n WiFi + Bluetooth 4.2 / BLE з 16 Мб флеш-пам'яті, 8 Мб PSRAM

- USB - 1x мікро USB порт для живлення і програмування
- Аудіо - 3.5 мм аудіо роз'єм TRS, підключений до ЦАП ESP32

Розширення:

- 40-контактний роз'єм GPIO, сумісний з популярними платами розширення HAT з сигналами харчування (5 В, 3.3 В, GND), I2S для аудіо ЦАП, I2C і SPI, послідовний UART, який доступний через роз'єм або мікро USB-порт, I2C GPIO розширювач з перериваннями і вибором адреси

- 8-контактних отворів з решти введів / висновків, доступних через розширювач GPIO

- 5-контактних отворів з аналоговими входами, що підключається до контактів ESP32 SENSOR_VP і SENSOR_VN

Різне: апаратні кнопки для скидання і вибору варіанта завантаження паралельно з контрольованими сигналами USB.

Живлення:

- 3.3 і 5V на GPIO роз'ємі
- 5V через мікро USB порт
- Додатковий вхід 5V через 2x контактних отвори на платі з кроком 2.54

мм

- 2x 3.3 V регулятора для ESP32 і GPIO

Розмір:

- 65 x 56 мм, відповідає форм-фактору Raspberry Pi Model A і сумісний з корпусами Raspberry Pi.

Дана плата зображена на рисунку 2.3.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

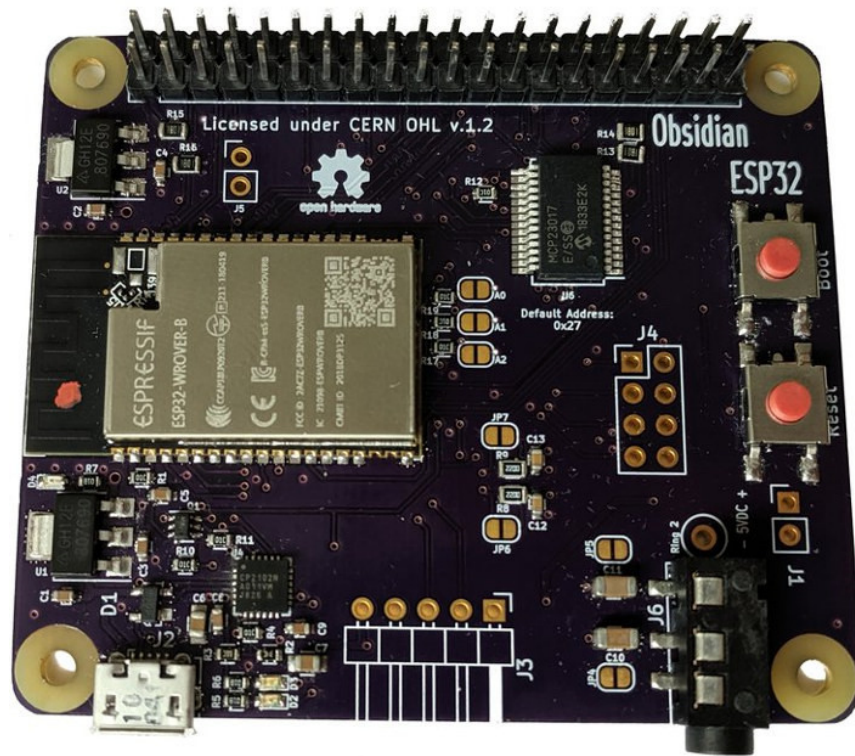


Рисунок 2.3 – Obsidian ESP32 [17]

Плата STM32F407VG Discovery.

Плата розробника STM32F4 DISCOVERY STM32F407VG побудована на базі контролера STM32F407VGT6 з ядром Cortex-M4F, яке можна розігнати до 168 МГц. Об'єм оперативної пам'яті складає 192 кб. Флеш пам'ять програм 1 мб. Є підтримка USB, причому контролер може бути як пристроєм так і хостом. Є апаратний генератор випадкових чисел. Крім звичних інтерфейсів DAC, ADC, SPI, I2C, PWM, RTC у чіпа є ще DCMІ для підключення камери і SDIO для підключення SD карт. Є підтримка Ethernet [18].

На борту розміщений вбудований цифровий мікрофон MP45DT02, стереофонічний 24х бітний ЦАП CS43L22 і вбудованим підсилювачем класу D з регулюванням гучності, тембру і т. і [14].

Також на платі встановлений цифровий акселерометр LIS302DL. Крім стандартних функцій, акселерометр додатково має можливість перемикає діапазони вимірювання прискорення, апаратно обробляти кліки і подвійні кліки, наприклад постукування і генерувати переривання. Є два інтерфейси: SPI або I2C і виходи для переривань. Для управління і контролю на платі розміщені

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

чотири різних світлодіода, розташованих навколо акселерометра, і дві кнопки, одна призначена для користувача, а друга апаратний сброс контролера.

Система живлення реалізована через стабілізатор напруги LD3985M33R. Для початку роботи цілком достатньо 150мА, однак для складних пристроїв з великою кількістю периферії потрібно більш потужний зовнішній стабілізатор 3.3В.

Дана плата зображена на рисунку 2.4.

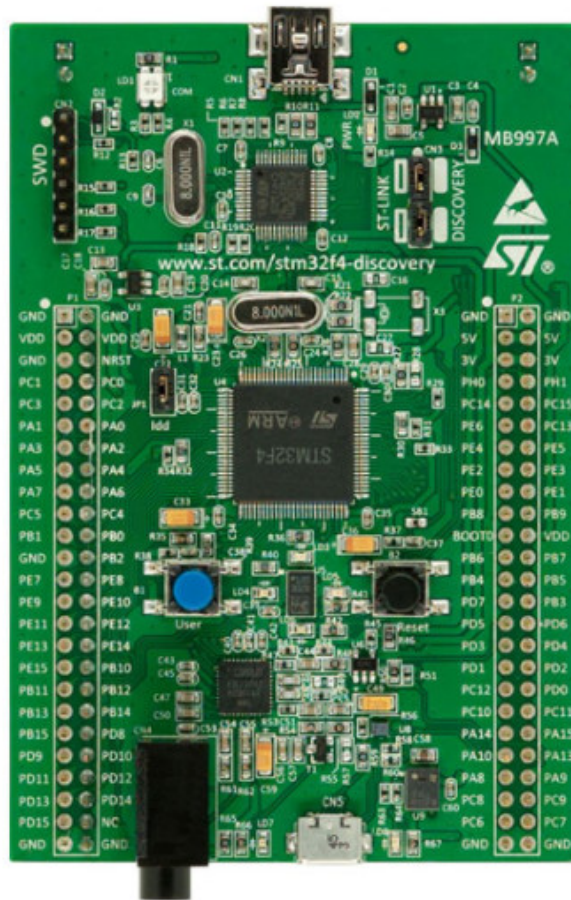


Рисунок 2.4 – STM32F407VG Discovery [18]

Порівняння описаних плат виробників.

Проведений аналіз ринку сучасних мікрокомп'ютерів показав можливість їх використання для обробки мультимедійного контенту в системі двостороннього зв'язку.

Як видно з таблиці 2.1, найбільш перспективним з приведених мікрокомп'ютерів на даний час є плата Olimex ESP32-ADF. Перевагою даного

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

модуля є відносно не висока ціна та висока продуктивність, вбудований мікрофон і можливість підключення дисплею. Технічні характеристики даної моделі дозволяють реалізувати двосторонній зв'язок по протоколу SIP в реальному часі. А оптимізація алгоритмів обробки даних для виконання на одноплатному комп'ютері дозволить скоротити час обробки до мілісекунд, і, відповідно, дозволить значно зменшити затрати пам'яті для збереження даних.

Всі мікрокомп'ютери що досліджувалися та аналізувалися представлені в таблиці 2.1.

Таблиця 2.1 – Детальна характеристика плат

Плата	Ціна \$	CPU	Пам'ять (RAM)
Olimex ESP32-ADF	22	ESP32-WROVER-B	8 Мб PSRAM
Obsidian ESP32	40	ESP32-WROVER-B	8 Мб PSRAM
STM32F407VG Discovery	28	Cortex-M4F	192 кб

3 РЕАЛІЗАЦІЯ І НАЛАШТУВАННЯ ПРИСТРОЮ ДВОСТОРОННЬОГО ЗВ'ЯЗКУ

3.1 Опис та налаштування пристрою на базі ESP32

ESP32 - це система на мікросхемі, яка інтегрує такі функції:

-Процесор: двоядерний (або одноядерний) Xtensa 32-розрядний мікропроцесор LX6, що працює на 160 або 240 МГц і працює на частоті до 600 DMIPS [19].

- Співпроцесор наднизької потужності (ULP)

- Пам'ять: 520 КіБ SRAM

Бездротове підключення:

- Wi-Fi: 802.11 б / г / н

- Bluetooth: v4.2 BR / EDR та BLE (ділиться радіо з Wi-Fi)

Периферійні інтерфейси:

- 12-розрядна SAR ADC до 18 каналів

- 2 × 8-біт ЦАП

- 10 × сенсорні датчики (ємнісне зондування GPIO)

- 4 × SPI

- 2 × I²S інтерфейси

- 2 × I²C інтерфейси

- 3 × UART

- SD/SDIO/CE-ATA/MMC/eMMC контролер хоста

- Ведений контролер SDIO / SPI

- Інфрачервоний пульт дистанційного керування (TX / RX, до 8 каналів)

- Датчик ефекту Холла

Працюючи на технології 40 нм, ESP32 забезпечує надійну, високо інтегровану платформу, яка допомагає задовольнити постійні вимоги щодо ефективного використання енергії, компактної конструкції, безпеки, високої продуктивності та надійності [20].

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Компанія Espressif надає базові апаратні та програмні ресурси, щоб допомогти розробникам програм реалізувати поставлені завдання. Для роботи з ESP32 необхідно використовувати:

-Інструментарій компіляції коду для ESP32.

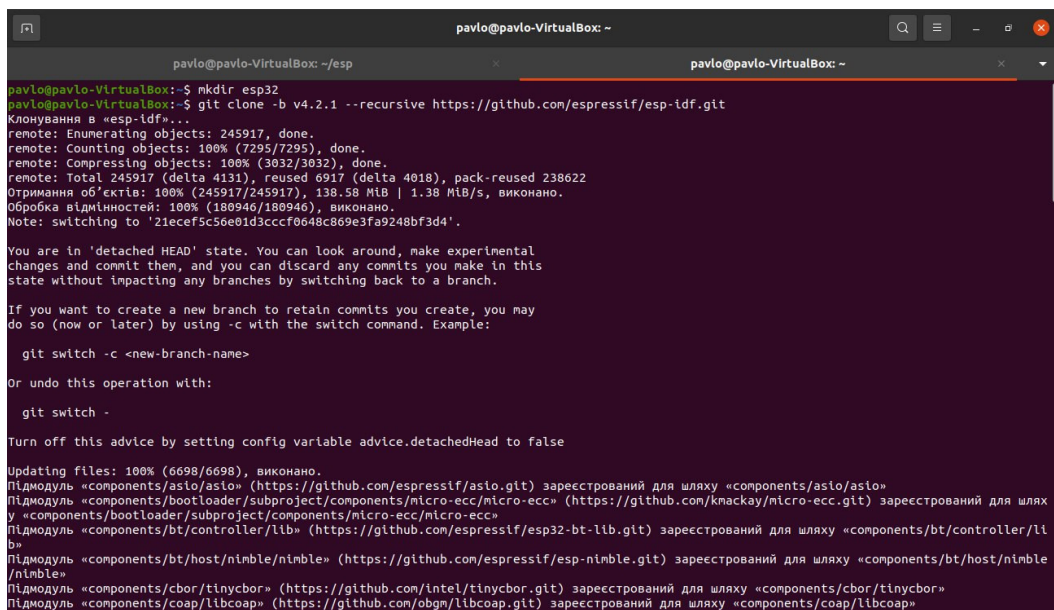
-Інструменти побудови - CMake та Ninja для створення повної програми для ESP32.

-ESP-IDF, який містить API (бібліотеки програмного забезпечення та вихідний код) для ESP32 та сценарії для роботи з ланцюгом інструментів.

Для початку роботи необхідно встановити ESP-IDF використовуючи такі команди:

```
cd ~/esp32
```

```
git clone -b v4.2.1 --recursive https://github.com/espressif/esp-idf.git
```



```
pavlo@pavlo-VirtualBox: ~  
pavlo@pavlo-VirtualBox:~$ mkdir esp32  
pavlo@pavlo-VirtualBox:~$ git clone -b v4.2.1 --recursive https://github.com/espressif/esp-idf.git  
Клонування в «esp-idf»...  
remote: Enumerating objects: 245917, done.  
remote: Counting objects: 100% (7295/7295), done.  
remote: Compressing objects: 100% (3032/3032), done.  
remote: Total 245917 (delta 4131), reused 6917 (delta 4018), pack-reused 238622  
Отримання об'єктів: 100% (245917/245917), 138.58 МБ | 1.38 МБ/с, виконано.  
Обробка відмінностей: 100% (180946/180946), виконано.  
Note: switching to '21ecf5c50e01d3ccc70648c869e3fa9248bf3d4'.  
  
You are in 'detached HEAD' state. You can look around, make experimental  
changes and commit them, and you can discard any commits you make in this  
state without impacting any branches by switching back to a branch.  
  
If you want to create a new branch to retain commits you create, you may  
do so (now or later) by using -c with the switch command. Example:  
  
  git switch -c <new-branch-name>  
  
Or undo this operation with:  
  
  git switch -  
  
Turn off this advice by setting config variable advice.detachedHead to false  
  
Updating files: 100% (6698/6698), виконано.  
Підмодуль «components/asio/asio» (https://github.com/espressif/asio.git) зареєстрований для шляху «components/asio/asio»  
Підмодуль «components/bootloader/subproject/components/micro-ecp/micro-ecp» (https://github.com/knackay/micro-ecp.git) зареєстрований для шляху  
у «components/bootloader/subproject/components/micro-ecp/micro-ecp»  
Підмодуль «components/bt/controller/lib» (https://github.com/espressif/esp32-bt-lib.git) зареєстрований для шляху «components/bt/controller/li  
b»  
Підмодуль «components/bt/host/nimble/nimble» (https://github.com/espressif/esp-nimble.git) зареєстрований для шляху «components/bt/host/nimble  
/nimble»  
Підмодуль «components/cbor/tinycbor» (https://github.com/intel/tinycbor.git) зареєстрований для шляху «components/cbor/tinycbor»  
Підмодуль «components/coap/libcoap» (https://github.com/obgn/libcoap.git) зареєстрований для шляху «components/coap/libcoap»
```

Рисунок 3.1 – Встановлення ESP-IDF

ESP-IDF доступний як набір компонентів для розширення функціональних можливостей, що вже надаються ESP-IDF (Espressif IoT Development Framework).

Для його встановлення вводимо:

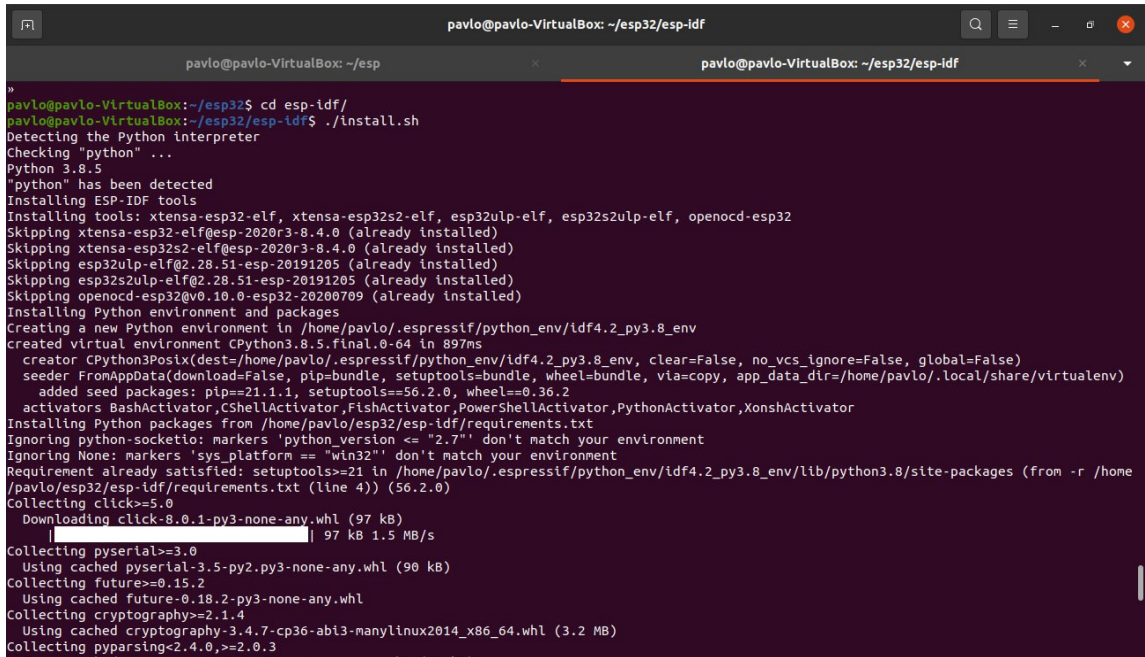
```
git clone --recursive https://github.com/espressif/esp-adf.git
```

Після чого встановлюємо усі необхідні інструменти для роботи ESP-IDF:

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

```
cd ~/esp32/esp-idf
```

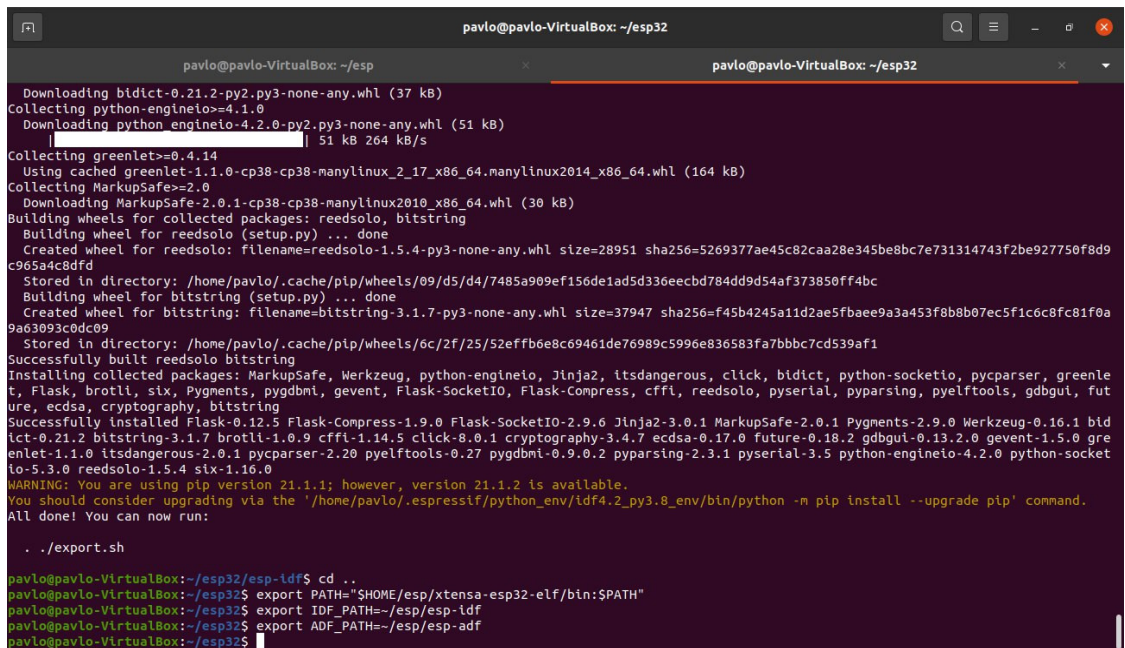
```
./install.sh
```



```
pavlo@pavlo-VirtualBox: ~/esp32/esp-idf
pavlo@pavlo-VirtualBox: ~/esp32/esp-idf$ ./install.sh
Detecting the Python interpreter
Checking "python" ...
Python 3.8.5
"python" has been detected
Installing ESP-IDF tools
Installing tools: xtensa-esp32-elf, xtensa-esp32s2-elf, esp32ulp-elf, esp32s2ulp-elf, openocd-esp32
Skipping xtensa-esp32-elf@esp-2020r3-8.4.0 (already installed)
Skipping xtensa-esp32s2-elf@esp-2020r3-8.4.0 (already installed)
Skipping esp32ulp-elf@2.28.51-esp-20191205 (already installed)
Skipping esp32s2ulp-elf@2.28.51-esp-20191205 (already installed)
Skipping openocd-esp32@v0.10.0-esp32-20200709 (already installed)
Installing Python environment and packages
Creating a new Python environment in /home/pavlo/.espressif/python_env/idf4.2_py3.8_env
created virtual environment CPython3.8.5.final.0-64 in 897ms
creator CPython3Posix(dest=/home/pavlo/.espressif/python_env/idf4.2_py3.8_env, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=21.1.1, setuptools=56.2.0, wheel=0.36.2, via=copy, app_data_dir=/home/pavlo/.local/share/virtualenv)
  added seed packages: pip==21.1.1, setuptools==56.2.0, wheel==0.36.2
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
Installing Python packages from /home/pavlo/esp32/esp-idf/requirements.txt
Ignoring python-socketio: markers 'python_version <= "2.7"' don't match your environment
Ignoring None: markers 'sys_platform == "win32"' don't match your environment
Requirement already satisfied: setuptools==21 in /home/pavlo/.espressif/python_env/idf4.2_py3.8_env/lib/python3.8/site-packages (from -r /home/pavlo/esp32/esp-idf/requirements.txt (line 4)) (56.2.0)
Collecting click==5.0
  Downloading click-8.0.1-py3-none-any.whl (97 kB)
    | 97 kB 1.5 MB/s
Collecting pyserial==3.0
  Using cached pyserial-3.5-py2.py3-none-any.whl (90 kB)
Collecting future==0.15.2
  Using cached future-0.18.2-py3-none-any.whl
Collecting cryptography==2.1.4
  Using cached cryptography-3.4.7-cp36-abi3-manylinux2014_x86_64.whl (3.2 MB)
Collecting pyparsing<2.4.0,>=2.0.3
```

Рисунок 3.2 – Встановлення інструментів ./install.sh

Щоб отримати програмам набору інструментів доступ до ESP-ADF та ESP-IDF за допомогою PATH змінної середовища необхідно вказати шляхи. У рисунку 3.3 наведено використання команд для отримання набору інструментів.



```
pavlo@pavlo-VirtualBox: ~/esp32
pavlo@pavlo-VirtualBox: ~/esp32$ ./export.sh
Downloading bidict-0.21.2-py2.py3-none-any.whl (37 kB)
Collecting python-engineio>=4.1.0
  Downloading python-engineio-4.2.0-py2.py3-none-any.whl (51 kB)
    | 51 kB 264 kB/s
Collecting greenlet>=0.4.14
  Using cached greenlet-1.1.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (164 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.0.1-cp38-cp38-manylinux2010_x86_64.whl (30 kB)
Building wheels for collected packages: reedsolo, bitstring
  Building wheel for reedsolo (setup.py) ... done
  Created wheel for reedsolo: filename=reedsolo-1.5.4-py3-none-any.whl size=28951 sha256=f5269377ae45c82caa28e345be8bc7e731314743f2be927750f8d9c96544c8d9d
  Stored in directory: /home/pavlo/.cache/pip/wheels/09/d5/d4/7485a909ef156de1ad5d336e6bd784dd9d54af373850ff4bc
  Building wheel for bitstring (setup.py) ... done
  Created wheel for bitstring: filename=bitstring-3.1.7-py3-none-any.whl size=37947 sha256=f45b4245a11d2ae5fbaee9a3a453f88b07ec5f1c6c8fc81f0a9a63093c0dc09
  Stored in directory: /home/pavlo/.cache/pip/wheels/6c/2f/25/52effb6e8c69461de76989c5996e836583fa7bbbc7cd539af1
Successfully built reedsolo bitstring
Installing collected packages: MarkupSafe, Werkzeug, python-engineio, Jinja2, itsdangerous, click, bidict, python-socketio, pycparser, greenlet, Flask, brotli, six, Pygments, pygdbmi, gevent, Flask-SocketIO, Flask-Compress, cffi, reedsolo, pyserial, pyparsing, pyelftools, gdbgui, future, ecdsa, cryptography, bitstring
Successfully installed Flask-0.12.5 Flask-Compress-1.9.0 Flask-SocketIO-2.9.6 Jinja2-3.0.1 MarkupSafe-2.0.1 Pygments-2.9.0 Werkzeug-0.16.1 bidict-0.21.2 bitstring-3.1.7 brotli-1.0.9 cffi-1.14.5 click-8.0.1 cryptography-3.4.7 ecdsa-0.17.0 future-0.18.2 gdbgui-0.13.2.0 gevent-1.5.0 greenlet-1.1.0 itsdangerous-2.0.1 pycparser-2.20 pyelftools-0.27 pygdbmi-0.9.0.2 pyparsing-2.3.1 pyserial-3.5 python-engineio-4.2.0 python-socketio-5.3.0 reedsolo-1.5.4 six-1.16.0
WARNING: You are using pip version 21.1.1; however, version 21.1.2 is available.
You should consider upgrading via the '/home/pavlo/.espressif/python_env/idf4.2_py3.8_env/bin/python -m pip install --upgrade pip' command.
All done! You can now run:
  ./export.sh
pavlo@pavlo-VirtualBox:~/esp32/esp-idf$ cd ..
pavlo@pavlo-VirtualBox:~/esp32$ export PATH="$HOME/esp/xtensa-esp32-elf/bin:$PATH"
pavlo@pavlo-VirtualBox:~/esp32$ export IDF_PATH=~/esp/esp-idf
pavlo@pavlo-VirtualBox:~/esp32$ export ADF_PATH=~/esp/esp-adf
pavlo@pavlo-VirtualBox:~/esp32$
```

Рисунок 3.3 – Встановлення шляхів до файлів проекту

					Арк.
					27
Змн.	Арк.	№ докум.	Підпис	Дата	

КР.КІ.071777/17.00.00.000 ПЗ

Після завершення завантаження файлів в проєкті створюємо папку VoIP у якій буде міститися програмна частина для здійснення IP телефонії.

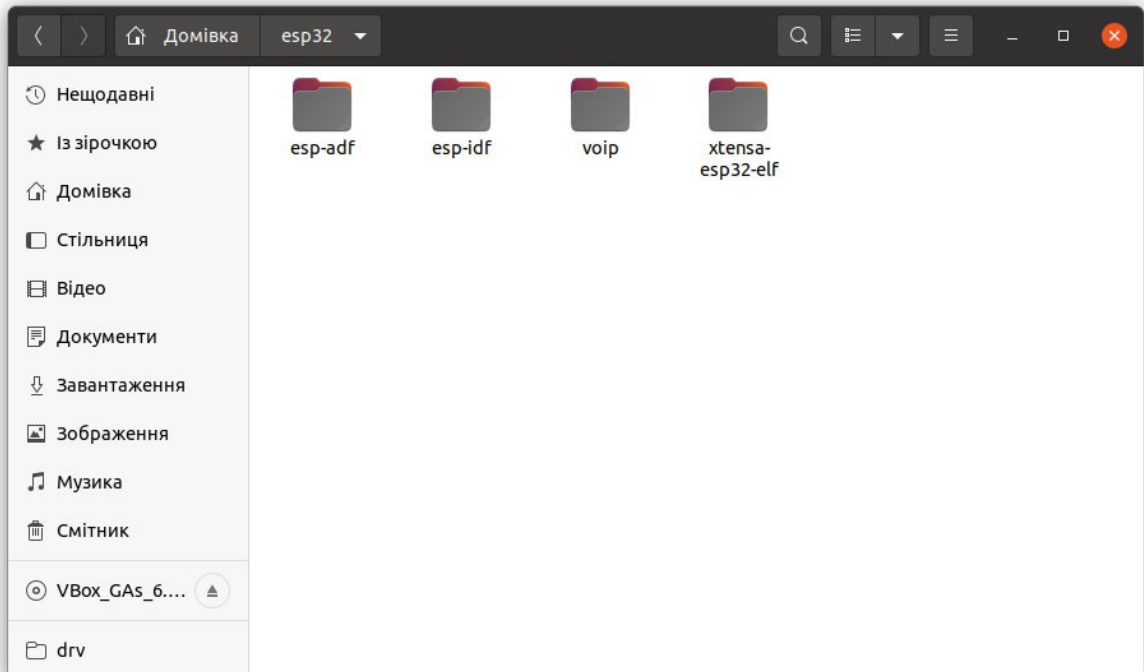


Рисунок 3.4 – Вміст файлів проєкту

3.2 Розробка основних функцій

При розробці коду, були використані такі бібліотеки, які дозволяють підключити стандартні модулі для роботи з інтернетом, flash пам'ятю, аудіо, системними файлами, кодуванням і декодуванням сигналу.

```
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "esp_log.h"
#include "nvs_flash.h"
#include "audio_element.h"
#include "audio_pipeline.h"
#include "audio_event_iface.h"
#include "audio_common.h"
```

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

```

#include "audio_mem.h"
#include "i2s_stream.h"
#include "esp_peripherals.h"
#include "periph_wifi.h"
#include "board.h"
#include "input_key_service.h"
#include "raw_stream.h"
#include "filter_resample.h"
#include "esp_sip.h"
#include "g711_decoder.h"
#include "g711_encoder.h"
#include "algorithm_stream.h"
#include "wifi_service.h"
#include "smart_config.h"
#include "audio_tone_uri.h"
#include "audio_player_int_tone.h"

```

Для встановлення зв'язку необхідно під'єднатися до мережі Інтернет, тому реалізуємо це у кодї де методом case перебираються варіанти встановлення з'єднання.

```

static int _sip_event_handler(sip_event_msg_t *event){
    ip4_addr_t ip;
    switch ((int)event->type) {
        case SIP_EVENT_REQUEST_NETWORK_STATUS:
            ESP_LOGD(TAG, "SIP_EVENT_REQUEST_NETWORK_STATUS");
            ip = _get_network_ip();
            if (ip.addr) {
                return true;}
            return ESP_OK;
        case SIP_EVENT_REQUEST_NETWORK_IP:
            ESP_LOGD(TAG, "SIP_EVENT_REQUEST_NETWORK_IP");
            ip = _get_network_ip();
            int ip_len = sprintf((char *)event->data, "%s",
ip4addr_ntoa(&ip));
            return ip_len;
        case SIP_EVENT_REGISTERED:
            ESP_LOGI(TAG, "SIP_EVENT_REGISTERED");

audio_player_int_tone_play(tone_uri[TONE_TYPE_SERVER_CONNECT]);
            break;
        case SIP_EVENT_RINGING:
            ESP_LOGI(TAG, "ringing... RemotePhoneNum %s", (char
*)event->data);
            audio_player_int_tone_play(tone_uri[TONE_TYPE_ALARM]);
            break;
        case SIP_EVENT_INVITING:
            ESP_LOGI(TAG, "SIP_EVENT_INVITING Remote Ring...");
            break;
        case SIP_EVENT_BUSY:

```

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

```

        ESP_LOGI(TAG, "SIP_EVENT_BUSY");
        break;
    case SIP_EVENT_HANGUP:
        ESP_LOGI(TAG, "SIP_EVENT_HANGUP");
        break;
    case SIP_EVENT_AUDIO_SESSION_BEGIN:
        ESP_LOGI(TAG, "SIP_EVENT_AUDIO_SESSION_BEGIN");
        player_pipeline_open();
        recorder_pipeline_open();
        break;
    case SIP_EVENT_AUDIO_SESSION_END:
        ESP_LOGI(TAG, "SIP_EVENT_AUDIO_SESSION_END");
        audio_pipeline_stop(player);
        audio_pipeline_wait_for_stop(player);
        audio_pipeline_deinit(player);
        audio_pipeline_stop(recorder);
        audio_pipeline_wait_for_stop(recorder);
        audio_pipeline_deinit(recorder);
        break;
    case SIP_EVENT_READ_AUDIO_DATA:
        return raw_stream_read(raw_read, (char *)event->data,
event->data_len);
    case SIP_EVENT_WRITE_AUDIO_DATA:
        return raw_stream_write(raw_write, (char *)event-
>data, event->data_len);
    case SIP_EVENT_READ_DTMF:
        ESP_LOGI(TAG, "SIP_EVENT_READ_DTMF ID : %d ", ((char
*)event->data)[0]);
        break;}
return 0;}

```

Методом `recorder_pipeline_open` здійснюємо запис звуку і його подальше надсилання користувачеві через сервер:

```

static esp_err_t recorder_pipeline_open(){
    audio_element_handle_t i2s_stream_reader;
    audio_pipeline_cfg_t pipeline_cfg =
DEFAULT_AUDIO_PIPELINE_CONFIG();
    recorder = audio_pipeline_init(&pipeline_cfg);
    AUDIO_NULL_CHECK(TAG, recorder, return ESP_FAIL);
    i2s_stream_cfg_t i2s_cfg = I2S_STREAM_CFG_DEFAULT();
    i2s_cfg.type = AUDIO_STREAM_READER;
    i2s_cfg.uninstall_drv = false;
#ifdef CONFIG_ESP_LYRAT_MINI_V1_1_BOARD
    i2s_cfg.i2s_port = 1;
    i2s_cfg.task_core = 1;
#endif
    i2s_cfg.i2s_config.sample_rate = I2S_SAMPLE_RATE;
    i2s_stream_reader = i2s_stream_init(&i2s_cfg);
#ifdef CONFIG_ESP_LYRAT_MINI_V1_1_BOARD

```

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

```

    algorithm_stream_cfg_t          algo_config          =
ALGORITHM_STREAM_CFG_DEFAULT();
    algo_config.input_type = ALGORITHM_STREAM_INPUT_TYPE1;
    algo_config.task_core = 1;
    audio_element_handle_t          element_algo        =
algo_stream_init(&algo_config);
#endif
    rsp_filter_cfg_t rsp_cfg = DEFAULT_RESAMPLE_FILTER_CONFIG();
    rsp_cfg.src_rate = I2S_SAMPLE_RATE;
#if defined CONFIG_ESP_LYRAT_MINI_V1_1_BOARD
    rsp_cfg.src_ch = 1;
#else
    rsp_cfg.src_ch = I2S_CHANNELS;
#endif
    rsp_cfg.dest_rate = CODEC_SAMPLE_RATE;
    rsp_cfg.dest_ch = CODEC_CHANNELS;
    rsp_cfg.complexity = 5;
    rsp_cfg.task_core = 1;
    audio_element_handle_t filter = rsp_filter_init(&rsp_cfg);
    g711_encoder_cfg_t g711_cfg = DEFAULT_G711_ENCODER_CONFIG();
    g711_cfg.task_core = 1;
    audio_element_handle_t          sip_encoder         =
g711_encoder_init(&g711_cfg);
    raw_stream_cfg_t raw_cfg = RAW_STREAM_CFG_DEFAULT();
    raw_cfg.type = AUDIO_STREAM_READER;
    raw_read = raw_stream_init(&raw_cfg);
    audio_element_set_output_timeout(raw_read, portMAX_DELAY);
    audio_pipeline_register(recorder, i2s_stream_reader, "i2s");
    audio_pipeline_register(recorder, filter, "filter");
    audio_pipeline_register(recorder, sip_encoder, "sip_enc");
    audio_pipeline_register(recorder, raw_read, "raw");
#if defined CONFIG_ESP_LYRAT_MINI_V1_1_BOARD
    audio_pipeline_register(recorder, element_algo, "algo");
    algo_stream_set_record_rate(element_algo,          I2S_CHANNELS,
I2S_SAMPLE_RATE);
    const char *link_tag[5] = {"i2s", "algo", "filter", "sip_enc",
"raw"};
    audio_pipeline_link(recorder, &link_tag[0], 5);
#else
    const char *link_tag[4] = {"i2s", "filter", "sip_enc", "raw"};
    audio_pipeline_link(recorder, &link_tag[0], 4);
#endif
#endif
}

```

Отримання звуку зі сервера і його подальша обробка

```

static esp_err_t player_pipeline_open(){
    audio_element_handle_t i2s_stream_writer;
    audio_pipeline_cfg_t          pipeline_cfg          =
DEFAULT_AUDIO_PIPELINE_CONFIG();
    player = audio_pipeline_init(&pipeline_cfg);
    AUDIO_NULL_CHECK(TAG, player, return ESP_FAIL);
    raw_stream_cfg_t raw_cfg = RAW_STREAM_CFG_DEFAULT();

```

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

```

raw_cfg.type = AUDIO_STREAM_WRITER;
raw_write = raw_stream_init(&raw_cfg);
g711_decoder_cfg_t g711_cfg = DEFAULT_G711_DECODER_CONFIG();
audio_element_handle_t sip_decoder =
g711_decoder_init(&g711_cfg);
rsp_filter_cfg_t rsp_cfg = DEFAULT_RESAMPLE_FILTER_CONFIG();
rsp_cfg.src_rate = CODEC_SAMPLE_RATE;
rsp_cfg.src_ch = CODEC_CHANNELS;
rsp_cfg.dest_rate = I2S_SAMPLE_RATE;
rsp_cfg.dest_ch = I2S_CHANNELS;
rsp_cfg.complexity = 5;
audio_element_handle_t filter = rsp_filter_init(&rsp_cfg);
i2s_stream_cfg_t i2s_cfg = I2S_STREAM_CFG_DEFAULT();
i2s_cfg.type = AUDIO_STREAM_WRITER;
i2s_cfg.uninstall_drv = false;
i2s_cfg.i2s_config.sample_rate = I2S_SAMPLE_RATE;
i2s_stream_writer = i2s_stream_init(&i2s_cfg);
audio_pipeline_register(player, i2s_stream_writer, "i2s");
const char *link_tag[4] = {"raw", "sip_dec", "filter", "i2s"};
audio_pipeline_link(player, &link_tag[0], 4);
audio_pipeline_run(player);
ESP_LOGI(TAG, "SIP player has been created");
return ESP_OK;}

```

Для встановлення зв'язку використовується протокол SIP. Він надасть змогу ініціювати сигнали і встановити з'єднання між двома користувачами.

Після створення коду, здійснюємо його компіляцію у терміналі.

```

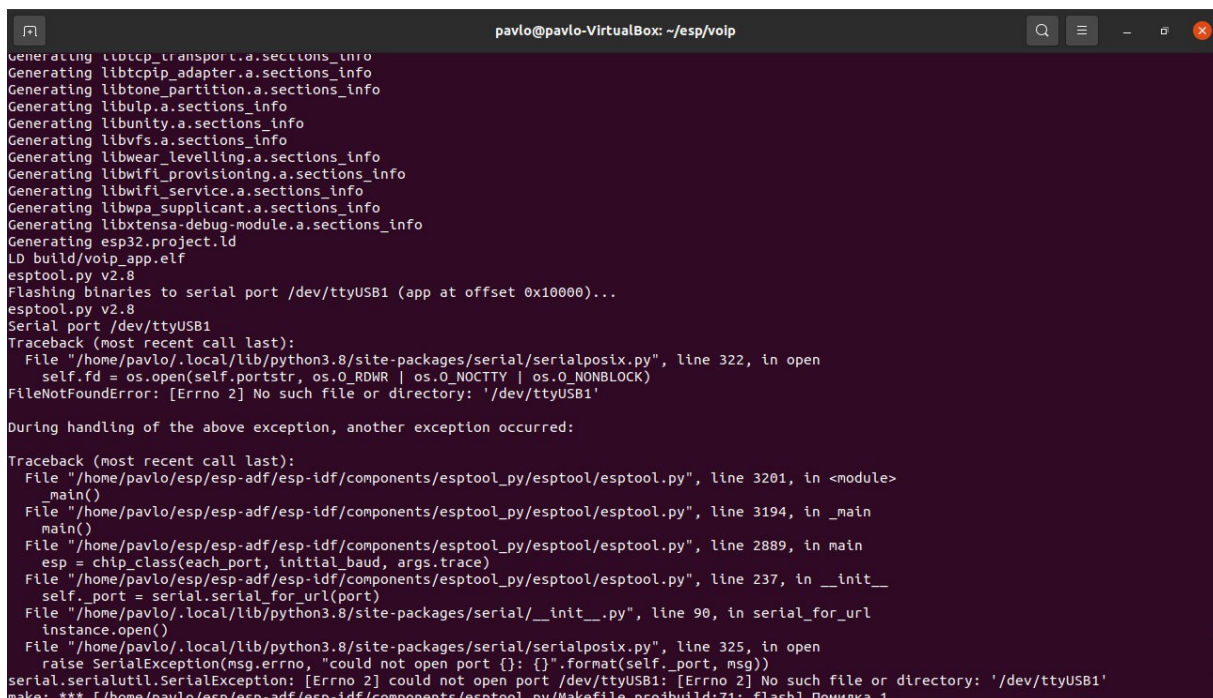
pavlo@pavlo-VirtualBox: ~/esp32/voip$ make flash
Toolchain path: /home/pavlo/esp32/xtensa-esp32-elf/bin/xtensa-esp32-elf-gcc
WARNING: Toolchain version is not supported: crosstool-ng-1.22.0-97-gc752ad5
Expected to see version: esp-2020r3
Please check ESP-IDF setup instructions and update the toolchain, or proceed at your own risk.
WARNING: Compiler version is not supported: 5.2.0
Expected to see version(s): 8.4.0
Please check ESP-IDF setup instructions and update the toolchain, or proceed at your own risk.
Python requirements from /home/pavlo/esp32/esp-idf/requirements.txt are satisfied.
Project is not inside a git repository, or git repository has no commits
will not use 'git describe' to determine PROJECT_VER.
App "voip_app" version: 1
WARNING: Toolchain version is not supported: crosstool-ng-1.22.0-97-gc752ad5
Expected to see version: esp-2020r3
Please check ESP-IDF setup instructions and update the toolchain, or proceed at your own risk.
WARNING: Compiler version is not supported: 5.2.0
Expected to see version(s): 8.4.0
Please check ESP-IDF setup instructions and update the toolchain, or proceed at your own risk.
CC build/bootloader/bootloader_support/src/bootloader_efuse_esp32.o
CC build/bootloader/bootloader_support/src/bootloader_flash.o
CC build/bootloader/bootloader_support/src/flash_qio_mode.o
CC build/bootloader/bootloader_support/src/bootloader_init.o
CC build/bootloader/bootloader_support/src/bootloader_random_esp32.o
CC build/bootloader/bootloader_support/src/bootloader_mem.o
CC build/bootloader/bootloader_support/src/bootloader_utility.o
CC build/bootloader/bootloader_support/src/bootloader_common.o
CC build/bootloader/bootloader_support/src/bootloader_clock.o
CC build/bootloader/bootloader_support/src/bootloader_flash_config_esp32.o
CC build/bootloader/bootloader_support/src/flash_partitions.o
CC build/bootloader/bootloader_support/src/esp_image_format.o
CC build/bootloader/bootloader_support/src/flash_encrypt.o
CC build/bootloader/bootloader_support/src/bootloader_random.o
CC build/bootloader/bootloader_support/src/esp32/bootloader_sha.o
CC build/bootloader/bootloader_support/src/esp32/bootloader_esp32.o
CC build/bootloader/bootloader_support/src/esp32/flash_encrypt.o
AR build/bootloader/bootloader_support/libbootloader_support.a
CC build/bootloader/log/log.o
CC build/bootloader/log/log_noos.o

```

Рисунок 3.5 – Компіляція коду проекту

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

При успішному складанні проекту відбувається спроба з'єднання з платою.



```
Generating libtcp_transport.a.sections_info
Generating libtcp_adapter.a.sections_info
Generating libtone_partition.a.sections_info
Generating libulp.a.sections_info
Generating libunity.a.sections_info
Generating libvfs.a.sections_info
Generating libwear_levelling.a.sections_info
Generating libwifi_provisioning.a.sections_info
Generating libwifi_service.a.sections_info
Generating libwpa_supplicant.a.sections_info
Generating libxtensa-debug-module.a.sections_info
Generating esp32.project.ld
LD build/voip_app.elf
esptool.py v2.8
Flashing binaries to serial port /dev/ttyUSB1 (app at offset 0x10000)...
esptool.py v2.8
Serial port /dev/ttyUSB1
Traceback (most recent call last):
  File "/home/pavlo/.local/lib/python3.8/site-packages/serial/serialposix.py", line 322, in open
    self.fd = os.open(self.portstr, os.O_RDWR | os.O_NOCTTY | os.O_NONBLOCK)
FileNotFoundError: [Errno 2] No such file or directory: '/dev/ttyUSB1'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/pavlo/esp/esp-adf/esp-idf/components/esptool_py/esptool/esptool.py", line 3201, in <module>
    _main()
  File "/home/pavlo/esp/esp-adf/esp-idf/components/esptool_py/esptool/esptool.py", line 3194, in _main
    main()
  File "/home/pavlo/esp/esp-adf/esp-idf/components/esptool_py/esptool/esptool.py", line 2889, in main
    esp = chip_class(each_port, initial_baud, args.trace)
  File "/home/pavlo/esp/esp-adf/esp-idf/components/esptool_py/esptool/esptool.py", line 237, in __init__
    self._port = serial.serial_for_url(port)
  File "/home/pavlo/.local/lib/python3.8/site-packages/serial/_init_.py", line 90, in serial_for_url
    instance.open()
  File "/home/pavlo/.local/lib/python3.8/site-packages/serial/serialposix.py", line 325, in open
    raise SerialException(msg.errno, "could not open port {}: {}".format(self._port, msg))
serial.serialutil.SerialException: [Errno 2] could not open port /dev/ttyUSB1: [Errno 2] No such file or directory: '/dev/ttyUSB1'
make: *** [/home/pavlo/esp/esp-adf/esp-idf/components/esptool_py/Makefile.projbuild:71: flash] Помилка 1
```

Рисунок 3.6 – Результат успішної компіляції

Наступним етапом роботи є налаштування сервера для встановлення з'єднання між двома користувачами.

3.2 Підбір і налаштування сервера

Asterisk - це повністю функціональна IP-телефонна станція, так звана IP АТС (Private Branch Exchange). Ініціатором та творцем цього рішення став Марк Спенсер. Дизайн був написаний на мові C і спочатку використовувався для Linux [25]. В даний час існують варіанти, які працюють для більшості операційних систем.

Asterisk підтримує ряд протоколів сигналізації:

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

- в мережі IP: протокол SIP, H.323, MGCP, CISCO SCCP і призначений для необхідності зв'язку з сервером Asterisk, протоколом Inter Exchange Asterisk;
- у мережах TDM: SS7 та DSS1.

Для управління сервером використовуємо FreePBX. FreePBX - це графічний інтерфейс користувача (GUI) з відкритим вихідним кодом, призначений для управління Asterisk. На сайті FreePBX завантажуюмо на комп'ютер останню версію програми.

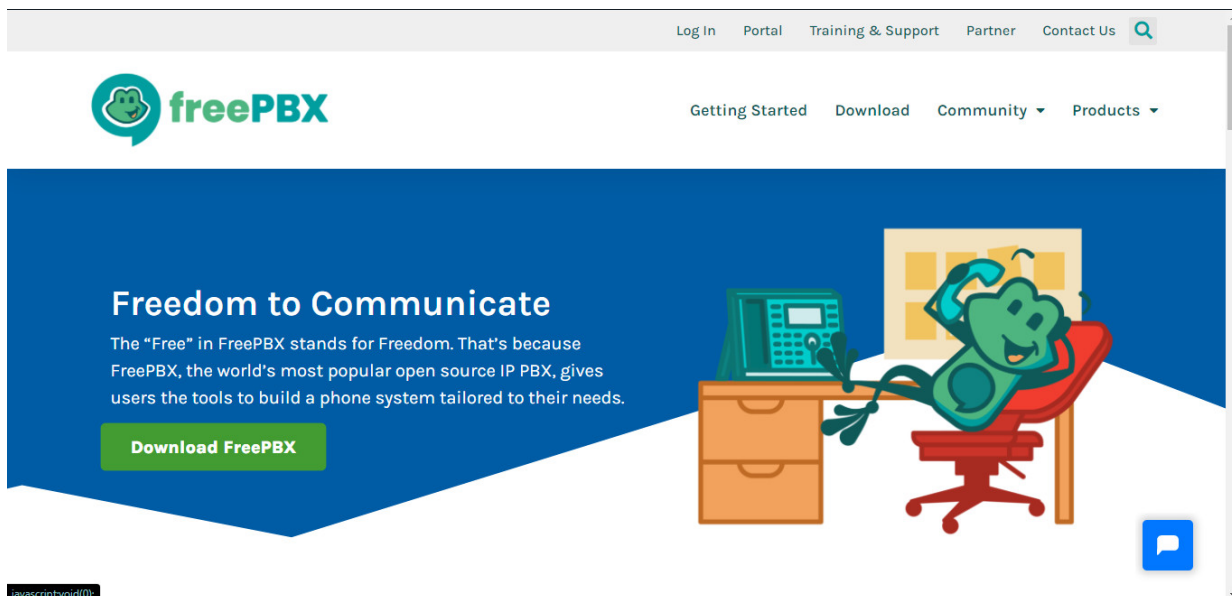


Рисунок 3.7 – Сайт FreePBX

Для роботи з веб інтерфейсом FreePBX використовуємо програму VirtualBox.

VirtualBox — це програма віртуалізації для операційних систем, розроблена німецькою фірмою Innotek, зараз вона належить Oracle Corporation. Вона встановлюється на наявну операційну систему, яка називається хостовою, усередину цієї програми встановлюється інша операційна система, яку називають гостьовою операційною системою [30].

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Підтримується основними операційними системами Linux, FreeBSD, Mac OS X, OS/2 Warp, Microsoft Windows, які підтримують роботу гостьових операційних систем FreePBX, Linux, OpenBSD,, Windows і Solaris [35].

У вікні програми створюємо і імпортуємо скачаний образ FreePBX. Після чого запускаємо сервер.

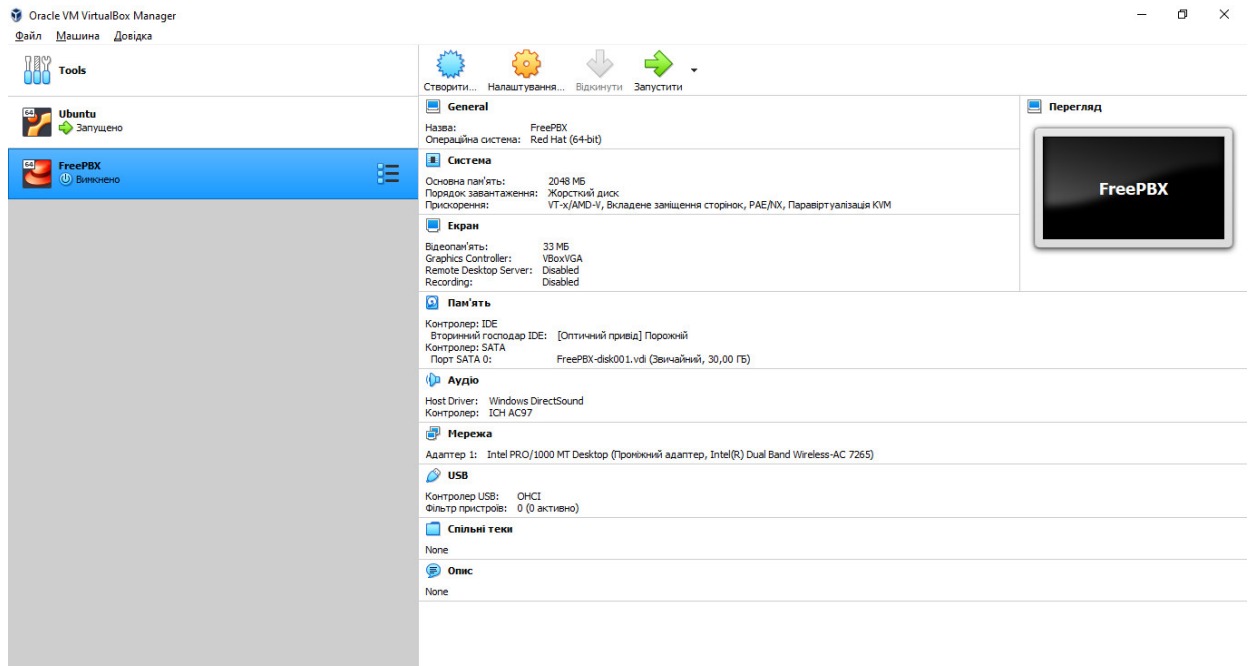


Рисунок 3.8– VirtualBox

При завантаженні системи, сервер вимагає вказати логін і пароль для входу для входу. Стандартні дані для входу це root : helloasterisk.

У разі успішної авторизації, відбувається запуск сервера, де вказується IP адреса, за якою ми можемо потрапити у головний кабінет для здійснення подальших налаштувань системи.

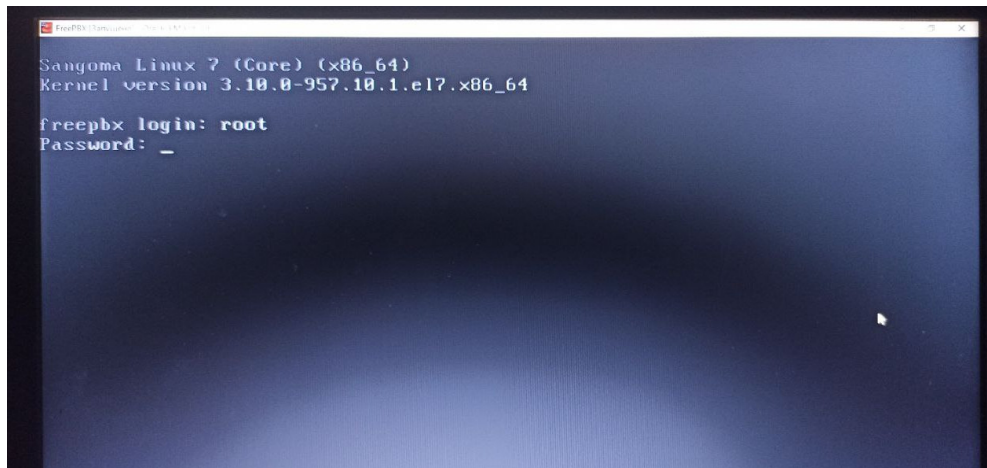


Рисунок 3.9 – Сервер FreePBX

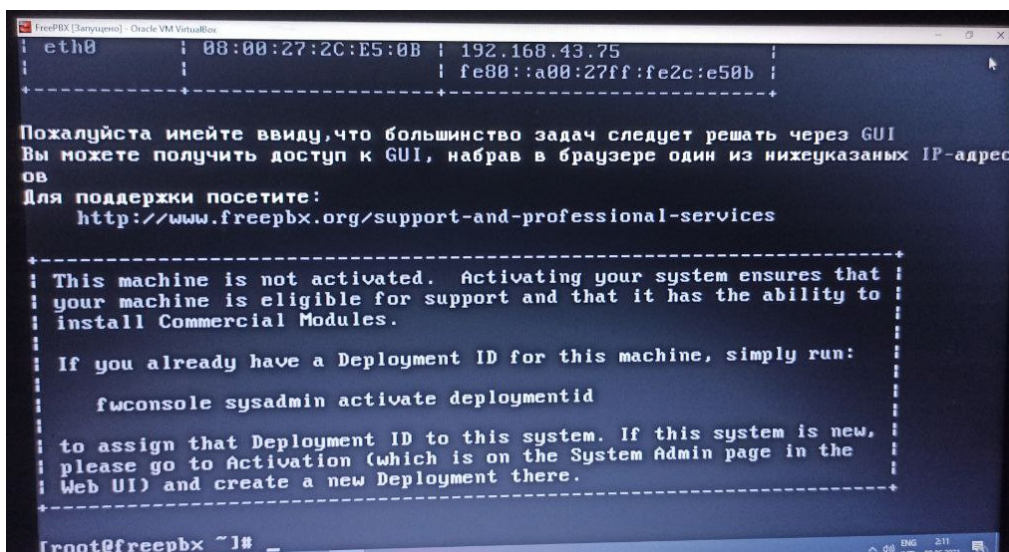


Рисунок 3.10 – Сервер FreePBX

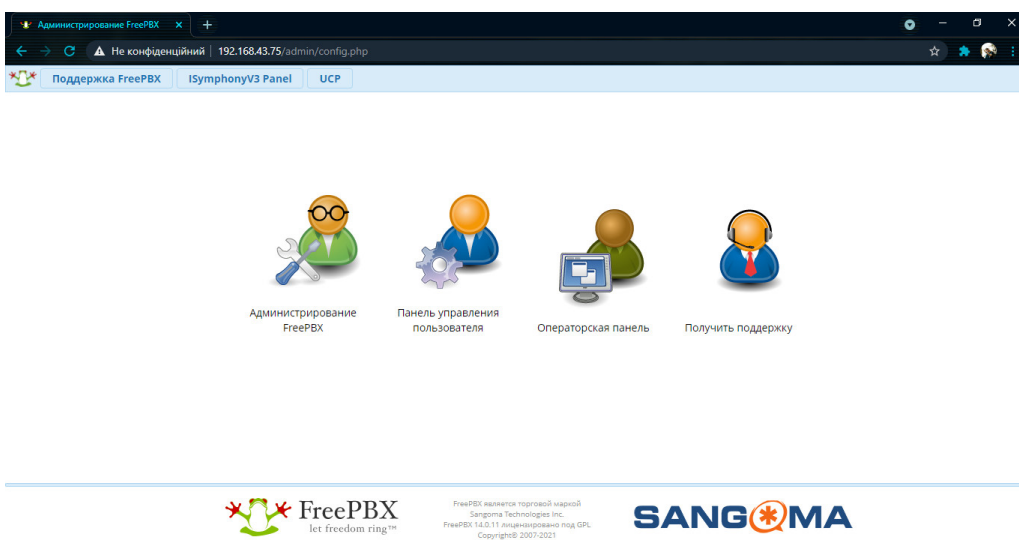


Рисунок 3.11 – Вхід у адмін панель

					КР.КІ.071777/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Щоб здійснити дзвінок між двома користувачами потрібно налаштувати внутрішні номери, де кожному абоненту призначається відповідний ідентифікатор в системі за яким ми його можемо його викликати.

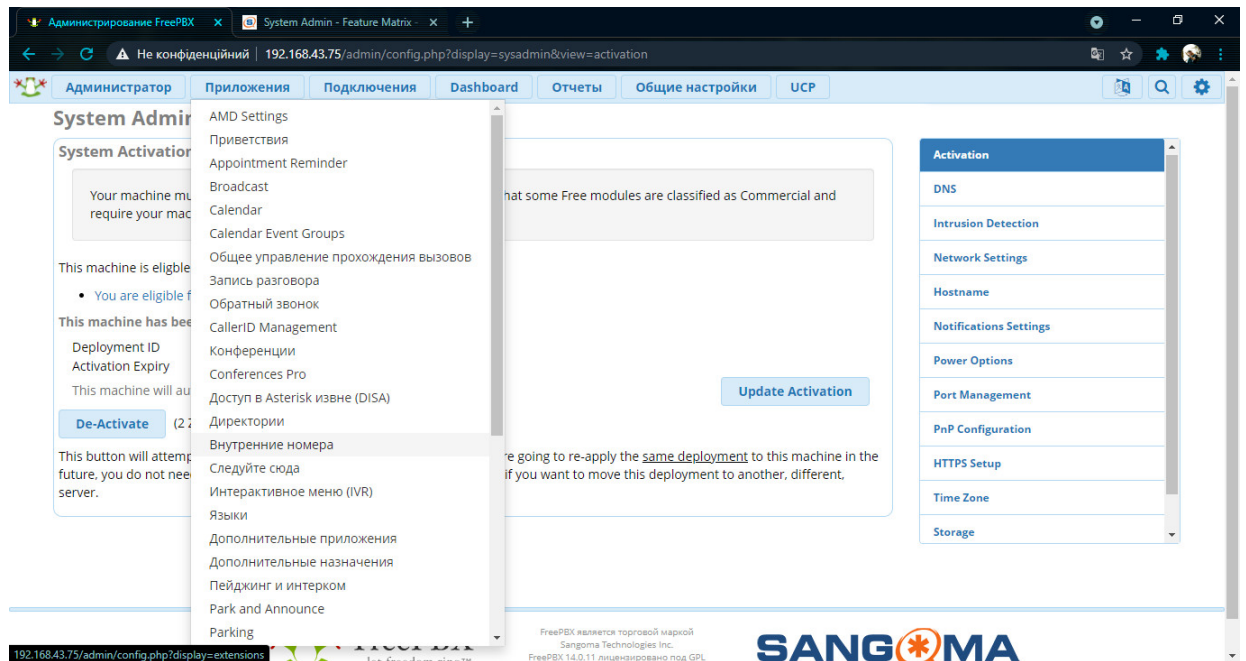


Рисунок 3.12 –Адмін панель

Для добавлення внутрішнього номера вибираємо вкладку Chan_SIP.

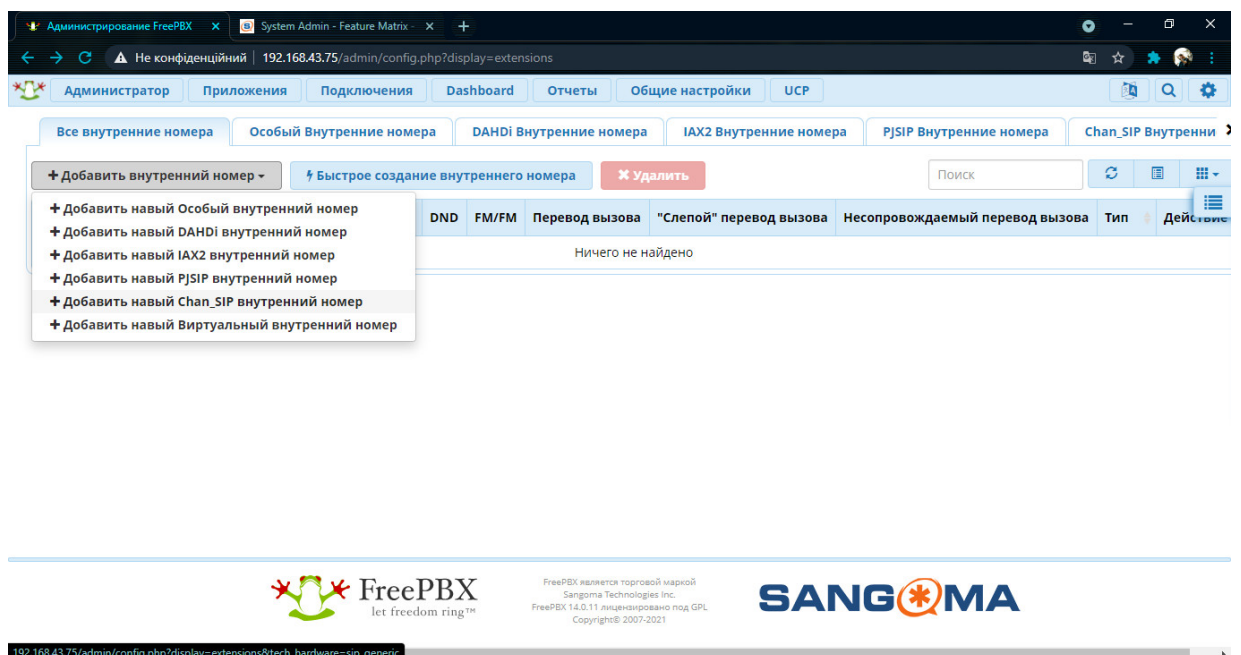


Рисунок 3.13 – Добавлення внутрішнього номера

При створенні внутрішнього номера система автоматично створює унікальний пароль для користувача, який в подальшому необхідно вказати при роботі з сервером.

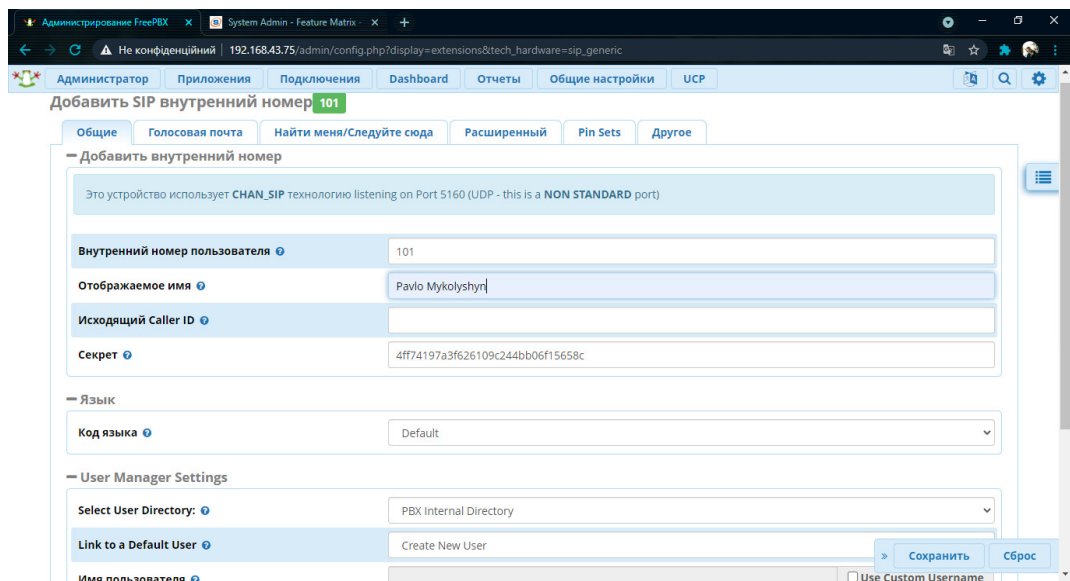


Рисунок 3.14 – Додавання внутрішнього номера

Також створимо номер для другого користувача.

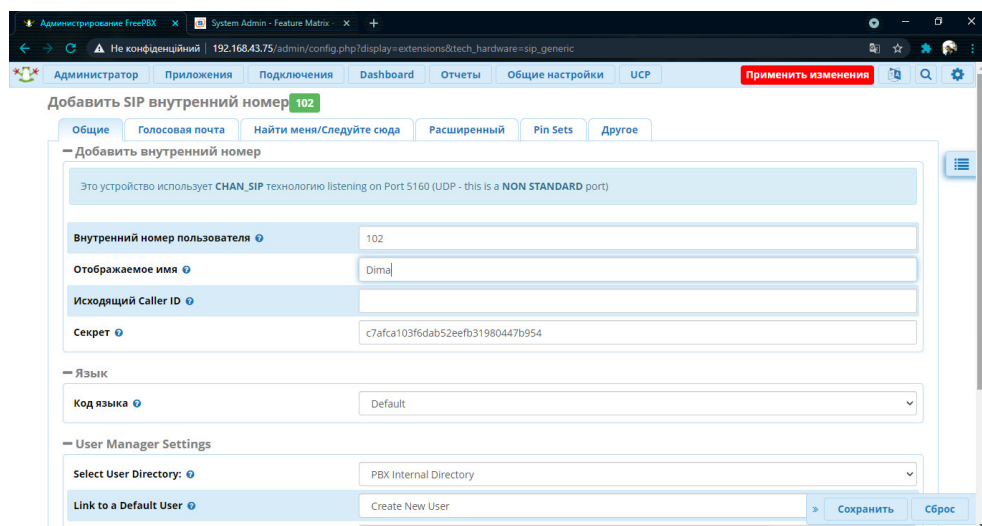


Рисунок 3.15 – Додавання внутрішнього номера для другого клієнта

Головним параметром при створенні внутрішнього номера, це вказати номер порта, за яким користувач має під'єднатися до сервера. Для цього вказуємо стандартний номер 5060. При успішному створенні номерів на

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

сторінці будуть відображатися добавлені користувачі. Наступним етапом роботи, це вказати створені адреси користувачів у меню налаштування для плати проекту і подальша компіляція файлів для завантаження їх у плату.

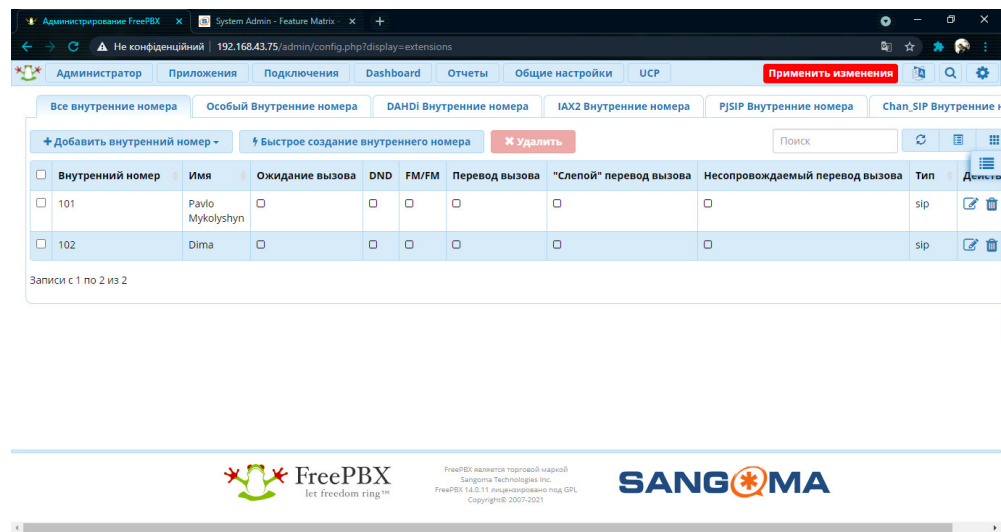


Рисунок 3.16 – Внутрішні номери

3.3 Тестування пристрою двостороннього зв'язку

Для початку, викликавши команду `make menuconfig` у випадаючому меню серед переліку плат вибираємо потрібний нам тип, а саме ESP-ADF Rev B.

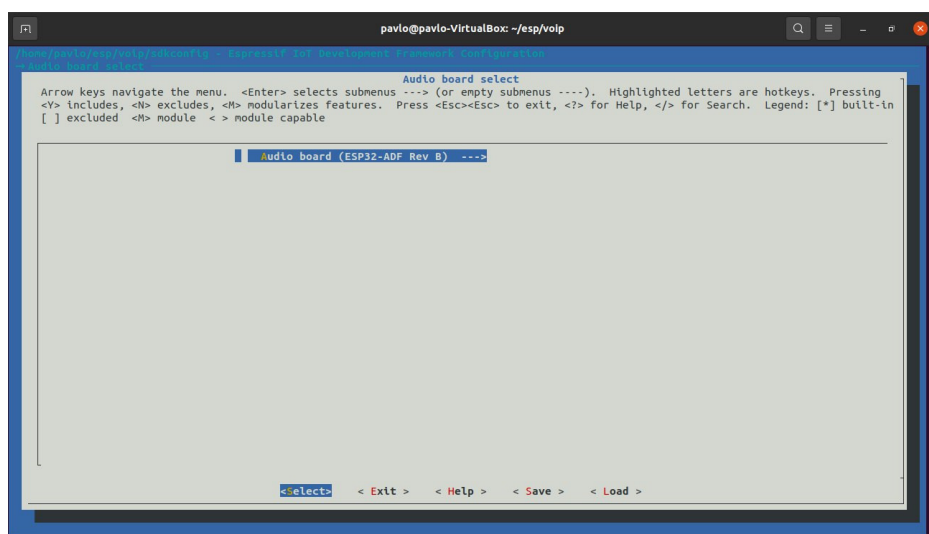


Рисунок 3.17 – Menuconfig

Далі у розділі Serial flash config вказуємо порт підключення USB і швидкість передачі даних.

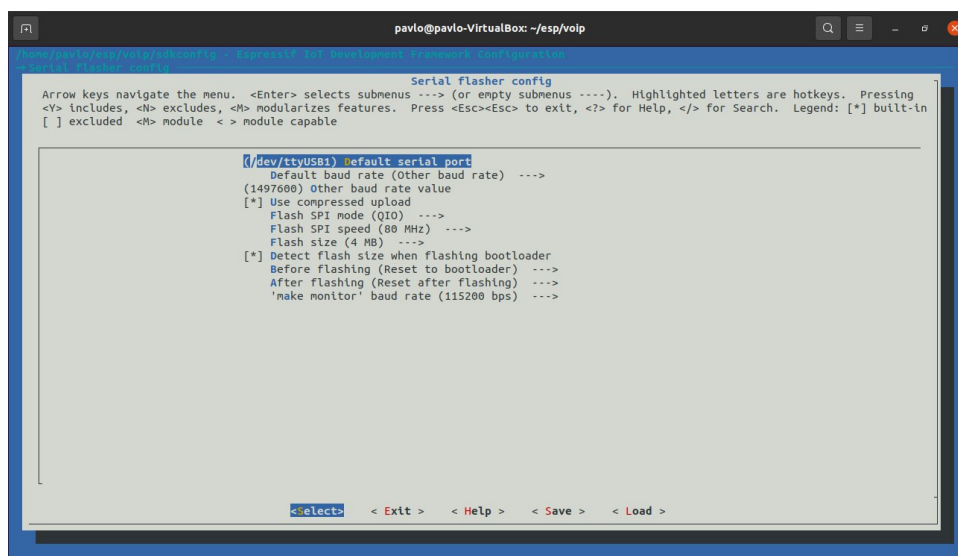


Рисунок 3.18 – Serial flash config

У розділі VoIP App Configuration вказуємо WiFi SSID і пароль для підключення. У вкладці SIP URI вводимо дані користувача, до якого ми підключимось через сервер.

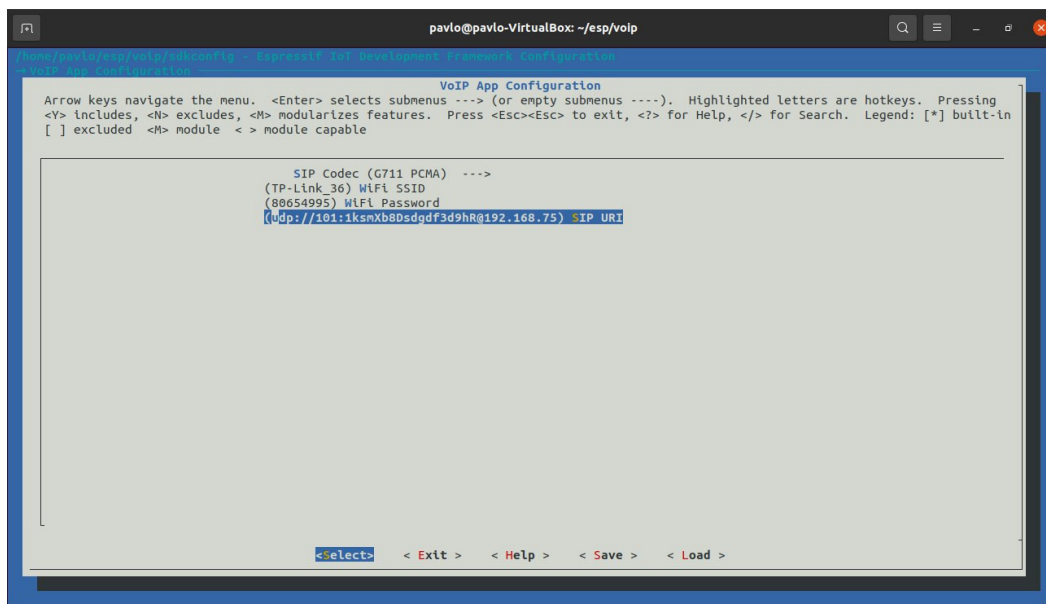


Рисунок 3.19 – VoIP App Configuration

Після внесення усіх змін необхідно зберегти налаштування і виконавши команду make flash здійснити завантаження коду у плату пристрою.

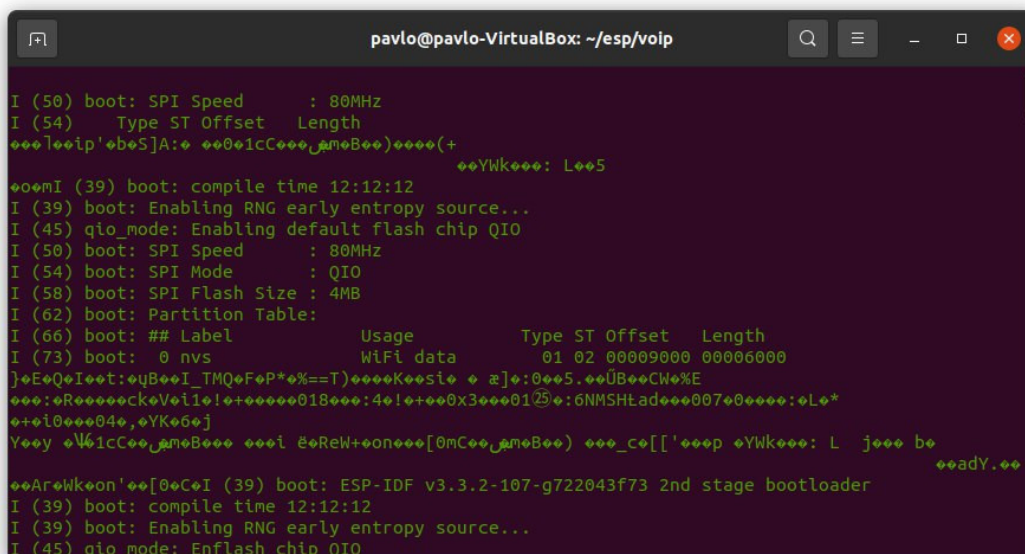


Рисунок 3.20 – Результат роботи коду

Початок сеансу (реєструється відправлення пакетів на сервер):

```
REGISTER sip:iptel.org:5060 SIP/2.0
Call-ID: e36cd7f583bd16f08d43c4bf55db76f3@212.191.78.149
CSeq: 1 REGISTER
From: <sip: msloma@iptel.org> ;tag=69
To: <sip: msloma@iptel.org>
Via: SIP/2.0/UDP
212.191.78.149:8193;branch=z9hG4bKda36626c48e78d5c0
ecab33b6cb411c0
Max-Forwards: 2
Contact: <sip: msloma@212.191.78.149: 8193; transport = udp>
Content-Length: 0
```

Відповідь сервера про необхідність авторизації:

```
SIP/2.0 401 Unauthorized Call-ID:
e36cd7f583bd16f08d43c4bf55db76f3@212.191.78.149
CSeq: 1 REGISTER
From: <sip: msloma@iptel.org>;tag=69
To: <sip:
msloma@iptel.org>;tag=b51ece8fd8c195776737473dd6552d43.6457
Via: SIP/2.0/UDP
212.191.78.149:8193;branch=z9Hg4bKda36626c48e78d5c0
ecab33b6cb411c0;rport=8193
WWW-Authenticate: Digest
```

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

```
realm=»iptel.org»,nonce=»4293708edad57d1bfb
e62ec3eec76ecdeda9eb92» Server: Sip Express router (0.9.0
(i386/linux))
Warning: 392 195.37.77.99:5060 «Noisy feedback tells: pid=15082
req _src_ip=212.191.78.149 req_src_port=8193
in_uri=sip:iptel.org:5060 out_uri=sip:i ptel.org:5060 via_cnt==1»
Content-Length: 0
```

Тепер програма має всю інформацію, необхідну для авторизації. Далі генерується автентична відповідь та поновлення запиту на голосовий сеанс зв'язку.

Спроба реєстрації на сервері:

```
REGISTER sip:iptel.org:5060 SIP/2.0
Call-ID: e36cd7f583bd16f08d43c4bf55db76f3@212.191.78.149
CSeq: 2 REGISTER
From: <sip: msloma@iptel.org>;tag=69
To: <sip: msloma@iptel.org>
Via: SIP/2.0/UDP
212.191.78.149:8193;branch=z9hG4bK1947973e08ac6f5e7
39d89a9ac23fae4
Max-Forwards: 2
Contact: <sip: msloma@212.191.78.149: 8193; transport = udp>
Authorization: Digest
username="msloma",realm="iptel.org",uri="sip:i
ptel.org:5060",algorithm=MD5,opaque="",nonce="4293708edad57d1bfb6
2ec3eec76ecded a9eb92",response="352b65e071115d54335f81958b9984a1"
Content-Length: 0
```

Позитивна відповідь сервера:

```
SIP/2.0 200 OK
Call-ID: e36cd7f583bd16f08d43c4bf55db76f3@212.191.78.149
CSeq: 2 REGISTER
From: <sip: msloma@iptel.org>;tag=69
To: <sip:
msloma@iptel.org>;tag=b51ece8fd8c195776737473dd6552d43.d305
Via: SIP/2.0/UDP
212.191.78.149:8193;branch=z9hG4bK1947973e08ac6f5e7
39d89a9ac23fae4;rport=8193 P-Role: master
Contact: <sip: msloma@iptel.org>; expires=600
Server: Sip EXpress router (0.9.0 (i386/linux))
Warning: 392 195.37.77.99:5060 "Noisy feedback tells: pid=15082
req _src_ip=212.191.78.149 req_src_port=8193
in_uri=sip:iptel.org:5060 out_uri=sip:i ptel.org:5060 via_cnt==1"
Content-Length: 0
```

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Користувач отримує пакет із повідомленням про те, що другий користувачем надає запрошення.

SIP/2.0 180 Ringing Record-Route:
Call-ID: c4a4de87260b3dd346ccf07809407fd0@212.191.78.149
CSeq: 1 INVITE
From: <sip: msloma@iptel.org>;tag=8696
To: <sip: msloma@iptel.org>
Via: SIP/2.0/UDP
212.191.78.149:9317;rport=9317;branch=z9hG4bKa85aa9
dedffc1b6950baa0b1b8bde06e Max-Forwards: 1
P-NAT-Check: YES
Content-Length: 0

Коли інший користувач приймає розмову, надсилається відповідь 200 ОК.

SIP/2.0 200 OK:
Record-Route: Call-ID:
c4a4de87260b3dd346ccf07809407fd0@212.191.78.149
CSeq: 1 INVITE
From: <sip: msloma@iptel.org>;tag=8696
To: <sip: msloma@iptel.org>;tag=5611
Via: SIP/2.0/UDP
212.191.78.149:9317;rport=9317;branch=z9hG4bKa85aa9
dedffc1b6950baa0b1b8bde06e Max-Forwards: 1
Contact: <sip: msloma@iptel.org>
Content-Type: application/sdp
P-NAT-Check: YES
Content-Length: 112

Користувач-ініціатор підтверджує отримання відповіді:

ACK sip:msloma@212.191.78.148:4029;transport=udp SIP/2.0
Call-ID: c4a4de87260b3dd346ccf07809407fd0@212.191.78.149 83
CSeq: 1 ACK
From: <sip: msloma@iptel.org>;tag=8696
To: <sip: msloma@iptel.org>;tag=5611
Via: SIP/2.0/UDP 212.191.78.149:9317;branch=1455210040
Max-Forwards: 2
Route: <sip: 195.37.77.99; ftag = 8696; lr = on>
Content-Length: 0

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

На цьому етапі встановлюється мультимедійний сеанс. Користувач, який бажає закінчити мультимедійний сеанс, надсилає повідомлення BYE і завершує сеанс сигналізації і визначає термінали, що беруть участь у них, щоб закінчити сеанс мультимедіа:

```
BYE sip:msloma@212.191.78.148:4029;transport=udp SIP/2.0
Call-ID: c4a4de87260b3dd346ccf07809407fd0@212.191.78.149
CSeq: 2 BYE
From: <sip: msloma@iptel.org>;tag=8696
To: <sip: msloma@iptel.org>; tag = 561;tag=5611
Via: SIP/2.0/UDP
212.191.78.149:9317;branch=z9hG4bKdbffd57f572e27b8f
084f260a2925423
Max-Forwards: 2
Route: Content-Length: 0
```

Отримуємо відповідь:

```
SIP/2.0 200 OK
Record-Route: <sip: 195.37.77.99; ftag = 8696; lr = on>
Call-ID: c4a4de87260b3dd346ccf07809407fd0@212.191.78.149
CSeq: 2 BYE
From: <sip: slomi@iptel.org>; ter = 8696;tag=8696
To: <sip: slomi@iptel.org>;tag=5611
Via: SIP/2.0/UDP
212.191.78.149:9317;rport=9317;branch=z9hG4bKdbffd5
7f572e27b8f084f260a2925423 Max-Forwards: 1
P-NAT-Check: YES
Content-Length: 0
```

У кінці сеансу зв'язку, обидві сторони завершують сесію.

Під час проведення тестування пристрою, не було виявлено проблем із надсиланням звукових даних через сервер користувачу. Пристрій для двостороннього зв'язку в результаті виконання тестів пройшов перевірку функціональності.

В результаті перевірки серйозних проблем не було виявлено. Результат роботи пристрою зображено на рисунку 3.21.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

У даному розділі кваліфікаційної роботи проводиться економічне обґрунтування доцільності розробки пристрою двостороннього зв'язку. Зокрема, здійснюється розрахунок витрат на реалізацію даного пристрою, експлуатаційних витрат, ціни на споживання проектного рішення, визначаються показники економічної ефективності нового рішення для інтернет телефонії, обґрунтовуються відповідні висновки.

Розроблений пристрій двостороннього зв'язку призначений для спілкування у мережі Інтернет згідно протоколу SIP.

4.1 Розрахунок витрат на реалізацію пристрою.

Експлуатаційними витратами є такі витрати, які забезпечують нормальне функціонування певного технічного рішення в період його експлуатації в розрахунку за рік.

Величина експлуатаційних витрат E за рік може бути розрахована за формулою:

$$E = k \cdot C \cdot \beta = k \cdot A \cdot S \cdot \beta, \quad (4.1)$$

де C – ціна реалізації нової розробки, якщо вона була відома або визначена раніше, грн./шт.;

k – коефіцієнт, який ураховує витрати на амортизацію, для обчислювальної техніки $k = 0,5 \dots 0,7$;

A – коефіцієнт, який ураховує прогнозований прибуток та податки, які повинен сплачувати виробник, $A = 1,7 \dots 2,3$;

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

S – собівартість нової розробки, розрахована спрощеним способом; грн.;

β – доля часу, який витрачає працівник на обслуговування нової технічної або інтелектуальної розробки в загальному часі своєї роботи.

$$E_{(\text{аналог})} = 0,7 * 1750 * 0,9 = 1102,5 \text{ грн}$$

$$E_{(\text{розробка})} = 0,7 * 2,3 * 307,7 * 0,9 = 445,86 \text{ грн}$$

Отже, витрати для розробки аналога становлять 1102,5 грн, а для нашої розробки – 445,86 грн.

Розрахунок витрат на створення програмного забезпечення і оплати праці розробників.

Розрахуємо середньогодинну оплату програміста. Для цього необхідно спочатку визначити його річний фонд грошового забезпечення. Це можна зробити, знаючи місячне грошове забезпечення програміста. Воно складає приблизно 15000,00 грн. Таким чином, річний фонд грошового забезпечення складає 180000 грн. Кількість робочих годин у році розраховуємо за формулою:

$$N_p = (N - N_n - N_g) * 8, \quad (4.2)$$

$$N_p = (365 - 14 - 104) * 8 = 1976$$

де N – загальна кількість днів у році,

N_n – кількість святкових днів у році,

N_v – кількість вихідних днів у році.

Приймається, що кількість святкових днів у році – 14, а вихідних – 104.

Середньогодинна оплата праці програміста визначається за формулою:

$$C_n = \frac{\Phi_p}{N_p}, \quad (4.3)$$

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

$$C_n = 200000 / 1976 = 101.21$$

де Φ_p – річний фонд грошового забезпечення.

Витрати на оплату праці розробників програми складають:

$$B_{оп} = C_n * T_3, \quad (4.4)$$

$$B_{оп} = 101.21 * 70 = 7084,7$$

де T_3 - загальна кількість годин роботи програміста над проектом.

Собівартість однієї години роботи ПК дорівнює:

$$C_{ПК} = \frac{B_e}{\Phi_{ПК}}, \quad (4.5)$$

$$C_{ПК} = 33000 / 1965 = 16,79$$

де B_e - річні поточні витрати на експлуатацію ПК,

$\Phi_{ПК}$ - річний фонд часу корисної роботи ПК. Розрахуємо річний фонд часу роботи ПК.

Суму річних амортизаційних відрахувань визначаємо за такою формулою:

$$B_{Ap} = C_{ПК} * H_A, \quad (4.6)$$

$$B_{Ap} = 16800 * 0,15 = 2520$$

де $C_{ПК}$ – балансова вартість ПК,

H_a – норма амортизаційних відрахувань (дорівнює 15% у квартал).

Балансову вартість ПК розраховуємо за формулою:

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

$$Ц_{ПК} = Ц_p * (1 + K_{УН}), \quad (4.7)$$

$$Ц_{ПК} = 15000 * (1 + 0,12) = 16800$$

де $Ц_p$ – ринкова вартість ПК,

$K_{УН}$ – коефіцієнт, що враховує витрати на установку й налагодження ПК (приймається рівним 12%).

Витрати на електроенергію, що споживає ПК, визначаємо за формулою:

$$B_{E_p} = P_{ПК} * \Phi_{ПК} * Ц_E * K_{ІВ}, \quad (4.8)$$

$$B_{E_p} = 0,2 * 1965 * 1,68 * 0,8 = 528.19$$

де $P_{ПК}$ – паспортна потужність ПК,

$\Phi_{ПК}$ – річний фонд корисного часу роботи ПК,

$Ц_E$ – вартість 1 кВт/год електроенергії,

$K_{ІВ}$ – коефіцієнт інтенсивного використання ПК (0,7 -1).

Таким чином, розрахункове значення витрат на електроенергію, що споживає ПК, складає:

- витрати на поточний і профілактичний ремонт (приймаються рівними 6% від вартості ПК):

$$B_{РЕМ_p} = Ц_{ПК} * 0,06, \quad (4.9)$$

$$B_{ремр} = 16800 * 0,06 = 1008$$

- витрати на додаткові комплектуючі витрати, необхідні для забезпечення експлуатації ПК (приймаються рівними 2% від вартості ПК):

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

$$B_{ДК_p} = Ц_{ПК} * 0,02, \quad (4.10)$$

$$B_{ДКp} = 16800 * 0,02 = 336$$

- інші витрати, тобто непрямі витрати, пов'язані з експлуатацією ПК (приймаються рівними 5-10% від вартості ПК):

$$B_{I_p} = Ц_{ПК} * 0,05, \quad (4.11)$$

$$B_{ip} = 16800 * 0,05 = 840$$

У ході розробки програмного комплексу, ПК використовується на таких етапах програмування:

- написання програми за готовою схемою алгоритму;
- налагодження програми на ПК;
- підготовки документації по задачі.

Таким чином, витрати машинного часу склали ($t_{маш}$):

$$t_{маш} = t_n + t_{отг}^k + t_{\partial}, \quad (4.12)$$

$$t_{маш} = 39,37 + 295,31 + 2067,18 = 2401,86$$

Розрахунок можливої ціни пристрою.

Собівартість одиниці нової продукції можна спрогнозувати за формулою:

$$S = \frac{(B_n \cdot K_n \cdot 100\%)}{П}, \quad (4.13)$$

де B_n – величина однієї із статей прямих витрат, яка вибрана за основу, в даному випадку це вартість інтелектуальної власності, грн.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

П – питома вага заробітної плати розробників в собівартості аналога, %;
 К_н – коефіцієнт, який враховує конструктивні та технологічні особливості програмного продукту (1..1,2).

$$S = (200 * 1 * 100 \%) / 65 \% = 307.7 \text{ грн}$$

Величина капітальних вкладень К може бути розрахована за формулою:

$$K = (1..3) \cdot B_p, \quad (4.14)$$

де (1..3) – це коефіцієнт, який враховує додаткові витрати;
 В_р – загальні витрати на проектування даної розробки;

$$K_{(\text{аналог})} = 2 * 8000 = 16000 \text{ грн}$$

$$K_{(\text{розробка})} = 2 * 3000 = 6000 \text{ грн}$$

Для зручності зведемо основні показники системи що розробляється, та аналога до таблиці 4.1.

Таблиця 4.1 – Основні показники

Показники	Одиниці виміру	Варіанти	
		Аналог	Нова розробка
Капітальні вкладення	грн.	16000	6000
Експлуатаційні витрати	грн./рік	1102,5	445,86

Оскільки, капітальні вкладення та експлуатаційні витрати аналога більші за аналогічні показники нової розробки, тоді абсолютний ефект на капітальних вкладеннях і на експлуатаційних витратах можна розрахувати за формулами:

$$K_{\text{еф}} = K_1 - K_2 \quad , \quad (4.15)$$

$$K_{\text{еф}} = 16000 - 6000 = 10000 \text{ грн}$$

$$E_{\text{еф}} = E_1 - E_2 \quad , \quad (4.16)$$

$$E_{\text{еф}} = 1102,5 - 445,86 = 656,64 \text{ грн}$$

На основі проведеного техніко-економічного обґрунтування доцільності реалізації пристрою та на підставі наведених співвідношень видно суттєві переваги нової розробки порівняно з аналогічним продуктом, що вже є на ринку.

Розробка є ефективною, оскільки забезпечує економію на капіталовкладеннях – 10000 грн., а на експлуатаційних витратах – 656,64 грн./рік.

Ціноутворення слід базувати на основі цін конкурента і встановити початкову ціну на розроблений продукт на 10 – 15% меншу за аналогічний. Далі, аналізуючи ситуацію на ринку, при можливості можна ще більше знизити її, тим самим залучаючи потенційних клієнтів.

Основним конкурентом можна вважати фірму виробник пристроїв для інтернет телефонії Fanvil. Ціна якої приблизно становить 1750 грн. Для завоювання позиції на ринку потрібно встановити меншу ціну на програмний продукт.

На основі проведеного попереднього аналізу, можна зробити висновок, що пристрій матиме попит на ринку і приносить прибуток.

Дослідження аналогів та порівняння їх вартість із ціною розробленої системи.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

При аналізі ринку були виявлені такі фірми конкуренти:

- Fanvil;
- Panasonic;
- Yealink;
- Alcatel-Lucent.

Головним конкурентом даного продукту на ринку є фірма Fanvil. Її головними особливостями є невелика ціна пристроїв, стильний і компактний дизайн у використанні їх продукції.

При порівнянні вартості реалізації пристрою для інтернет телефонії з аналогами, виявилось, що дана розробка є дешевшою чим в конкурента.

Таблиця 4.2 – Основні показники

Аналог (грн)		Нова розробка (грн)
Капітальні вкладення	16000	6000
Експлуатаційні витрати	1102,5	445,86

Відрахування на соціальні заходи.

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат.

Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати:

$$V_{\phi} = 0.205 * 7084,7 = 1452.36 \text{ грн}$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення (таблиця 4.3).

Таблиця 4.3 – Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	7084,7
2	Відрахування у спеціальні державні фонди	1452.36
3	Амортизаційних відрахування	2520
4	Витрати на електроенергію	528.19
Разом		11585,25

Оцінка соціально-економічних результатів функціонування продукту.

Беручи до уваги потенціал країни, в близькій перспективі можна очікувати створення ринку інформаційних продуктів і послуг, аналогічного ринкам Європи. Створення товарів для інтернет телефонії є однією із сфер діяльності, де реально може бути використаний великий науково-технічний потенціал. Для розвитку такої галузі Україна має високоосвічені, кваліфіковані і порівняно дешеві трудові ресурси, а також достатню телекомунікаційну інфраструктуру.

Дана розробка є актуальною, оскільки телекомунікація відіграє важливу роль у роботі компаній.

4.2 Розрахунок ціни проекту

При розробці проекту, а саме – пристрою на базі ESP32 для здійснення інтернет дзвінків, необхідно використати таке обладнання і програмне забезпечення:

- комп'ютер (1 шт.);
- програмне забезпечення для здійснення дзвінків по протоколу SIP (1 шт.);
- телефон (1 шт.);
- плата Olimex ESP32 з під'єднаними до неї компонентами (1 шт.);
- програмне забезпечення для налаштування ESP32 (1 шт.).

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Отже, для розробки проекту, нам потрібно використати 5 одиниць обладнання і програмного забезпечення.

Балансову вартість ПК розраховуємо за формулою:

$$Ц_{ПК} = Ц_p * (1 + K_{УН}), \quad (4.17)$$

$$Ц_{ПК} = 15000 * (1 + 0,12) = 16800$$

де $Ц_p$ – ринкова вартість ПК,

$K_{УН}$ – коефіцієнт, що враховує витрати на установку й налагодження ПК (приймається рівним 12%).

Вартість продажу розробленого продукту розраховують за формулою:

$$B_i = B_{заг} \cdot \left(1 + \frac{p}{100}\right), \quad (4.18)$$

$$B_i = 47411,92 * (1 + 20 / 100) = 56894,3$$

де p - середній рівень рентабельності на поточний період. Показник рентабельності інвестицій.

Обґрунтування техніко-економічної ефективності розробки.

Беручи до уваги потенціал країни, в близькій перспективі можна очікувати створення ринку інформаційних продуктів і послуг, аналогічного ринкам Європи. Створення програмних засобів є однією із сфер діяльності, де може бути використаний великий науково-технічний потенціал. Для розвитку такої галузі Україна має високоосвічені, кваліфіковані і порівняно дешеві трудові ресурси, а також достатньо розвинену телекомунікаційну інфраструктуру. Отже, реалізація пристрою є актуальною, оскільки телекомунікація розвивається швидко і завжди буде актуальною у сучасному світі.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

4.3 Визначення показників економічної ефективності

Визначення трудомісткості розробки програмного забезпечення.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ розраховують за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \quad (4.19)$$

$$t = 50 + 15,75 + 39,37 + 39,37 + 196,87 + 2067,18 = 2408,54$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ - витрати праці на налагодження програми на ПК;

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється. Умовне число операторів (підпрограм) розраховують за формулою:

$$Q = q \cdot C \cdot (1 + p), \quad (4.20)$$

$$Q = 500 * 1.2 * (1 + 0.05) = 630$$

де q - передбачуване число операторів;

C - коефіцієнт складності програми;

P - коефіцієнт кореляції програми в ході її розробки.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \quad (4.21)$$

$$t_u = 630 \cdot 1.5 / (75 \cdot 0.8) = 15,75$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Витрати праці на розробку алгоритму розв'язку задачі розраховують за формулою:

$$t_a = \frac{Q}{(20...25) \cdot k}, \quad (4.22)$$

$$t_a = 630 / (20 \cdot 0.8) = 39.37$$

Витрати на складання програми по готовій блок-схемі розраховують за формулою:

$$t_n = \frac{Q}{(20...25) \cdot k}, \quad (4.23)$$

$$t_n = 630 / (20 \cdot 0.8) = 39.37$$

Витрати праці на налагодження програми на ПК розраховують:

- за умови автономного налагодження одного завдання:

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

$$t_{\text{отл}} = \frac{Q}{(4.5) \cdot k}, \quad (4.24)$$

$$t_{\text{отл}} = 630 / (4 * 0.8) = 196,87$$

- за умови комплексного налагодження завдання:

$$t_{\text{отл}}^k = 1,5 \cdot t_{\text{отл}}, \quad (4.25)$$

$$t_{(k)\text{отл}} = 1.5 * 196,87 = 295,31$$

Витрати праці на підготовку документації розраховують за формулою:

$$t_{\text{д}} = t_{\text{др}} + t_{\text{до}}, \quad (4.26)$$

$$t_{\text{д}} = 52,5 * 39,37 = 2067,18$$

де $t_{\text{др}}$ - трудомісткість підготовки матеріалів і рукопису, що розраховують за формулою:

$$t_{\text{др}} = \frac{Q}{(15..20) \cdot k}, \quad (4.27)$$

$$t_{\text{др}} = 630 / (15 * 0.8) = 52,5$$

де $t_{\text{до}}$ - трудомісткість редагування, друкування й оформлення документації:

$$t_{\text{до}} = 0,75 \cdot t_{\text{др}}, \quad (4.28)$$

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

$$t_{до} = 0.75 * 52,5 = 39,37$$

Витрати, пов'язані з розробкою програми на ПК розраховуються:

$$B_{ПК} = T_{ПК} * t_c, \quad (4.29)$$

$$B_{ПК} = 70 * 3000 = 210000$$

де $T_{ПК}$ - час використання ПК для розробки програми,

$B_{ПК}$ - собівартість машинного часу обчислювальної техніки (розраховує бухгалтерія підприємства).

Витрати на оплату машинного часу розраховуємо за формулою:

$$B_{маш} = t_{маш} * C_{ПК}, \quad (4.30)$$

$$B_{маш} = 2401,86 * 16,79 = 40327,22$$

Загальні витрати на розробку програмного комплексу складають:

$$B_{заг} = B_{оп} + B_{маш}, \quad (4.31)$$

$$B_{заг} = 7084,7 + 40327,22 = 47411,92$$

Розрахунок показників економічної ефективності розробки програмного продукту.

Дохід від розробки ПЗ у першому періоді розраховуємо за формулою:

$$D_i = J_i(B_i - C_i), \quad (4.32)$$

$$D_i = 1 * (56894,3 - 47411,92) = 9482,38$$

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

де V_i - ціна продажу програмного продукту в першому періоді;

C_i - собівартість програмного продукту (фактично дорівнює сумі витрат на розробку ПЗ);

J_i - кількість ПЗ.

Економічна ефективність полягає у відношенні результату від розробленого програмного продукту до затрачених ресурсів:

$$E = \frac{D_i}{B_{\text{заг}}}, \quad (4.33)$$

$$E = 9482,38 / 47411,92 = 0,2$$

Тоді термін окупності можна розрахувати за такою формулою:

$$T = \frac{1}{E}, \quad (4.34)$$

$$T = 1 / 0.2 = 5$$

Таблиця 4.4 - Економічні показники розробки

№	Показник	Значення
1	Собівартість, грн	56894,3
2	Плановий прибуток, грн.	9482,38
3	Ціна, грн.	47411,92
4	Економічна ефективність	0.2
5	Термін окупності, рік	5

Отже, можна зробити висновок щодо доцільності даної розробки. Термін окупності капітальних вкладень не перевищує 10 років. А це означає, що така розробка є економічно вигідною та конкурентоздатною на ринку подібних ІТ продуктів.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи отримано такі висновки.

1. Тестування проводилось на комп'ютері: Intel Pentium CPU (2.5 ГГц) / RAM 8 ГБ / HDD 500 ГБ / Intel HD Graphics 520 / без ОД / LAN / Wi-Fi / Bluetooth.

2. У даній роботі я представив функціонування пристрою для двостороннього зв'язку згідно протоколу SIP. Спроектував загальну структуру системи та описав алгоритм її роботи. Також обгрунтовано вибір одноплатних комп'ютерів, їх переваги та недоліки.

3. Здійснено аналіз апаратних засобів, які будуть використовуватися в системі двостороннього зв'язку. Також я описав можливості технології сигналізації, а саме протоколу SIP.

4. У практичній частині було проведено налаштування середовища розробки для плати ESP32, здійснено підбір, встановлення і можливості використання сервера для IP-телефонії.

5. Метою роботи є розробка, підбір і налаштування пристрою для роботи з сервером. В роботі наведені можливості використання плати Olimex ESP32-ADF при здійсненні дзвінків через мережу Інтернет для подальшого її використання у корпоративних цілях.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Миколишин П.П. Пристрій двостороннього зв'язку згідно протоколу SIP. Науково–практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Частина II. Тернопіль, 2021. с. 9

2. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Бакалавр” спеціальності 123 «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ЗУНУ, 2020. 60с.

3. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем»/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.

4. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.

5. Семенов Ю. А. Телекоммуникационные технологии. Интернет университет информационных технологий [Электронный ресурс]. – Режим доступа: <http://book.itep.ru>

6. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов: 3-е изд. / В. Г Олифер., Н. А. Олифер // – СПб.: Питер, 2006.

7. Зайченко Ю. П. Анализ и оптимизация характеристик сетей MPLS по заданным показателям качества / Ю. П. Зайченко, Ахмед А. М. Шарадка // Вісник національного технічного університету України КПІ сер. Інформатика управління та обчислювальна техніка. Вип. 43. – 113–123 с.

8. Будылдина Н. В. Разработка программного обеспечения для оптимизации мультисервисных сетей / Н. В. Будылдина., П. А. Коновалов // Открытое образование, июнь 2006. – 58 с.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

9. Зайцев Д. А. Моделирование телекоммуникационных сетей в системе NS. / Д. А. Зайцев, Т. Н. Шинкарчук // Наукові праці ОНАЗ ім. О. С. Попова. – 2006.– № 2.
10. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет / Е.А. Кучерявый // – М.: Наука и Техника. – 2007. – 336 с.
11. Panwar Li. Y. S. On the Performance of MPLS TE Queues for QoS Routing // Panwar Li. Y. Liu C.J. Simulation series. – 2004. – Vol. 36; part 3. – P. 170–174.
12. Роберт І.В. Сучасні інформаційні технології в освіті: дидактичні проблеми, перспективи використання — М.: Школа-Пресс, 1994— 205с.
13. Розділ 2.Основи UML – діаграм: [Електронний ресурс] Режим доступу:<https://docs.kde.org/trunk4/uk/kdesdk/umbrello/uml-basics.html>
14. Електронна енциклопедія Вікіпедія: Модульне тестування [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/Модульне>
15. Гольдштейн А.Б., Гольдштейн Б.С. SOFTSWITCH. БХВ – Санкт – Петербург, 2006. – 368 с.: ил.
16. Гольдштейн В.С., Пинчук А.В., Суховицкий А.Л. IP – телефония. – М.: Радио и связь, 2001. – 336с.: ил.
17. FusionPBX Documentation. Mark J Grane, Len Graham. Apr 3, 2020: <https://docs.fusionpbx.com/en/latest/>
18. Гольдштейн В.С., Пинчук А.В., Суховицкий А.Л. IP – телефония. https://www.studmed.ru/goldshteyn-bc-pinchuk-av-suhovickiy-al-iptelefoniya_3bd453debf.html
19. Интернет ресурс Вікіпедія. <https://ru.wikipedia.org/wiki/IP-телефония>
20. http://ekmair.ukma.edu.ua/bitstream/handle/123456789/13473/Cherkasov_Osnovy_tekhnologii_VoIP_ta_IP_telefonii.pdf?sequence=1&isAllowed=y
21. TCP/IP // [Електронний ресурс] – режим доступу: <https://uk.wikipedia.org/wiki/TCP/IP>
22. IPv6 // [Електронний ресурс] – режим доступу: <https://uk.wikipedia.org/wiki/IPv6>

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

23. Исчерпание _ IPv4-адресов// [Электронный ресурс] – режим доступа: https://ru.wikipedia.org/wiki/Исчерпание_ IPv4-адресов
24. PON // [Электронный ресурс] – режим доступа: https://uk.wikipedia.org/wiki/PON#Топології_мереж_доступу
25. Б.С . Гольдштейн, А .В. Пинчук, А .Л . Суховицкий, IP-телефония. - СПб.: БХВ — Санкт-Петербург, 2014.— 336 с.
26. Конопелько, В. К. К64 Измерение и анализ трафика IP-телефонии : метод. пособие по курсу «Цифровая коммутация каналов, пакетов и IPтелефония» для студ. спец. «Системы распределения мультимедийной информации» всех форм обуч. / В. К. Конопелько, С. М. Лапшин, В. Ю. Цветков. – Минск : БГУИР, 2011. – 56 с.
27. Гольдштейн, Б. С. Сети связи постNGN / Б. С. Гольдштейн, А. Е. Кучерявый. — СПб.: БХВПетербург, 2014. —160 с
28. Гольдштейн В.С., Пинчук А.В., Суховицкий А.Л. IP-Телефония. — М.: Радио и связь, 2001. — 336 с.: ил.
29. Прогноз розміру ринку VoIP на 2019-2025 роки. Звіт про аналіз частки галузі, дата публікації: квітень 2019, 400 с. Ідентифікатор звіту: GMI2989.
30. VoIP Market Size Forecast 2019-2025. Industry Share Analysis Report – Назва з екрану. – [Электронный ресурс]. – Режим доступа: <https://www.gminsights.com/industry-analysis/voice-over-internet-protocol-voip-market>
31. Global Business VoIP Market Report 2020– Назва з екрану. – [Электронный ресурс]. – Режим доступа: <https://www.cognitivemarketresearch.com/medicaldevices--consumables/business-voip-market-report>
32. Девидсон Д., Питерс Д., Бхатия М. Основы передачи голосовых данных по сетям IP. 2-е изд. — М.: Вильямс, 2007. — 400 с.
33. ITU-T Recommendation H.323, "Packet-Based Multimedia Communications Systems", February 1998.
34. M. Handley, H. Schulzrinne, E. Scholler, J. Rosenberg, " SIP: Session Initiation Protocol", RFC2543, IETF, March 1999.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

35. H. Schulzrinne, J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", Network and Operating System Support for Digital Audio and Video (NOSSDAV), Cambridge, England, July 1998.

36. Steve Donovan, Matthew Cannon, "A Functional Description of SIP-PSTN Gateway", Internet Draft, IETF, November 1998.

37. ITU-T Recommendation X.691, "Information Technology - ASN.1 encoding rules – Specification of Packed Encoding Rules (PER)," Dec. 1997.

38. Телекоммуникационные системы и сети: Учеб. пособие: В 3 т./под. ред. В.П. Шувалова. – М. : Горячая линия – Телеком, 2005. – Т.3: Мультисервисные сети / В.В. Величко [и др.]. – 2005.

39. Гольдштейн Б.С. Протокол SIP: справ. / Б. С. Гольдштейн, А. А. Зарубин, В. В. Саморезов. – СПб. : БХВ- СПб., 2005. – 456 с.

40. Мережева академія Cisco. – Назва з екрану. – [Електронний ресурс]. – Режим доступу: <https://www.netacad.com/>

41. Коллинс Д. IP telephony. - К.: Издательство «Мак Гроу Хилл», 2007. – 736 с.:

42. Таненбаум Э. Компьютерные сети. - СПб. : Питер, 2005.- 4-е изд.- 992 с.

43. Understanding Direct-Inward-Dial (DID) on IOS Voice Digital (T1/E1) Interfaces – Назва з екрану. – [Електронний ресурс]. – Режим доступу: <https://www.cisco.com/c/en/us/support/docs/voice/digital-ccs/14072-direct-inward-dial.html>

44. Ready To Get Started With Asterisk?– Назва з екрану. – [Електронний ресурс].– Режим доступу: <https://www.asterisk.org/>

45. FreePBX is an open source community – Назва з екрану. – [Електронний ресурс]. – Режим доступу: <https://www.freepbx.org/>

46. Microsip. Open source portable SIP softphone for Windows based on PJSIP stack– Назва з екрану. – [Електронний ресурс]. – Режим доступу: <https://www.microsip.org/downloads>

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

47. Н. Schulzrinne, J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", Network and Operating System Support for Digital Audio and Video (NOSSDAV), Cambridge, England, July 1998.

48. Семенов Ю. А. Телекоммуникационные технологии. Интернет университет информационных технологий [Электронный ресурс]. – Режим доступа: <http://book.itep.ru>

49. Девидсон Д., Питерс Д., Бхатия М. Основы передачи голосовых данных по сетям IP. 2-е изд. — М.: Вильямс, 2007. — 400 с.

50. ITU-T Recommendation X.691, "Information Technology - ASN.1 encoding rules – Specification of Packed Encoding Rules (PER)," Dec. 1997.

51. Будылдина Н. В. Разработка программного обеспечения для оптимизации мультисервисных сетей / Н. В. Будылдина., П. А. Коновалов // Открытое образование, июнь 2006. – 58 с.

					КР.КІ.07177/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66