

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИПЕНКО

(підпис)

“\_\_” \_\_\_\_\_ 2023 р.

**Дипломний проєкт**

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”  
спеціальності 123 “Комп’ютерна інженерія”

на тему: Засоби для віддаленої розробки вбудованої системи на платі

STM Discovery

Виконав : студент 4 курсу, групи ІВ-92  
(шифр групи)

Дмитришин Андрій Дмитрович

(прізвище, ім’я, по батькові)

(підпис)

Керівник асистент, Гайдай А.Р.

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

(підпис)

Консультант (нормоконтроль) ст. викладач Виноградов Ю. М.

(назва розділу)

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

(підпис)

Рецензент професор кафедри ІСТ, д.т.н. Корнієнко Б.Я.

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

(підпис)

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць  
інших авторів без відповідних  
посилань. Студент \_\_\_\_\_  
(підпис)

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)  
Освітньо-професійна програма  
“Комп’ютерні системи та мережі  
спеціальності 123 “Комп’ютерна інженерія”

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**  
Сергій СТИРЕНКО

(підпис)

“\_\_” \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

на бакалаврський дипломний проєкт студента

Дмитришина Андрія Дмитровича

1. Тема проєкту Засоби для віддаленої розробки вбудованої системи на  
платі STM Discovery

керівник проєкту \_\_\_\_\_ Гайдай Анатолій Русланович, асистент,

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 31 травня 2023 року №2102-с

2. Термін здачі студентом закінченого проєкту 8 червня 2023 р.

3. Вихідні дані до проєкту технічна документація, теоретичні дані.

4. Зміст пояснювальної записки (перелік питань, які розробляються)

Розділ 1. Опис та аналіз об’єкту дослідження.

Розділ 2. Огляд технологій для проектування системи.

Розділ 3. Деталі розробки системи.

Розділ 4. Тестування системи.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) взаємодія компонентів (структурна схема), блок інтерфейсний UML-діаграма (функціональна схема), алгоритм дій системи (принципова схема).
6. Консультанта проекту, з вказівкою розділів проекту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Виноградов Ю. М.		

7. Дата видачі завдання «15» грудня 2022 р.

#### Календарний план

№ п/п	Найменування етапів дипломного проекту	Терміни виконання етапів проекту	Примітки
1.	<i>Затвердження теми проекту</i>	<i>10.12.2022-15.12.2022</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>16.12.2022-23.04.2023</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>24.04.2023-30.04.2023</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>01.05.2023-07.05.2023</i>	
5.	<i>Програмна реалізація системи</i>	<i>08.05.2023-14.05.2023</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.05.2023-27.05.2023</i>	
7.	<i>Захист програмного продукту</i>	<i>04.06.2023</i>	
8.	<i>Передзахист</i>	<i>05.06.2023</i>	
9.	<i>Захист</i>	<i>19.06.2023</i>	

Студент-дипломник \_\_\_\_\_ Андрій ДМИТРИШИН  
(підпис)

Керівник проекту \_\_\_\_\_ Анатолій ГАЙДАЙ  
(підпис)

## АНОТАЦІЯ

Ця дипломна робота присвячена розробці засобів для віддаленої розробки вбудованої системи на платі STM Discovery в навчальному процесі. Метою роботи є створення продукту, який дозволить студентам отримати практичний досвід розробки вбудованих систем навіть на віддаленій основі. У роботі розглянуті області застосування засобів для віддаленої розробки, зокрема в навчальному процесі.

У цій роботі буде проведений аналіз різних засобів і технологій, що забезпечують віддалену розробку вбудованих систем, з фокусом на платі STM Discovery. Буде досліджено різні протоколи, програмне забезпечення та інструменти, які дозволяють забезпечити взаємодію з платою STM Discovery через мережу. Продукт, який буде розроблено в рамках цієї роботи, надасть студентам можливість віддалено виконувати практичні завдання з вбудованих систем, навчатися на реальному апаратному забезпеченні та розвивати навички розробки в цій області.

Ключові слова: віддалена розробка, вбудована система, STM Discovery.

## ANNOTATION

This thesis is dedicated to the development of tools for remote development of embedded systems on the STM Discovery board in the educational process. The aim of this work is to create a product that enables students to gain practical experience in embedded systems development even in a remote setting. The research explores the areas of application for remote development tools, particularly in the educational process.

The thesis conducts an analysis of various tools and technologies that facilitate remote development of embedded systems, with a focus on the STM Discovery board. Different protocols, software, and tools that enable interaction with the STM Discovery board over a network will be investigated. The product developed as part of this work will provide students with the opportunity to remotely perform practical tasks in embedded systems, learn on real hardware, and develop their skills in this field.

Keywords: remote development, embedded system, STM Discovery.



**ТЕХНІЧНЕ ЗАВДАННЯ**  
**ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Засоби для віддаленої розробки вбудованої системи на платі STM  
Discovery»

# ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ .....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ .....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. ВИМОГИ ДО РОЗРОБЛЕНОГО ПРОДУКТУ .....	3
5.2. ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	3
5.3. ВИМОГИ ДО АПАРАТНОЇ ЧАСТИНИ .....	3
6. ЕТАПИ РОЗРОБКИ .....	3

					<b>ІАЛЦ.467200.002 ТЗ</b>			
		№ докум.	Підпис	Дата				
Розробив		Дмитришин А.Д.			Засоби для віддаленої розробки вбудованої системи на платі STM Discovery	Літ.	Аркуш	Аркушів
Перевірив		Гайдай А.Р.				1	3	
Н. Контр.		Виноградов Ю. М.				КПІ ім. Ігоря Сікорського		
Затвердив						ФІОТ ІВ-92		
					Технічне завдання			

# 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання поширюється на розробку засобів для віддаленої розробки вбудованої системи на платі STM Discovery.

Областю застосування модуля є галузі, де необхідна взаємодія з вбудованими системами віддалено.

## 2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки даної системи є завдання для виконання роботи кваліфікаційно-освітнього рівня «бакалавр комп'ютерної інженерії», яке було затверджено факультетом “Інформатики та обчислювальної техніки” кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського».

## 3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою та призначенням даної роботи є розробка системи для віддаленої розробки вбудованої системи на платі STM Discovery з метою використання в навчальних закладах, університетах та інших освітніх установах для покращення навчального процесу.

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки даного дипломного проекту є офіційні документації, публікації та статті в мережі Інтернет на дану тему, а також науково-технічна література.

					ІАЛІЦ.467200.002 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1. Вимоги до розробленого продукту

- Забезпечення можливості віддаленого доступу до плати STM Discovery та апаратного забезпечення для розробки вбудованих систем.
- Інтерфейс повинен бути інтуїтивно зрозумілим та підтримувати основні функції розробки вбудованих систем.
- Розроблена система повинна забезпечувати високий рівень безпеки віддаленого доступу до апаратного забезпечення.

### 5.2. Вимоги до програмного забезпечення

- Операційна система Windows, MacOS або Linux.
- Інтегроване середовище розробки STM32CubeIDE
- Програма для роботи з SSH протоколом

### 5.3. Вимоги до апаратної частини

- Плата STM32F407G-DISC1
- Логічний аналізатор - AZ-Delivery
- ЦП не менше ніж Intel® Core (TM) i3-2100T.
- ROM не менше ніж 64 ГБ.
- RAM не менше ніж 4 ГБ.
- Веб-камера HD 1080P

## 6. ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	10.12.2022-15.12.2022
Вивчення та аналіз завдання	16.12.2022-23.04.2023
Розробка архітектури та загальної структури системи	24.04.2023-30.04.2023
Розробка структур окремих частин системи	01.05.2023-07.05.2023
Програмна реалізація системи	08.05.2023-14.05.2023
Оформлення пояснювальної записки	15.05.2023-27.05.2023

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Засоби для віддаленої розробки вбудованої системи на платі STM  
Discovery»

# ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ .....	2
ВСТУП .....	3
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ .....	5
1.1 ПОНЯТТЯ ВБУДОВАНИХ СИСТЕМ .....	5
1.2 ЗАГАЛЬНИЙ ОГЛЯД STM. ....	7
1.3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ НА STM .....	10
1.4 ПРОТОКОЛИ ДЛЯ КЕРУВАННЯ ТА ПРОШИВКИ STM .....	14
1.5 БЕЗПЕКА ВІДДАЛЕНОЇ РОЗРОБКИ .....	15
ВИСНОВОК ДО РОЗДІЛУ 1 .....	17
РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ ПРОЕКТУВАННЯ СИСТЕМИ .....	19
2.1 ПРОШИВКА STM МІКРОКОНТРОЛЕРІВ ЧЕРЕЗ КОНСОЛЬ .....	19
2.2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДЕОТРАНСЛЯЦІЇ .....	26
2.3 ПРОТОКОЛИ ПЕРЕДАЧІ ДАНИХ ЧЕРЕЗ МЕРЕЖУ .....	33
ВИСНОВОК ДО РОЗДІЛУ 2 .....	41
РОЗДІЛ 3. ДЕТАЛІ РОЗРОБКИ СИСТЕМИ .....	43
3.1 РОБОТА З ІНСТРУМЕНТОМ STM32CUBEPROGRAMMER CLI .....	43
3.2 НАЛАШТУВАННЯ SSH ПІДКЛЮЧЕННЯ ДЛЯ ВІДДАЛЕНОГО ДОСТУПУ .....	44
3.3 ПЕРЕСІЛКА ФАЙЛІВ ЗА ДОПОМОГОЮ SCP .....	48
3.4 РОБОТА З AZ DELIVERY LOGIC ANALYZER .....	49
ВИСНОВОК ДО РОЗДІЛУ 3 .....	52
РОЗДІЛ 4. ТЕСТУВАННЯ СИСТЕМИ .....	53
ВИСНОВОК ДО РОЗДІЛУ 4 .....	57
ВИСНОВКИ .....	58
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	59

					<b>ІАЛЦ.467200.003 ПЗ</b>			
		№ докум.	Підпис	Дата				
Розробив	Дмитришин А.Д.				Засоби для віддаленої розробки вбудованої системи на платі STM Discovery	Літ.	Аркуш	Аркушів
Перевірив	Гайдай А.Р.					1	62	
Н. Контр.	Виноградов Ю. М.					КПІ ім. Ігоря Сікорського		
Затвердив						ФІОТ ІВ-92		
					Пояснювальна записка			

\

UART (Universal Asynchronous Receiver-Transmitter) - універсальний асинхронний передавач-приймач.

USART (Universal Synchronous/Asynchronous Receiver-Transmitter) - універсальний синхронний/асинхронний передавач-приймач.

SPI (Serial Peripheral Interface) - послідовний інтерфейс для зовнішніх пристроїв.

I2C (Inter-Integrated Circuit) - міжінтегральна шина для зв'язку між електронними пристроями.

USB (Universal Serial Bus) - універсальна послідовна шина для підключення пристроїв до комп'ютера або інших електронних пристроїв.

IDE (Integrated Development Environment) - інтегроване середовище розробки.

SWD (Serial Wire Debug) - послідовна шина для налагодження.

SSH (Secure Shell) - безпечний протокол оболонки.

					ІАЛІЦ.467200.003 ПЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Сучасний світ переповнений технологіями, які з кожним днем стають все більш інтегрованими і складними. Важливою частиною цього технологічного розвитку є вбудовані системи, які знайшли широке застосування в різноманітних галузях - від промисловості до розваг. В рамках цього дипломного проекту, основна увага приділена дослідженню вбудованих систем на базі мікроконтролерів STM32.

STM32 - це серія високопродуктивних мікроконтролерів від компанії STMicroelectronics, які засновані на ядрах ARM Cortex-M. Ці мікроконтролери демонструють вражаючу гнучкість і відмінні характеристики, що робить їх вибором для багатьох розробників вбудованих систем.

Однак, не дивлячись на свої сильні сторони, процес розробки вбудованих систем на основі STM32 вимагає глибоких знань і певного досвіду. Тому метою цього проекту є розробка ефективного підходу до віддаленої розробки вбудованих систем на платі STM32, з акцентом на безпеку, ефективність і зручність для розробників.

У першому розділі ми розглянемо загальні поняття вбудованих систем, зосереджуючись на мікроконтролерах STM32, їх програмному забезпеченні та протоколах для керування та прошивки. В другому розділі детальніше описані технології, які будуть використані для проектування системи. Третій розділ присвячений деталям розробки системи, включаючи роботу з основними інструментами та налаштування для віддаленого доступу. В останньому розділі ми зосередимося на тестуванні системи, оцінюючи її продуктивність і надійність.

Важливо зазначити, що цей проект орієнтований на дослідження віддаленої розробки вбудованих систем, що є актуальною темою в сучасному світі, де віддалена робота стає все більш поширеною.

Дана розробка має значний потенціал для підвищення ефективності навчання студентів, які вивчають вбудовані системи в університеті.

					ІАЛЦ.467200.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Навички роботи з віддаленими системами стануть цінним додатком до компетенцій студента. Це допоможе їм адаптуватися до сучасних тенденцій на ринку праці. Незважаючи на віддалений доступ, студенти все ще матимуть змогу працювати з реальним обладнанням. Це важливо для отримання практичного досвіду та глибокого розуміння роботи вбудованих систем

Таким чином, розробка системи віддаленої розробки для вбудованих систем на базі STM32 може значно покращити навчальний процес та підготовку студентів університету до вирішення реальних інженерних завдань. Нарешті, така система може дати студентам можливість реалізувати власні проекти, пов'язані з вбудованими системами. Вони зможуть побачити, як їхні ідеї втілюються в життя, що є важливим мотиватором для подальшого навчання.

					ІАЛІЦ.467200.003 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

# РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Поняття вбудованих систем

Вбудовані системи є основою сучасних технологій. Вони представляють собою спеціалізовані комп'ютерні системи, які є частиною більшої системи або продукту і виконують певну, дуже специфічну задачу. Вбудовані системи мають велике розмаїття форм і розмірів, від мікроконтролерів у пристроях для носіння до великих комп'ютерних систем, що керують промисловими процесами. Вбудовані системи відрізняються від загального призначення комп'ютерів декількома характеристиками. Вони часто розроблені для виконання одного конкретного завдання, хоча деякі можуть виконувати кілька задач. Вбудовані системи часто оптимізуються для максимальної продуктивності, надійності або ефективності, а не для загальної гнучкості або здатності виконувати широкий спектр задач. Вбудовані системи можуть бути базовими, виконуючи дуже прості задачі без особливих вимог до продуктивності або енергоспоживання. Проте вони також можуть бути дуже високотехнологічними і складними, виконуючи складні задачі, такі як керування двигунами, відео обробка, штучний інтелект, тощо.[1]

Вбудовані системи є усюди в нашому оточенні. Вони використовуються в автомобілях для контролю двигуна, в телекомунікаційному обладнанні для обробки сигналів, в медичних приладах для моніторингу пацієнтів та в багатьох інших застосуваннях. Процес розробки вбудованих систем може включати різні етапи, включаючи проектування апаратного забезпечення, написання програмного забезпечення, тестування та валідацію. Залежно від специфіки проекту, ці етапи можуть вимагати спеціалізованих знань і навичок. Віддалена розробка стає все більш поширеною у сфері вбудованих систем, особливо в контексті ІоТ. Здатність віддалено оновлювати програмне забезпечення вбудованих пристроїв, керувати ними та отримувати діагностичну інформацію з них може значно

					ІАЛЦ.467200.003 ПЗ	Арк.
						5
Зм.	Арк.	№ док.ум.	Підпис	Дата		

полегшити процес розробки та підтримки. Це також може поліпшити безпеку, дозволяючи швидко виправляти будь-які виявлені проблеми.

Програмне забезпечення вбудованих систем може включати операційну систему, драйвери периферійних пристроїв, застосунки та проміжне програмне забезпечення. Операційна система управляє ресурсами системи, драйвери периферійних пристроїв забезпечують інтерфейс між апаратним забезпеченням та програмним забезпеченням, застосунки виконують основні функції системи, а проміжне програмне забезпечення може надавати додаткові служби, такі як комунікація мережі або обробка даних. Розробка вбудованих систем може представляти ряд викликів. Одним з них є необхідність оптимізувати систему для високої продуктивності, низького споживання енергії або обмеженого простору. Інший виклик полягає в забезпеченні надійності і безпеки, особливо в критичних застосуваннях, таких як медичні прилади або автомобільні системи.

Розробка вбудованих систем потребує набору спеціалізованих навичок. Розробники повинні розуміти апаратне забезпечення, на якому вони працюють, а також мати глибоке розуміння програмування. Вони також повинні знати, як ефективно використовувати ресурси, доступні для вбудованої системи, щоб оптимізувати її продуктивність та ефективність.

Тим не менше, віддалена розробка та управління вбудованими системами вимагають уважного підходу до питань безпеки. Несанкціонований доступ до вбудованих систем може призвести до серйозних наслідків, особливо якщо ці системи використовуються в критичних застосуваннях. Розробники повинні враховувати безпеку на кожному етапі розробки, включаючи вибір протоколів комунікації, розробку механізмів аутентифікації та шифрування, а також розробку стратегій для виявлення та реагування на можливі атаки.

					ІАЛЦ.467200.003 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

## 1.2 Загальний огляд STM.

STM - це родина мікроконтролерів, розроблених компанією STMicroelectronics. Вони базуються на ядрі ARM Cortex-M, що є високоефективною та широко використовуваною архітектурою для вбудованих систем.[2]

Архітектура ARM Cortex-M включає варіанти Cortex-M0, Cortex-M3, Cortex-M4 та Cortex-M7. Кожне з цих ядер використовує RISC-архітектуру, яка забезпечує високу продуктивність при низькому енергоспоживанні.

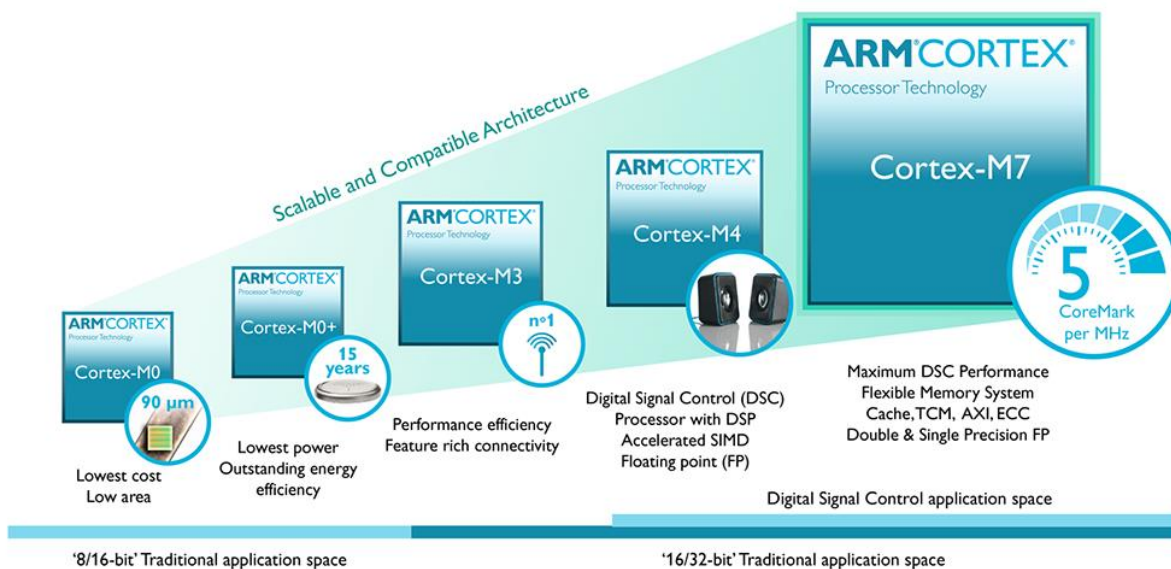


Рисунок 1.1 – Лінійка продуктів arm cortex m [2]

Cortex-M0 - це найпростіше ядро в сімействі, ідеально підходить для простих вбудованих систем. Cortex-M3 має додаткові інструкції та вищу продуктивність, що робить його вибором для більш складних застосувань. Cortex-M4 додає підтримку DSP та плаваючої точки, тоді як Cortex-M7 пропонує найвищу продуктивність в сімействі та покращену підтримку DSP та плаваючої точки.

STM32 мікроконтролери пропонують широкий спектр вбудованих периферійних пристроїв, що включають UART, USART, SPI, I2C, USB, CAN, Ethernet, SDIO, HDMI CEC і багато інших.

					ІАЛЦ.467200.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

UART, USART, SPI та I2C - це загальноприйняті протоколи передачі даних, які використовуються для комунікації з іншими пристроями або мікроконтролерами. USB, CAN, Ethernet і SDIO - це складніші протоколи, що надають високу швидкість передачі даних та підтримують складніший обмін даними.[3]

Standard	Tx Type	# Signal Wires	Data Rate & Distance	Hardware \$	Scalability	Application Example
UART	Asynchronous	2	20kbps @ 15m	Medium (transceiver)	Low (point-to-point)	Diagnostic display
LIN	Asynchronous	2	20kbps @ 40m	Medium (transceiver)	High (identifier)	Washing machine subsystem network
SPI	Synchronous	4+	25Mbps @ 0.1m	Low	Medium (chip selects)	High speed chip to chip link
I2C	Synchronous	2	1Mbps @ 0.5m	Low (resistors)	High (identifier)	System sensor network

Рисунок 1.2 – Різниця між USART, LIN, SPI, I2C [3]

Периферійні пристрої STM32 включають блоки управління живленням, модулі управління захистом, також таймери різних типів (основний, захоплення/порівняння, PWM, тощо) для часових задач та управління сигналами.

PWM контролери використовуються для генерації аналогових вихідних сигналів за допомогою цифрових засобів, що є особливо корисним при управлінні електромеханічними пристроями, такими як мотори.

Блоки управління живленням дозволяють контролювати режими живлення мікроконтролера, такі як режими сну і стримування, щоб оптимізувати споживання енергії.

Модулі управління захистом можуть бути використані для моніторингу стану системи та реагування на помилки або аномальні умови. Це може включати захист від перевищення температури, перевищення струму або перенапруги.

Родина STM32 включає в себе ряд серій, кожна з яких відрізняється своїми характеристиками. Наприклад, серія STM32F1 включає моделі з ядром Cortex-M3 та набором периферійних пристроїв, ідеально підходить для багатьох

загальних завдань у вбудованих системах. Серія STM32F4 включає моделі з ядром Cortex-M4F, що мають підтримку плаваючої точки та більш продуктивну обробку сигналів, ідеально підходить для застосувань, що вимагають високої обчислювальної потужності або обробки сигналів, такий мікроконтролер ми використовуємо. Серія STM32L має у своєму асортименті моделі, які в основному спрямовані на додатки з низьким енергоспоживанням. Їх режими економії енергії, такі як режим сну, режим пробудження та інші, дозволяють їм працювати в умовах з обмеженою енергією, як-то батарейні пристрої та інші портативні системи. STM32H7 – це найпотужніша серія в родині STM32. Вона базується на ядрі Cortex-M7 і має найвищу тактову частоту, а також включає в себе найбільший набір периферії. Ці мікроконтролери використовуються в найбільш вимогливих застосуваннях, таких як промислові контролери, авіоніка, автомобільні системи та системи медичного обладнання.

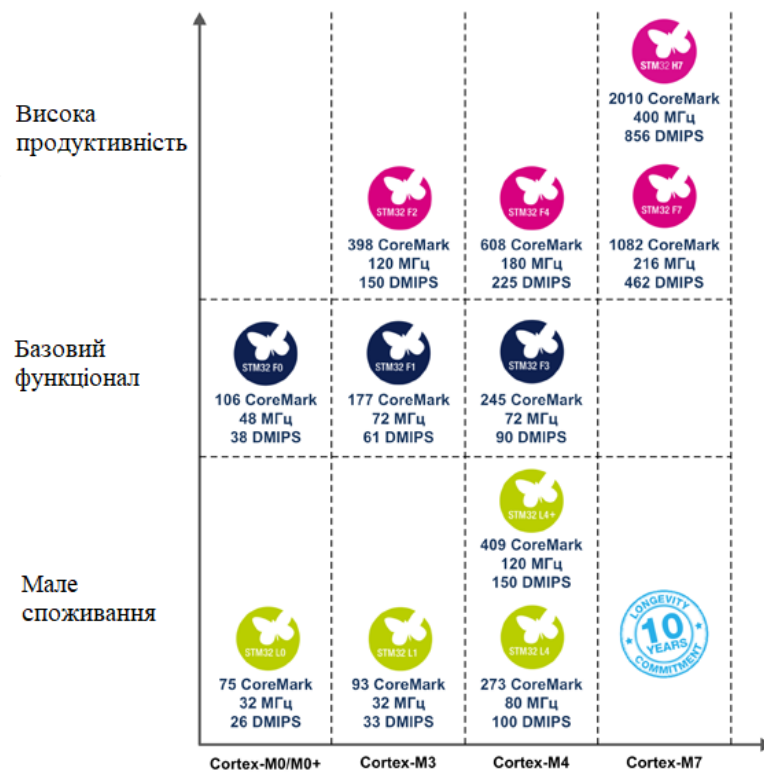


Рисунок 1.3 – Сімейство 32-розрядних мікроконтролерів STM32 [4]

Загалом, мікроконтролери STM32 відомі своєю гнучкістю, високою продуктивністю та широким набором периферійних пристроїв. Вони підтримують широкий спектр протоколів комунікації, що робить їх ідеально підходящими для вбудованих застосувань, в яких потрібне взаємодія з іншими пристроями або системами. Завдяки своїм енергоефективним характеристикам та потужним можливостям обробки, мікроконтролери STM32 можуть служити основою для широкого спектру продуктів і систем, включаючи IoT-пристрої, розумні домашні системи, автономні транспортні засоби, промислові контролюючі системи та багато іншого.[5]

Додаткові переваги STM32 включають підтримку різних джерел живлення, включаючи батареї, сонячні панелі та інші форми відновлюваної енергії, що робить їх ідеальними для використання в умовах, де доступ до електроенергії обмежений або відсутній. Окрім того, вони підтримують широкий діапазон температур робочого середовища, що дозволяє використовувати їх в різних умовах, від дуже холодних до дуже гарячих.

Для розробників програмного забезпечення STM32 пропонує багатий набір інструментів розробки, включаючи STM32Cube, екосистему, що включає графічні інтерфейси для конфігурації периферії, автоматичне генерування коду, вбудовані бібліотеки для розробки мультимедійних додатків, драйвери периферії, проміжне ПЗ для протоколів зв'язку, а також інструменти для налагодження і профілювання.

### **1.3 Програмне забезпечення для розробки на STM**

STM32CubeIDE - це основне офіційне IDE від STMicroelectronics. Воно базується на Eclipse і інтегровано з STM32CubeMX, графічним інструментом для ініціалізації коду та конфігурації периферійних пристроїв. STM32CubeIDE має багатий набір інструментів для відладки, включаючи вбудований відладчик, інструменти для аналізу профілювання та ін.[6] Воно безкоштовне для використання, має вбудований набір інструментів, спеціально розроблених

					ІАЛЦ.467200.003 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

для STM32.

STM32CubeIDE вбудовано в STM32CubeMX, що є графічним інструментом для ініціалізації коду та конфігурації периферійних пристроїв. Цей інструмент дозволяє розробникам легко налаштувати різні параметри системи, такі як тактова частота, периферійні інтерфейси, переривання та ін., а потім автоматично генерує відповідний вихідний код.



Рисунок 1.4 – Логотип STM32CubeIDE [6]

STM32CubeIDE використовує ARM GCC компілятор, що дозволяє розробникам використовувати повний потенціал процесорів STM32 та використовувати всі сучасні стандарти мови програмування C і C++.

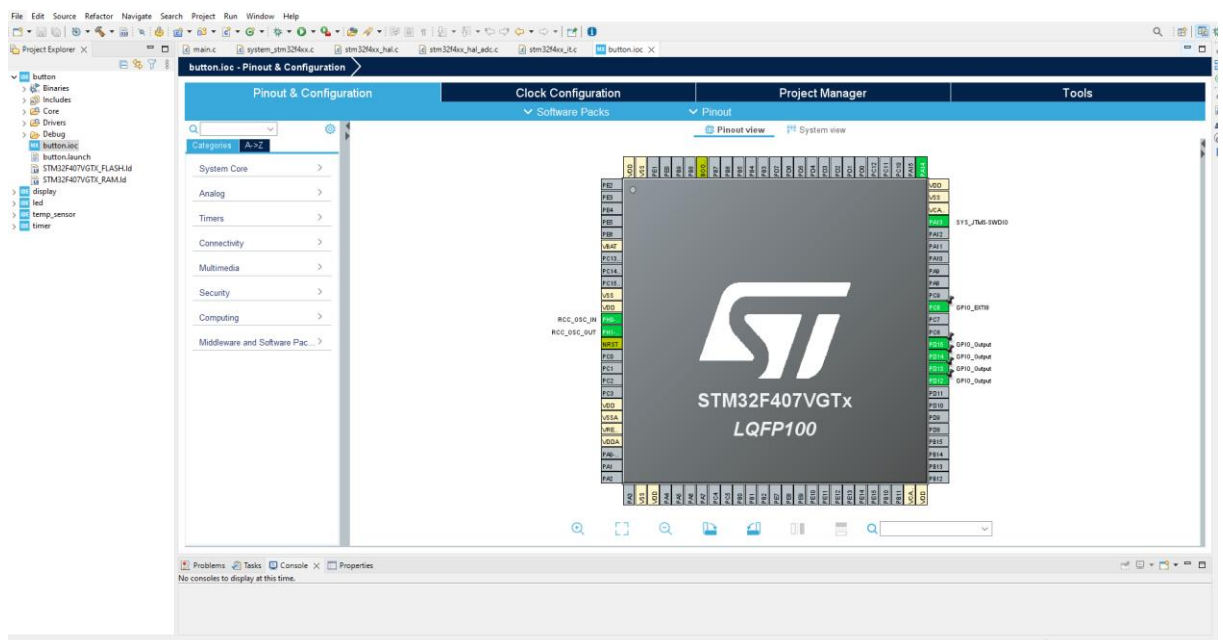


Рисунок 1.5 – Інтерфейс STM32CubeIDE

Середовище має вбудований відладчик, який підтримує JTAG та SWD інтерфейси для апаратної відладки. Він дозволяє встановлювати точки

Зм.	Арк.	№ докум.	Підпис	Дата

зупинки, виконувати крокове виконання, переглядати змінні та реєстри, аналізувати стек та ін. Також STM32CubeIDE підтримує вбудовану інтеграцію з FreeRTOS, популярною операційною системою в реальному часі, яка дозволяє розробникам легко впроваджувати багатозадачність в свої проекти.

Keil MDK-ARM - це одне з найпопулярніших комерційних IDE для розробки на базі ARM. Воно має потужний компілятор, високопродуктивний відладчик, розширений аналізатор коду та великий набір периферійних драйверів і прикладів. Це комерційний продукт від компанії Keil, що належить компанії ARM.



Рисунок 1.6 – Логотип Keil MDK [7]

Keil MDK-ARM включає в себе потужний компілятор C/C++, який оптимізований для процесорів ARM. Він підтримує всі сучасні стандарти C/C++ та надає ряд оптимізацій для покращення продуктивності та ефективності коду. Середовище розробки  $\mu$ Vision є частиною MDK-ARM і надає гнучке та зручне середовище для розробки та відладки програм. Воно включає в себе редактор коду, відладчик, симулятор, а також менеджер проектів. [7]

Keil MDK-ARM має розширені інструменти відладки, включаючи підтримку JTAG і SWD інтерфейсів, а також розширені можливості трасування, які дозволяють розробникам аналізувати виконання своїх програм у реальному часі. MDK-ARM включає бібліотеки middleware для впровадження різних протоколів та інтерфейсів, таких як USB, TCP/IP, файлові системи та ін. Воно також підтримує різні реалізації RTOS, включаючи RTX, FreeRTOS.

					ІАЛЦ.467200.003 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

IAR Embedded Workbench - інший популярний комерційний IDE, який має набір інструментів, що спрямовані на покращення ефективності коду та продуктивності розробки. Його відладчик C-SPY є одним з найпотужніших на ринку і має розширений набір можливостей для відстеження та аналізу виконання програми.[8] Embedded Workbench включає в себе C/C++ компілятор IAR, який є одним з найкращих у галузі вбудованих систем. Він надає широкий спектр оптимізацій, що дозволяють збільшити продуктивність та зменшити обсяг коду. IAR Embedded Workbench надає повністю інтегроване середовище з редактором коду, відладчиком, симулятором та менеджером проектів. Воно підтримує різні види периферії, що дозволяє розробникам швидко та ефективно створювати вбудовані системи.



**IAR Embedded Workbench**

Рисунок 1.7 – Логотип IAR Embedded Workbench [8]

IAR Embedded Workbench має розширені можливості відладки, включаючи апаратне та програмне трасування, точки зупинки, перегляд змінних та регістрів та ін. Воно також підтримує відладку на рівні ядра, що дозволяє розробникам аналізувати поведінку їхніх програм на низькому рівні. Також IAR Embedded Workbench надає розширені можливості для розробки безпечних вбудованих систем. Воно включає в себе інструменти для аналізу безпеки коду, такі як статичний аналіз коду (C-STAT) та інструменти для захисту від реверс-інжинірингу.

Всі ці IDE добре підходять для розробки вбудованих систем на мікроконтролерах STM32. Вони надають інструменти для створення, відлагодження програмного забезпечення для вбудованих пристроїв.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

## 1.4 Протоколи для керування та прошивки STM

При розробці вбудованих систем важливим аспектом є можливість віддаленого керування та прошивки мікроконтролерів. Використання спеціальних протоколів для цих задач забезпечує надійність, безпеку та ефективність процесу.

Один з таких протоколів - це UART (Universal Asynchronous Receiver-Transmitter), що використовується для асинхронного обміну даними. Завдяки своїй простоті і невеликому набору необхідних ліній UART широко використовується в вбудованих системах. Однак він має обмежену дальність і швидкість передачі.

SPI (Serial Peripheral Interface) - це інший протокол, що забезпечує швидкий обмін даними між мікроконтролером і периферійними пристроями. Він підтримує високу швидкість передачі даних, але вимагає більше ліній порівняно з UART.

I2C (Inter-Integrated Circuit) - це дволінійний протокол для обміну даними між різними компонентами в системі. I2C підтримує множинні пристрої на одній шині, але має обмежену швидкість передачі даних.

USB (Universal Serial Bus) - це широко використовуваний протокол для обміну даними і зарядки пристроїв. USB підтримує високу швидкість передачі даних, але вимагає складного апаратного та програмного інтерфейсу.

Ці технології та протоколи забезпечують різні рівні зв'язку, швидкості передачі даних, безпеку та енергоефективність для віддаленого керування та прошивки вбудованих систем. Вибір конкретного протоколу залежить від потреб і вимог конкретного застосування.

Для забезпечення безпеки віддаленого керування та прошивки вбудованих систем можуть використовуватись різні механізми, такі як шифрування даних, аутентифікація та авторизація, цифрові підписи та інші. Застосування безпекових протоколів та механізмів допомагає запобігти несанкціонованому доступу до системи та зберегти цілісність даних.

					ІАЛЦ.467200.003 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

Для ефективного процесу віддаленого керування та прошивки вбудованих систем також можуть використовуватись спеціалізовані програмні інструменти, які дозволяють віддалено керувати, моніторити та оновлювати пристрої. Ці інструменти можуть надавати графічний інтерфейс для взаємодії з вбудованою системою, включати функціональність віддаленого відладчика, забезпечувати моніторинг стану пристроїв, а також дозволяти віддалено завантажувати нові версії програмного забезпечення.

У загальному, використання спеціалізованих протоколів, технологій та інструментів дозволяє забезпечити ефективне та безпечне віддалене керування та прошивку вбудованих систем на мікроконтролерах STM32. Кожен вибір залежить від вимог конкретного проекту, рівня зв'язку, енергоефективності та інших факторів, які варто враховувати при розробці вбудованих систем.

## 1.5 Безпека віддаленої розробки

Віддалена розробка вбудованих систем на STM32 мікроконтролерах збільшує продуктивність та гнучкість, але також накладає додаткові вимоги до безпеки. Важливо забезпечити захист від несанкціонованого доступу, шифрування даних та захист від помилок. Ось декілька аспектів безпеки, що варто розглянути:

Розробники мають гарантувати, що тільки уповноважені особи мають доступ до ресурсів віддаленої розробки. Всі передані та збережені дані повинні бути зашифровані, щоб запобігти їх перехопленню або злому. Це включає в себе дані, що передаються між розробником та вбудованою системою під час процесу відлагодження або оновлення прошивки, а також будь-які дані, збережені в хмарі.

Вбудовані системи повинні бути захищені від помилок, які можуть виникнути під час процесу віддаленої розробки. Наприклад, системи OTA оновлення повинні мати механізми відновлення в разі невдачі оновлення, такі як двійкова прошивка (dual-bank firmware) або безпечний режим завантаження

					ІАЛЦ.467200.003 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

(safe boot mode).

Моніторинг важливі для виявлення та розслідування можливих проблем безпеки. Журнали доступу, журнали операцій та системи моніторингу в реальному часі можуть надати важливу інформацію про ненормальну поведінку або спроби несанкціонованого доступу. Системи обмеження та детекції вторгнень (IDS) можуть стежити за мережевим трафіком та поведінкою системи, щоб виявити та відреагувати на ворожі дії. Це може бути особливо важливо для вбудованих систем, які з'єднані з Інтернетом і тому відкриті для потенційних атак. Нарешті, не забуваймо про фізичну безпеку мікроконтролера і асоційованого обладнання. Віддалена розробка може включати доступ до пристроїв, які знаходяться в відкритих або слабо захищених місцях, і фізичний захист цих пристроїв повинен бути врахований.

Зрештою, безпека є критично важливим аспектом віддаленої розробки вбудованих систем на STM32, і вона вимагає постійного розуміння та уваги до поточних найкращих практик та стандартів безпеки.

					ІАЛЦ.467200.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВОК ДО РОЗДІЛУ 1

У даному розділі дипломної роботи було розглянуто різні аспекти віддаленої розробки вбудованих систем на платі STM Discovery. Починаючи з огляду поняття вбудованих систем, було встановлено їх важливість та широке застосування у сучасному світі. Далі було проведено загальний огляд плати STM Discovery, яка є популярним інструментом для розробки вбудованих систем, і були висвітлені її основні характеристики.

Також було досліджено програмне забезпечення для розробки на STM, включаючи інтегроване середовище розробки (IDE) які допомагають розробникам в ефективному створенні програмного забезпечення для вбудованих систем на платі STM Discovery. Під час дослідження було встановлено, що плата STM Discovery має широкий набір інструментів та можливостей для розробки вбудованих систем. Інтегроване середовище розробки, таке як STM32CubeIDE, надає зручний інтерфейс для написання коду, налагодження та тестування програмного забезпечення. Мови програмування, такі як C або C++, є популярними і ефективними інструментами для розробки на STM.

В рамках розділу було також розглянуто протоколи для керування та прошивки STM, що є важливим етапом розробки вбудованих систем. Були розглянуті різні протоколи з комунікації, які можуть бути використані для взаємодії з вбудованою системою на платі STM Discovery, а також процес прошивки, який дозволяє завантажити програмне забезпечення на пристрій. Протоколи для керування та прошивки STM, такі як UART, SPI, I2C, USB, дозволяють забезпечити зручну і надійну комунікацію між вбудованою системою та зовнішніми пристроями або системами. Процес прошивки є необхідним етапом розробки, що дозволяє завантажити програмне забезпечення на плату STM Discovery для подальшого тестування та експлуатації.

					ІАЛЦ.467200.003 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім того, було приділено увагу безпеці віддаленої розробки, оскільки це актуальна проблема у сучасному світі, де розробка вбудованих систем все частіше здійснюється віддалено. Було розглянуто різні аспекти безпеки, такі як захист від несанкціонованого доступу, шифрування комунікації та захист від зловживань. Віддалена розробка вбудованих систем ставить питання безпеки на передній план. Захист від несанкціонованого доступу, шифрування комунікації та захист від зловживань є важливими аспектами безпеки віддаленої розробки. Застосування відповідних заходів безпеки може гарантувати захищеність розробки і використання вбудованої системи.

Загальною метою даного розділу був аналіз різних аспектів віддаленої розробки, починаючи з вибору відповідного програмного забезпечення для розробки на STM, використання протоколів для керування та прошивки, а також забезпечення безпеки процесу.

					ІАЛЦ.467200.003 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

# РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ ПРОЕКТУВАННЯ СИСТЕМИ

## 2.1 Прошивка STM мікроконтролерів через консоль.

Розглянемо різні інструменти, які можна використовувати для прошивки мікроконтролерів STM. Прошивка мікроконтролерів є важливим кроком у розробці вбудованих систем і додаванні нових функціональних можливостей до пристроїв. Зосередимося на використанні наступних інструментів: STM32CubeProgrammerCLI, STM32Flash, OpenOCD та ST-Link Utility. Кожен з цих інструментів має свої особливості, переваги та обмеження.

STM32CubeProgrammerCLI є командним рядком (CLI) інтерфейсом для STM32CubeProgrammer, який є офіційним інструментом від STMicroelectronics для програмування та керування мікроконтролерами STM32. Він надає розширені можливості для автоматизації процесу прошивки та керування вбудованими системами на мікроконтролерах STM32.[9]

STM32CubeProgrammerCLI підтримує різні команди та параметри, що дозволяють виконувати специфічні операції залежно від потреб розробки. Цей інтерфейс може бути використаний у командному рядку оперативної системи або вбудований в скрипти для автоматичного виконання певних дій.

Наприклад, використовуючи STM32CubeProgrammerCLI, розробники можуть програмувати мікроконтролер STM32 з використанням команди "program", передаючи шлях до файлу прошивки та інші параметри. Вони також можуть сконфігурувати пини та периферійні пристрої, використовуючи команду "setpin" або "setoption", або зчитати налаштування пам'яті, використовуючи команду "readmem".

STM32CubeProgrammerCLI підтримує різні типи підключення до мікроконтролера, включаючи USB, UART, SWD, JTAG та інші.[9] Це дозволяє розробникам використовувати відповідний інтерфейс для підключення та взаємодії з мікроконтролером залежно від доступних можливостей та

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

конфігурації системи.

Окрім основних операцій програмування та керування, STM32CubeProgrammerCLI також надає додаткові функції, такі як розширена інформація про мікроконтролер, зчитування та запис налаштувань Flash за допомогою XML-файлів, визначення властивостей пристрою та багато іншого.

STM32CubeProgrammerCLI також підтримує автоматизацію за допомогою сценаріїв або скриптів. Це означає, що розробники можуть створювати скрипти, які виконують послідовність команд для програмування та керування мікроконтролерами. Це особливо корисно при розгортанні та віддаленому керуванні великою кількістю пристроїв.

STM32CubeProgrammerCLI також підтримує різні формати файлів для програмування мікроконтролерів, включаючи бінарні файли, файливе представлення Intel Hex та Motorola S-record. Це дозволяє використовувати різні інструменти та середовища для створення програмного коду та конфігурацій, а потім використовувати STM32CubeProgrammerCLI для програмування цих файлів на мікроконтролери STM32.

STM32CubeProgrammerCLI є потужним інструментом для автоматизації процесу програмування та керування мікроконтролерами STM32. Він забезпечує розробникам можливість виконувати різноманітні операції з вбудованими системами з використанням командного рядка, що спрощує та прискорює розробку та тестування вбудованих систем на мікроконтролерах STM32.

Плюси:

- Це офіційний інструмент від STMicroelectronics, тому він гарантує сумісність зі всіма STM32 мікроконтроллерами.
- Підтримує різні інтерфейси, включаючи JTAG, SWD, USART, USB (DFU mode).
- Має і графічний, і командний інтерфейс.

					ІАЛЦ.467200.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

Мінуси:

- Не є відкритим кодом.
- Може бути складним для новачків через багатий набір функцій.

STM32Flash є інструментом командного рядка, який призначений для прошивки (запису) програмного забезпечення на мікроконтролери STM32. Він є відкритим програмним забезпеченням і доступний для використання на різних операційних системах, включаючи Windows, Linux і macOS.

Основна функціональність STM32Flash полягає у записі програмного забезпечення (фірмвару) на флеш-пам'ять мікроконтролерів STM32. Він підтримує різні формати файлів, такі як бінарні файли, файлове представлення Intel Hex і Motorola S-record. Крім того, він може читати, видаляти та перевіряти флеш-пам'ять на мікроконтролері.

STM32Flash підтримує різні інтерфейси для з'єднання з мікроконтролерами STM32, включаючи RS232, USB-Serial адаптери, ST-Link/V2 програматори, CMSIS-DAP програматори та інші. Це дає розробникам гнучкість у виборі методу підключення до мікроконтролера залежно від доступного обладнання.

Крім звичайного процесу прошивки, STM32Flash також має деякі додаткові можливості. Наприклад, він може виконувати запис частини фірмвару, що дозволяє оновлювати лише певні частини програми, не впливаючи на решту. Він також підтримує параметризовані команди для зчитування та запису регістрів мікроконтролера, що дозволяє здійснювати операції з внутрішніми регістрами безпосередньо через STM32Flash.

STM32Flash є корисним інструментом для розробників, які працюють з мікроконтролерами STM32 і потребують швидкого та зручного способу прошивки програмного забезпечення. Він надає простий та ефективний спосіб виконати операції з флеш-пам'яттю мікроконтролера, включаючи запис, читання, видалення та перевірку. Завдяки своїй підтримці різних інтерфейсів і форматів файлів, він дозволяє розробникам легко взаємодіяти з мікроконтролерами STM32 та виконувати операції прошивки з різних

					ІАЛЦ.467200.003 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

платформ.

STM32Flash має простий та зрозумілий синтаксис командного рядка, що спрощує використання інструмента. Розробники можуть виконувати команди STM32Flash безпосередньо з командного рядка або включати його в скрипти для автоматизації процесу прошивки та керування мікроконтролерами.

Загалом, STM32Flash є потужним інструментом для прошивки мікроконтролерів STM32 з використанням командного рядка. Він надає розробникам гнучкість і контроль над процесом прошивки, дозволяючи ефективно працювати з програмним забезпеченням мікроконтролерів STM32.

OpenOCD (Open On-Chip Debugger) - це відкрите програмне забезпечення, яке забезпечує можливість відлагодження та програмування вбудованих систем через інтерфейс JTAG, SWD або другі підтримувані інтерфейси. Воно є популярним інструментом для роботи з мікроконтролерами, включаючи мікроконтролери STM32.[10]

OpenOCD надає широкий набір функціональності для розробників, включаючи такі можливості:

- Відладка: OpenOCD дозволяє використовувати інтерфейс JTAG або SWD для здійснення відладки програми на мікроконтролері. Він підтримує встановлення точок зупинки, крокове виконання коду, перегляд регістрів та пам'яті, а також аналіз стеку в реальному часі. Це дозволяє розробникам виявляти та усувати помилки в програмному забезпеченні та виконувати детальний аналіз роботи мікроконтролера.
- Програмування: OpenOCD дозволяє завантажувати програмне забезпечення (прошивку) на мікроконтролери. Він підтримує різні формати файлів, такі як Intel Hex, Binary та ELF, що дозволяє розробникам легко завантажувати програмне забезпечення на мікроконтролер з використанням OpenOCD.
- Керування ресурсами: OpenOCD дозволяє керувати ресурсами

					ІАЛЦ.467200.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

мікроконтролера, такими як тактова частота, периферійні пристрої та інші налаштування. Він надає можливість змінювати налаштування мікроконтролера в режимі реального часу, що є корисним для тестування різних конфігурацій та оптимізації роботи системи.

- Мультицільова підтримка: OpenOCD підтримує багато різних мікроконтролерів та платформ, що робить його мультицільовим інструментом для відладки та програмування вбудованих систем. Він підтримує широкий спектр мікроконтролерів STM32, включаючи різні серії, такі як STM32F, STM32L, STM32H і багато інших.
- Розширені можливості: OpenOCD має багато розширених можливостей для розробників. Він підтримує роботу зі зовнішніми цілями (target) для відлагодження та програмування, можливість налаштування параметрів JTAG та SWD, підтримку різних типів адаптерів JTAG/SWD та ін.
- Відкрите програмне забезпечення: OpenOCD є відкритим проектом з вільною ліцензією, що дозволяє його використання та модифікацію розробниками. Відкрите програмне забезпечення сприяє співпраці та спільному вдосконаленню інструменту з боку спільноти розробників.

Загалом, OpenOCD є потужним інструментом для відладки та програмування мікроконтролерів, зокрема мікроконтролерів STM32. Він надає розробникам широкий набір функціональності, гнучкість та можливість роботи з багатьма цілями. Будучи відкритим проектом, OpenOCD також стимулює співпрацю та обмін знаннями серед спільноти розробників.

Плюси:

- OpenOCD є відкритим кодом, тому ви можете налаштувати його відповідно до своїх потреб.
- Підтримує багато мікроконтролерів, не лише STM32, але й багато інших.
- Може використовуватися для прошивки через JTAG або SWD інтерфейси.

					ІАЛЦ.467200.003 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

Мінуси:

- Може бути складним для налаштування, особливо для новачків.

ST-Link Utility - це інструмент, розроблений компанією STMicroelectronics, який надає можливості для програмування та відладки мікроконтролерів STM32, які підтримують протокол ST-Link. Він пропонує користувачу зручний графічний інтерфейс, який дозволяє виконувати різноманітні операції з мікроконтролерами, такі як прошивка, зчитування, запис та відлагодження програмного коду.

Основні функції ST-Link Utility включають:

- Прошивка (Flash): Інструмент дозволяє завантажувати бінарні файли програмного коду (файли прошивки) на мікроконтролер STM32. Користувач може обрати різні опції прошивки, такі як очищення пам'яті, встановлення захисту від запису або виконання, перевірка контрольної суми та багато інших.
- Зчитування (Read): За допомогою ST-Link Utility можна зчитувати програмний код з мікроконтролерів STM32. Це може бути корисним для резервного копіювання програмного коду або аналізу наявного коду.
- Запис (Write): Інструмент дозволяє записувати новий програмний код на мікроконтролер STM32. Це може бути використано для оновлення програми або встановлення нової версії програмного забезпечення на мікроконтролер.
- Видалення (Erase): ST-Link Utility надає можливість видалення програмного коду з пам'яті мікроконтролера. Це може бути корисним при необхідності повного стирання програми або при виправленні помилок під час прошивки.
- Відлагодження (Debugging): Інструмент підтримує функцію відлагодження для мікроконтролерів STM32, які підтримують протокол ST-Link. Користувач може встановлювати точки зупинки, кроку виконання коду, спостерігати за значеннями регістрів та змінними, а

					ІАЛЦ.467200.003 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

також аналізувати стек викликів і режими працездатності. ST-Link Utility надає зручний інтерфейс для цих операцій, що дозволяє розробникам ефективно відлагоджувати програми на мікроконтролерах STM32.

- Конфігурація і налаштування: Інструмент ST-Link Utility також дозволяє налаштовувати різні параметри пристрою ST-Link, такі як швидкість передачі даних, тип підключення (JTAG або SWD), напругу живлення та інші. Це дозволяє забезпечити оптимальну роботу мікроконтролера та ST-Link пристрою під час прошивки та відлагодження.

Після аналізу дослідження різних інструментів для прошивки STM мікроконтролерів, прийнято рішення використовувати STM32CubeProgrammerCLI. Це рішення базується на декількох факторах:

- Підтримка мікроконтролерів STM32. STM32CubeProgrammerCLI розроблений самою компанією STMicroelectronics, яка є виробником мікроконтролерів STM32.
- Командний рядок інтерфейс. STM32CubeProgrammerCLI надає командний рядок інтерфейс, що дозволяє автоматизувати процес прошивки та інших операцій з мікроконтролерами STM32.
- Розширені функції. STM32CubeProgrammerCLI має широкий спектр функціональних можливостей, які дозволяють виконувати різні операції з мікроконтролерами STM32.

Враховуючи ці фактори, вибір STM32CubeProgrammerCLI є обґрунтованим, оскільки він забезпечує надійну підтримку мікроконтролерів STM32, має командний рядок інтерфейс для автоматизації та надає розширені функціональні можливості для роботи з мікроконтролерами.

					ІАЛЦ.467200.003 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2.2 Програмне забезпечення для відеотрансляції.

У цьому підрозділі розглянемо різні програмні засоби для відеотрансляції, що дозволяють нам отримати, відеодані про те що відбувається з платою.

Zoom є веб-платформою, що надає можливість спілкуватися з людьми в режимі реального часу через відео- та аудіоконференції. Це дозволяє людям з різних місць зустрітися в одному віртуальному просторі і провести обговорення, зустрічі або навіть викладати онлайн-уроки.[11]

За допомогою Zoom ви можете проводити конференції з однією або багатьма учасниками. Платформа пропонує велику кількість корисних функцій, таких як відео та аудіо збори, чат, обмін файлами та показ екрана. Це дає можливість учасникам спілкуватися, обговорювати питання, спільно працювати над проектами і багато іншого.

Zoom підтримує якісне відео та аудіо, що залежить від швидкості та якості вашого Інтернет-підключення. Ви можете підключатися до Zoom з різних пристроїв, таких як комп'ютери, смартфони та планшети, завдяки спеціальним додаткам, які доступні для операційних систем Windows, macOS, Linux, iOS та Android.

У зусиллях забезпечити безпеку, Zoom постійно вдосконалює свої заходи захисту. Наприклад, вони використовують шифрування даних, що забезпечує приватність під час пересилання інформації. Також можна встановити паролі для конференцій та обмежити доступ до зустрічей за допомогою ID конференцій та спеціальних лінків.

Zoom знайшов широке застосування у різних сферах. Він використовується для особистого спілкування з родиною та друзями, а також для бізнес-зустрічей, онлайн-уроків, вебінарів та інших форматів спілкування.

У Zoom є різні режими перегляду, які дозволяють вам налаштувати те, як ви бачите учасників конференції.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

#### Переваги:

- **Сумісність:** Zoom є сумісним з багатьма операційними системами, включаючи Windows, MacOS, Android і iOS. Це дозволяє користувачам спілкуватися майже з будь-якого пристрою.
- **Якість відео:** Zoom відомий своєю високою якістю відео та аудіо, навіть при використанні з поганою інтернет-з'єднанням.
- **Функції співпраці:** Zoom має багато інструментів для співпраці, включаючи спільне використання екрану, віртуальні дошки та систему повідомлень в реальному часі.
- **Масштабованість:** Zoom може підтримувати велику кількість учасників - від невеликих команд до великих вебінарів на тисячі осіб.

#### Недоліки:

- **Проблеми з безпекою:** Zoom зіткнувся з кількома питаннями щодо безпеки та конфіденційності в минулому, включаючи неправильне управління даними користувачів і "Zoombombing", коли незапрошені гості вторгалися в приватні зустрічі.
- **Комплексність установки:** Деякі користувачі зазначають, що початкова настройка та використання Zoom може бути дещо складним.
- **Обмеження безкоштовної версії:** Безкоштовна версія Zoom має обмеження в 40 хвилин на групові зустрічі, що може бути незручно для більших зустрічей або вебінарів.

Vimeo Livestream[12] - це сервіс, що дозволяє користувачам стрімити відео в реальному часі через інтернет. Він належить компанії Vimeo, яка відома своєю платформою для завантаження, обміну і перегляду відео високої якості.

З використанням Vimeo Livestream, користувачі можуть надсилати живий відеопотік великому числу глядачів, незалежно від їх місцезнаходження. Сервіс підтримує широкий спектр відеоформатів, включаючи відео високої чіткості та 4К.

					ІАЛЦ.467200.003 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

Vimeo Livestream надає велику гнучкість для трансляції відео. Користувачі можуть транслювати відео прямо зі своєї веб-камери, використовуючи вбудоване веб-програмне забезпечення, або вони можуть використовувати спеціалізоване обладнання для професійної відеопродукції.

Vimeo Livestream також надає різні інструменти для інтеракції з глядачами. Наприклад, він включає вбудовану функцію чату, яка дозволяє глядачам коментувати і задавати питання в реальному часі під час трансляції.



Рисунок 2.1 – Логотип Vimeo-Livestream [12]

Щодо налаштувань приватності, користувачі можуть вибрати, хто зможе дивитися їх трансляції. Вони можуть обмежити доступ до трансляції за допомогою пароля, або вони можуть зробити трансляцію приватною і надіслати посилання тільки обраним особам.

Vimeo Livestream також дозволяє користувачам інтегруватися з різними соціальними медіа платформами. Користувачі можуть одночасно транслювати своє відео на Vimeo Livestream і на інших платформах, таких як Facebook, YouTube або LinkedIn, щоб досягти більшої аудиторії. Vimeo Livestream має вбудовану аналітику, яка дозволяє користувачам отримувати важливу інформацію про їх глядачів і про те, як вони взаємодіють з відео.

Однією з ключових особливостей Vimeo Livestream є його здатність до автоматичного архівування відео після трансляції. Це означає, що користувачі можуть легко переглядати, завантажувати та поділитися своїми відео після трансляції. Крім того, їх глядачі також можуть переглядати пропущені трансляції за своїм розсудом.

					ІАЛЦ.467200.003 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

Сервіс Vimeo Livestream надає великий спектр можливостей для підтримки відеострімінгу, з яким можна працювати. Він має вбудовані інструменти для управління відео, створення плейлистів, планування трансляцій та навіть для створення сповіщень про початок трансляцій.

Крім того, Vimeo Livestream пропонує преміальні сервіси, такі як "Vimeo Premium" та "Vimeo Enterprise". Вони включають додаткові можливості, такі як підтримка технічної служби 24/7, автоматичне кодування відео, віртуальна продакшн-студія, персоналізовані аналітичні панелі та багато іншого.

Незалежно від того, які потреби у користувача, Vimeo Livestream має широкий спектр інструментів та функцій, які можуть допомогти користувачам ефективно стрімити відео в реальному часі та залучати свою аудиторію.

Переваги:

- **Якість відео:** Vimeo Livestream підтримує відео високої якості, включаючи 4K відео.
- **Масштабованість:** Vimeo Livestream може підтримувати тисячі одночасних переглядів, що робить його відмінним вибором для великих заходів.
- **Налаштування приватності:** Vimeo Livestream дозволяє користувачам контролювати, хто може переглядати їхні стріми, завдяки набору налаштувань приватності.
- **Інтеграції:** Vimeo Livestream може інтегруватися з різними платформами, такими як Facebook, YouTube, LinkedIn, що дозволяє користувачам транслювати відео на декількох платформах одночасно.
- **Гнучкість:** Користувачі можуть транслювати відео безпосередньо з веб-камери або використовувати більш складне обладнання для професійного відеострімінгу.

					ІАЛЦ.467200.003 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

Недоліки:

- Вартість: В порівнянні з деякими іншими платформами, Vimeo Livestream може бути дорожчим. Безкоштовна версія має досить обмежені можливості.
- Комплексність: Деякі користувачі зазначають, що Vimeo Livestream може бути складним для використання, особливо для тих, хто вперше працює з відеострімінгом.
- Технічна підтримка: Деякі користувачі скаржаться, що технічна підтримка може бути повільною або недостатньою, особливо у вихідні.
- Вимоги до швидкості інтернету: Для трансляції відео високої якості потрібен швидкий інтернет-з'єднання, що може бути проблемою для деяких користувачів.
- Безпека: Хоча Vimeo Livestream пропонує кілька налаштувань приватності, він не має такого рівня безпеки, як деякі інші платформи. Наприклад, він не підтримує двофакторну автентифікацію.

OBS Studio, відомий також як Open Broadcaster Software Studio[13], є високоякісним інструментом для запису та потокового відтворення відео. Програмний продукт є відкритим, що означає, що користувачі можуть безкоштовно завантажити, використовувати та модифікувати його.

Це програмне забезпечення створене з великою гнучкістю та потужністю, що дозволяє користувачам налаштовувати власні сцени та джерела відео. Користувачі можуть обирати та міксувати відео з різних джерел, таких як веб-камери, вікна програм, відеофайли, екрани комп'ютера та інше.

В OBS Studio також включені інструменти змішування аудіо, які дозволяють користувачам контролювати звукові потоки з різних джерел, регулюючи гучність, застосовуючи фільтри тощо.

					ІАЛЦ.467200.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

У OBS Studio є можливість розширення за допомогою плагінів. Користувачі можуть встановлювати плагіни, щоб додати додаткові функції або покращити існуючі. OBS Studio також підтримує потокову трансляцію відео на популярні платформи, такі як Twitch, YouTube та Facebook. Користувачі можуть налаштувати параметри трансляції та навіть планувати трансляції в майбутньому.

Загалом, OBS Studio - це надзвичайно гнучкий та потужний інструмент для запису та потокового відтворення відео, що має широкий спектр можливостей і використовується в різних галузях, від ігрової індустрії до освітніх установ та корпоративного сектору.

#### Переваги OBS Studio:

- **Безкоштовність:** OBS Studio є повністю безкоштовним та відкритим програмним забезпеченням, що робить його доступним для всіх користувачів.
- **Гнучкість налаштувань:** OBS Studio пропонує гнучкі налаштування зняття і трансляції відео, дозволяючи користувачам змінювати такі параметри як роздільна здатність, частоту кадрів, бітрейт, і багато іншого.
- **Підтримка множини джерел відео:** OBS Studio підтримує велику кількість відеоджерел, включаючи веб-камери, екрани, окремі вікна, зображення, текст, браузері та багато іншого.
- **Сцени та джерела:** OBS дозволяє вам створювати різні сцени з різними джерелами, що дає вам можливість легко перемикатися між різними відеоконтентами під час стрімінгу.
- **Підтримка різних платформ:** OBS Studio підтримує багато платформ трансляцій, включаючи YouTube, Twitch, Facebook та інші.

#### Недоліки OBS Studio:

- **Складність використання:** OBS Studio може бути досить складним для новачків через багато функцій та налаштувань. Його інтерфейс також

					ІАЛЦ.467200.003 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

може бути неінтуїтивним для деяких користувачів.

- Вимоги до системних ресурсів: OBS Studio може вимагати значних системних ресурсів, особливо при високій роздільній здатності або частоті кадрів. Це може призвести до проблем з продуктивністю на слабкіших системах.
- Відсутність підтримки: Оскільки OBS Studio є безкоштовним програмним забезпеченням, він не має формальної служби підтримки. Хоча існує спільнота, що підтримує користувачів, пошук відповідей на специфічні питання може бути складним.

У цьому підрозділі ми розглянули три основні програми для відео трансляції - Vimeo Livestream, Zoom та OBS Studio. Кожна з них має свої унікальні характеристики та можливості, що робить їх ідеальними для певних ситуацій.

Vimeo Livestream має потужні можливості, що включають високоякісну трансляцію з різноманітними настройками. Проте, ця програма може бути надмірною для більш простих завдань та потребує платної підписки.

OBS Studio є відкритим джерелом і пропонує безмежні налаштування та кастомізацію. Проте, її інтерфейс може бути складним для новачків і вимагає більше технічної експертизи.

Zoom, з іншого боку, має досить простий в обробці інтерфейс і легко інтегрується з багатьма платформами. Це робить Zoom ідеальним для прямих відео трансляцій процесу прошивки плати, що можуть бути легко організовані і спостережені.

Отже, на основі нашого аналізу, Zoom виявився найкращою програмою для наших конкретних потреб. Ефективність, простота використання та гнучкість Zoom роблять його ідеальним інструментом для відео трансляцій процесу прошивки плати.

					ІАЛЦ.467200.003 ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2.3 Протоколи передачі даних через мережу

SSH [14], що розшифровується як Secure Shell, - це мережевий протокол, який надає безпечне середовище для обміну даними між двома системами по небезпечному каналу, такому як Інтернет. SSH підтримує безпечний віддалений вхід, безпечне копіювання файлів та інші схожі функції.

В основному, протокол SSH працює так:

1. Установка з'єднання: Коли клієнт підключається до сервера, вони обмінюються інформацією про версію протоколу і набори шифрувальних алгоритмів, які вони підтримують.
2. Узгодження ключів: За допомогою алгоритму обміну ключами вони створюють спільний секретний ключ, який використовується для шифрування всіх подальших спілкувань в цьому сеансі.
3. Аутентифікація: Зазвичай, сервер аутентифікується перед клієнтом, надсилаючи свій відкритий ключ, який клієнт може перевірити. Потім клієнт аутентифікується перед сервером, зазвичай за допомогою пароля або використовуючи пару ключів.
4. Засідання: Після вдалої аутентифікації обидві сторони можуть почати обмін даними. Всі дані шифруються за допомогою спільного ключа, який було згенеровано раніше.
5. Завершення засідання: Після завершення обміну даними сеанс завершується. Ключі, які були використані в цьому сеансі, зазвичай видаляються і не використовуються в майбутніх сеансах.

Існує багато реалізацій SSH, включаючи вільно доступні та комерційні варіанти. Деякі з них включають OpenSSH, PuTTY та Tectia SSH.

SSH може бути налаштований для підтримки різних методів аутентифікації, включаючи аутентифікацію за допомогою паролю, аутентифікацію на основі відкритого ключа, а також багатофакторну аутентифікацію.

					ІАЛЦ.467200.003 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

SSH також підтримує тунелювання, що дозволяє енкапсулювати інші мережеві протоколи і передавати їх через безпечне з'єднання SSH. Це може включати просту пересилку портів, X11 тунелювання для віддалених графічних сеансів, або навіть комплексніше тунелювання, таке як VPN на основі SSH.

Під час свого життя, SSH пройшов через декілька версій. Перша версія була випущена в 1995 році, але вона мала декілька відомих проблем з безпекою. Сучасна версія, SSH-2, випущена в 2006 році, і її використовують як стандарт у більшості сучасних систем.

SSH-2, що використовується сьогодні, є значно більш безпечною та гнучкою, ніж її попередниця[15]. Вона підтримує кілька різних методів шифрування, включаючи 3DES, Blowfish, AES, і CAST128. Вона також підтримує використання різних алгоритмів обміну ключами та алгоритмів аутентифікації.

Загалом, SSH допомагає забезпечити приватність і цілісність даних при передачі між двома системами. Шифрування допомагає захистити передавані дані від перехоплення, а цифровий підпис, що створюється при аутентифікації, гарантує, що дані не були змінені під час передачі.

Останнє, але не менш важливе, SSH також підтримує передачу файлів за допомогою протоколів SFTP або SCP, що стають незамінними інструментами для адміністраторів систем і розробників, яким потрібно безпечно переміщати файли між різними системами.

SSH надає багато переваг, але також має деякі недоліки. Нижче наведено детальний опис.

Переваги SSH:

- **Безпека:** SSH використовує потужні методи шифрування для захисту даних від перехоплення і зміни. Він також використовує аутентифікацію на основі пароля або ключа для перевірки ідентифікації користувача.

					ІАЛЦ.467200.003 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

- Тунелювання: SSH може захищати різноманітні види мережевого трафіку через тунелювання, включаючи X11 та TCP/IP з'єднання.
- Передача файлів: SSH підтримує безпечну передачу файлів за допомогою протоколів SCP та SFTP.
- Універсальність: SSH підтримується більшістю операційних систем, включаючи Unix/Linux, MacOS та Windows.

#### Недоліки SSH:

- Складність налаштування: Для забезпечення максимальної безпеки SSH може вимагати складного налаштування, особливо якщо використовуються методи аутентифікації на основі ключів.
- Потенційні проблеми з безпекою: Незважаючи на загальну безпеку SSH, існують певні потенційні слабкі місця, такі як атаки "людина посередині" або проблеми з безпечним зберіганням приватних ключів.
- Продуктивність: Шифрування та дешифрування даних може вимагати значних ресурсів обчислювальної потужності, що може сповільнити передачу даних.
- Безпека стосовно пристрою користувача: Якщо пристрій користувача скомпрометовано, зловмисники можуть отримати доступ до приватних ключів або паролів.

Telnet (Telecommunication Network) - це мережевий протокол, який використовується в Інтернеті або місцевих мережах для надання двостороннього інтерактивного текстового спілкування за допомогою віртуального терміналу[16]. Зазвичай він використовується для віддаленого керування серверами або для доступу до мережевого обладнання.

Телнет створений як симетричний протокол: на різних кінцях передачі немає відмінностей, крім того, що одна сторона надає інтерфейс користувача, а інша - виконує дії. Всі спілкування між сторонами здійснюються в текстовому форматі.

					ІАЛЦ.467200.003 ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

Телнет використовує механізм неготованого з'єднання, який дозволяє обмінюватися опціями між клієнтом і сервером. Наприклад, сервер може попросити клієнта використовувати певні типи хешування символів або режимів введення даних.

Після встановлення з'єднання, дані передаються у вигляді тексту між клієнтом і сервером. Клієнт вводить команди, які передаються на сервер для виконання, а результати повертаються назад до клієнта.

Зверніть увагу, що протокол Telnet не надає жодного шифрування для переданих даних, що означає, що всі передані дані, включаючи паролі або іншу чутливу інформацію, можуть бути перехоплені і прочитані безпосередньо.

Велику частину свого часу Telnet був стандартним протоколом для віддаленого керування системами в Інтернеті. Однак, через свої проблеми з безпекою, він в основному був замінений на SSH в сучасних системах.

Існує багато реалізацій Telnet, як вільно доступні, так і комерційні. Найпоширенішою вільно доступною реалізацією є util-linux, яка доступна для більшості дистрибутивів Linux. В Windows Telnet також доступний як вбудований компонент, хоча він може бути відключений за замовчуванням.

Протокол Telnet використовує порт 23 за замовчуванням, хоча може бути налаштований на використання будь-якого іншого порту. Це полегшує встановлення з'єднання з сервером Telnet, якщо обидва кінці знають, який порт використовується.

Telnet дозволяє встановити з'єднання з віддаленим сервером і ввести команди текстом, які виконуються на сервері. Результати цих команд потім повертаються клієнту, дозволяючи користувачеві бачити вивід команди, як би він працював безпосередньо на сервері.

Однак, через відсутність шифрування в Telnet, всі ці команди і вивід видно в явному вигляді всім, хто перехоплює трафік між клієнтом та сервером. Це ставить користувача на ризик перехоплення його інформації, включаючи імена користувачів і паролі.

					ІАЛЦ.467200.003 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Телнет, хоча й вважається застарілим у порівнянні з більш безпечними протоколами, такими як SSH, все ще широко використовується для певних застосувань, зокрема для діагностики мережевих проблем або для роботи з певним мережевим обладнанням.

#### Переваги Telnet:

- Сумісність: Telnet є одним з найстаріших протоколів для віддаленого керування, і він підтримується майже всіма мережевими пристроями.
- Простота: Telnet простий в використанні та налаштуванні. Він не вимагає складного налаштування або додаткового програмного забезпечення.
- Текстовий інтерфейс: Telnet працює з текстовими інтерфейсами, які зазвичай менш ресурсоємкі, ніж графічні інтерфейси.

#### Недоліки Telnet:

- Відсутність шифрування: Telnet передає всі дані, включаючи імена користувачів та паролі, в явному вигляді, що робить його вразливим для атак перехоплення.
- Відсутність аутентифікації: Telnet не має вбудованих механізмів аутентифікації, що робить його вразливим для несанкціонованого доступу.
- Відсутність захисту від атак: Telnet не має вбудованих механізмів захисту від атак типу "людина посередині" або "повторення".
- Обмежена функціональність: Telnet є простим протоколом, який не має багатьох сучасних функцій, доступних в інших протоколах віддаленого доступу, таких як тунелювання або перенаправлення портів.

Remote Desktop Protocol (RDP) – це протокол, розроблений Microsoft, що дозволяє користувачам віддалено підключатися та керувати іншими комп'ютерами або серверами.[17] RDP в основному використовується в операційній системі Windows, але також існують клієнти для інших операційних систем, включаючи macOS та Linux. RDP надає віддалений доступ до графічного інтерфейсу користувача, який містить інтерфейс вікон,

					ІАЛЦ.467200.003 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

як ви бачите його на вашому власному робочому столі. Він також надає доступ до системних ресурсів, таких як файли, друк, та мультимедіа.

Для встановлення з'єднання, сервер RDP відкриває порт (зазвичай 3389) і чекає на вхідні з'єднання. Клієнт RDP встановлює з'єднання, використовуючи IP-адресу або ім'я домену сервера. Після встановлення з'єднання користувач може виконувати операції на віддаленому комп'ютері, наче він працює безпосередньо на ньому.

RDP використовує передачу даних на основі TCP/IP та здатний адаптуватися до ширини каналу мережі та мережевих затримок. Він використовує методи стиснення для зменшення кількості даних, що передаються, і покращення продуктивності.

Захист в RDP забезпечується за допомогою шифрування. Доступ до сервера контролюється через механізми аутентифікації, які використовують паролі або цифрові сертифікати.

RDP також включає функції для перенаправлення портів, підключення локальних пристроїв до віддаленого сервера, та передачі аудіо та відео. Це робить його корисним для багатьох сценаріїв віддаленого доступу.

RDP дозволяє віддалено виконувати додатки, навіть якщо вони не встановлені на клієнтському комп'ютері. Він передає лише зображення інтерфейсу користувача, клавіатурного введення та дії миші, тому не вимагає великої пропускну здатності.

RDP має набір основних служб, які включають службу віртуального каналу для розширення функціональності, службу авторизації для аутентифікації користувачів, службу графічного виведення для відображення інтерфейсу користувача та службу передачі даних для передачі введення та виведення між сервером та клієнтом.

Додатково, RDP включає функції для забезпечення кращого користувацького досвіду. Наприклад, перенаправлення звуку дозволяє користувачам слухати аудіо від віддаленого комп'ютера на своєму місцевому

					ІАЛЦ.467200.003 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

пристрої. Це особливо корисно при використанні мультимедійних додатків.

RDP також підтримує перенаправлення принтерів, що дозволяє користувачам друкувати документи на місцевому принтері, навіть якщо документ знаходиться на віддаленому сервері.

Для більшої безпеки, RDP підтримує Network Level Authentication (NLA), яка вимагає аутентифікації користувача перед встановленням сеансу RDP.

RDP підтримує багато сеансів, що дозволяє кільком користувачам одночасно працювати на одному і тому ж сервері без взаємного втручання. Це особливо корисно для великих організацій, які потребують підтримки великої кількості віддалених користувачів.

Протокол передачі даних RDP здатний динамічно адаптуватися до змін умов мережі. Наприклад, він може автоматично зменшувати якість графіки, щоб покращити продуктивність при поганих умовах мережі.

RDP включає спеціальні властивості для роботи в обмежених умовах мережі. Наприклад, він може використовувати методи стиснення даних для зменшення кількості переданих даних, а також алгоритми оптимізації для зменшення затримки.

В цілому, Remote Desktop Protocol є потужним і гнучким інструментом для віддаленого керування системами.

Переваги Remote Desktop Protocol (RDP):

- Повний графічний інтерфейс: RDP дозволяє користувачам взаємодіяти з віддаленою системою, як би вони працювали локально, з повним графічним інтерфейсом та доступом до віддалених програм.
- Перенаправлення ресурсів: RDP дозволяє перенаправлення аудіо, відео, принтерів та USB-пристроїв, що забезпечує багатофункціональність та гнучкість.
- Багатосеансовий : Дозволяє кільком користувачам одночасно підключатися та працювати на одному сервері.

					ІАЛЦ.467200.003 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

- Адаптивність до умов мережі: RDP може адаптуватися до ширини каналу мережі та мережевих затримок, динамічно коригуючи якість графіки для оптимальної продуктивності.
- Шифрування даних: RDP використовує шифрування для захисту даних, що передаються між клієнтом та сервером, що забезпечує безпеку.

#### Недоліки Remote Desktop Protocol (RDP):

- Залежність від Microsoft: Хоча існують клієнти RDP для інших платформ, протокол був розроблений Microsoft і найкраще працює з операційними системами Windows.
- Безпека: RDP часто стає ціллю кібератак. Порт RDP (зазвичай 3389) є одним з найбільш цільових для атак.
- Висока вимогливість до пропускної здатності: Незважаючи на оптимізацію, RDP може вимагати значної пропускної здатності для плавної роботи, особливо при передачі великих відеофайлів або використанні додатків з високою графічною вимогливістю.
- Затримка: Незважаючи на оптимізацію, можуть виникнути значні затримки при високій навантаженості мережі або слабкій зв'язковості.
- Потреба в конфігурації: RDP може вимагати певної конфігурації сервера та клієнта для оптимальної роботи, що може бути складним для не досвідчених користувачів.

SSH (Secure Shell) вибирається як найкращий варіант для віддаленого управління системами за допомогою командного рядка на основі наступних чинників:

- Безпека.
- Легкість використання
- Широке застосування.
- Підтримка SCP, який дозволяє безпечно передавати файли між системами.

					ІАЛЦ.467200.003 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВОК ДО РОЗДІЛУ 2

У цьому розділі були розглянуті різні технології для проектування системи, включаючи методи прошивки STM мікроконтролерів, програмне забезпечення для відеотрансляції та протоколи передачі даних через мережу.

Ми розглядали різні інструменти для прошивки плати STM через консоль, включаючи STM32CubeProgrammerCLI, STM32Flash, OpenOCD та ST-Link Utility.

- STM32CubeProgrammerCLI це інструмент в командному рядку, який дає можливість розробникам просто та ефективно завантажувати прошивку на STM мікроконтролери.
- STM32Flash, хоча і є простим у використанні, має обмежену підтримку різних типів пам'яті.
- OpenOCD - це більш складний інструмент, який вимагає більше налаштувань, але він надзвичайно потужний та гнучкий.
- ST-Link Utility - це офіційний інструмент від ST, який пропонує графічний інтерфейс для прошивки STM32 мікроконтролерів.

Вибравши STM32CubeProgrammerCLI, було отримано простий у використанні, але все ж потужний інструмент, що здатний відповідати всім нашим потребам.

Наступним етапом було розгляд програмного забезпечення для відеотрансляції. Ми вивчали Vimeo Livestream, Zoom та OBS Studio.

- Vimeo Livestream пропонує якісну відеотрансляцію, але його основний фокус - це трансляція великих подій.
- Zoom - це добре відома платформа для відеоконференцій, яка також включає можливості відеотрансляції.
- OBS Studio - це вільний і відкритий програмний продукт для запису відео та трансляції в реальному часі, що дозволяє розширений контроль над потоками.

					ІАЛЦ.467200.003 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

Ми обрали Zoom через його широку підтримку, стабільність та інтеграцію з багатьма іншими сервісами, що ми плануємо використовувати.

В третьому підрозділі ми розглядали різні протоколи передачі даних через мережу: SSH, Telnet і Remote Desktop Protocol (RDP).

- SSH (Secure Shell) - це криптографічний протокол мережевого рівня для безпечного з'єднання з віддаленим комп'ютером.
- Telnet - це старий протокол, який не забезпечує шифрування, тому його зараз рідко використовують.
- RDP (Remote Desktop Protocol) - це протокол, який забезпечує графічний інтерфейс для віддаленого з'єднання, але він може бути зайвим для наших потреб.

Наш вибір - SSH, через його безпеку, надійність та широку підтримку.

					ІАЛІЦ.467200.003 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 3. ДЕТАЛІ РОЗРОБКИ СИСТЕМИ

### 3.1 Робота з інструментом STM32CubeProgrammer CLI

STM32CubeProgrammerCLI є інструментом командного рядка від STMicroelectronics, який дозволяє прошивати, витирати, читати та верифікувати флеш-пам'ять мікроконтролерів STM32. Цей інструмент можна використовувати в різних операційних системах, включаючи Windows, macOS та Linux.

Встановлення STM32CubeProgrammerCLI.

Для встановлення STM32CubeProgrammerCLI потрібно завантажити архів з веб-сайту STMicroelectronics. Після завантаження розархівуйте його.

**Get Software**

Part Number	General Description	Latest version	Download	All versions
+ STM32CubePrg-Lin	STM32CubeProgrammer software for Linux	2.13.0	<a href="#">Get latest</a>	Select version ▾
+ STM32CubePrg-Mac	STM32CubeProgrammer software for Mac	2.13.1	<a href="#">Get latest</a>	Select version ▾
+ STM32CubePrg-W32	STM32CubeProgrammer software for Win32	2.13.0	<a href="#">Get latest</a>	Select version ▾
+ STM32CubePrg-W64	STM32CubeProgrammer software for Win64	2.13.0	<a href="#">Get latest</a>	Select version ▾

Рисунок 3.1 – Сторінка завантаження STM32CubeProgrammerCLI

Запустіть інсталятор STM32CubeProgrammerCLI, слідуючи інструкціям:

```
/SetupSTM32CubeProgrammer-2.13.0.linux
```

Зауважте, що після інсталяції STM32CubeProgrammerCLI потрібно в кінець файлу ~/.bashrc додати наступний рядок:

```
export PATH="$PATH:<path_to_STM32CubeProgrammer_install_directory>/bin/"
```

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

Основні команди STM32CubeProgrammerCLI.

STM32\_Programmer\_CLI.exe -c port=SWD - ця команда встановлює з'єднання з пристроєм через SWD порт. Ви можете замінити "SWD" на інший порт, якщо ви використовуєте інший тип з'єднання.

STM32\_Programmer\_CLI.exe -c port=SWD -w file.bin 0x8000000 - ця команда записує файл file.bin на адресу 0x8000000 в пам'яті мікроконтролера через SWD порт.

STM32\_Programmer\_CLI.exe -c port=SWD -r file.bin 0x8000000 1024 - ця команда читає 1024 байтів пам'яті з адреси 0x8000000 і зберігає їх у файл file.bin.

STM32\_Programmer\_CLI.exe -c port=SWD -v file.bin 0x8000000 - ця команда перевіряє, чи відповідає вміст пам'яті на адресі 0x8000000 файлу file.bin.

STM32\_Programmer\_CLI.exe -c port=SWD -e all - ця команда видаляє весь вміст флеш-пам'яті мікроконтролера.

## 3.2 Налаштування SSH підключення для віддаленого доступу

Secure Shell (SSH) є важливим протоколом, який дозволяє безпечно управляти віддаленими комп'ютерами. В контексті нашого проекту, він може бути використаний для встановлення віддаленого доступу до комп'ютера, на якому прошиваються плати STM Discovery.

### 1. Встановлення SSH-сервера

Першим кроком є встановлення SSH-сервера на віддаленому комп'ютері. SSH-сервер, зазвичай, вже є встановленим на багатьох дистрибутивах Linux. Якщо ж це не так, його можна встановити за допомогою пакетного менеджера системи. Для систем, які використовують apt, таких як Ubuntu, можна використати наступні команди:

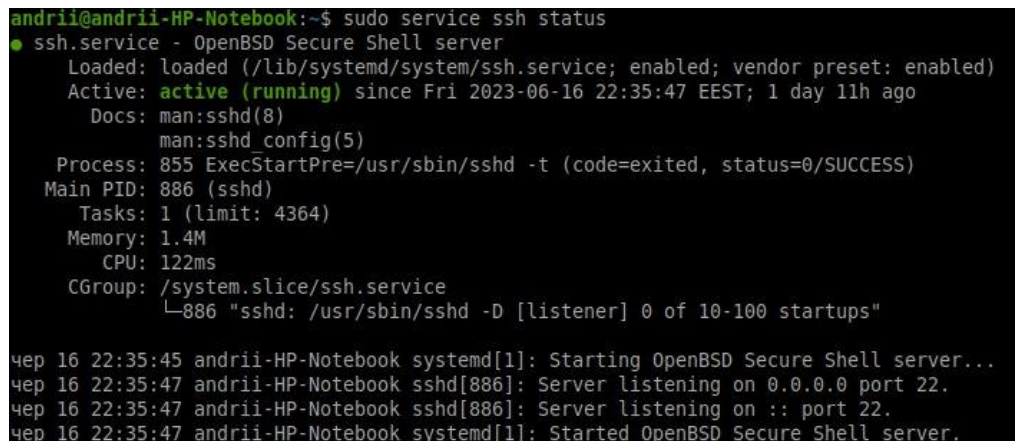
```
sudo apt-get update
```

```
sudo apt-get install openssh-server
```

					ІАЛЦ.467200.003 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

Після встановлення потрібно перевірити, чи SSH-сервер запущено. Це можна зробити за допомогою команди:

```
sudo service ssh status
```



```
andrii@andrii-HP-Notebook:~$ sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-06-16 22:35:47 EEST; 1 day 11h ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 855 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 886 (sshd)
    Tasks: 1 (limit: 4364)
   Memory: 1.4M
      CPU: 122ms
   CGroup: /system.slice/ssh.service
           └─886 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

чер 16 22:35:45 andrii-HP-Notebook systemd[1]: Starting OpenBSD Secure Shell server...
чер 16 22:35:47 andrii-HP-Notebook sshd[886]: Server listening on 0.0.0.0 port 22.
чер 16 22:35:47 andrii-HP-Notebook sshd[886]: Server listening on :: port 22.
чер 16 22:35:47 andrii-HP-Notebook systemd[1]: Started OpenBSD Secure Shell server.
```

Рисунок 3.2– Перевірка, чи SSH-сервер запущено.

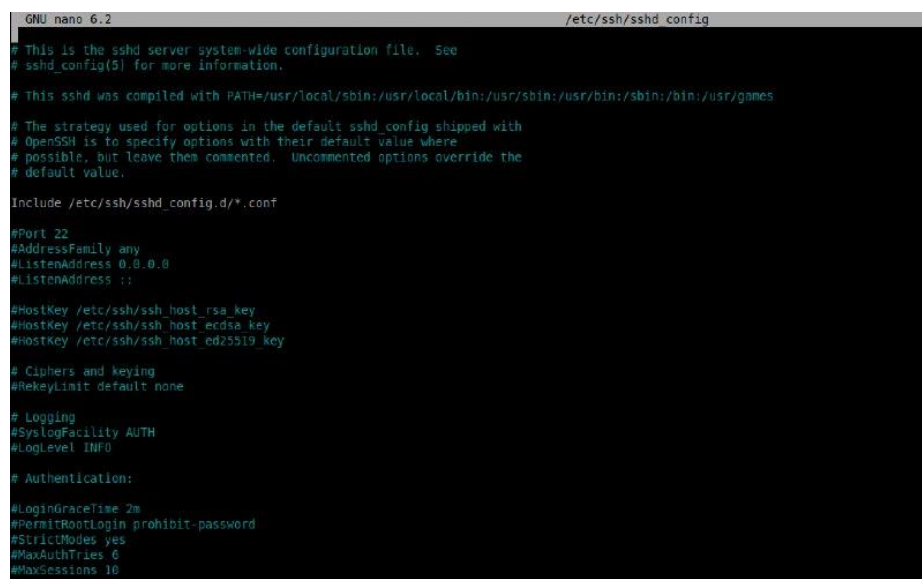
Якщо сервер SSH не запущено, його можна запусити за допомогою команди:

```
sudo service ssh start
```

## 2. Налаштування SSH-сервера

Наступним кроком є налаштування SSH-сервера. Конфігураційний файл SSH-сервера знаходиться в `/etc/ssh/sshd_config`. Цей файл можна відкрити в текстовому редакторі для редагування:

```
sudo nano /etc/ssh/sshd_config
```



```
GNU nano 6.2 /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Рисунок 3.3– Конфігураційний файл SSH-сервера.

За замовчуванням, більшість параметрів можуть залишитися без змін. Однак, можливо, потрібно змінити порт, на якому SSH-сервер слухає з'єднання (за замовчуванням це порт 22), або відключити вхід за паролем і включити тільки вхід за ключем для підвищення безпеки.

Після внесення змін потрібно перезавантажити SSH-сервер:

```
sudo service ssh restart
```

3. Встановлення SSH-клієнта на стороні локального комп'ютера

Третім кроком є встановлення SSH-клієнта на локальному комп'ютері.

```
sudo apt-get update
```

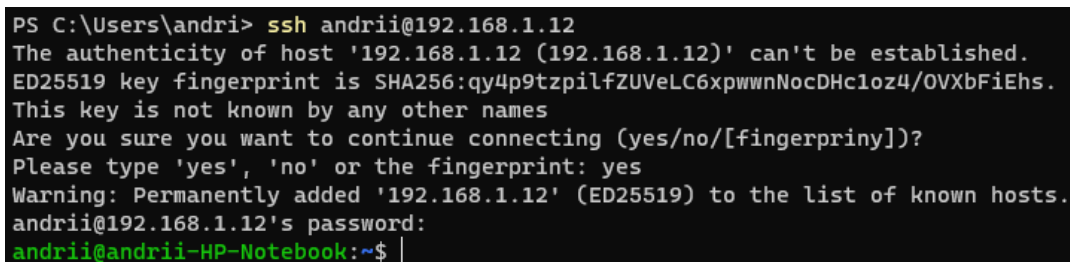
```
sudo apt-get install openssh-client
```

4. Підключення до SSH-сервера

Для встановлення з'єднання з SSH-сервером можна використовувати наступну команду:

```
ssh username@hostname
```

Тут username - це ім'я користувача на віддаленому клієнті, а hostname - IP-адреса віддаленого клієнта.



```
PS C:\Users\andri> ssh andrii@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ED25519 key fingerprint is SHA256:qy4p9tzipilfZUveLC6xpwwnNocDHc1oz4/OVXbFiEhs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.12' (ED25519) to the list of known hosts.
andrii@192.168.1.12's password:
andrii@andrii-HP-Notebook:~$ |
```

Рисунок 3.4– Успішне підключення до віддаленого клієнта.

5. Налаштування авторизації за допомогою ключів

SSH дозволяє використовувати пари ключів для аутентифікації, що є значно безпечнішим, ніж використання паролів. Для створення нової пари ключів можна використати команду ssh-keygen:

```
ssh-keygen
```

Ця команда створить нову пару ключів і збереже їх в домашньому каталозі користувача в .ssh/id\_rsa (приватний ключ) і .ssh/id\_rsa.pub (публічний ключ). Приватний ключ має залишатися в таємниці.

					ІАЛЦ.467200.003 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

```

andrii@andrii-HP-Notebook:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/andrii/.ssh/id_rsa):
Created directory '/home/andrii/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/andrii/.ssh/id_rsa
Your public key has been saved in /home/andrii/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vcKPSfX6KraZEhuVXnsqVPjvatsXuwzMz2I4E9Kytnc andrii@andrii-HP-Notebook
The key's randomart image is:
+----[RSA 3072]-----+
|
|   o
|  + o
| o + .
| o o o..
| = .S+o
| o *.+++.o.
| . X++= .o
| =.B*E*o
| .,+o==*.
+----[SHA256]-----+

```

Рисунок 3.5–Створення нової пари ключів.

Публічний ключ можна скопіювати на SSH-сервер за допомогою команди `ssh-copy-id`:

`ssh-copy-id username@hostname`

### 6. Використання SSH для прошивки STM Discovery

Нарешті, SSH можна використовувати для запуску команд на віддаленому сервері, включаючи команди для прошивки STM Discovery.

```

andrii@andrii-HP-Notebook:~/STM32CubeIDE/workspace_1.12.0/Led/Debug$ STM32_Programmer_CLI -c port=swd -w led.bin 0x08080808 -V -HardRst
-----
STM32CubeProgrammer v2.13.0
-----
ST-LINK SN : 8671FF885356867767122827
ST-LINK FW : V2J41H27
Board : STM32F4DISCOVERY
Voltage : 3.22V
SWD Freq : 4888 KHz
Connect mode: Normal
Reset mode : Software reset
Device ID : 0x413
Revision ID : Rev Y
Device name : STM32F405xx/F407xx/F415xx/F417xx
Flash size : 1 MBytes (default)
Device type : MCU
Device CPU : Cortex-M4
BL Version : 0x31

Memory Programming ...
Opening and parsing file: led.bin
File : led.bin
Size : 5,89 KB
Address : 0x08080808

Erasing memory corresponding to segment 0:
Erasing internal memory sector 0
Download in Progress:
[-----] 100%

File download complete
Time elapsed during download operation: 00:00:00.412

Verifying ...

Read progress:
[-----] 100%

Download verified successfully

Hard reset is performed

```

Рисунок 3.6–Результат успішної прошивки.

Використання SSH для віддаленого керування процесом прошивки STM Discovery дозволяє підвищити гнучкість та ефективність цього процесу.

### 3.3 Пересилка файлів за допомогою SCP

Secure Copy Protocol (SCP) - це протокол, який базується на SSH і використовується для передачі файлів між серверами по безпечному з'єднанню.

#### 1. Використання SCP для передачі файлів

Передача файлів за допомогою SCP відбувається за допомогою командної стрічки. Ось загальний формат команди:

```
scp source_file username@destination:directory/target_file
```

Де:

source\_file - шлях до файлу, який ви хочете передати.

username - ім'я користувача на віддаленому сервері.

destination - ім'я хоста або IP-адреса віддаленого сервера.

directory/target\_file - шлях, куди файл буде скопійовано на віддаленому сервері.

#### 2. Приклад пересилки файлу з локального комп'ютера на віддалений сервер.

Наприклад, якщо ви хочете передати файл file.txt з локального комп'ютера на віддалений сервер з IP-адресою 192.168.0.2, ви можете використати наступну команду:

```
scp file.txt username@192.168.0.2:/home/username/
```

Ця команда скопіює file.txt до домашнього каталогу користувача username на віддаленому сервері.

За допомогою SCP, ви можете безпечно передавати файли між локальним та віддаленим серверами через SSH, що дозволяє вам легко обмінюватися даними та ресурсами.

					ІАЛІЦ.467200.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3.4 Робота з AZ Delivery Logic Analyzer

Програма Logic - це дуже потужний інструмент від Saleae, який широко використовується для аналізу цифрових сигналів. Навіть якщо ви використовуєте AZ Delivery Logic Analyzer, цей програмний продукт може бути сумісний з вашим аналізатором.

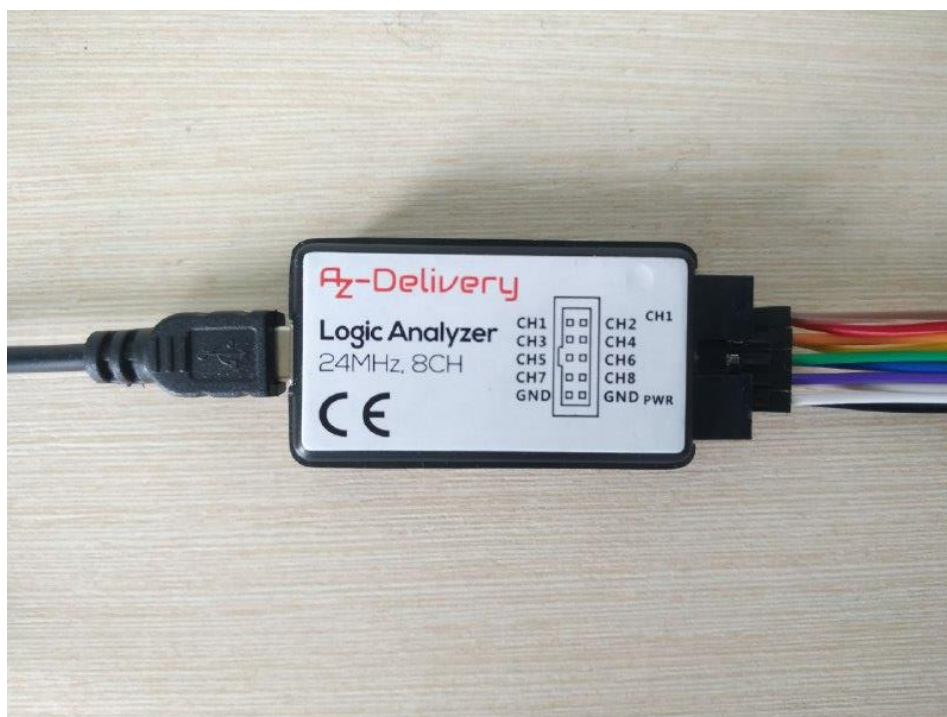


Рисунок 3.7– AZ Delivery Logic Analyzer.

#### 1. Встановлення програми Logic

Завантажте програму Logic з офіційного веб-сайту Saleae. Ви можете вибрати версію для Windows, Mac або Linux в залежності від вашої операційної системи. Після завантаження встановіть програму, слідуючи інструкціям.

#### 2. Підключення AZ Delivery Logic Analyzer

Підключіть AZ Delivery Logic Analyzer до свого комп'ютера за допомогою USB-кабелю. Програма Logic має автоматично визначити ваш пристрій.

Зм.	Арк.	№ докум.	Підпис	Дата

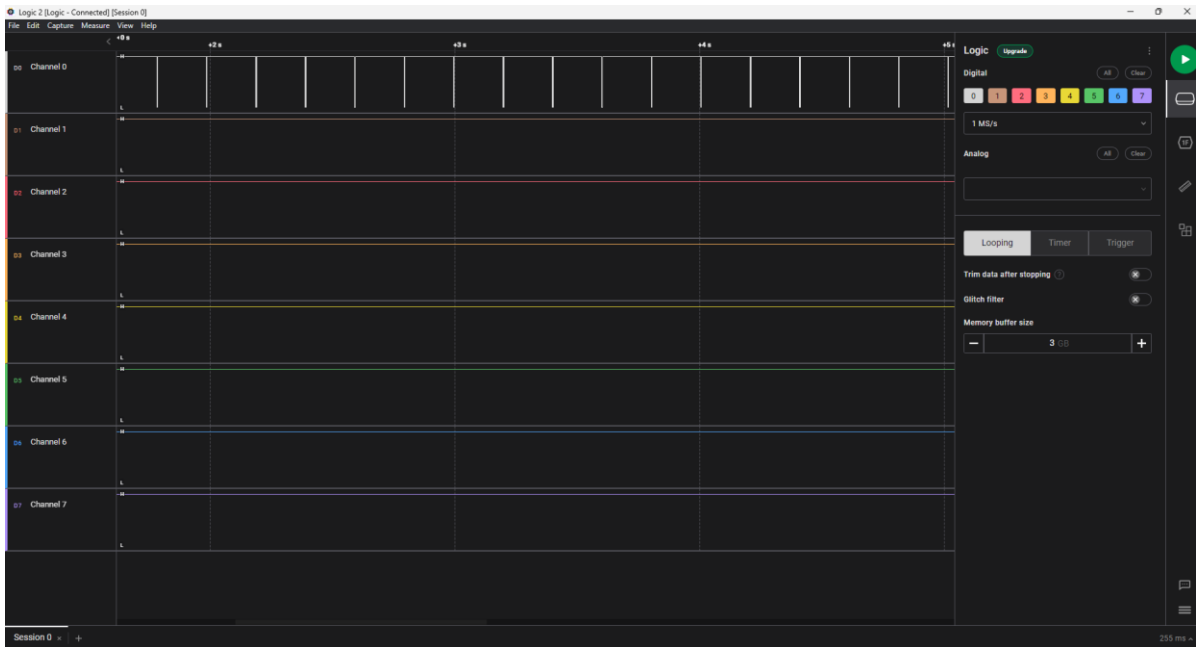


Рисунок 3.8– Програма Logic.

### 3. Захоплення даних

Перед початком захоплення даних, ви маєте визначити, які канали ви хочете використовувати. Ви також повинні вибрати швидкість дискретизації та тривалість захоплення. Після встановлення цих параметрів, натисніть на кнопку "Start" ("Почати") для початку процесу захоплення.

На Рисунку 3.7 показано приклад зчитування першого каналу який підключений до піна блимаючої лампочки.

### 4. Аналіз даних

Після завершення захоплення, ви зможете аналізувати дані. Програма Logic надає вам можливість дивитися на сигнали у вигляді графіка, а також використовувати різні інструменти для аналізу даних, такі як декодери протоколів.

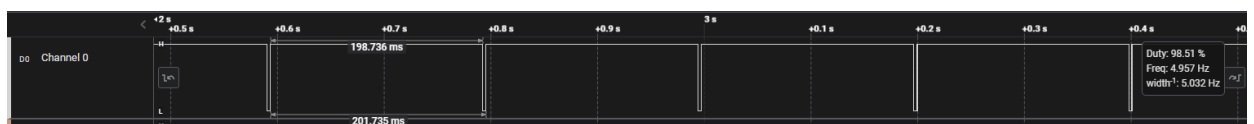


Рисунок 3.9–Графік сигналу.

Також можливо зберегти дані для подальшого аналізу або експортувати їх у форматі, який можна використовувати в інших програмах.

## 5. Налаштування

Програма Logic також має потужні засоби для налаштування. Ви можете використовувати маркери для вимірювання інтервалів часу, а також використовувати тригери для автоматичного початку захоплення при виявленні певних подій.

Ці засоби можуть бути надзвичайно корисними при налаштуванні складних цифрових систем, таких як STM Discovery.

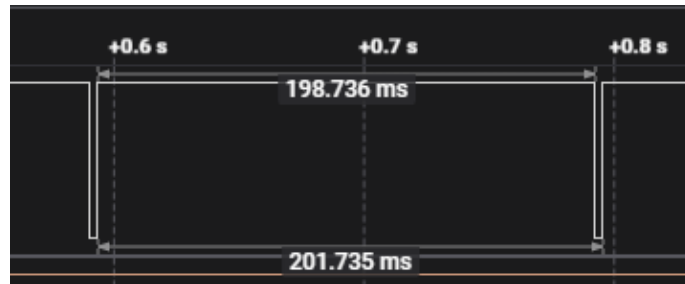


Рисунок 3.10–Маркери вимірювання інтервалів часу.

## 6. Зчитування різних протоколів.

Logic може зчитувати та аналізувати різні протоколи, включаючи, але не обмежуючись I2C, SPI, UART/Serial, CAN, OneWire, Digital Audio (I2S, PCM, TDM), а також багато інших.

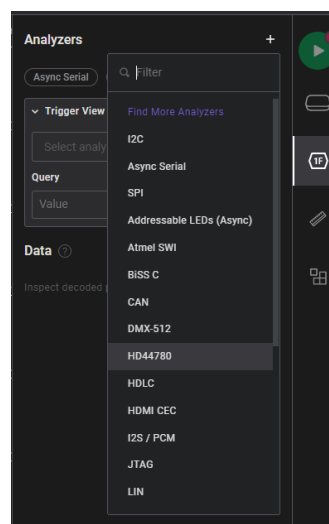


Рисунок 3.11–Протоколи які може зчитувати програма Logic.

Коли програма почне зчитування даних, ви зможете бачити візуалізовані дані на екрані. Залежно від обраного протоколу, ви зможете бачити різні відомості про передачу даних, такі як передачі байтів, коди помилок тощо.

					ІАЛІЦ.467200.003 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВОК ДО РОЗДІЛУ 3

У третьому розділі нашого дослідження було розглянуто ключові аспекти розробки системи для віддаленої прошивки плат. Було проаналізовано ряд важливих інструментів та технологій, які допомагають оптимізувати цей процес.

Зокрема, було вивчено роботу з інструментом STM32CubeProgrammer CLI, що надає потужний набір засобів для прошивки STM32 мікроконтролерів. Завдяки його зручному командному рядку та багатьом опціям конфігурації, цей інструмент є незамінним для ефективної розробки.

Наступним кроком було налаштування SSH підключення для віддаленого доступу до системи. SSH забезпечує безпечний спосіб взаємодії з віддаленими системами, дозволяючи проводити багато операцій, таких як запуск команд або перегляд файлів.

Ще одним важливим аспектом, що було розглянуто, є пересилка файлів за допомогою SCP. SCP дозволяє безпечно копіювати файли між віддаленими системами, що є важливим для забезпечення необхідних файлів для прошивки.

Нарешті, було розглянуто використання AZ Delivery Logic Analyzer. Цей інструмент дозволяє аналізувати логічні стани сигналів у системі, що є незамінним для налагодження та виявлення проблем.

Таким чином, цей розділ надає детальний огляд ключових інструментів та технологій, що використовуються у процесі розробки системи для віддаленої прошивки плат.

					ІАЛЦ.467200.003 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 4. ТЕСТУВАННЯ СИСТЕМИ

Учасниками системи, розглянутої в даному дослідженні, є студент, що працює на своєму локальному комп'ютері, і викладач, на комп'ютері якого запущений SSH і за допомогою програми Zoom ведеться відеотрансляція з платами STM Discovery та демонстрацією екрана з програмою Logic. Наступні аспекти цієї системи були піддані тестуванню.

SSH підключення: Викладач налаштував SSH сервер на своєму комп'ютері, а студент намагається підключитись до нього.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\andri> ssh andrii@192.168.1.12
andrii@192.168.1.12's password:
Last login: Sun Jun 18 10:09:39 2023 from 192.168.1.4
andrii@andrii-HP-Notebook:~$ |
```

Рисунок 4.1–Студент успішно підключився до віддаленого комп'ютера

Підключення було успішним, і студент може виконувати команди на віддаленому сервері.



Рисунок 4.2–Розміщення компонентів на робочому місці

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

Відеотрансляція плат і екрану комп'ютера викладача:

Викладач використовує Zoom для відеотрансляції плат STM та програму Logic. Студент може чітко бачити весь процес і розуміти, як плата реагує на взаємодію. Використання програми Logic: Викладач підключив логічний аналізатор до плати STM і запустив програму Logic. Студент зміг бачити в режимі реального часу, як програма зчитує і аналізує сигнали з плати.

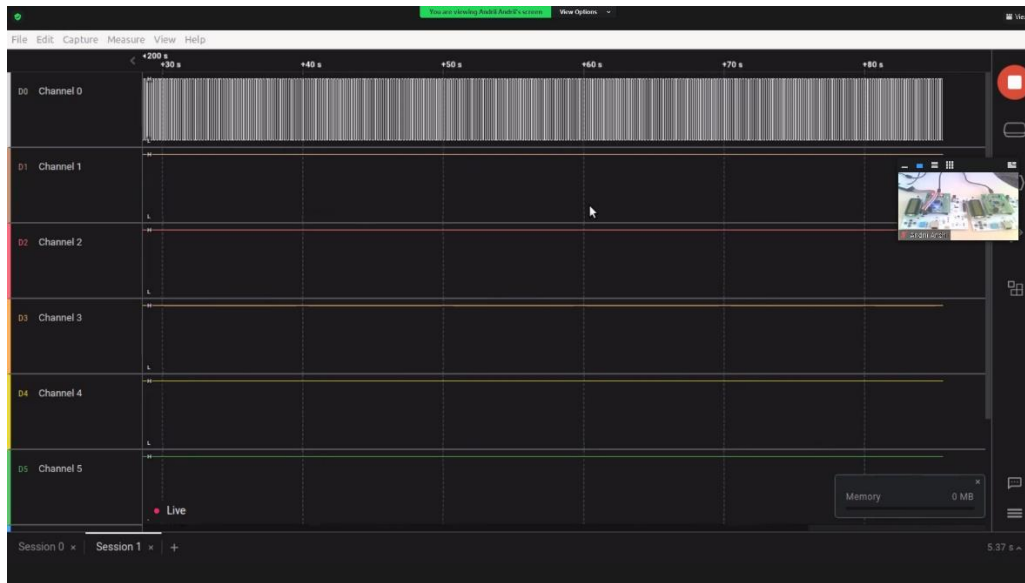


Рисунок 4.3– Демонстрація на якій студент може спостерігати графіки сигналів

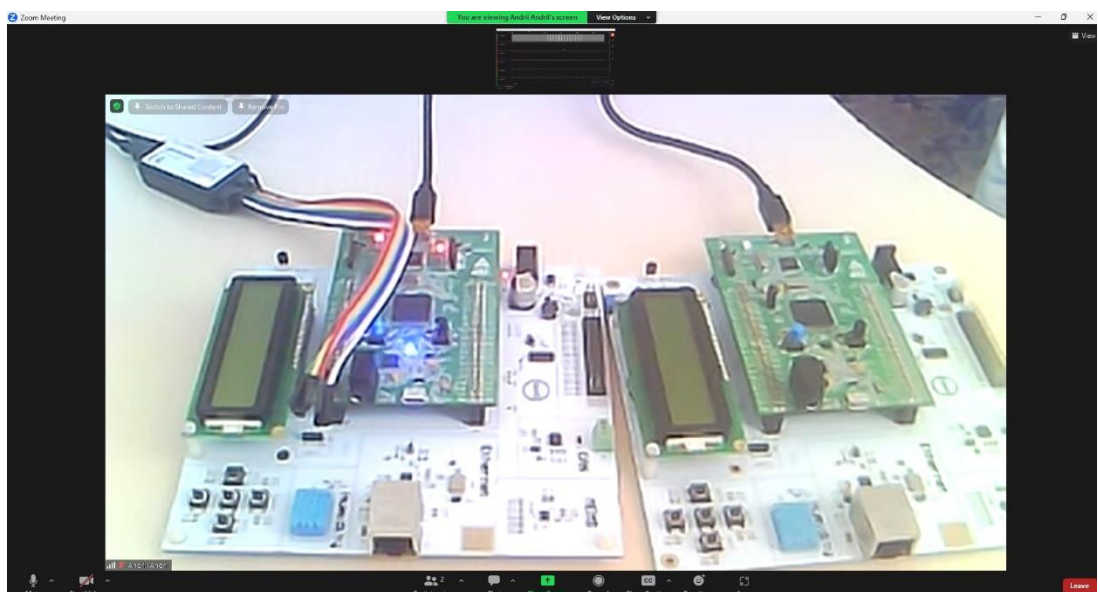


Рисунок 4.4– Відеотрансляція на якій студент може спостерігати за платами

Зм.	Арк.	№ докум.	Підпис	Дата

Пересилка прошивки за допомогою SCP: Студент використовував SCP для надсилання файлу прошивки на віддалений сервер викладача. Файл успішно доставлено, і викладач міг бачити його в потрібній директорії.

```
PS C:\Users\andri\STM32CubeIDE\workspace_1.12.0\led\Debug> scp led.bin andrii@192.168.1.12:/home/andrii/
andrii@192.168.1.12's password:
led.bin                                     100% 6064   315.8KB/s   00:00
PS C:\Users\andri\STM32CubeIDE\workspace_1.12.0\led\Debug> |
```

Рисунок 4.5– Студент успішно надіслав файл за допомогою SCP

Запуск прошивки за допомогою SSH: За допомогою SSH студент запустив прошивку на віддаленому сервері. Процес було успішно завершено, і студент міг бачити результати віддалено.

```
PS C:\Users\andri\STM32CubeIDE\workspace_1.12.0\led\Debug> ssh andrii@192.168.1.12
andrii@192.168.1.12's password:
Last login: Sun Jun 18 12:16:56 2023 from 192.168.1.4
andrii@andrii-HP-Notebook:~$ cd $HOME
andrii@andrii-HP-Notebook:~$ ls
Відео      Завантаження  Зображення  Стільниця  development  led.bin      STM32CubeIDE  Warpinator
Документи  Загальнодоступні  Музика      Шаблони   git           STM32Cube   STMicroelectronics
andrii@andrii-HP-Notebook:~$ STM32_Programmer_CLI -c port=swd -w led.bin 0x08000000 -V -HardRst

-----
STM32CubeProgrammer v2.13.0
-----

ST-LINK SN   : 0671FF505356867767122827
ST-LINK FW   : V2J41M27
Board        : STM32F4DISCOVERY
Voltage      : 3,22V
SWD freq     : 4000 KHz
Connect mode : Normal
Reset mode   : Software reset
Device ID    : 0x413
Revision ID  : Rev Y
Device name  : STM32F405xx/F407xx/F415xx/F417xx
Flash size   : 1 MBytes (default)
Device type  : MCU
Device CPU   : Cortex-M4
BL Version   : 0x31

Memory Programming ...
Opening and parsing file: led.bin
File         : led.bin
Size        : 5,92 KB
Address     : 0x08000000

Erasing memory corresponding to segment 0:
Erasing internal memory sector 0
Download in Progress:
[=====] 100%

File download complete
Time elapsed during download operation: 00:00:00.398

Verifying ...
```

Рисунок 4.6– Студент успішно скористався SSH та прошив плату

					ІАЛЦ.467200.003 ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

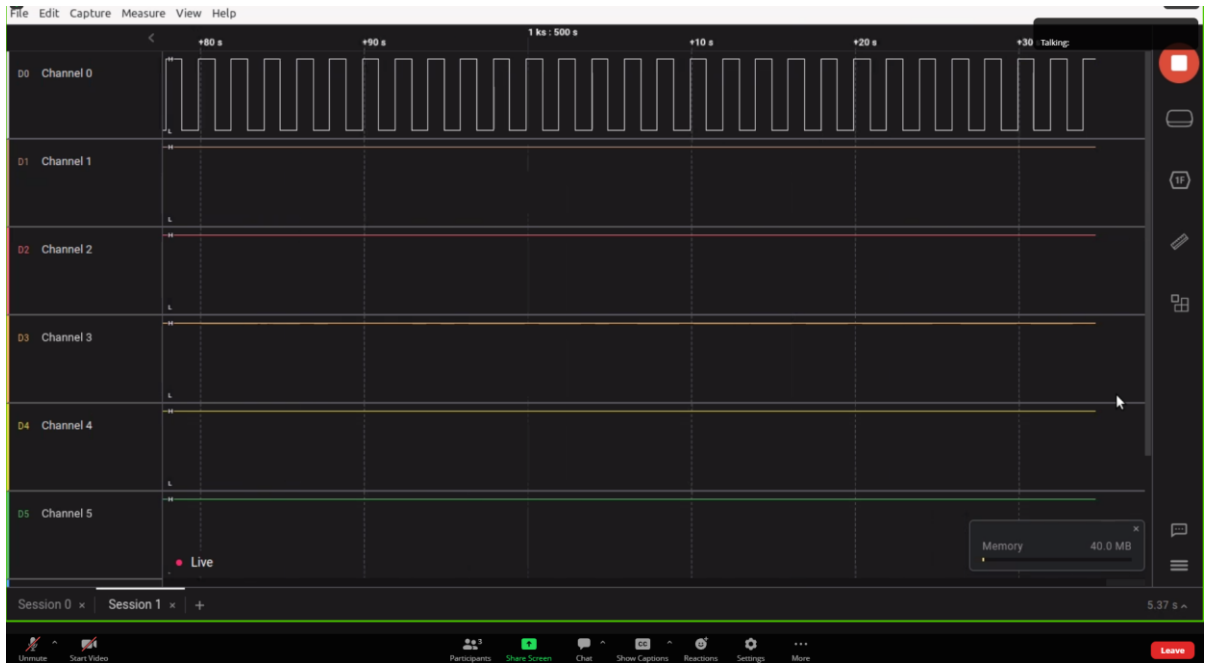


Рисунок 4.6– Студент спостерігає що графік сигналу змінився, а значить плата прошита успішно

Отже, система, що була представлена і протестована в цьому розділі, виявилася вдалою і ефективною.

Зм.	Арк.	№ докум.	Підпис	Дата

## ВИСНОВОК ДО РОЗДІЛУ 4

Система для віддаленої розробки і прошивки плат STM, представлена в даному дослідженні, продемонструвала свою ефективність під час проведення тестів. Вона включає в себе різні аспекти віддаленої взаємодії між студентом та викладачем, включаючи віддалене виконання команд через SSH, відеотрансляції в режимі реального часу через Zoom, використання програми Logic для аналізу сигналів плати, пересилку файлів через SCP, та віддалене запуск прошивки.

Кожен з цих компонентів був підданий тестуванню, і всі вони показали успішні результати. SSH підключення виявилось надійним, пересилка файлів через SCP була ефективною, а програма Logic виявила себе зручним інструментом для аналізу сигналів плати в режимі реального часу.

Демонстрація процесу розробки в режимі реального часу за допомогою Zoom виявилася цінним доповненням до системи, що дозволило студенту безпосередньо спостерігати за процесом роботи викладача і отримувати знання в більш практичному та візуальному форматі.

Крім того, віддалений запуск прошивки через SSH дозволив студенту безпосередньо впливати на роботу плати, навіть перебуваючи на відстані.

Система підтвердила свою здатність до надійного використання в навчальному процесі, що дозволяє студентам розробляти і прошивати плати STM віддалено, при цьому отримуючи повноцінний навчальний досвід, що супроводжується візуальними демонстраціями та реальною взаємодією з обладнанням.

					ІАЛЦ.467200.003 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

# ВИСНОВКИ

У цьому дипломному проекті детально розглядалися аспекти вбудованих систем, включаючи STM мікроконтролери, їх програмне забезпечення, протоколи для керування та прошивки, технології для проектування систем та інструменти для розробки. Особлива увага була приділена безпеці віддаленої розробки та способам віддаленого доступу до мікроконтролерів, включаючи використання SSH та SCP для управління та передачі файлів.

Було розглянуто детальне використання інструменту STM32CubeProgrammer CLI та Az Delivery Logic Analyzer, що дозволяє максимально ефективно використовувати можливості STM мікроконтролерів.

Також було розглянуто програмне забезпечення для відеотрансляції та протоколи передачі даних через мережу, що є важливим для розробки вбудованих систем, які вимагають високої пропускної спроможності та надійності.

У проекті були розроблені та випробувані різні методи та техніки, що дозволяють ефективно працювати з вбудованими системами.

Проте, необхідно врахувати, що хоча цей проект пропонує важливі висновки та рекомендації з приводу вбудованих систем, дослідження в цій області повинні продовжуватися, оскільки технології швидко розвиваються.

Основою цього дипломного проекту є теоретичні та практичні дослідження, що можуть бути впроваджені в навчальний процес в університеті. Це може стати важливою складовою навчального курсу з вбудованих систем, допоможе студентам глибше зрозуміти концепції та практики, пов'язані з STM мікроконтролерами та віддаленою розробкою. Впровадження цього дослідження до навчального плану допоможе підготувати студентів до реальних викликів в області вбудованих систем, сприяючи їхньому професійному зростанню та розвитку в цій динамічній сфері.

					ІАЛЦ.467200.003 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. What is an Embedded System? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/iotagenda/definition/embedded-system>.
2. ARM Cortex-M [Електронний ресурс] – Режим доступу до ресурсу: [https://hexus.net/static/arm\\_cortex\\_m/](https://hexus.net/static/arm_cortex_m/).
3. Difference Between USART, UART, RS232, USB, SPI, I2C, TTL [Електронний ресурс] – Режим доступу до ресурсу: <https://onebyzeroelectronics.blogspot.com/2016/03/difference-between-uart-uart-rs232-usb.html>
4. Overview of STM32 32-bit MCU Family [Електронний ресурс] – Режим доступу до ресурсу: <https://www.digikey.co.za/en/product-highlight/s/stmicroelectronics/stm32-overview>.
5. Arm® 32-bit Microcontrollers [Електронний ресурс] – Режим доступу до ресурсу: [https://www.st.com/content/st\\_com/en/arm-32-bit-microcontrollers.html](https://www.st.com/content/st_com/en/arm-32-bit-microcontrollers.html)
6. STM32CubeProgrammer software for all STM32 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.st.com/en/development-tools/stm32cubeprog.html>
7. Keil MDK [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.arm.com/Tools%20and%20Software/Keil%20MDK>.
8. IAR Embedded Workbench for Arm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.iar.com/ewarm>
9. STM32CubeProgrammer software description [Електронний ресурс] – Режим доступу до ресурсу: [https://www.st.com/resource/en/user\\_manual/um2237-stm32cubeprogrammer-software-description-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um2237-stm32cubeprogrammer-software-description-stmicroelectronics.pdf)
10. OpenOCD User's Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://openocd.org/doc-release/pdf/openocd.pdf>

					ІАЛЦ.467200.003 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

11. About Zoom [Електронний ресурс] – Режим доступу до ресурсу: <https://explore.zoom.us/en/about/>.
12. Professional live streaming is here [Електронний ресурс] – Режим доступу до ресурсу: <https://vimeo.com/blog/post/vimeo-live-is-here-professional-live-streaming/>.
13. Everything you need to know about OBS [Електронний ресурс] – Режим доступу до ресурсу: <https://obsproject.com/kb/>.
14. What is SSH (Secure Shell)? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ssh.com/academy/ssh#the-ssh-protocol>.
15. SSH2 vs. SSH1 and why SSH versions still matter [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/searchsecurity/tip/An-introduction-to-SSH2>.
16. Telnet [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/searchnetworking/definition/Telnet>.
17. Understanding the Remote Desktop Protocol (RDP) [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol>.

					ІАЛЦ.467200.003 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

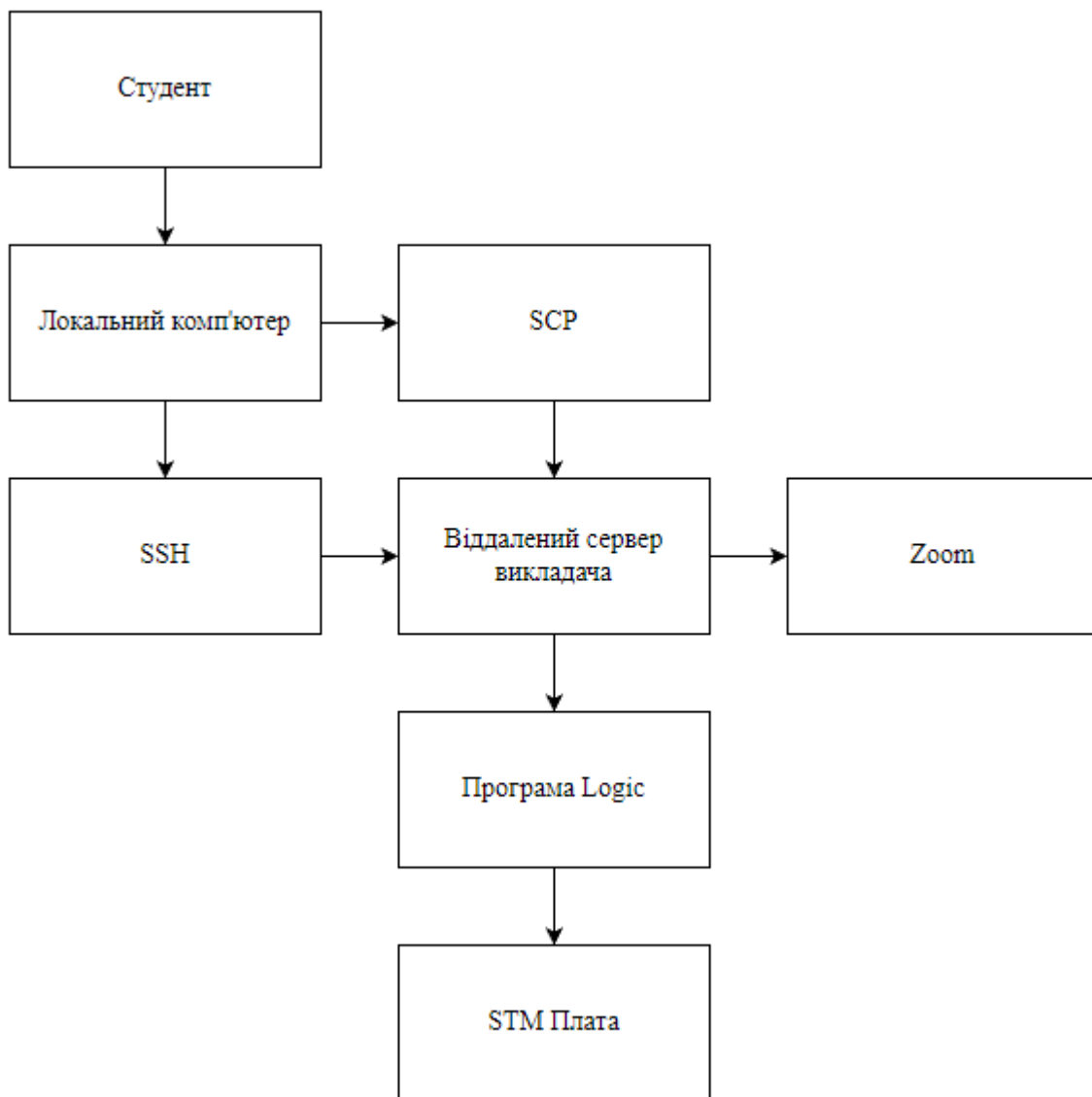
# **ДОДАТОК 1**

Засоби для віддаленої розробки вбудованої системи на платі STM Discovery

**Взаємодія компонентів (структурна схема)**

ІАЛЦ.467200.004 Д1

Аркушів 1



					ІАЛЦ.467200.004 Д1			
		№ докум.	Підпис	Дата				
Розробив	Дмитришин А.Д.				Літ.	Аркуш	Аркушів	
Перевірив	Гайдай А.Р.					1	1	
Н. Контр.	Виноградов Ю. М.				КПІ ім. Ігоря Сікорського ФІОТ ІВ-92			
Затвердив								
Засоби для віддаленої розробки вбудованої системи на платі STM Discovery Взаємодія компонентів (структурна схема)								

# **ДОДАТОК 2**

Засоби для віддаленої розробки вбудованої системи на платі STM Discovery

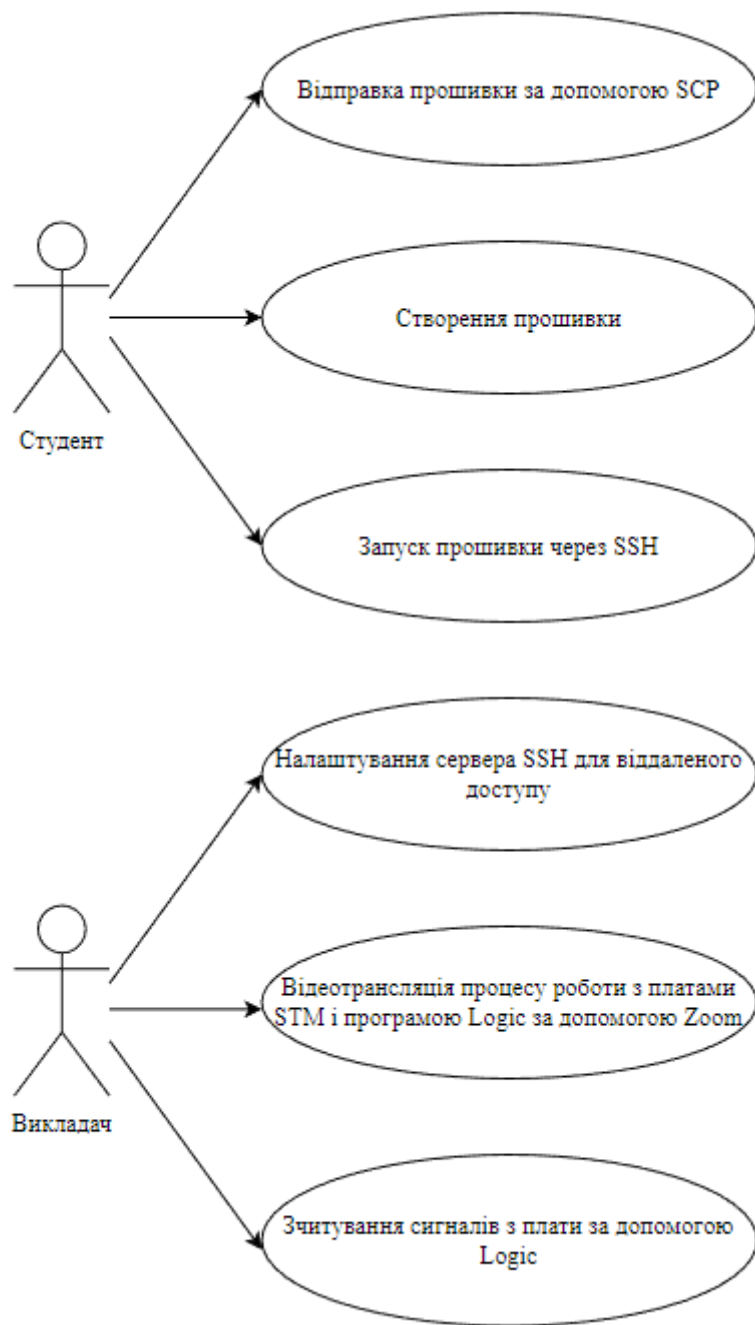
**Блок інтерфейсний UML-діаграма**

**(функціональна схема)**

**ІАЛЦ.467200.005 Д2**

Аркушів 1

**Київ 2023**



					ІАЛЦ.467200.005 Д2		
		№ докум.	Підпис	Дата			
Розробив	Дмитришин А.Д.				Літ.	Аркуш	Аркушів
Перевірив	Гайдай А.Р.					1	1
Н. Контр.	Виноградов Ю. М.				КПІ ім. Ігоря Сікорського ФІОТ ІВ-92		
Затвердив							
Засоби для віддаленої розробки вбудованої системи на платі STM Discovery Блок інтерфейсний UML-діаграма (функціональна схема)							

# **ДОДАТОК 3**

Засоби для віддаленої розробки вбудованої системи на платі STM Discovery

**Алгоритм дії системи (принципова схема)**

ІАЛЦ.467200.006 ДЗ

Аркушів 1

**Київ 2023**



					ІАЛЦ.467200.006 ДЗ			
		№ докум.	Підпис	Дата				
Розробив	Дмитришин А.Д.				Засоби для віддаленої розробки вбудованої системи на платі STM Discovery Алгоритм дії системи (принципова схема)	Літ.	Аркуш	Аркушів
Перевірив	Гайдай А.Р.						1	1
Н. Контр.	Виноградов Ю. М.					КПІ ім. Ігоря Сікорського ФІОТ ІВ-92		
Затвердив								