

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Автоматики та управління в технічних системах
(повна назва кафедри)

«На правах рукопису»
УДК _____

«До захисту допущено»

завідувач кафедри
_____ Ролік О.І.
(підпис) (ініціали, прізвище)

“04” грудня 2018 р.

Магістерська дисертація

зі спеціальності (спеціалізації) _____ 121 програмна інженерія _____
(код і назва спеціальності)

на тему: Система завадостійкої передачі даних на базі віртуального
послідовного порту _____

Виконав (-ла): студент (-ка) б курсу, групи ІТ-73мп
(шифр групи)

_____ Поліщук Олексій Андрійович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник _____ к.т.н., доцент, Катін П.Ю _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2018 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет інформатики та обчислювальної техніки

Кафедра автоматизації та управління в технічних системах

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою
Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.І. Ролік

«__» _____ 20__ р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Поліщуку Олексію Андрійовичу

1. Тема дисертації «Система завадостійкої передачі даних із використанням віртуального послідовного порту», науковий керівник дисертації Катін Павло Юрійович, к.т.н., доцент, затверджені наказом по університету від «__» _____ 20__ р. № _____
2. Термін подання студентом дисертації _____
3. Об'єкт дослідження – система передачі даних на базі віртуального послідовного порту.
4. Предмет дослідження – програмне забезпечення для завадостійкої системи передачі даних на базі віртуального послідовного порту.
5. Перелік завдань, які потрібно розробити: аналіз існуючих рішень та програмних комплексів СЗПД на базі ВПП; класифікація СЗПД; опис технологічних особливостей для СЗПД на базі ВПП; побудова UML діаграм, стартап проект.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: ілюстрація аналогів програмних комплексів та апаратних рішень для віртуалізації портів, діаграма прецедентів; структурна схема; діаграма розгортання, діаграма робочого процесу, функціональна схема.

РЕФЕРАТ

Магістерська дисертація містить 108 сторінок, 27 рисунків, 23 таблиці, 9 додатків, 20 джерел.

Об'єктом дослідження даної роботи є система передачі даних на базі віртуального послідовного порту.

Метою роботи є отримання прототипу для ВПП.

В ході виконання роботи було проведено дослідження існуючих пристроїв для конвертації сигналу, програмного для віртуалізації процесу, виконано їх аналіз та класифікацію. Було розглянуто ключові елементи архітектури таких комплексів та обрано архітектуру для створення моделі комплексу.

Результатом роботи стало створення моделі програмно-апаратного комплексу для передачі інформації через віртуальний ПП, побудовано основні діаграми для формалізації комплексу.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ, АРХІТЕКТУРА ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ, ПОСЛІДОВНИЙ ІНТЕРФЕЙС.

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ.....	6
ВСТУП.....	7
1 ПРИЗНАЧЕННЯ І СФЕРА ЗАСТОСУВАННЯ	9
2 ОБГРУНТУВАННЯ ПОТРЕБИ У РОЗРОБЦІ СИСТЕМИ ЗАВАДОСТІЙКОЇ ПЕРЕДАЧІ ДАНИХ ІЗ ВИКОРИСТАННЯМ ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ І ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	10
2.1 Актуальність використання СЗПД на базі ВПП	10
2.2 Існуючі технічні рішення	11
2.2.1 Рішення «TCP COM BRIDGE»	11
2.2.2 Рішення «USB to Serial Adapter»	13
2.2.3 Рішення «Адаптер COM RS232»	14
2.2.3 Рішення «Промисловий комутатор Dacrol».....	16
2.2.4 Рішення «Advantech ADAM-4570».....	17
2.2.5 Рішення «Комутатор MikroTik».....	20
3 ОПИС ТЕХНОЛОГІЧНИХ ОСОБЛИВОСТЕЙ СИСТЕМИ ЗАВАДОСТІЙКОЇ ПЕРЕДАЧІ ДАНИХ ІЗ ВИКОРИСТАННЯМ ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ	22
3.1 Опис технологічних складових СЗПД	22
3.2 Описання роботи елементів на рівні ПК-ПІ	25
3.3 Опис віртуальних реєстрів ПІ	27
3.4 Стек протоколу ВПП.....	33
3.5 Загальна схема керованої комп'ютерної системи.....	35
3.6 Варіант реалізації диспетчера сеансів.....	37
3.7 Буферизація даних ПІ	41
4 МАТЕМАТИЧНА КОНЦЕПЦІЯ ЗАВАДОСТІЙКОГО КАНАЛУ	49
4.1 Теоретичні засади СЗПД	49
4.2 Математичні моделі каналу зв'язку	53
4.3 Циклічні надлишкові коди	56

5	ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ЗАВАДОСТІЙКОЇ ПЕРЕДАЧІ ДАНИХ ІЗ ВИКОРИСТАННЯМ ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ	64
5.1	Типове рішення для створення ВПП на основі протоколу TCP/IP.....	64
5.2	Сценарій використання ПЗ.....	66
5.2	Веб-сервер «Flask»	70
6	РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ.....	76
6.1	Опис ідеї проекту	76
6.2	Технологічний аудит ідеї проекту	78
6.3	Аналіз ринкових можливостей запуску стартап-проекту.....	79
6.4	Розроблення ринкової стратегії проекту.....	87
6.5	Розроблення маркетингової програми стартап-проекту	91
	ВИСНОВКИ.....	96
	ПЕРЕЛІК ПОСИЛАНЬ	97
	Додаток А Діаграма прецедентів користувача.....	100
	Додаток Б Діаграма прецедентів пристрою.....	101
	Додаток В Структурна схема	102
	Додаток Г Функціональна схема	103
	Додаток Д Діаграма послідовностей	104
	Додаток Е Діаграма компонентів	105
	Додаток Ж Діаграма розгортання.....	106
	Додаток З Діаграма робочого процесу пристрою.....	107
	Додаток И Акт впровадження.....	108

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

СЗПД –система завадостійкої передачі даних

СПД – система передачі даних

ВПП – віртуальний послідовний порт

ПІ – послідовний інтерфейс

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ОС – операційна система

FIFO – first in, first out

UART – Universal Asynchronous Receiver-Transmitter

USART – Universal synchronous and asynchronous receiver-transmitter

USB – Universal Serial Bus

PoE – Power over Ethernet

TCP/IP – Transmission Control Protocol/Internet Protocol

PCI – Peripheral component Interconnect

BIOS – Basic Input-Output System

MTTTY – Multi Threaded Tele TType

ВСТУП

Актуальність теми

На теперішній час пристрої, що використовують ПІ все ще достатньо широко використовуються у промисловості, особливо інтерфейси UART, USART. На сучасних ПК фізичні інтерфейси на базі UART майже зникли, проте зручність використання та значна кількість існуючого ПЗ, що використовувала доступ до зовнішніх пристроїв через ПІ, привело до створення ВПП.

Мета і задачі дослідження

Метою роботи є розробка прототипу системи завадостійкої передачі даних на базі віртуального послідовного порту.

Було проведено аналіз відомих технологій та дослідження аналогів адаптерів. На базі цього аналізу потрібно сформулювати узагальнену структуру моделі зв'язку із пристроєм з ВПП, із використанням технології Ethernet.

Для досягнення цієї мети, виконано наступні завдання:

- огляд існуючих рішень адаптерів ВПП;
- огляд програмних засобів комунікації з використанням ВПП;
- на основі огляду рішень та створеної класифікації, детально розглянута технологія ВПП;
- розглянуті типові рішення ПЗ з використанням ВПП.

Об'єктом дослідження даної роботи є система завадостійкої передачі даних на базі віртуального послідовного порту.

Предметом дослідження є програмне забезпечення для системи завадостійкої передачі даних.

Наукова новизна отриманих результатів.

Аналіз наукових робіт і літератури, показує що моделювання пристрою для віртуалізації зв'язку із послідовним портом, є задачею достатньо не простою, тому у

даній роботі запропоновано узагальнений варіант моделювання програмно-апаратного засобу у вигляді UML діаграм, що може бути використано як основа для реалізації аналогічних рішень і продовження подальших досліджень.

Практичне застосування одержаних результатів

На базі результатів досліджень, отримано прототип ПЗ для СЗПД на базі ВПП. Розроблений програмний макет може бути використаний у навчальному процесі.

Апробація результатів роботи.

Впроваджено прототип у роботу підприємства.

1 ПРИЗНАЧЕННЯ І СФЕРА ЗАСТОСУВАННЯ

У промисловості на теперішній час активно використовуються ПІ, наприклад: термінальні сервери Advantech, промислові комутатори Dacrol, перетворювачі RS232 на TCP/IP та ін. Раніше ПІ використовувався для підключення терміналу, пізніше для модему або миші. Зараз він використовується для з'єднання з джерелами безперебійного живлення, для зв'язку з апаратними засобами розробки вбудованих обчислювальних систем, супутниковими ресиверами, касовими апаратами, програматорами, з приладами систем безпеки об'єктів, а також з багатьма іншими пристроями [1].

ПІ використовувався з часів MS-DOS для передачі файлів з одного комп'ютера на інший, в UNIX для термінального доступу до іншої машини, а в Windows (навіть сучасної) - для відладчика рівня ядра.

Перевагою ПІ є простота обладнання і ПЗ. Недоліком є низька швидкість, великі розміри роз'ємів, а також часто високі вимоги до часу відгуку ОС і драйвера і велика кількість переривань (одне на половину апаратної черги, тобто 8 байт).

Широко поширений в IBM PC-сумісних комп'ютерах, ПІ не придатний для розважальних і офісних завдань, зате широко використовується в прикладних задачах, тому спроби позбутися в сучасних материнських платах від спадщини старих інтерфейсів приречені. Специфікація PC99 - один з безуспішних прикладів нажитися на масштабній заміні «застарілого» обладнання. Про неї пишуть, що вона «нині застаріла і представляє виключно історичний інтерес», а виробництво «застарілого» обладнання процвітає. В офісних і домашніх комп'ютерах, де відстані і перешкоди невеликі, а швидкість обміну затребувана, RS-232 практично витіснений інтерфейсом USB [2].

2 ОБГРУНТУВАННЯ ПОТРЕБИ У РОЗРОБЦІ СИСТЕМИ ЗАВАДОСТІЙКОЇ ПЕРЕДАЧІ ДАНИХ ІЗ ВИКОРИСТАННЯМ ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ І ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Актуальність використання СЗПД на базі ВПП

Існують стандарти на емуляцію послідовного порту над USB і над Bluetooth (ця технологія в значній мірі і проектувалася як «бездротовий послідовний порт»).

Проте програмна емуляція даного порту широко використовується і сьогодні. Так, наприклад, практично всі мобільні телефони емулюють всередині себе класичний COM-порт і модем для реалізації тетерінга - доступу комп'ютера в Інтернет через GPRS / EDGE / 3G / 4G обладнання телефону. При цьому для фізичного підключення до комп'ютера використовується USB, Bluetooth або Wi-Fi [3].

Також програмна емуляція даного порту надається «гостям» віртуальних машин VMWare і Microsoft Hyper-V, основна мета при цьому - підключення відладчика рівня ядра Windows до «гостя».

У вигляді UART, що відрізняється рівнями напруги і відсутністю додаткових сигналів, присутній практично у всіх мікроконтролерах, крім самих-самих маленьких, SoC, платах розробників, а також присутній на платах більшої частини пристроїв, хоча роз'єм і не виведений на корпус. Така популярність пов'язана з простотою цього інтерфейсу, як з фізичної точки зору, так і з легкістю доступу до порту з боку ПЗ в порівнянні з іншими інтерфейсами.

2.2 Існуючі технічні рішення

2.2.1 Рішення «TCP COM BRIDGE»

TCP COM Bridge це утиліта, що працює як програмний конвертер RS232-TCP / IP і дозволяє направити дані від реального (physical) або віртуального ПІ в мережу через Ethernet або Інтернет [4]. Використовуючи TCP COM Bridge, можливо перетворити будь-який комп'ютер в "Serial Device Server". За допомогою цієї програми можливо з'єднати два ПІ на різних комп'ютерах, в тому числі і через Internet. Це дозволяє побудувати розподілені системи збору даних, використовуючи існуюче ПЗ, яке працює тільки з ПІ.

Програма дозволяє заощадити кошти на покупку апаратних термінальних серверів, використовуючи комп'ютер з TCP COM Bridge для того ж самого завдання. Можливо підключити або створити до 256 ПІ на одному комп'ютері, який буде працювати з усіма пристроями одночасно.

Програма TCP COM Bridge, запускається за допомогою іконки в меню "Пуск -> Програми". Програма запропонує створити нове з'єднання, з використанням віртуального або реального ПІ. Необхідно вибрати номер порту і параметри TCP з'єднання в діалоговому вікні, програма встановить і налаштує з'єднання, і якщо треба - додасть віртуальний ПІ в систему.

Універсальність. Можливо створити з'єднання 256 реальних або віртуальних ПІ.

Можливості. За допомогою програми можна з'єднувати ПІ по мережі, отримувати доступ до реального ПІ з віддаленого комп'ютера, розділяти ПІ на два і більше ПІ, отримати доступ до одного порту з двох додатків, забезпечити передачу даних по мережі в додатках, які спочатку можуть працювати тільки з ПІ. ВПП. Наявність реальних портів у системі не обов'язкова. 99.9% емуляція. Інші додатки ніколи не побачать відмінності між реальним послідовним портом і ВПП, створеним TCP COM Bridge.

Вид програми:

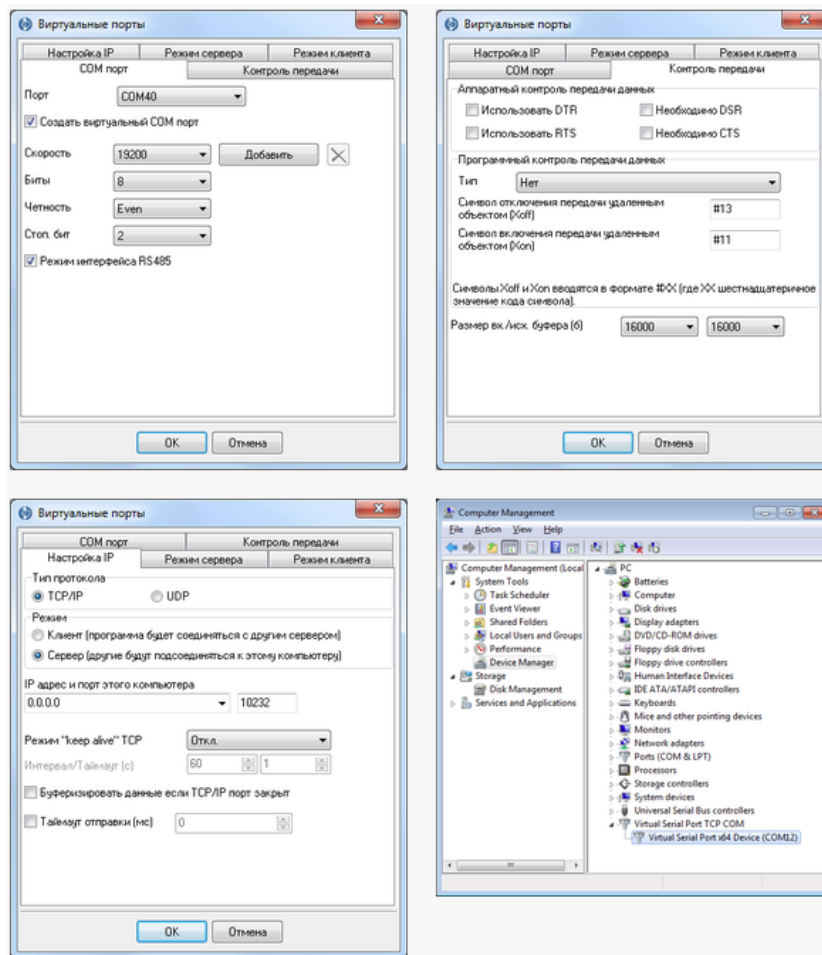


Рис 2.1 – TCP COM Bridge [4]

По мережі дані можуть передаватися по протоколам TCP і UDP як в локальній мережі, так і через Internet. У мережі Internet програма може працювати з динамічними IP адресами, використовуючи доменне ім'я. Локальне використання. Можливо створювати локальні з'єднання в межах одного комп'ютера. Поділ портів. Дані від одного COM порту можна розділити на два або кілька напрямків (serial port splitter). Програма може автоматично запускатися як служба і створювати ВПП, встановлювати мережеві з'єднання ще до логіна користувача в систему. Сумісна із Windows 7, Windows Server 2008 R2.

2.2.2 Рішення «USB to Serial Adapter»

Адаптер розроблений на основі популярного та надійного чіпа FT232RL від FTDI [5]. Крім того, він має світлодіодні індикатори, що показують активність TX та RX.



Рис 2.2 – USB to Serial Adapter – pro mini [5]

Маленькі розміри дозволяють використовувати пристрій для використання із мобільними пристроями, легко встановлюється в будь-який користувацький пристрій, такий як ноутбук або ПК, може використовуватись як в звичайних офісах так і на підприємствах. Після встановлення драйверів ВПП автоматично налаштовується як додатковий ПП, який відображається в диспетчері пристроїв операційної системи. Цей ПП може використовуватись як звичайний вбудований ПП. Чіп та драйвери FT232RL також дозволяють користувачеві регулювати кілька параметрів порту, таких як швидкість передачі даних, біти даних, керування потоком, номер ПП, а також час затримки та інше. Цей USB-последовний адаптер сумісний з більшістю серійних пристроїв, таких як модеми, принтери, сканери, цифрові камери, офісного обладнання, тощо. Функція Plug and Play адаптера дозволяє легко встановлювати і не потребує додаткових ресурсів системи, тобто можна приєднувати більше пристроїв без сутичок із пристроями та конфліктами ресурсів.

Повністю сумісний з усіма масовими ОС.

2.2.3 Рішення «Адаптер COM RS232»

Кабель-перехідник USB-RS232 призначений для підключення до USB порту комп'ютера пристроїв, які використовують ПІ RS232 і призначені для підключення до ПІ DB9 [6].



Рис 2.3 - Адаптер COM RS232[6]

Багато сучасних модифікацій материнських плат та моделей ноутбуків не оснащені ПІ. Між тим, пристрої даного стандарту досить поширені (окремі види принтерів, термінали ISDN та багато іншого). Кабель-перехідник успішно вирішує завдання сумісності пристроїв, що працюють на інтерфейсі RS232. Пристрій, що підключається, матиме можливість використовувати функцію «гарячого» підключення до робочого комп'ютера. Перехідник має нерозбірний корпус.

Для використання кабелю потрібно спочатку встановити драйвер, що йде в комплекті. Потім потрібно вставити кабель в USB-порт комп'ютера і дочекатися, поки ОС «підхопить» потрібні драйвера. Якщо «підхоплення» не відбулося, потрібно зробити це вручну за допомогою диспетчера пристроїв. Кабель має відобразитись як пристрій USB-SERIAL CH340.

Управління кабелем-перехідником здійснюється з комп'ютера або ноутбука за допомогою спеціальних програм (терміналів).

Кабель-перехідник USB - RS232 має два інтерфейси для підключення до комп'ютера і для підключення пристрою:

інтерфейс USB підтримує специфікації 1.1 і 2.0 і має швидкість передачі даних 0,5-480 Мбіт / с.

інтерфейс RS232 виконаний в 9-ти контактному роз'єму стандарту DB9 (він же COM порт) типу «папа», має швидкість передачі даних до 115200 біт / с.

RI - індикатор виклику. Вказує на прийом модемом сигналу виклику по телефонній мережі.

Повний набір сигналів використовується тільки для підключення до комп'ютера зовнішнього модему.

Цей кабель-перехідник підходить для вирішення будь-яких завдань, пов'язаних з використанням інтерфейсу RS232.

Характеристики:

- зібраний на чіпі: CH340;
- довжина кабелю: 75 см;
- вага комплекту: 58 г;
- підтримка ОС: Windows / 98 / ME / 2000 / NT / XP / Vista / 7/8 як 32 так і 64-бітними, Linux, iMac, iMac OS X;
- швидкість передачі даних по RS232: до 115200 біт / с;
- підтримка всіх сигналів інтерфейсу RS232;
- підтримка функції Plug-in-Play;
- інтерфейс USB підтримує специфікації: 1.1 і 2.0;
- інтерфейс RS-232 виконаний в 9-ти контактному роз'ємі стандарту DB9 (він же COM порт) типу «тато».

2.2.3 Рішення «Промисловий комутатор Daprol»



Рис 2.4 – Промисловий комутатор Daprol 6KM [7]

Комутатори промислові залізничні з функцією Power over Ethernet дозволяють мережевим пристроям у мобільних транспортних засобах зменшити займаний простір і обмежити прокладку кабелів [7]. Ця функція забезпечує надійне з'єднання та зменшує витрати на встановлення та економить час. Основні параметри моделі 6KM:

- Повна конфігурація додаткових портів;
- Стійкий корпус металевого вимикача, який працює як радіатор із захистом IP52;
- Різні джерела постійного струму: подвійне джерело 110В, 12В, 24В, 48В, 125В, 250В; PoE; за бажанням іншого джерела живлення змінного струму;
- До 8 налаштованих портів з функцією PoE;
- Наведені стандарти EN50155 та EN50121-4;
- Повна сумісність з NEMA TS-2 & TEES;
- Перемикач у відповідності до вимог ІЕС 61373 (Шок та вібрація), ІЕС 60068-2-27 (шок) та ІЕС 60068-2-6-FC (вібрації);
- Сертифікат DNV;
- Перемикач відповідає вимогам стандартів ІЕС 61850 та ІЕЕЕ 1613 для стандартів захисту навколишнього середовища для електричних підстанцій;

- Повна швидкість на всіх портах, пріоритет QoS 802.1p;
- Повне програмне забезпечення Magnum MNS-6K включає в себе: GUI, SNMPv2, v3, RMON, розширену захищеність, SNTP, журнал подій, BootP, DHCP;
- Додаткове програмне забезпечення MNS-6K-SECURE для підвищення безпеки;
- IGMP Snooping з повідомленнями про фільтрацію або обробку. IGMP-L2 для керування групою груп;
- RSTP і Dual-Homing програмне забезпечення, а також кероване подвійне Homing Dual-Homing;
- Захисне покриття електроніки від пилу та вологості. • Стандартний монтаж панелі, необов'язковий для DIN-рейки.

2.2.4 Рішення «Advantech ADAM-4570»

Термінальний сервер Advantech ADAM-4570 це пристрій, що є шлюзом даних між двома незалежними портами RS232, RS422 або RS485 і мережевим інтерфейсом Ethernet [8].



Рис 2.5 – Термінальний сервер Advantech ADAM-4570 [8]

Термінальний сервер Advantech ADAM-4570 це пристрій, що є шлюзом даних між двома незалежними портами RS232, RS422 або RS485 і мережевим інтерфейсом

Ethernet [8]. Завдяки високій швидкості передачі даних до 230Кб/с, наявності Ethernet-порту і підтримки протоколів TCP і UDP, шлюз ADAM-4570 дає можливість віддаленого доступу для контролю і дистанційного керування сторонніми пристроями, в тому числі і через Інтернет. Разом з пристроєм Advantech поставляє драйвери для MS Windows, що дозволяють зручно конфігурувати параметри роботи шлюзу, використовуючи лише можливості операційної системи і не вдаючись до додаткового складного програмування. Крім того, ADAM-4570 забезпечений системою автовизначення поточної конфігурації та налаштування будь-яких параметрів знаходження в мережах Ethernet. Термінальний сервер ADAM-4570 оснащений високотехнологічною системою захисту від несанкціонованого доступу, системою захисту від стрибків вхідної напруги і системою автоматичного контролю даних. Використання ADAM -4570 дозволяє без особливих тимчасових і матеріальних витрат розширювати мережу в майбутньому, підключаючи до неї все нові пристрої. Даний шлюз заслужено користується великою популярністю при автоматизації промислових підприємств, монтажі складних систем безпеки, забезпеченні діяльності транспортних компаній і диспетчерських служб.

- Порт: два послідовний порти (інтерфейс RS232 / RS422 / RS485, в L-версії RS232)

- Мережа: стандартна Ethernet 10/100 Mbps

- Швидкість передачі даних: до 230 Кб / с

- Роз'єми: мережевий - RJ-45; послідовний порт - RJ-48

- Ізоляційне напруга: 15 000 В для будь-яких вхідних сигналів

- Захист від перенапруги на лініях передачі даних по протоколу RS-485 і подачі

живлення

- Автоматичний контроль передачі даних по інтерфейсу RS-485

- Протоколи: TCP / IP, UDP

- Покращений механізм захисту для запобігання несанкціонованого доступу

- Автоматичне відновлення підключення до мережі

- Можливість віддаленого завантаження нових версій вбудованого ПЗ
- Функція автовизначення мережевого інтерфейсу
- Зручна програма для налаштування портів
- Драйвери для ОС Windows® 98 / NT / 2000 / XP
- Розміри 70 x 130 x 30 mm
- Вага 160 гр

2.2.5 Рішення «Комутатор MikroTik»

У деяких роутерах MikroTik вбудований консоль-порт для управління пристроєм. В цей порт пристрою віддають інформацію при завантаженні, так само через нього можна увійти в меню RouterBOOT для усунення неполадок завантаження і тонкої настройки завантажувача [19].



Рис 2.6 – Комутатор MikroTik [19]

У деяких роутерах MikroTik вбудований консоль-порт для управління пристроєм. В цей порт пристрою віддають інформацію при завантаженні, так само через нього можна увійти в меню RouterBOOT для усунення неполадок завантаження і тонкої настройки завантажувача [19].

Послідовна консоль та термінал - це інструменти, які використовуються для зв'язку з пристроями та іншими системами, які взаємопов'язані через ПІ. Послідовний термінал може використовуватися для моніторингу та налаштування багатьох пристроїв, включаючи модем, мережеві пристрої (включаючи маршрутизатори MikroTik) та будь-який пристрій, яке може бути підключено до ПІ. Функція Serial Console призначена для налаштування засобів конфігурації прямого доступу (монітор / клавіатура та ПІ), які в основному використовуються для початкової або конфігурації відновлення. Якщо ви не плануєте використовувати послідовний порт для доступу до іншого пристрою або для з'єднання даних через модем, ви можете налаштувати його

як послідовну консоль. Перший ПП конфігурується як послідовна консоль, але ви можете вибрати, щоб не налаштувати його, щоб звільнити його для інших програм. Вільний ПП також може використовуватися для доступу до інших маршрутизаторів (або інших пристроїв, таких як комутатори) серійних консолей від маршрутизатора MikroTik RouterOS. Для підключення двох хостів (наприклад, двох ПК або двох маршрутизаторів, а не модемів) потрібен спеціальний нуль-модемний кабель. Зверніть увагу, що для доступу до послідовної консолі на іншому комп'ютері потрібна програма емуляції терміналу (наприклад, HyperTerminal у Windows або minicom на linux).

3 ОПИС ТЕХНОЛОГІЧНИХ ОСОБЛИВОСТЕЙ СИСТЕМИ ЗАВАДОСТІЙКОЇ ПЕРЕДАЧІ ДАНИХ ІЗ ВИКОРИСТАННЯМ ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ

3.1 Опис технологічних складових СЗПД

Важливі деталі цього проекту стосуються обчислювальних систем, які можуть використовувати схеми послідовної передачі через мережу та, зокрема, обчислювальні системи, які можуть використовувати схему послідовного зв'язку через мережу з віртуальним послідовним портом та протокол, призначений для такої програми .

Частим випадком є, коли комп'ютерна система вимагає процедури діагностики або технічного обслуговування, натискання клавіш повинні бути введені в обчислювальні системи. Наприклад, щоб оновити налаштування в базовій системі вводу / виводу обчислювальної системи, користувач повинен, як правило, бути присутнім у системі, щоб вводити натискання клавіш на екрани BIOS. Звичайно, інші види діагностичних або техобслуговування вимагають однакової форми набору клавіш. У контексті великої мережі комп'ютерів, наприклад, в робочому середовищі з багатьма офісами, може бути громіздкою і дорогою, щоб генерувати навіть простий набір клавіш для обчислювальної системи. За таких обставин фахівці з інформаційних технологій (ІТ) можуть фізично переходити на комп'ютер, який потребує діагностики або технічного обслуговування, щоб використовувати клавіатуру та монітор. Цей параметр займає багато часу, і може бути дорогим. Крім того, спеціаліст з інформаційних технологій може намагатися телефонувати за допомогою комп'ютера користувача, який фізично знаходиться за наявності комп'ютера, так що користувач комп'ютера може вводити натискання клавіш за допомогою клавіатури комп'ютера. Знову ж таки, цей параметр може бути помилковим і трудомістким.

Для вирішення вищезазначеної проблеми деякі комп'ютерні системи дозволяють перенаправляти дані екрана в текстовому режимі на послідовний порт і дозволяти перенаправлення даних клавіш із послідовного порту. Рис. 1 зображена така обчислювальна система. Обчислювальна система включає в себе центральний процесор (ЦП), який перенаправляє дані (наприклад, дані екрана текстового режиму) через універсальний асинхронний приймач / передавач (UART) до контролера управління платою. Контролер управління платою - це система, яка контролює змінні середовища обчислювальної системи (наприклад, контролює та контролює температуру комп'ютерної системи). Контролер управління плати містить мікроконтролер, який з'єднаний з іншим UART. Таким чином, процесор може переадресувати дані через UART, який серіалізує дані, які проходять через перемикач (далі обговорення перемикача слід), який спрямовує дані через інший UART. Другий UART десеріалізує дані та передає дані мікроконтролеру на контролер керування плати. Таким чином, мікроконтролер може передавати дані екрана текстового режиму в локальну мережу (LAN) через мережеву карту інтерфейсу (NIC). З іншого боку, дані клавіш можуть бути отримані через локальну мережу та перенаправлені до ЦП через UART [10].

Перемикач може передавати дані від центрального процесора як на мікроконтролер контролера керування на планшетах, так і на роз'єм послідовного порту (коннектор П). Роз'єм П - це фізичне розташування контактів (наприклад, роз'єм DB-9 або DB-25, який використовується з послідовним портом RS-232), який зазвичай доступний зовні комп'ютера. Стан вимикача знаходиться під контролем мікроконтролера. Коли перемикач забезпечує підключення до П, П може бути використаний, тому що будь-який звичайний П буде використовуватися в іншому випадку. Коли перемикач забезпечує підключення до мікроконтролера (через UART), П недоступний для використання. Тому П недоступний для використання, коли центральний процесор перенаправляє дані екрана текстового режиму або дані клавіш через контролер управління плати [11].

Вищезгадана схема виявляє певні недоліки. Включення перемикача додає складність і вартість для дизайну контролера управління плати. Крім того, необхідність керувати станом комутатора додає складності програмному забезпеченню / прошивці, виконаному мікроконтролером. Крім того, дані, перенаправлені з ЦП на мікроконтролер, обмежені швидкістю передачі даних за швидкістю передачі даних, що підтримується UART. З наведених вище причин видно, що існує потреба в схемі перенаправлення, яка стосується одного або декількох вищезгаданих недоліків.

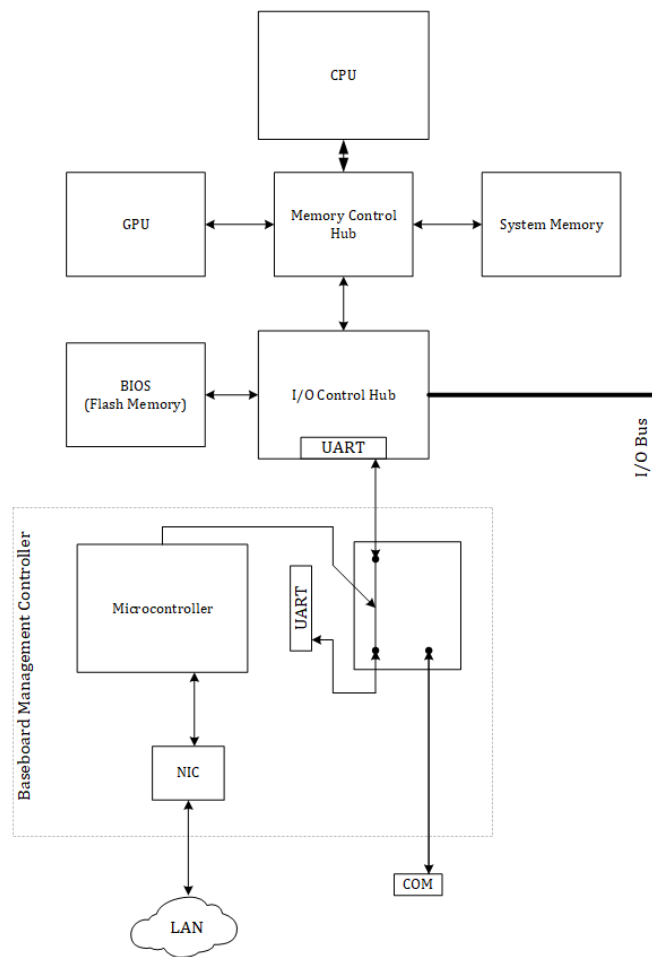


Рис 3.1 – Система, що здатна передавати текстові дані через UART

3.2 Описання роботи елементів на рівні ПК-ПІ

На рис 3.2 показано приклад комп'ютерної системи, яка переспрямовує дані екрана текстового режиму та дані клавіш без використання перемикача. Обчислювальна система включає в себе процесор, який з'єднаний з концентратором управління пам'яттю. Контрольний концентратор пам'яті являє собою схему управління, яка управляє доступом до системної пам'яті, відеокарти та контрольного вузла вводу / виводу (вводу / виходу). Контрольний концентратор вводу-виводу, у свою чергу, управляє та контролює доступ до пристрою флеш-пам'яті, в якому зберігається основна система вводу / виводу (BIOS). Він також управляє та контролює доступ до UART, який може бути втілений як частина контрольного вузла вводу-виводу, а також керує та контролює доступ до шини вводу-виводу, наприклад, шини міжсмугового периферійного компонента (PCI). (Контрольний концентратор вводу-виводу також може керувати та контролювати доступ до аудіоканалів, USB-портів та інших пристроїв вводу-виводу, які відомі в даній галузі техніки, але не є важливими в контексті цього розкриття та не відображені тут) [12].

Взаємодія з шиною вводу-виводу є вбудованим багатофункціональним пристроєм. Як більш докладно обговорюється нижче, інтегрований багатофункціональний пристрій - це єдине пристрій, що забезпечує кілька функцій. У конкретному прикладі, показаному на рис. 3.2, інтегрований багатофункціональний пристрій - це єдиний пристрій, що пропонує функцію ПІ та функцію контролера локальної мережі. Такий інтегрований багатофункціональний пристрій може бути представлений на ринку як контролер локальної мережі з вбудованими функціями керування.

Вбудований багатофункціональний пристрій може містити мікроконтролер, з'єднаний з UART та контролером локальної мережі. Як альтернатива, схема UART може бути фізично відсутня і присутня лише у віртуальному сенсі, як це описано нижче.

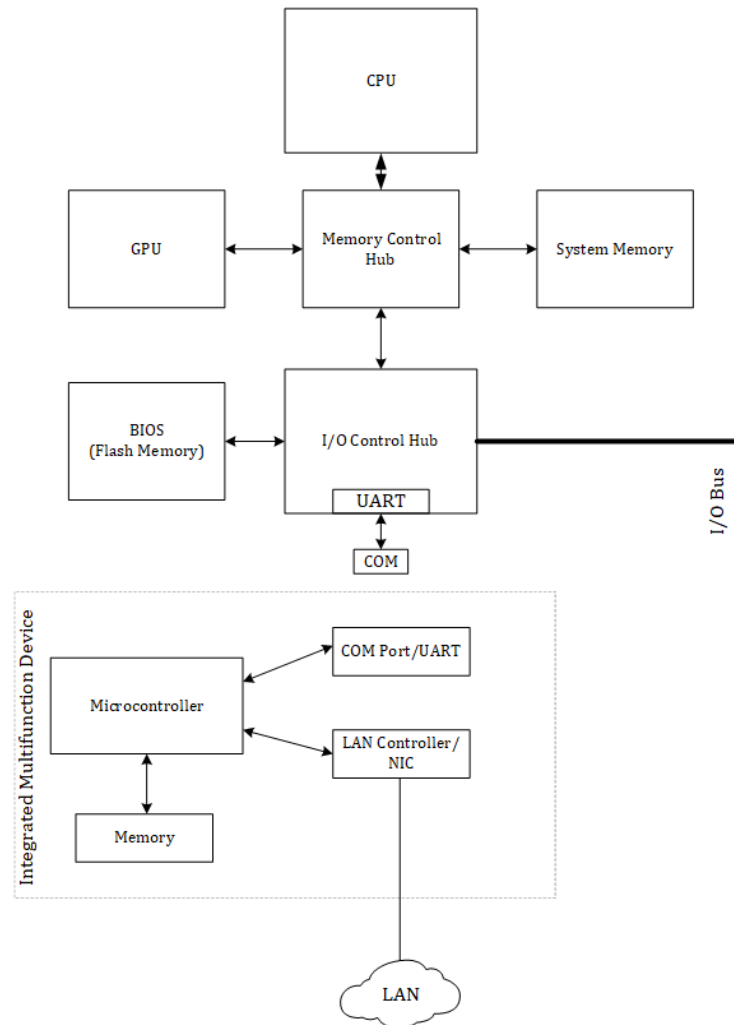


Рис. 3.2 – Обчислювальна система, здатна перенаправляти дані через інтегрований багатофункціональний пристрій

Функція UART включає набір з дванадцяти регістрів (не показаних на рис.3.2), доступних ЦП. Реєстри служать інструментом для взаємодії з функцією ПІ та програмним забезпеченням, що працює на ЦП. Іншими словами, дані читаються та

записуються у функцію ПІ, читаючи їх та записуючи до одного з дванадцяти регістрів. Крім того, поведінка функції ПІ контролюється шляхом запису та читання з решти семи регістрів [10].

Зокрема, в одному варіанті здійснення, інтегрований багатофункціональний пристрій насправді не містить ПІ. Скоріше, інтегрований багатофункціональний пристрій містить набір з дванадцяти регістрів, які з виду процесора з'являються як регістри ПІ звичайного ПІ. Прошивка або програмне забезпечення, що працює на мікроконтролері, контролює регістри таким чином, щоб імітувати поведінку ПІ за допомогою приєднаного до нього пристрою, хоча ніякий такий ПІ чи пристрій насправді відсутні. Таке розташування регістрів і прошивки або програмного забезпечення може бути названо "віртуальним ПІ" або ВПІ. Таким чином, програмне забезпечення, що працює на ЦП, може бути структуроване таким чином, щоб забезпечувати взаємодію з регістрами так, ніби вони є транспортним засобом для взаємодії з фактичним ПІ, хоча жоден фізичний ПІ не існує.

3.3 Опис віртуальних регістрів ПІ

На рис.3.3 зображено інтегрований багатофункціональний пристрій більш детально, включаючи віртуальні регістри ПІ [9]. Мікроконтролер, що знаходиться на інтегрованому багатофункціональному пристрої, з'єднаний з віртуальними регістрами ПІ, який може включати дванадцять регістрів, доступних через вісім входів / виходів. Реєстри наступні: регістр передачі / утримання (THR), регістр активності переривання (IER), регістр ідентифікації переривань (IRR), регістр керування лінією (LCR), регістр керування модемами (MCR), регістр стану лінії (LSR), регістр стану модему (MSR), зарезервований реєстр (RSV), регістр буферу приймача (RBR), регістр керування FIFO (FCR), регістр фіксації дільника, найменший значущий байт (DLL) та регістр фіксації дільника, найбільший значущий байт (DLM). Інші реалізації ПІ можуть мати

більшу чи меншу кількість регістрів, що мають різні функції. Такі реалізації входять до сфери дії цього розкриття.

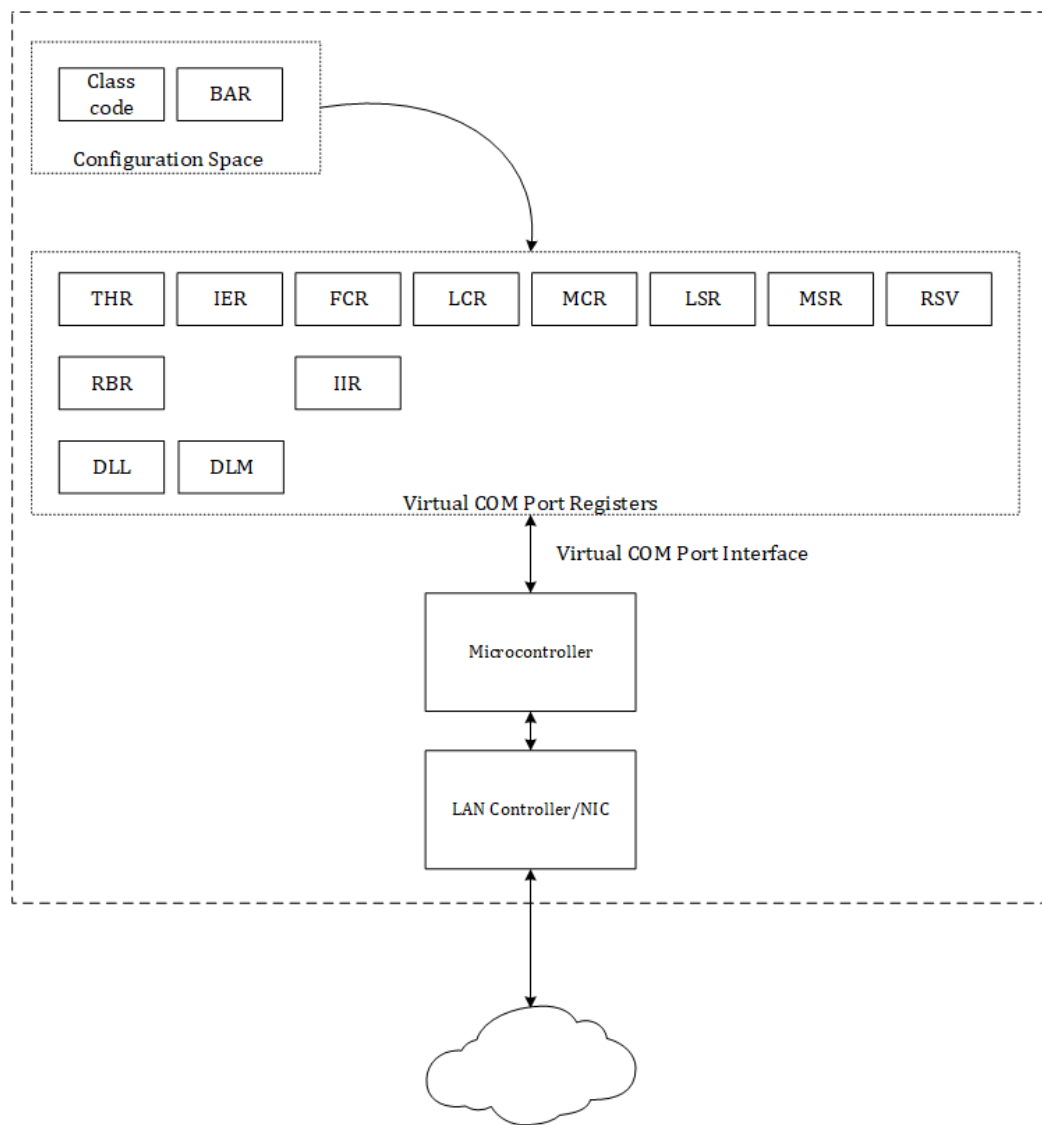


Рис 3.3 – Інтегрований багатофункціональний пристрій з набором віртуальних регістрів COM-порту

Для кожного з вищезазначених регістрів зазвичай призначається певна функція в контексті звичайного ПІ. Коротко кажучи, їх функції є наступними [9].

Регістр передачі / утримання (THR) зазвичай використовується для зберігання байтів вихідних даних і використовується тільки для запису. Буферний регістр

приймача (RBR) містить вхідний байт даних і використовується тільки для читання. Реєстри THR та RBR діляться адресою вводу-виводу. Спроба прочитати із загальної адреси вводу-виводу призводить до прочитання з реєстру RBR; спроба записати на неї призводить до запису в реєстр THR.

Реєстр увімкнення переривань (IER) зазвичай використовується для включення та вимкнення переривань, які можуть бути згенеровані під час використання пов'язаного ПП.

Реєстр ідентифікації переривань (IPP) зазвичай використовується для ідентифікації джерела певного переривання і є тільки для читання. Реєстр контролю FIFO (FCR) контролює внутрішню операцію FIFO в UART, і це тільки для запису. Реєстри IIR та FCR діляться адресою вводу-виводу. Спроба прочитати з загальної адреси вводу-виводу призводить до читання з реєстру IIR; спроба записати на неї призводить до запису в реєстр FCR.

Реєстр керування лініями (LCR) зазвичай використовується для зберігання параметрів зв'язку параметрів (паритет, кількість стрибкових бітів тощо). Восьмий біт реєстру керування лінією називається бітового дробу (біт DLAB). Коли біт DLAB підтверджується, команда запису або читання для реєстру THR та IER призводить до запису або читання відповідно до реєстрів DLL та DLM.

База часу UART може бути розділена, щоб дозволити програмам, що вимагають повільної швидкості. Дільник (найменш значущий байт і найбільш значущий байт) можуть бути введені в реєстри DLL та DLM, коли біт DLAB в LCR підтверджено. Швидкість бази часу поділяється на дільник, введений в ці реєстри, тим самим сповільнюючи швидкість зв'язку UART.

Реєстр керування модемами (MCR) використовується для відправлення певних керуючих сигналів (наприклад, цей реєстр може використовуватися для встановлення сигналу керування RTS, коли центральний процесор хоче сигналізувати модем, яким він хоче надсилати дані, або цей реєстр може бути використаний щоб покласти порт

у режим зворотного зв'язку, в якому повертаються дані, які будуть передані, також отримані дані).

Реєстр статусу лінії (LSR) використовується для позначення подій, які можуть виникати на лінії даних (наприклад, отримані дані готові, отримані дані пошкоджені, співвідношення помилок та інше).

Нарешті, реєстр стану модему (MSR) використовується для зберігання різних індикаторів стану модему (наприклад, вказівка кільця, вказівка на готовність набору даних та індикація, відоме як прозорий для надсилання, а це означає, що модем готовий отримувати дані від ПК тощо).

Як зазначено вище, мікроконтролер виконує вбудоване програмне забезпечення або програмне забезпечення, що зберігається на пристрої пам'яті, що змушує мікроконтролер читати та записувати до реєстрів так, наче присутній фактичний UART. (Пристрій пам'яті - це один з прикладів комп'ютера, який може зберігати матеріали, розкриті в даному документі. Інші приклади включають магнітні або оптичні носії інформації.) Нижче наведено приклад того, як реальний пристрій використовує реєстри при реальному UART присутній, а потім приклад, що описує віртуальний послідовний порт, який використовує реєстри паралельно. Приклади призначені для того, щоб забезпечити смак способу, з яким мікроконтролер може читати та записувати до реєстрів, щоб емулювати поведінку фактичного ПІ.

Звичайний набір реєстрів ПІ з'єднаний з UART. Процесор записує дані, що передаються через ПІ до реєстру передачі / утримання (згадані вище). У відповідь UART зазвичай вводить дані у внутрішній буфер FIFO. Після цього дані видаляються з буфера FIFO і передаються один біт за один раз (тобто дані серіалізуються) через коннектор (наприклад, з'єднувач DB-9) вздовж лінії передачі послідовного порту, після чого дані отримані пристроєм (наприклад, модемом), з'єднаним з іншим кінцем лінії електропередачі. Якщо пристрій, з'єднаний із ПІ, зайнятий, він може залишити чітке значення, щоб надіслати біт у реєстрі стану модему (згадане вище). Це, як правило, сигналізує програмі, що використовує ПІ, що необхідно зупинити

надсилання даних на пристрій. Як викладено нижче, ВПП може поводитися однаково у цьому відношенні та в будь-якому іншому відношенні [10].

Повертаючись до рис.3.3, на ньому зображено ВПП. Таким чином, мається на увазі, що, хоча існують реєстри послідовних портів, не існує UART або роз'єму, пов'язаного з регістрами. Проте, з точки зору процесора, не видно, що існує фактичний UART або коннектор. Мікроконтролер зчитує та записує до регістрів таким чином, щоб імітувати той, який використовується UART, з пристроєм, що з ним пов'язано. Таким чином, наприклад, коли центральний процесор записує дані на ВПП, він робить це, записуючи в регістр передачі / утримання (THR) - так само, як він записує дані на звичайний ПП. Мікроконтролер реагує, читаючи дані, записані в регістр передачі / утримання (THR), і передаючи дані через контролер локальної мережі, і, зрештою, до мережі. Якщо мікроконтролер стає зайнятий, мікроконтролер може відмовитися від передачі біт у регістрі стану модему (MSR), тим самим сигналізуючи, що процесор відмовляється відсилати дані до набору реєстрів ПП, а також міміці поведінки пристрою, пов'язаного з ПП. Коротше кажучи, поведінка ВПП зареєстрована, ідентична поведінці звичайного набору реєстрів ПП.

Повертаючись до рис.3.2 та обговорення структури інтегрованого багатофункціонального пристрою, інтегрований багатофункціональний пристрій може також включати локальний контролер. Контролер LAN містить набір регістрів, через які процесор взаємодіє з функцією контролера локальної мережі. Звичайно, він також включає схему для виконання низькорівневих функцій, включаючи взаємодію з фізичним середовищем зв'язку. Вбудований багатофункціональний пристрій може втілюватися як один чіп або може бути втілений як декілька чіпів, які взаємодіють один з одним.

Система, зображена на рис.3.2 використовує той факт, що деякі комерційно доступні BIOS та ОС включають можливість переадресації даних екрана текстового режиму на ПП і приймати дані з ПП, і розглядати його як дані вхідної клавіатури. В такому випадку, процесор може або надіслати текстові дані екрана тексту на ПП, а не

на екран, або може записувати екранні дані в текстовому режимі як на екран, так і на ПІ. Аналогічно, дані можуть бути прочитані з ПІ і розглядаються як локально створені дані клавіш. Використовуючи комп'ютерну систему з ВПП, через який можна отримати доступ до локальної мережі, послідовні дані можуть бути перенаправлені через локальну мережу до іншої комп'ютерної системи.

Якщо інтегрований багатофункціональний пристрій є сумісним з PCI пристроєм, ВПП може бути доступним для ЦП, вказавши конфігураційний простір, який повідомляє про присутність функції послідовного порту в пристрої. Наприклад, такий PCI-сумісний пристрій містить конфігураційний простір PCI, який являє собою набір реєстрів, включаючи реєстр коду класу і базовий реєстр адреси (BAR). Реєстр кодів класу містить значення, що визначає вид функції, наданої пристроєм. Таким чином, в контексті пристрою, що забезпечує ВПП (або звичайний ПІ), реєстр кодів класу містить значення, що ідентифікує функцію ПІ. Реєстр базової адреси наданий у конфігураційному просторі, так що BIOS може зберігати в ній адресу вводу-виводу, що вказує на реєстри ПІ. Під час запуску BIOS перевіряє кожен шини вводу-виводу (наприклад, шини PCI) і шукає кожен пристрій. Кожне знайдене пристрій перераховано. Крім того, BIOS шукає кожен функцію, запропоновану кожним пристроєм. Кожна функція кожного пристрою також перераховується. Під час цього процесу BIOS зберігає адресу вводу-виводу в базовому реєстрі адреси конфігураційного простору, пов'язаного з кожною функцією кожного пристрою. Виходячи з адреси вводу-виводу, що зберігаються в реєстрі базової адреси, BIOS та операційна система можуть визначати, як звертатися до певної функції на конкретному пристрої.

3.4 Стек протоколу ВПП

Вбудований багатофункціональний пристрій може виконувати стек протоколу, такий, як показано на рис.3.4.

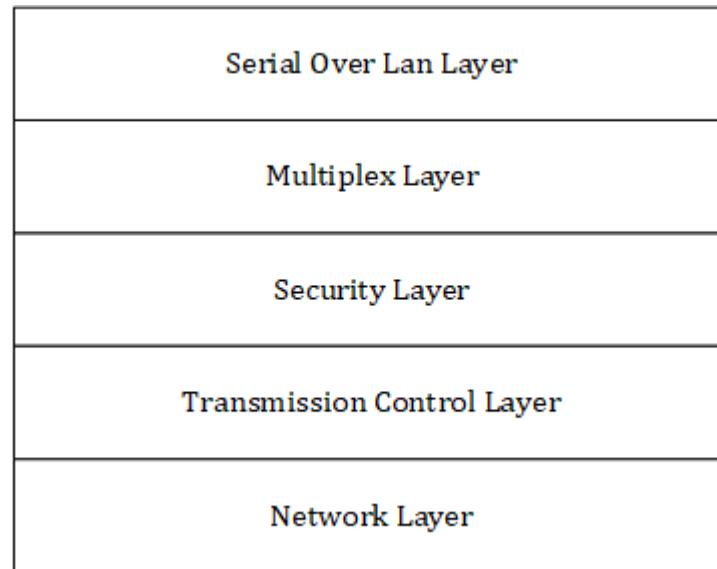


Рис 3.4 – стек протоколу для виконання інтегрованим багатофункціональним пристроєм

Кожний шар надає послуги шарам, розташованих над ним. Як видно, стек містить мережевий рівень [13]. Він включає в себе програмне забезпечення / прошивку, що забезпечує функціональність Інтернет-протоколу IP. IP-функціональність, надана мережевим рівнем, доступна для керуючого рівня передачі. Функціональність TCP або UDP, надана рівнем керування передачею, забезпечується шару безпеки. Рівень безпеки - необов'язковий рівень, який може бути повністю відсутнім, може бути активований або деактивований або може бути завжди активований. Рівень захисту включає програмне забезпечення / прошивку, що забезпечує функціональність безпеки транспортного рівня (TLS) або включає в себе програмне забезпечення / вбудоване програмне забезпечення, яке забезпечує безпеку, визначену Інтелектуальною платформою управління інтерфейсом (IPMI). Функціональність безпеки, надана рівнем безпеки, доступна для мультиплексного

шару. Мультиплексний шар знову є додатковим шаром. Він може бути присутнім на комп'ютерній системі, що працює під інші форми перенаправлення (наприклад, перенаправлення IDE, розглянуті нижче). Мультиплексний шар відповідає за визначення того, чи слід отримувати дані, отримане через контролер локальної мережі, в кінцевому підсумку до реєстру віртуальних послідовних портів або до іншого набору реєстрів віртуальних пристроїв (таких як набір віртуальних реєстрів пристроїв IDE). Функціональність, надана мультиплексним шаром, доступна для шару III понад LAN. Рівень III понад LAN виконує багато завдань, спрямованих на мету одержувати дані з набору віртуальних реєстрів послідовних портів та доставляти їх до контролера локальної мережі, і навпаки. Рівень послідовного переходу через мережу обговорюється докладніше нижче.

Програмне забезпечення / прошивку, що складають мережевий рівень, може виконуватися контролером локальної мережі або мікроконтролером. Програмне забезпечення / прошивку, що складають інші шари, може виконуватися мікроконтролером у вбудованому багатofункціональному пристрої.

У процесі роботи послідовний шар за допомогою LAN приймає дані з реєстрів і пакує дані в них до команд, які передаються вниз через стек протоколу, призначені для прийому консолі керування (розглянуті нижче). Коли виконується стек протоколу, команди пакетуються і в кінцевому підсумку передаються через мережу, які в кінцевому підсумку отримують вищезгадана консоль керування.

Консоль керування - це комп'ютерна система, яка взаємодіє з керованою обчислювальною системою (рис.3.2). Консоль управління керує ідентичним стеком протоколу, щоб він міг правильно інтерпретувати команди, отримані від керованої обчислювальної системи. Отже, персонал ІТ на консолі управління може переглядати дані екрана текстового режиму, перенаправлені з керованої обчислювальної системи. Крім того, персонал ІТ може вводити дані клавіатури на консолі керування, а консоль управління передає дані через мережу керованій обчислювальній системі, після чого вона приймає вбудований багатofункціональний пристрій і розпаковується

(послідовно через мережу шар) у віртуальний регістр П. BIOS або операційна система, яка виконується ЦП керованої обчислювальної системи, потім отримує дані з регістрів віртуального послідовного порту та обробляє дані як вхідні дані клавіш. Таким чином, віддалений персонал ІТ, розташований на консолі управління, здатний діагностувати проблеми та переналаштувати керовану комп'ютерну систему без необхідності фізично бути присутнім у системі. Консоль керування також може передавати сигнал керування потоком (це більш детально обговорюється нижче), наприклад, детектор даних (DCD), готове набір даних (DSR), вказане кільце (RI) тощо.

3.5 Загальна схема керованої комп'ютерної системи

Як показано на рис.3.5, керована комп'ютерна система (така як обчислювальна система) виконує наступні дії [13].

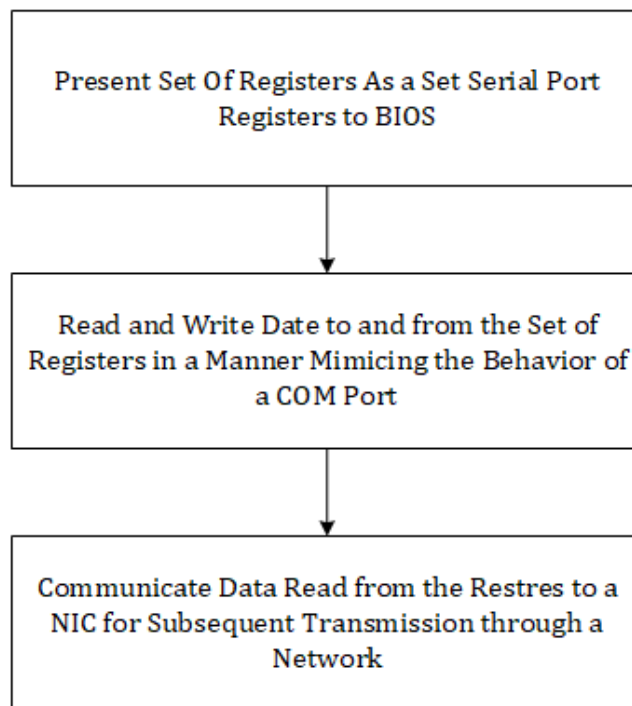


Рис 3.5 – спосіб забезпечення віртуального послідовного порту

По-перше, як показано в операції, набір реєстрів, розмірених за розміром і числом, що імітує те, що з набору реєстрів послідовного порту, представлені в BIOS та / або операційній системі, що виконуються ЦП як звичайний набір реєстрів послідовних портів. У контексті виконання цього завдання з пристроєм PCI це включає в себе надання конфігураційного простору PCI, що має реєстр кодів класу, який ідентифікує пристрій як функцію ПІ, а також має реєстр базового адреси, який може бути запрограмований для вказування на вищезгаданий набір реєстрів. Далі, як показано в роботі, мікроконтролер, з'єднаний із вищезгаданим набором реєстрів, запрограмований на читання та запис даних у реєстри та з них, імітуючи поведінку звичайного ПІ з приєднаним до нього пристроєм. У контексті використання послідовної локальної мережі це може включати в себе надання прикладного рівня послідовного інтерфейсу через мережу, як показано на рис.3.4, і, як детально обговорюється, нижче. Нарешті, як показано на зображенні операції, дані, що зчитуються з вищезгаданого набору реєстрів, передаються контролеру LAN для передачі через мережу консолі керування або іншої обчислювальної системи. Звичайно, дані, отримані від консолі керування через локальний контролер, розпаковуються у вищезгадані реєстри для завантаження ЦП. Операція може включати в себе виконання стека протоколу, такого, як показано в шарах на рис. 3.4

Рис. 3.6 зображує один можливий варіант здійснення стека протоколу, показаного на рис. 3.4. Як показано на рис. 3.6, шар послідовного з'єднання через мережу, показаний на рис. 3.4 може керуватися диспетчером сеансів. Коротко кажучи, диспетчер сесії відповідає за відкриття сеансу перенаправлення (наприклад, сеанс переадресації послідовної передачі через мережу) і несе відповідальність за те, щоб було відкрито менше, ніж деяку максимальну кількість таких сеансів (приклад: обмеження кількості одночасних серійних перехідних сеансів для одного такого сеансу). Роботу диспетчера сесій докладніше розглянемо нижче.

3.6 Варіант реалізації диспетчера сеансів

Як видно з рис. 3.6, багатофункціональний інтегрований пристрій може також підтримувати інший сеанс перенаправлення, крім послідовної через мережу [10].

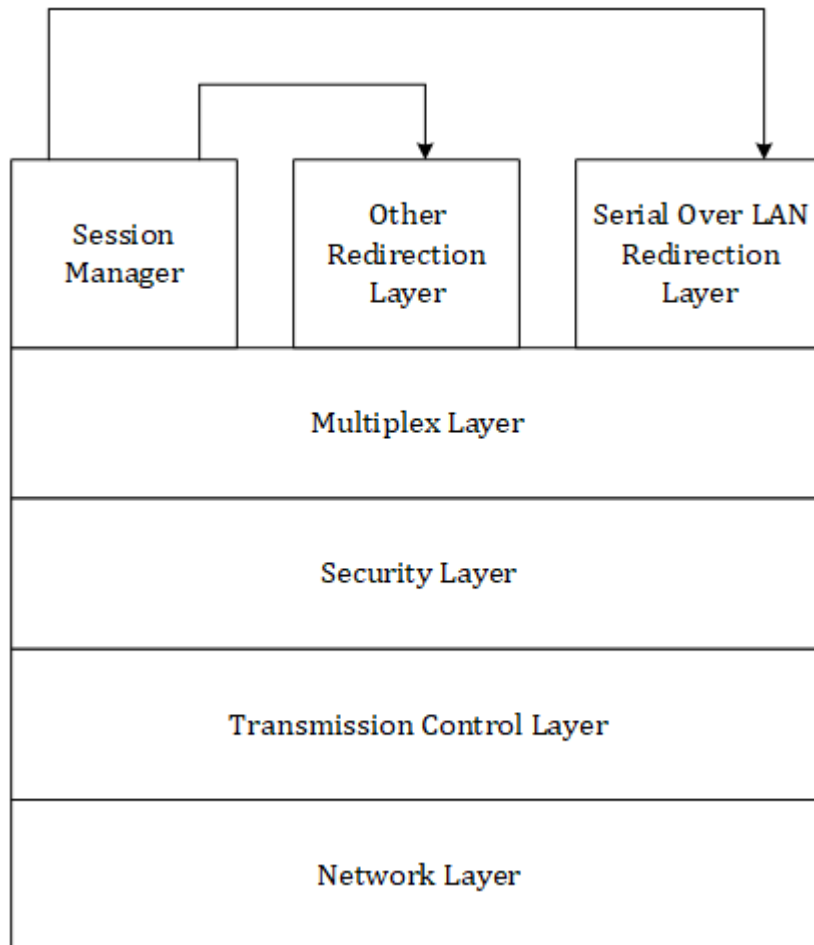


Рис 3.6 – Можливий варіант здійснення стека протоколу

Наприклад, інтегрований багатофункціональний пристрій також може забезпечувати набір регістрів, розмір яких відповідає розміру та номеру, ідентичним тому, який встановлений у регістрі пристроїв інтегрованого приводу електроніки (IDE). Іншими словами, інтегрований багатофункціональний пристрій може надавати віртуальне середовище IDE. Мікроконтролер в інтегрованому

багатофункціональному пристрої може бути запрограмований для взаємодії з пультом дистанційного керування таким чином, щоб мікроконтролер зчитував та записує до набору реєстрів таким чином, що імітує поведінку звичайного пристрою IDE. Така функціональність забезпечується шаром перенаправлення, який забезпечує функцію перенаправлення, відмінну від перенаправлення послідовної через локальну мережу.

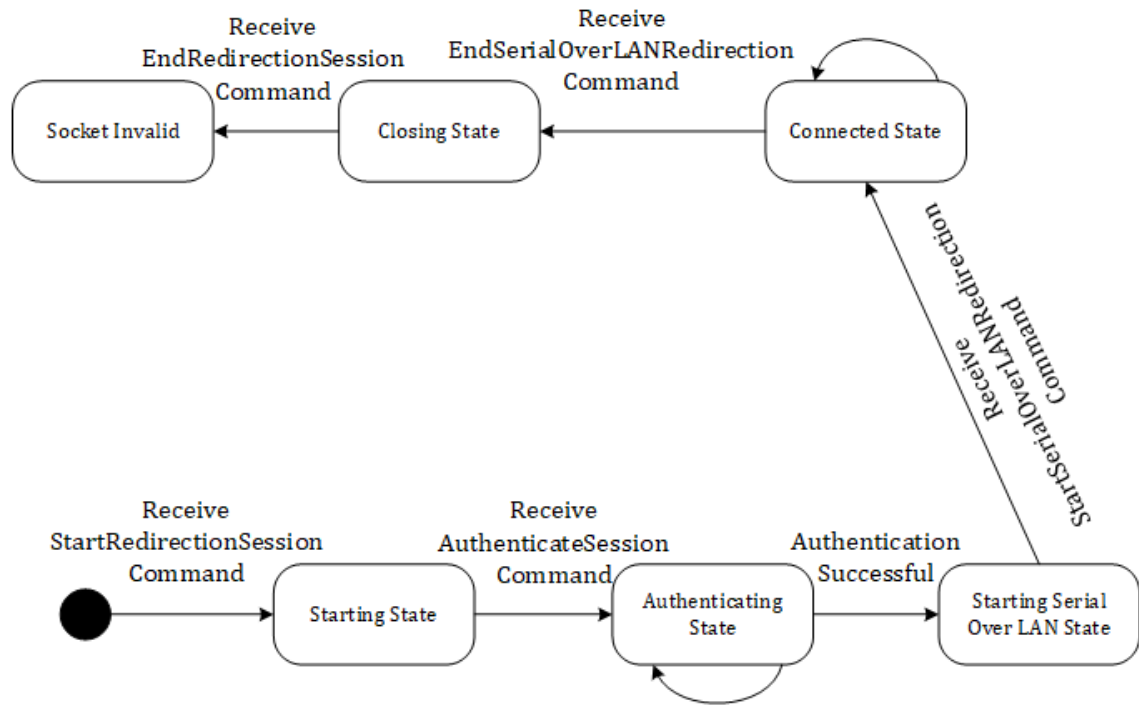


Рис 3.7 – Потік роботи диспетчера сеансів та шар III понад LAN

Як показано на Рис. 3.7, диспетчер сеансу може спочатку бути запущений при отриманні команди `StartRedirectionSession` з консолі керування. (Звичайно, команди з консолі керування приймаються через контролер LAN в інтегрованому багатофункціональному пристрої). Команда `StartRedirectionSession` - це перша команда, яку консоль управління зазвичай надсилає до обчислювальної системи під управлінням. В основному, ця команда визначає, який тип сеансу перенаправлення слід запустити (тобто, якщо запускається сеанс серії по локальній мережі або має бути запущена інша форма сеансу перенаправлення, наприклад сеанс перенаправлення IDE). Таким чином, команда `StartRedirectionSession` може містити принаймні один

аргумент, щоб виконати свою роль у визначенні того, який сеанс перенаправлення буде запущений.

Команда `StartRedirectionSession` отримується диспетчером сеансів [14], що змушує диспетчера сеансів переходити до початкового стану. У початковому стані диспетчер сесії визначає, чи відкриття запитуваного сеансу перенаправлення спричинить більше, ніж максимальну кількість дозволених сеансів для відкриття. Якщо так, диспетчер сеансів не відкриває запитуваний сеанс перенаправлення. З іншого боку, якщо відкриття запитуваної сесії не призведе до більш ніж максимальної кількості дозволених сеансів, диспетчер сесії продовжуватиме свою процедуру, щоб відкрити сеанс. Так чи інакше, диспетчер сеансів відповідає на консоль управління, відправивши повідомлення `StartRedirectionSessionReply`. Це повідомлення вказує, чи вже відкрита максимальна кількість сеансів перенаправлення, а також вказує на те, чи підтримується обчислювальна система під керуванням певної різноманітності сеансу перенаправлення, що вимагається консоллю керування.

Диспетчер сеансів залишається в початковому стані, поки не отримає команду `AuthenticateSession` з консолі керування, після чого вона переходить у стан автентифікації. Команда `AuthenticateSession` визначає процес, за допомогою якого консоль керування може ідентифікувати себе самостійно, так що керована комп'ютерна система може перевірити, чи дійсно консоль має повноваження для запуску сеансу перенаправлення. Наприклад, команда `AuthenticateSession` може містити аргументи, такі як ім'я користувача та пароль для вищезазначеної мети. Якщо автентифікація не виконана (наприклад, ім'я користувача чи пароль, надані консоллю керування, невірні), диспетчер сеансу залишається в стані автентифікації, очікуючи на команду `AuthenticateSession` з правильним ім'ям і паролем. Повідомлення констеляції `AuthenticateSessionReply` повертається на консоль управління, що означає, що автентифікація не виконана. З іншого боку, якщо автентифікація успішна, повідомлення консолі керування, яке показує успіх, повертається в повідомлення

AuthenticateSessionReply, а перехід послідовного по локальній мережі відбувається в початковому послідовному режимі через мережу (SOL).

У початковому стані SOL стан серії по локальній мережі чекає команди StartSOLRedirection з консолі керування. Команда StartSOLRedirection повідомляє про те, що він повинен почати передачу послідовних по локальній мережі повідомлень на консолі керування. Іншими словами, це означає початок сеансів послідовного інтерфейсу. Команда StartSOLRedirection може містити певні аргументи, які налаштовують шар ПІ понад LAN (Рис. 3.8). Коротше кажучи, цей шар включає буфер передачі та буфер прийому. Рівень ПІ понад LAN може також включати чотири таймери: (1) таймер буфера передачі; (2) таймер буферного прийому; (3) таймер серцебиття; і (4) таймер очікування сеансу.

Під час роботи перенаправлені дані з ЦП отримують з реєстру ПІ (який може бути набором віртуальних реєстрів ПІ, як показано на рис.3.3), і зберігається в буфері передачі. Буфер передачі може бути буфером FIFO. Як більш докладно обговорюється нижче, після виникнення певних подій дані видаляються з буфера передачі, упаковуються в команди та повідомляються мультиплексному шару для передачі на консоль управління. Довжина буфера передачі може бути визначена аргументом команди StartSOLRedirection.

3.7 Буферизація даних ПІ

Команди з консолі керування приймаються з мультиплексного шару і поміщаються в буфер прийому [15]. Як і буфер передачі, буфер прийому може бути буфером FIFO (рис. 3.8).

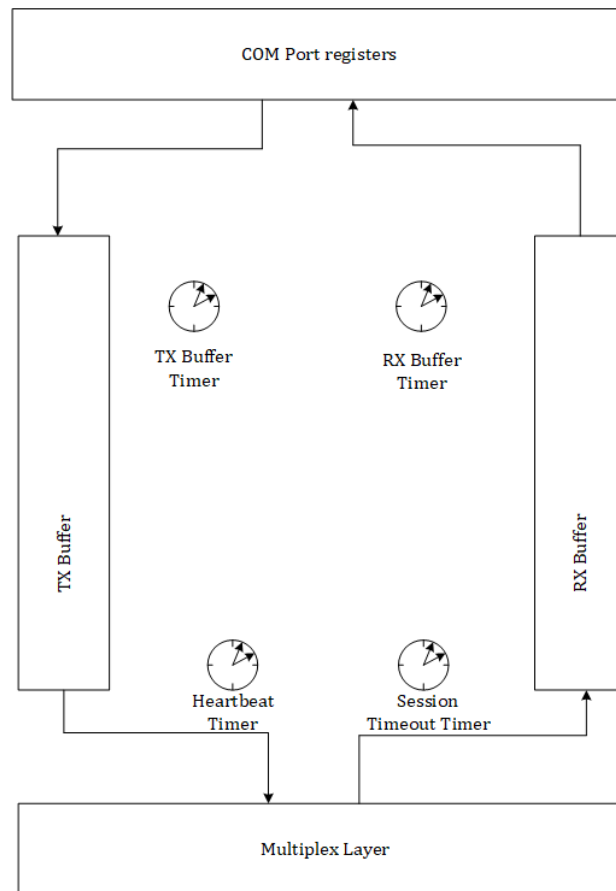


Рис 3.8 – Загальна структура шару ПІ понад LAN

Як більш докладно обговорюється нижче, після появи певних подій команд видаляються з буферу прийому та розпакуються в регістри ПІ. Такі команди можуть містити байти даних, які передаються центральному процесору-хазяїну через регістр RBR або можуть містити керуючі сигнали, передані через регістр MSR.

Рівень III понад LAN включає буферний таймер передачі. Коли таймер буферу передачі закінчується, дані у буфері передачі видаляються та упаковуються в команди, які передаються мультиплексному шару. Команда StartSOLRedirection може налаштувати таймер буферу передачі, щоб закінчився після певного періоду часу (наприклад, команда StartSOLRedirection може налаштувати буфер передачі, що закінчиться через 10 або 20 мс). Дані також видаляються з буфера передачі, коли буфер заповнюється. Кожного разу, коли буфер передачі вичерпується, таймер буферу передачі скидається.

Як згадувалося вище, шар III понад LAN також містить буфер прийому. Коли прийом буферного таймера минув, байти даних у буфері прийому скидаються, тому що хост, здається, занадто повільний для читання даних. Команда StartSOLRedirection може налаштувати таймер буферу прийому, щоб пройти після певного періоду часу. Дані також видаляються з буферу прийому, коли буфер заповнюється. Кожного разу, коли буфер прийому вичерпується, таймер буферного прийому скидається.

Команда StartSOLRedirection також може налаштувати таймер серцевого ритму, що закінчився після вибраного періоду часу. Після закінчення таймера серцебиття, повідомлення про серцебиття надсилається на консоль управління, щоб показати, що сеанс серійного інтернету все ще активний. Таймер серцебиття скидається щоразу, коли надсилається повідомлення серцебиття або будь-яке інше повідомлення / команда надсилається на консоль управління.

Команда StartSOLRedirection також може налаштувати таймер очікування сеансу, щоб закінчитися після вибраного періоду часу. Після закінчення таймера очікування сеансу завершується сеанс III, оскільки виходить висновок, що консоль управління більше не підтримує зв'язок з обчислювальною системою під керуванням з тієї чи іншої причини. Таймер таймера сеансу скидається при кожному отриманні повідомлення з консолі керування.

Команда StartSOLRedirection також може встановити початкове значення для послідовного номера консолі. Кожне наступне повідомлення з консолі управління

включає в себе номер послідовності консолі, причому кожне наступне повідомлення містить послідовний номер, який був збільшений на одиницю. Послідовний номер консолі гарантує, що його повідомлення надходять і виконуються в правильному порядку в керованій комп'ютерній системі.

Після отримання команди `StartSOLRedirection` ПІ понад LAN переходить із початкового стану SOL в приєднаний стан. Щоб вказати, що такий перехід був зроблений, повідомлення на `StartSOLRedirectionReply` надсилається на консоль керування. Повідомлення `StartSOLRedirectionReply` інформує консоль керування про те, що початок сеансу по локальній мережі. Крім того, це може вказувати розмір буфера прийому. Консоль управління не повинен надсилати дані в блоках, більших за розмір буфера прийому, тому що рівень ПІ понад LAN не зможе ввести такий блок даних у буфер.

Хоча потік переходу стану, зображений на рис. 3.7 описує консоль керування, яка ініціює сеанс послідовної локальної мережі, керована система може також запускати сеанс. У такому випадку керована система може передавати команди `StartRedirectionSession` та `StartSOLRedirection` на консоль управління замість консолі керування, яка передає такі команди керованій системі.

Під час перебування підключеного стану, послідовний шар за шаблоном очікує на випадок певних подій [15]. У відповідь на ці події шар за шаром ПІ реагує, як показано на рис. 3.9.

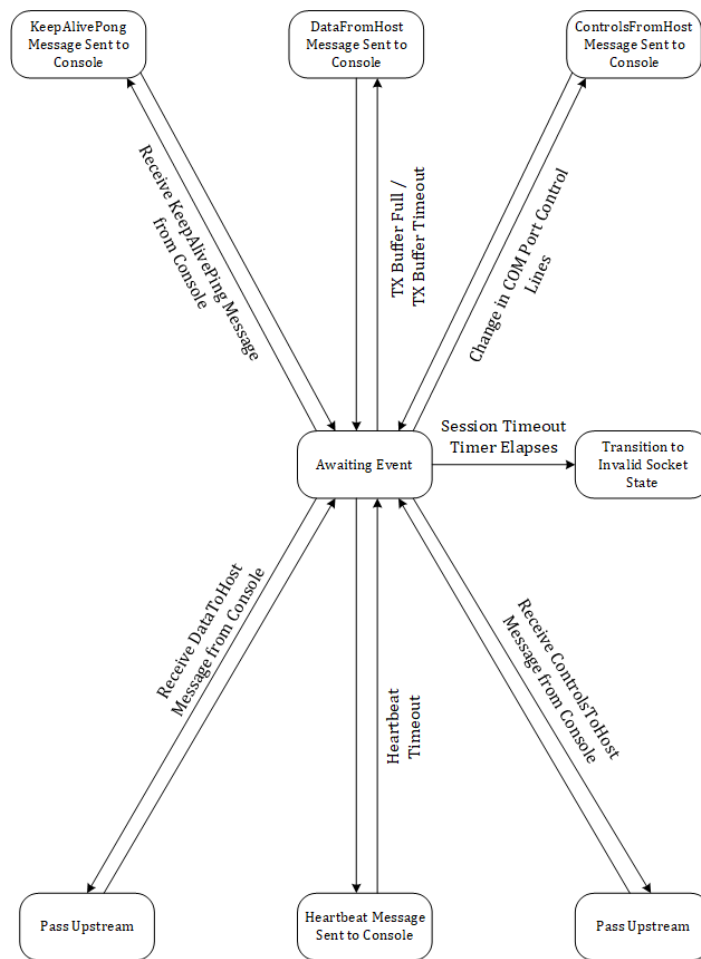


Рис 3.9 – Діаграма реакцій ПІ понад LAN

Як коротко описано вище, коли буфер передачі стає повним або коли проходить таймер буферу передачі, приєднаний стан переходить від операції очікування до передачі операції надсилання даних DataFromHost. Під час цієї операції повідомлення DataFromHost надсилається з шару ПІ понад LAN до мультиплексного шару для остаточної доставки на консолі керування. Повідомлення DataFromHost передає дані, отримані з буфера передачі, на консоль управління, використовуючи ідентифікатор типу повідомлення, щоб ідентифікувати блок даних як перенаправлені дані (наприклад, дані перенаправлення даних екрана тексту), на відміну від деяких інших типів даних, наприклад, дані керування з ПІ. Після надсилання повідомлення

DataFromHost до мультиплексного шару, приєднане стан повертається до операції, що очікує на появу.

Як показано на рис. 3.9, консоль керування може передавати повідомлення KeepAlivePing. Консоль керування може надсилати таке повідомлення керованій обчислювальній системі, коли проходить певний проміжок часу, не отримуючи повідомлення від системи або не надсилаючи повідомлення до системи. У відповідь на отримання повідомлення KeepAlivePing приєднане стан переходить від операції очікування до операції надсилання операції KeepAlivePong. Під час цієї операції повідомлення KeepAlivePong повертається на консоль управління (за допомогою мультиплексного шару). Після передачі повідомлення KeepAlivePong приєднане стан повертається до операції, що очікує на появу. Завдяки цьому обміну "ping" та "pong" консоль управління може перевірити, що сеанс все ще активний. Якщо після передачі повідомлення KeepAlivePing проходить певний проміжок часу без консолі керування, яка отримує повідомлення KeepAlivePong, консоль управління закриває сеанс ПІ понад LAN, виходячи з припущення, що сеанс більше не активний з однієї причини або іншої.

Як також показано на рис. 3.9, після закінчення таймера серцебиття, перехід з підключеного стану від операції очікування до передачі сигналу серцебиття. Це повідомлення виконує аналогічну мету для KeepAlivePong, тобто він підтверджує консолі керування, що сеанс перенаправлення ще є дійсним. Після передачі сигналу серцебиття, пов'язаний стан повертається до очікуваної рівномірної операції.

Коли стан керування ПІ змінює стан, приєднаний стан переходить від операції очікування до операції надсилання команди ControlsFromHost. Під час цієї операції повідомлення ControlsFromHost надсилається на консоль управління. Повідомлення ControlsFromHost містить ідентифікатор типу повідомлення, який вказує на те, що повідомлення доставляє керуючі дані (на відміну від даних екрана текстового режиму). Контрольні дані слідує за ідентифікатором типу повідомлення. Передаючи керуючі дані на консолі керування, консоль керування може бути

інформована про стандартну інформацію про стан ПІ (наприклад, запит на передачу (RTS) у ПІ підтверджено або скасовано, що означає, що процесор не готовий до передачі даних або лінія готовності до терміналу даних (DTR) у ПІ підтверджена або відхилена, що свідчить про те, що процесор є або не готовий до зв'язку).

Коли перенаправлені дані клавіш приймаються з консолі управління, вони вставляються в повідомлення DataToHost. Після одержання такого повідомлення, приєднаний стан переходить від операції очікування до операції, що передається вгору. Під час цієї операції повідомлення DataToHost розпаковується, що означає, що вбудовані в нього дані клавіш витягуються і розміщуються у відповідний регістр у наборі регістрів ПІ для передачі ЦП. Крім того, дані також можуть бути поміщені в буфер прийому, якщо така опція включена. Після завершення операції, пов'язана стан повертається до очікуваної операції події.

Коли контрольні дані ПІ отримуються з консолі управління, вони вбудовуються в повідомлення ControlsToHost. Після одержання такого повідомлення приєднаний стан переходить від операції очікування до операції, що передається вгору, як показано на рис. 3.9. Під час цієї операції повідомлення ControlsToHost розпаковується, що означає, що керуючі дані ПІ вбудовані в нього, витягуються та поміщаються у відповідний регістр (register of status modem-MSR) в межах набору реєстрів ПІ для передачі ЦП. Після завершення операції, пов'язана стан повертається до очікуваної операції події.

Як коротко згадано вище, шар ПІ понад LAN включає таймер таймера сеансу. Таймер таймера сеансу скидається при кожному отриманні повідомлення з консолі керування. Якщо таймер таймера сеансу закінчується, то рівень ПІ понад LAN переходить на недійсний стан сокету (див. рис. 3.7), як показано операцією. Таймер очікування сеансу дозволяє керованій комп'ютерній системі зробити висновок про те, що сеанс перенаправлення був скомпрометований певною мірою (наприклад, зв'язок було втрачено або консоль керування зазнав аварії і т. Д.), А також відповісти на недійсний сеанс перенаправлення .

Коли, нарешті, IT-фахівець на консолі управління завершив свою дистанційну діагностику або роботу з конфігурацією на керованій комп'ютерній системі, він може закрити сеанс перенаправлення. Консоль управління називає закриття сеансу перенаправлення відправкою команди `EndSOLRedirection`, як показано на рис. 3.7. Команда `EndSOLRedirection` отримує послідовний шар через мережу та інформує шар, що завершився сеанс перенаправлення, тобто цей шар повинен припинити пересилання будь-яких повідомлень на консолі керування. Отримання команди `EndSOLRedirection` призводить до того, що рівень ПІ понад LAN переходить у стан закриття. Після переходу в стан закриття, рівень ПІ понад LAN передає консоль управління консоллю керування `EndSOLRedirection`, це повідомлення, яке підтверджує консолі керування, що рівень завершення сеансу перенаправлення завершився послідовним інтерфейсом.

Рівень ПІ понад LAN залишається в стані закриття до отримання команди `EndRedirectionSession`. Команда `EndRedirectionSession` отримує диспетчер сеансів і інформує диспетчера сеансів про завершення сеансу перенаправлення. У відповідь диспетчер сеансів закриває комунікаційний сокет, який використовувався шаром ПІ понад LAN під час сеансу перенаправлення. Тому диспетчер сеансу переходить на роз'єм неправильного стану. Хоча це не зображено, отримання команди `StartRedirectionSession` змушує диспетчера сеансів переходити від недійсного стану розетки до початкового стану, а вище згадане поточне стан починається знову.

Таким чином, у варіанті здійснення, пристрій може отримувати з віддаленого комп'ютера першу команду для зв'язку через локальну мережу з віддаленим комп'ютером. Перша команда може містити принаймні один аргумент для ідентифікації типу сеансу перенаправлення, який вимагається запустити (наприклад, видавши першу команду). Тип сеансу перенаправлення, який може бути ідентифікований аргументом, може включати сеанс послідовної локальної мережі або іншу форму перенаправлення (наприклад, сеанс перенаправлення IDE). Сеанс по локальній мережі може полягати в передачі даних тексту на віддалений комп'ютер.

Текстові дані можуть бути перенаправленні, починаючи від передбачуваного прийому на послідовному порту (наприклад, ПП) до сеансу послідовного по локальній мережі. Пристрій може відповісти на віддалений комп'ютер, чи підтримується цим пристроєм тип перенаправлення, який потрібно запустити. Якщо пристрій підтримує тип запитованого сеансу перенаправлення, пристрій може у відповідь відкрити, принаймні частково, першій команді цей сеанс перенаправлення, за винятком випадків, коли попередньо визначена максимальна кількість дозволених сеансів переспрямування вже відкритий. Якщо тип запитованого сеансу перенаправлення містить сеанс послідовної локальної мережі, пристрій також може (1) отримати другу команду з віддаленого комп'ютера, щоб зупинити переадресацію текстових даних на послідовний перехід на локальну область мережеве сеанс; (2) припинити, у відповідь, принаймні частково, на другу команду, перенаправлення; (3) приймати третю команду з віддаленого комп'ютера, щоб закрити сеанс, і (4) закривають сеанс.

4 МАТЕМАТИЧНА КОНЦЕПЦІЯ ЗАВАДОСТІЙКОГО КАНАЛУ

4.1 Теоретичні засади СЗПД

Під інформаційною розумітимемо будь-яку систему, яка за допомогою технічних засобів виконує одну або кілька таких функцій, як збирання, передавання, перетворення, накопичення, зберігання та оброблення інформації [16].

За функціональною ознакою інформаційні системи можна поділити на: системи електрозв'язку; системи передачі даних; інформаційно-вимірювальні системи; системи перетворення інформації; інформаційно-пошукові системи; системи зберігання інформації; автоматизовані системи управління; системи експериментальних досліджень [16].

Найпоширенішими в повсякденному житті є системи елект-розв'язку та передачі даних, які можна об'єднати назвою систем передані інформації (СПІ). Як приклад розглянемо роботу одноканальної системи передачі даних, структурну схему якої для передачі інформації в одному напрямку зображено на рис. 4.1. Тут ДП і ОП – відповідно джерело і одержувач повідомлення; Пвх і Пвих – відповідно вхідний та вихідний перетворювачі; К1, К2 – кодери; ДК1, ДК2 – декодери; М – модулятор; ДМ – демодулятор [16].

Лінія зв'язку – це фізичне середовище, в якому поширюються сигнали.

Каналом зв'язку називається сукупність технічних засобів, що забезпечує передачу повідомлень від джерела до одержувача.

По одній лінії зв'язку незалежно від передачі повідомлення від інших джерел до інших одержувачів.

Сукупність лінії зв'язку, модулятора та демодулятора (пристроїв перетворення сигналів) утворює неперервний канал передачі інформації, а якщо до цієї сукупності додати ще кодер і декодер, то дістанемо дискретний канал цієї передачі.

Модулятор і демодулятор, конструктивно об'єднані в одному блоці, називаються модемом, а конструктивне об'єднання кодера та декодера – кодеком [16].

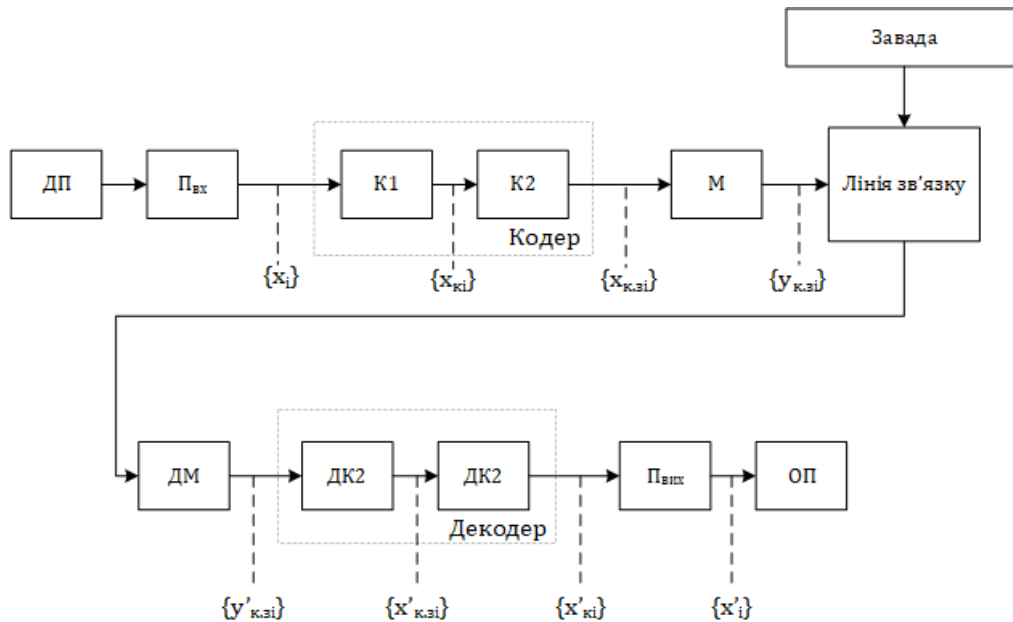


Рис 4.1 – Канал зв'язку [16]

Нехай джерело повідомлень ДП за допомогою вхідного перетворювача $P_{вх}$ створює випадкову дискретну послідовність сигналів $\{x_i\}$, яка подається на кодувальний пристрій кодера $K1$, призначеного для перетворення первинної послідовності сигналів $\{x_i\}$ на послідовність $\{x_{ki}\}$ (див. рис. 4.1). Це перетворення необхідне для більш ефективного використання лінії зв'язку, а також для перетворення дискретної послідовності сигналів одного алфавіту на дискретну послідовність іншого алфавіту [16].

Далі послідовність сигналів $\{x_{ki}\}$ кодером $K2$ перетворюється на дискретну випадкову послідовність сигналів $\{x_{kzi}\}$. Кодер $K2$ виконує функції пристрою захисту інформації від помилок, кодуючи комбінації первинного коду коректувальним кодом, який виявляє та виправляє помилки, що дає змогу зменшити вплив завад і спотворень на інформацію, яка передається. У свою чергу, послідовність сигналів $\{x_{kzi}\}$ після їх

модуляції модуляторі М перетворюється на випадкову послідовність сигналів $\{x_{k,zi}\}$ то однозначно відповідає послідовності $\{x_{kz}\}$ [16].

Під способом передачі сигналів розуміють сукупність операцій перетворення повідомлення на сигнал [5], яку можна подати так:

$$\{y_{k,zi}(t)\} = \Lambda_{\text{прд}}\{x_i\} = \Lambda_M \Lambda_K \{x_i\} \quad (4.1)$$

де $\Lambda_{\text{прд}}$ – оператор способу передачі сигналів;

Λ_M – оператор їх модуляції;

Λ_K – оператор кодування сигналів.

Сигнали при передачі по лінії зв'язку загасають, піддаючись дії завад і спотворень, що спричинює значні відхилення послідовності сигналів на вході приймача $\{y_{п,zi}(t)\}$ від переданої в лінію послідовності сигналів $\{y_{кз}(i)\}$, тобто

$$\{y_{п,zi}(t)\} = \Lambda_{\text{л}}\{y_{к,zi}(t)\} = \Lambda_{\text{л}}\Lambda_{\text{прд}}\{x_i\} \quad (4.2)$$

де $\Lambda_{\text{л}}$ – оператор лінії зв'язку.

Якщо в лінії є адитивна завада у вигляді випадкового процесу $\omega(t)$ то на вході приймача діятиме неперервний випадковий процес

$$y_i(t) = \{y_{п,zi}(t)\} + \omega(t). \quad (4.3)$$

У приймачі після підсилення сигналів, яке необхідне для компенсації загасання їх у лінії зв'язку, сигнали демодулюються в демодуляторі ДМ [16]. На виході останнього утворюється дискретна послідовність сигналів $\{y'_{п,zi}(t)\}$, яка має відповідати послідовності сигналів $\{y_{к,zi}(t)\}$ на виході модулятора М. Цього, однак, може й не бути через дію завад і спотворень у лінії та похибки перетворень сигналів у модуляторі та демодуляторі. Після декодування сигналів декодером ДК2 послідовність $\{y'_{п,zi}\}$ перетворюється на послідовність кодових комбінацій сигналів $\{x'_{ki}\}$, яка має відповідати переданій послідовності сигналів $\{x_{ki}\}$. Ця послідовність

залежатиме від властивостей лінії зв'язку, способу приймання сигналів і коректувального коду, що використовується для їх передачі [16].

Після декодера ДК1, який перетворює послідовність сигналів $\{x'_{ki}\}$ на послідовність $\{x'_i\}$, дискретна послідовність сигналів подається (в разі необхідності перетворення дискретних сигналів на неперервну форму) на вихідний перетворювач $P_{вих}$, з виходу якого вона спрямовується до одержувача повідомлення ОП [16].

Сукупність операцій перетворення сигналів на повідомлення називається способом їх приймання, який можна відобразити так:

$$\{x'_i\} = \Lambda_{прм}[y_i(t)] = \Lambda_{прм}[\{y_{п.зі}(t)\} + \omega(t)], \quad (4.4)$$

де $\Lambda_{прм} = \Lambda_{дм}\Lambda_{дк}$ – оператор способу приймання сигналів;

$\Lambda_{дм}$ – оператор їх демодуляції;

$\Lambda_{дк}$ – оператор декодування сигналів.

З урахуванням (4.1) і (4.4) процес передавання дискретної інформації можна подати у вигляді

$$\{x'_i\} = \Lambda_{дм}\Lambda_{дк}[\Lambda_{л}\Lambda_{м}\Lambda_{к}\{x_i\} + \omega(t)] \quad (4.5)$$

Завдання, яке необхідно вирішити при побудові СПІ, полягає в тому, щоб дістати послідовність сигналів $\{x'_i\}$, яка найменше відрізняється від переданої послідовності сигналів $\{x_i\}$, і забезпечити при цьому високі техніко-економічні показники системи – швидкість передачі інформації, її вірогідність, прийнятну вартість тощо. При побудові СПІ, як правило, задаються ансамблем повідомлень джерела та параметрами лінії (каналу) зв'язку [16].

Одним з основних завдань при проектуванні СПІ є вибір способу передавання інформації, від якого значною мірою залежатимуть рішення щодо вибору окремих вузлів і блоків системи.

У багатоканальних СПІ на відміну від одноканальної забезпечується одночасна та взаємно незалежна передача по одній загальній лінії повідомлень від багатьох джерел (відправників). Спрощену структурну схему СПІ з частотним поділом сигналів показано на рис. 4.2, де $УП_{\text{прд}}$ і $УП_{\text{прм}}$ – відповідно ущільнювальні пристрої передавача та приймача. Решта блоків багатоканальної СПІ за призначенням не відрізняються від одноканальної. За допомогою ущільнювальних передавальних і приймальних пристроїв розносять спектри сигналів, які передаються різними каналами, в діапазоні частот, що відводиться в лінії зв'язку для організації п каналів передачі інформації. В цілому процес математичного опису такої системи не відрізняється від аналогічного опису одноканальної СПІ [16].

4.2 Математичні моделі каналу зв'язку

Від вибору каналу зв'язку залежить не тільки кількість інформації, яку можна передати від передавача до одержувача повідомлень, а й швидкість передачі інформації та її вірогідність.

У той же час точний математичний опис будь-якого реального каналу зв'язку досить складний. Тому на практиці, як правило, користуються більш спрощеними математичними моделями, ЯКІ дають змогу визначити основні закономірності каналів. При побудові таких моделей враховуються найсуттєвіші особливості каналу зв'язку і відкидаються другорядні чинники, що майже не впливають на якість зв'язку [16].

Ідеальний канал зв'язку без завад – це лінійне коло зі сталою функцією передачі, що звичайно зосереджена в обмеженій смузі частот, де завади будь-якого виду відсутні. Вихідний сигнал при заданому вхідному буде детермінованим.

Ця модель каналу зв'язку частіше застосовується для опису кабельних каналів дуже короткої довжини, де наявність адитивних завад ще не відчувається [16].

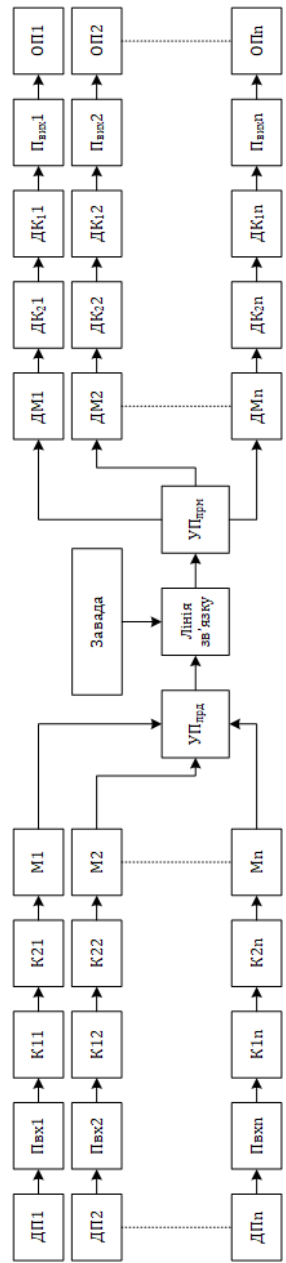


Рис 4.2 – система передачі інформації [16]

Канал зв'язку з адитивним гауссовим шумом – це канал з «білим» або «квазібілим» (з рівномірною спектральною щільністю в смузі спектра сигналу) шумом, у якому коефіцієнт передачі та час затримки сигналів не залежать від часу й

є відомими детермінованими величинами [16]. Вихідний сигнал такого каналу визначається виразом

$$z(t) = k_n s(t - \tau) + \omega(t), \quad (4.6)$$

де $s(t)$ – вхідний сигнал;

k_n – коефіцієнт передачі;

τ – час затримки сигналу в каналі;

$\omega(t)$ – гауссовий адитивний шум з нульовим математичним сподіванням і заданою кореляційною функцією.

Якщо змінити початок відліку часу на виході каналу, то запізнення τ сигналу можна не враховувати.

Модель (4.6) застосовується для опису реальних провідних каналів зв'язку й однопроменевих радіоканалів без завмирань. На метрових хвилях для зв'язку в межах прямої видимості, а також радіоканалів з повільними завмираннями, для яких можна надійно передбачити значення k_n та τ [16].

Канал з невизначеною фазою сигналу – це канал, який відрізняється від попереднього тим, що в ньому запізнення τ є випадковою величиною. Вихідний сигнал такого каналу визначається виразом

$$z(t) = k_n s(t) \cos \theta_k + \tilde{s}(t) \sin \theta_k + \omega(t), \quad (4.7)$$

де $\tilde{s}(t)$ – перетворення Гільберта від $s(t)$;

$\theta_k = \omega_0(t)$ випадкова початкова фаза, причому розподіл імовірностей θ_k найчастіше задається рівномірним в інтервалі від 0 до 2π .

Модель (4.7) використовується для опису тих самих каналів зв'язку, що й попередня, якщо фаза сигналу в них змінюється. Ця флуктуація фази пояснюється незначними змінами довжини каналу, фазовою нестабільністю генераторів носійної

частоти, а також незначними змінами властивостей середовища, через яке передається сигнал [16].

4.3 Циклічні надлишкові коди

Циклічний надлишковий код (англ. Cyclic redundancy check, CRC) — алгоритм обчислення контрольної суми, призначений для перевірки цілісності даних. CRC є практичним додатком завадостійкого кодування, заснованому на певних математичних властивостях циклічного коду [17].

Поняття циклічні коди достатньо широке. У англійській літературі CRC розшифровується двояко в залежності від контексту: Cyclic Redundancy Code або Cyclic Redundancy Check. Під першою розшифровкою розуміють математичний феномен циклічних кодів, під другою — конкретне застосування цього феномену як геш-функції [17].

Перші спроби створення кодів з надлишковою інформацією почалися задовго до появи сучасних ПК. До прикладу, ще в шістдесятих роках минулого століття Рідом і Соломоном була розроблена ефективна методика кодування — код Ріда-Соломона. Використання її у ті часи не представлялося можливим, так як провести операцію декодування за розумний час першими алгоритмами не вдавалося. Крапку в цьому питанні поставила фундаментальна робота Берлекампа, опублікована в 1968 році. Ця методика, на практичне застосування якої вказав через рік Мессі, і донині використовується в цифрових пристроях, що забезпечують приймання RS-кодованих даних. Більш того: дана система дозволяє не тільки визначати позиції, але й виправляти невірні кодові символи (найчастіше октети) [17].

Але далеко не скрізь від коду потрібна корекція помилок. Сучасні канали зв'язку мають прийнятні характеристики, і часто достатньо лише перевірити, чи успішно пройшла передача або виникли будь-які складності; структура ж помилок і конкретні позиції невірних символів абсолютно не цікавлять сторону, яка приймає дані. І в цих

умовах дуже вдалим рішенням виявилися алгоритми, що використовують контрольні суми. CRC як найкраще підходить для подібних задач: невисокі витрати ресурсів, простота реалізації і вже сформований математичний апарат з теорії лінійних циклічних кодів забезпечили їй величезну популярність.

У найзагальнішому своєму вигляді контрольна сума являє собою деяке значення, побудоване за певною схемою на основі кодованого повідомлення. Перевірочна інформація при систематичному кодуванні дописується, найчастіше, на кінець повідомлення — після корисних даних. З приймального боку абонент знає алгоритм обчислення контрольної суми: відповідно, програма має можливість перевірити коректність прийнятих даних [17].

При передачі пакетів по реальному каналу, зрозуміло, можуть виникнути спотворення вихідної інформації внаслідок різних зовнішніх впливів: електричних наведень, поганих погодних умов і багатьох інших. Сутність методики в тому, що при хороших характеристиках хеш-функції в переважній кількості випадків помилка в повідомленні призведе до зміни обчисленого на прийомі значення CRC. Якщо вихідна і обчислена суми не рівні між собою, ухвалюється рішення про недостовірність отриманих даних, і можна запитати повторну передачу пакета [17].

Алгоритм CRC базується на властивості ділення з остачею двійкових многочленів, тобто многочленів над скінченним полем. Значення CRC є по суті остачею від ділення многочлена, відповідного вхідним даним, на деякий фіксований породжувальний многочлен [17].

Кожній послідовності бітів взаємно однозначно зіставляється двійковий многочлен

$$\sum_{n=0}^{N-1} a_n x^n, \quad (4.8)$$

послідовність коефіцієнтів якого являє собою початкову послідовність. Наприклад, послідовність бітів 1011010 відповідає многочлену:

$$P(x) = 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0 = x^6 + x^4 + x^3 + x^1. \quad (4.9)$$

Кількість різних многочленів степені меншоїдорівнює , що збігається з числом всіх двійкових послідовностей довжини . Значення контрольної суми в алгоритмі з породжуючим многочленом степенізадається як бітова послідовність довжини , яка представляє многочлен , отриманий в остачі при діленні многочлена , який представляє вхідний потік біт, на многочлен :

$$R(x) = P(x) \cdot x^N \text{ mod } G(x) \quad (4.10)$$

де

$R(x)$ — многочлен, який представляє значення CRC;

$P(x)$ — многочлен, коефіцієнти якого представляють вхідні дані;

$G(x)$ — породжувальний многочлен;

N — степінь породжувального многочлена.

Множення x^N здійснюється приписуванням нульових бітів до вхідної послідовності, що покращує якість гешування для коротких вхідних послідовностей [17].

При діленні з остачею початкового многочлена на породжуючий многочлен $G(x)$ степені N можна отримати 2^N різних остач від ділення. $G(x)$ найчастіше є незвідним многочленом. Зазвичай його підбирають відповідно до вимог до геш-функції у контексті кожного конкретного застосування. Проте, існує багато стандартизованих породжувальних многочленів, що володіють добрими математичними та кореляційними властивостями (мінімальне число колізій, простота обчислення), деякі з котрих приведені нижче [17].

Обчислення CRC. Параметри алгоритму

Одним з основних параметрів CRC є породжувальний многочлен. З породжувальним многочленом пов'язаний інший параметр — його степінь, який

визначає кількість бітів, застосованих для обчислення значення CRC. На практиці найбільш поширені 8-, 16- та 32-бітові слова, що є наслідком особливостей архітектури сучасної обчислювальної техніки [17].

Ще одним параметром є початкове (стартове) значення слова. Вказані параметри повністю визначають «традиційний» алгоритм обчислення CRC. Існують також модифікації алгоритму, наприклад, які використовують зворотний порядок обробки бітів.

Опис процедури

З файлу береться перше слово — це може бути бітовий (CRC-1), байтовий (CRC-8) або будь-який інший елемент. Якщо старший біт у слові «1», то слово зсувається вліво на один розряд з подальшим виконанням операції XOR з породжувальним многочленом. Відповідно, якщо старший біт у слові «0», то після зсуву операція XOR не виконується. Після зсуву втрачається старий старший біт, а молодший біт звільняється — його значення встановлюється рівним нулю. На місце молодшого біту завантажуються черговий біт із файлу, й операція повторюється до тих пір, поки не завантажиться останній біт файлу. Після проходження всього файлу, в слові залишається остача, яка і є контрольною сумою [17].

Популярні й стандартизовані многочлени

В той час, як циклічні надлишкові коди є частиною стандартів, у цього терміну не існує загальноприйнятого визначення — трактування різних авторів нерідко суперечать один одному.

Цей парадокс стосується й вибору многочлена-генератора: найчастіше стандартизовані многочлени не є найбільш ефективними в плані статичних властивостей відповідного їм *check redundancy code*.

При цьому багато широко використовуваних многочленів не є найефективнішими із всіх можливих. У 1993—2004 роках група вчених займалася

дослідженням породжувальних многочленів розрядності до 16, 24 та 36 біт й знайшла многочлени, які дають кращу, ніж стандартизовані многочлени, продуктивність у сенсі кодової відстані. Один із результатів цього дослідження вже знайшов своє застосування в протоколі iSCSI.

Найпопулярніший та рекомендований IEEE многочлен для CRC-32 використовується в Ethernet, FDDI; також цей многочлен є генератором коду Геммінга. Використання іншого многочлену — CRC-32C — дозволяє досягти такої ж продуктивності при довжині вихідного повідомлення від 58 біт до 131 кбіт, а в деяких діапазонах довжини вхідного повідомлення може бути навіть більше — цьому в наш час він також користується популярністю. Наприклад, стандарт ITU-T G.hn використовує CRC-32C з ціллю виявлення помилок в корисному навантаженні [17].

Таблиця 4.1 – найбільш розповсюджені многочлени [17].

Назва	Многочлен	Використання	Представлення		
			Нормальне	Реверсоване	Реверсоване від зворотнього
CRC-1		використовується в апаратному контролі помилок; також відомий як біт парності	0x1	0x1	0x1
CRC-4-ITU		ITU G.704 [4]	0x3	0xC	0x9
CRC-5-EPC		Gen 2 RFID ^[5]	0x09	0x12	0x14
CRC-5-ITU		ITU G.704 ^[6]	0x15	0x15	0x1A
CRC-5-USB		USB token packets	0x05	0x14	0x12

Продовження таблиці 4.1

CRC-6-ITU		ITU G.704 ^[7]	0x03	0x30	0x21
CRC-7		системи телекомунікації, ITU-T G.707 ^[8] , ITU-T G.832 ^[9] , MMC, SD	0x09	0x48	0x44
CRC-8-CCITT		ATM HEC, ISDN Header Error Control and Cell Delineation ITU-T I.432.1 (02/99)	0x07	0xE0	0x83
CRC-8-Dallas/Maximum		1-Wire bus	0x31	0x8C	0x98
CRC-8		ETSI EN 302 307, 5.1.4 ^[10]	0xD5	0xAB	0xEA
CRC-8-SAE J1850			0x1D	0xB8	0x8E
CRC-10			0x233	0x331	0x319
CRC-11		FlexRay ^[11]	0x385	0x50E	0x5C2
CRC-12		системи телекомунікації	0x80F	0xF01	0xC07
CRC-15-CAN			0x4599	0x4CD1	0x62CC
CRC-16-IBM		Bisync, Modbus, USB, ANSI X3.28 ^[20] , багато інших; також відомий як <i>CRC-16</i> та <i>CRC-16-ANSI</i>	0x8005	0xA001	0xC002
CRC-16-CCITT		X.25, HDLC, XMODEM, Bluetooth, SD та інші	0x1021	0x8408	0x8810
CRC-16-T10-DIF		SCSI DIF	0x8BB7	0xEDD1	0xC5DB
CRC-16-DNP		DNP, IEC 870, M-Bus	0x3D65	0xA6BC	0x9EB2
CRC-24		FlexRay	0x5D6DCB	0xD3B6BA	0xAE B6E5
CRC-24-Radix-64		OpenPGP	0x864CFB	0xDF3261	0xC3267D
CRC-30		CDMA	0x2030B9C7	0x38E74301	0x30185CE3

Продовження таблиці 4.1

CRC-32-IEEE 802.3		V.42, MPEG-2, PNG, POSIX cksum	0x04 C11 DB7	0xED B883 20	0x82 608E DB
CRC-32C (Castagnoli)		iSCSI, G.hn payload	0x1E DC6 F41	0x82 F63B 78	0x8F 6E37 A0
CRC-32K (Koopman)			0x74 1B8C D7	0xEB 31D8 2E	0xB A0D C66B
CRC-32Q		авіація; AIXM	0x81 4141 AB	0xD5 8282 81	0xC0 A0A 0D5
CRC-64-ISO		HDLC — ISO 3309	0x00 0000 0000 0000 1B	0xD8 0000 0000 0000 00	0x80 0000 0000 0000 0D
CRC-64-ECMA			0x42 F0E1 EBA 9EA3 693	0xC9 6C57 95D7 870F 42	0xA1 7870 F5D4 F51B 49
CRC-1		використовується в апаратному контролі помилок; також відомий як біт парності	0x1	0x1	0x1

Специфікації алгоритмів CRC [17]

Однією з найвідоміших є методика Ross N. Williams^[12]. У ній використовуються наступні параметри:

- Назва алгоритму (name);
- Ступінь породжує контрольну суму многочлена (width);
- Сам виготовляючий поліном (poly). Для того, щоб записати його у вигляді значення, його спочатку записують як бітову послідовність, при цьому старший біт опускається - він завжди дорівнює 1. Наприклад, многочлен в даній нотації буде записаний числом. Для зручності отримане двійкове подання записують в шістнадцятковій формі. Для нашого випадку воно буде дорівнює або 0x11;

- Стартові дані (init), тобто значення регістрів на момент початку обчислень;
- Прапор (RefIn), який вказує на початок і напрямок обчислень. Існує два варіанти: False - починаючи зі старшого значущого біта (MSB-first),^[13] або True - з молодшого (LSB-first);^[14]
- Прапор (RefOut), що визначає, інвертується чи порядок бітів регістра при вході на елемент XOR;
- Число (XorOut), з яким складається по модулю 2 отриманий результат;
- Значення CRC (check) для рядка «123456789».

5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ЗАВАДОСТІЙКОЇ ПЕРЕДАЧІ ДАНИХ ІЗ ВИКОРИСТАННЯМ ВІРТУАЛЬНОГО ПОСЛІДОВНОГО ПОРТУ

5.1 Типове рішення для створення ВПП на основі протоколу TCP/IP

Більшість сучасних настільних і портативних комп'ютерів більше не мають старих ПП DB9 RS-232, які традиційно використовувались для безпосереднього підключення до апаратного забезпечення, що використовується в комерційних, лабораторних або промислових системах безпеки та управління. У багатьох випадках, навіть якщо вони є, кількість доступних портів дуже обмежена. Це може бути особливо проблематично, якщо ви все ще покладаетесь на застарілі програмні додатки, які спеціально спілкуються через ці неіснуючі (або дуже обмежені) послідовні порти. Це трохи інша проблема, ніж просто перетворення інтерфейсу послідовного пристрою в USB або інтерфейс Ethernet, оскільки застаріле програмне забезпечення не вміє розпізнавати чи читати USB або Ethernet - і воскресити інженера-програміста, який спочатку кодував його, ймовірно не вибір.

Нам все ж потрібно періодично оновлювати комп'ютери, щоб відповідати найновішим вимогам ОС та безпеки, але це не означає, що ваші застарілі активи серійного програмного забезпечення повинні раптом бути марними. ВПП допоможе вам продовжити використання вашої системи ПП, що може використовуватися для передачі послідовних даних через мережу (Інтернет або LAN) до серверів. Це заощаджує час і гроші, дозволяючи продовжувати використовувати існуючі застарілі програми.

Розглянемо базову віртуальну конфігурацію ПП, щоб відобразити їх.



Рис 5.1 – Схема роботи ПЗ

У нашому сценарії "застарілий серійний додаток" - це програма MTTTY, встановлена на мікрокомп'ютер, що має інтерфейси Ethernet та USB-порт. По-перше, ми створимо віртуальний послідовний порт на COM20, який надсилатиме дані до серійного порту нашого застосунку до сервера пристроїв Ethernet (S2E) через Ethernet з'єднання. Пристрій потім надсилає отримані дані мережі з власного послідовного порту (DB9) на ноутбук. Ці дані будуть прочитати через порт USB ноутбука за допомогою USB-послідовного конвертера. Ми будемо використовувати другий термінал MTTTY для читання послідовних даних з нашого перетворюваного USB послідовного порту COM6. Очікувана остаточна поведінка полягає в тому, що якщо ми надсилатимемо дані з одного вікна, воно з'явиться в іншому, і навпаки.

5.2 Сценарій використання ПЗ

Типовий сценарій використання ПЗ включає наступні кроки:

1. Встановлення ПЗ на сервер, зокрема драйвера ВПП.
2. Налаштування IP-адреси пристрою та номеру порту прослуховування. Для цього можна використати зокрема утиліту IPSetup, яка є у вільному доступі в мережі Інтернет.
3. Перевірка підключення пристрою ПІ понад LAN або основного модуля до маршрутизатора.

Приклад використання IPSetup.

Відкрити ПЗ IPSetup. Пристрій має з'явитись у полі "Select a unit", але якщо не натиснути кнопку "Search again", він має з'явитися. Якщо ваш пристрій підтримує бездротовий зв'язок, натисніть значок "+", щоб розгорнута інформація відображала поточну IP-адресу пристрою. В іншому випадку він повинен бути вказаний у головному рядку запису пристрою. На малюнку нижче вибраний рядок IP 10.1.1.130. Змінити цю IP-адресу для подальшого використання.

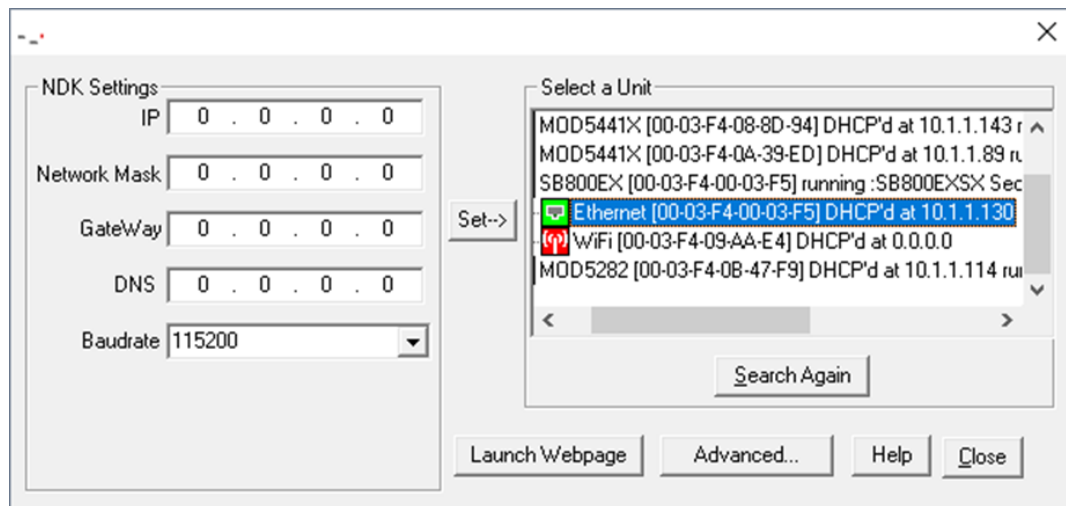


Рис 5.2 – Вікно налаштувань мережевих адрес

4. Перейти на домашню сторінку вашого пристрою та відкрити його прослуховуючий порт. Ви можете просто натиснути кнопку "Launch webpage" в

IPSetup. Потім натиснути на гіперпосилання "TCP" в меню. На малюнку нижче показана сторінка TCP для пристрою ПІ понад LAN, що має два послідовних порти, порт 0 та порт 1, відповідно.

TCP		Port 0	Port 1
Listen for incoming network connections	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Listening network port:	<input type="text" value="23"/>	<input type="text" value="24"/>	<input type="text" value="24"/>
Timeout and disconnect after this many seconds of inactivity.	<input type="text" value="60"/>	<input type="text" value="60"/>	<input type="text" value="60"/>
Allow new connection if the existing connection has been idle for this many seconds.	<input type="text" value="30"/>	<input type="text" value="30"/>	<input type="text" value="30"/>
When to begin making outgoing tcp connections:	<input type="text" value="Never"/>	<input type="text" value="Never"/>	<input type="text" value="Never"/>
Connect on network port:	<input type="text" value="1000"/>	<input type="text" value="1000"/>	<input type="text" value="1000"/>
Connect to this address:	<input type="text" value="(Enter IP Address)"/>	<input type="text" value="(Enter IP Address)"/>	<input type="text" value="(Enter IP Address)"/>
Alternate address:	<input type="text" value="(Enter IP Address)"/>	<input type="text" value="(Enter IP Address)"/>	<input type="text" value="(Enter IP Address)"/>
Timeout and disconnect after this many seconds of inactivity.	<input type="text" value="60"/>	<input type="text" value="60"/>	<input type="text" value="60"/>
Retry failed outgoing connections after this many seconds.	<input type="text" value="360"/>	<input type="text" value="360"/>	<input type="text" value="360"/>
Check and maintain valid connection at intervals in seconds.	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Use custom packetization logic (below)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Number of characters to accumulate before sending TCP packet:	<input type="text" value="32"/>	<input type="text" value="32"/>	<input type="text" value="32"/>
Number of msec to wait for accumulated characters: 0 waits forever.	<input type="text" value="100"/>	<input type="text" value="100"/>	<input type="text" value="100"/>
Flush TCP frame when this character is received (Enter NA to disable):	<input type="text" value="NA"/>	<input type="text" value="NA"/>	<input type="text" value="NA"/>
USE SSL rather than TCP for connections:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Always Save Serial Chars regardless of connection status:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Network Settings on Serial Port - Advanced Serial Settings			

Рис 5.3 – Вікно налаштувань портів

5. Повернутись у програму Virtual COM Port, клацніть на «Add» у вікні ВПІ, в якому відкриється вікно «Edit connection».

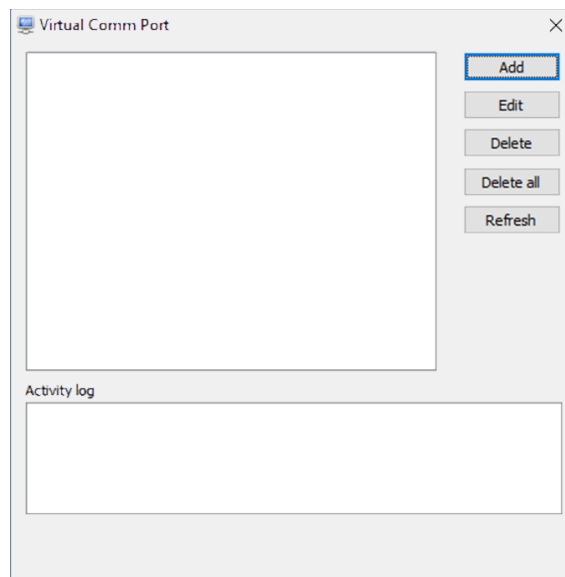


Рис 5.4 – список існуючих ПП

6. Далі у спадному меню "Select serial port" виберіть кому порт, який потрібно зв'язати, і дайте йому нове ім'я. У цьому випадку обрано COM20 і, відповідно, названо "Client_COM20". Після цього необхідно ввести IP-адресу пристрою у поле з міткою «Remote host name/port», після чого слід обрати прослуховуючий порт, визначений під час попередніх кроків.

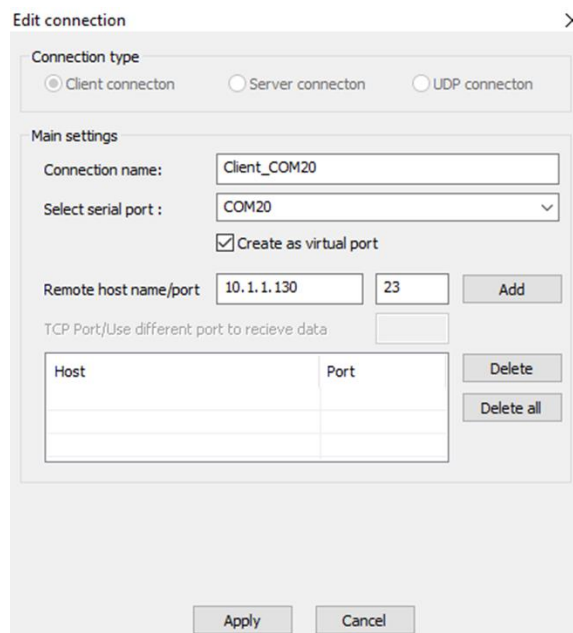


Рис 5.5 – Вікно налаштувань з'єднань

7. Натиснути "Add", переконатись, що запис був доданий до списку, а потім натиснути "Apply".

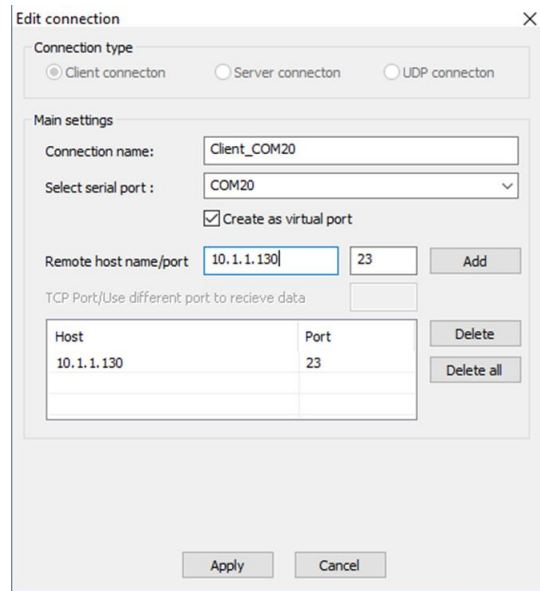


Рис 5.6 – Додане з'єднання

8. Тепер можливо побачити, що наш ВПП був успішно доданий до головного вікна програми Virtual Comm Port.

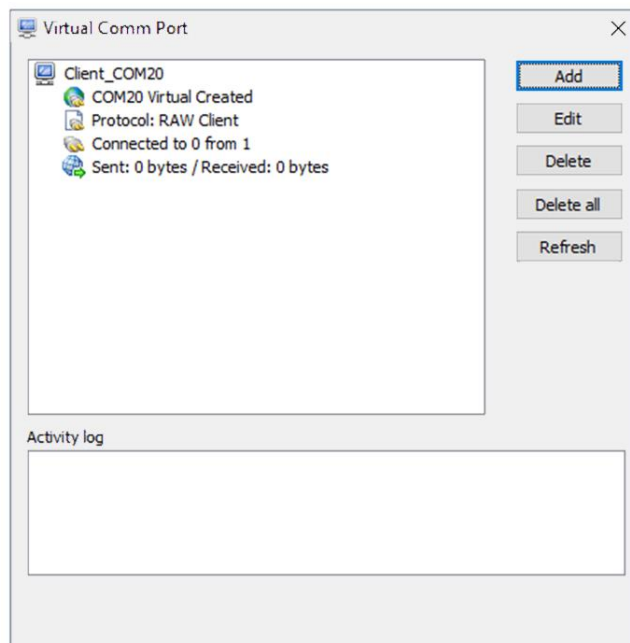


Рис 5.7 – Головне вікно програми

9. Відкрити два багатопотокові вікна TTY. Для послідовного порту USB буде використано перше вікно, а для ВПП буде використано друге вікно. На рис 5.8, COM20 – це встановлений нами ВПП, а COM6 підключається безпосередньо до пристрою через ПІ. Натиснути “connect” та надіслати дані з ВПП або фізичного ПІ, щоб підтвердити, що дані успішно передаються.

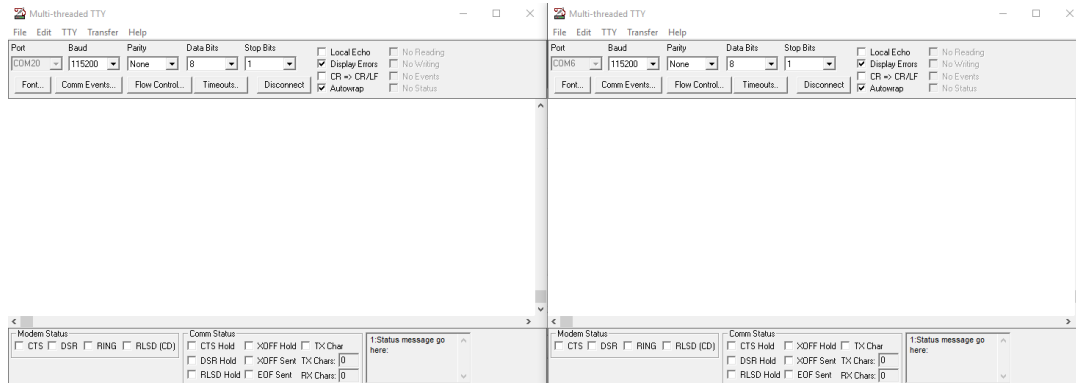


Рис 5.8 – демонстрація роботи ВПП

5.2 Веб-сервер «Flask»

Максимум результатів при мінімальних витратах

Це ідеал усіх бізнес-процесів, включаючи розробку програмного забезпечення. Як отримати цей ефект для проектів, які мають короткі терміни та високі очікування щодо продуктивності? На жаль, немає простої відповіді: якість програмного забезпечення вимагає як часу, так і грошей, але ... вибір технологій, інструментів та моделей може виявитися критичним для життєвого циклу продукту. Відповідні рішення можуть істотно поліпшити кінцевий час та вартість проекту. Рішення REST API в технології Python може стати основою для ефективного розробки програмного забезпечення з поглибленим аналізом та урахуванням досвіду побудування проектів [18].

Технологія досліджень та орієнтири

Наведена нижче таблиця показує результат тесту, проведений TechEmpower . Він показує кількість відповідей у форматі JSON, на HTTP-запит за 1 секунду для різних технологій, включаючи Python. Для окремих рішень були досягнуті наступні результати [18]:

Latency of JSON responses, dedicated i7 hardware			
Framework	Average latency	σ (SD)	Max
vertx	0.4 ms 0.2%	0.5 ms	2.0 ms
servlet	0.6 ms 0.4%	0.4 ms	1.1 ms
gemini	0.9 ms 0.5%	0.3 ms	1.1 ms
netty ✓	1.1 ms 0.6%	0.2 ms	1.6 ms
go ✓	1.4 ms 0.8%	0.9 ms	3.9 ms
compojure ✓	1.8 ms 1.1%	7.9 ms	55.1 ms
spring ✓	2.2 ms 1.3%	5.5 ms	34.6 ms
http-kit ✓	2.3 ms 1.4%	1.4 ms	5.1 ms
wicket	2.7 ms 1.6%	1.1 ms	8.3 ms
nodejs ✓	3.0 ms 1.8%	1.3 ms	5.7 ms
tapestry ✓	3.4 ms 2.1%	11.8 ms	61.8 ms
play-scala ✓	4.5 ms 2.7%	2.6 ms	15.9 ms
webgo	5.0 ms 3.0%	4.4 ms	17.7 ms
grails ✓	5.4 ms 3.2%	15.9 ms	144.3 ms
express	6.2 ms 3.7%	2.7 ms	16.1 ms
play ✓	8.1 ms 4.9%	14.9 ms	212.5 ms
wsgi ✓	10.2 ms 6.2%	2.2 ms	20.1 ms
php ✓	10.5 ms 6.3%	16.5 ms	154.2 ms
rack-jruby	11.0 ms 6.6%	28.1 ms	287.4 ms
yesod ✓	12.5 ms 7.5%	14.2 ms	221.8 ms
flask ✓	21.8 ms 13.1%	3.1 ms	41.5 ms
rack-ruby	33.1 ms 19.9%	1.6 ms	42.6 ms
django-stripped ✓	46.2 ms 27.8%	61.3 ms	201.3 ms
sinatra-ruby	55.2 ms 33.2%	6.5 ms	79.2 ms
django ✓	59.2 ms 35.6%	93.2 ms	1420.0 ms
rails-stripped-ruby ✓	69.8 ms 42.0%	7.9 ms	111.2 ms
rails-stripped-jruby ✓	81.6 ms 49.1%	38.3 ms	212.4 ms
rails-ruby	82.5 ms 49.6%	8.9 ms	124.2 ms
sinatra-jruby	92.4 ms 55.6%	52.1 ms	405.8 ms
rails-jruby	95.9 ms 57.7%	42.0 ms	216.7 ms
cake ✓	166.2 ms 100.0%	147.2 ms	1010.0 ms

Рис 5.9 – Продуктивність рішень відповідей у форматі JSON на HTTP-запити

Після аналізу цих результатів вбачаються декілька способів їх поліпшення, зокрема:

- Оптимізація бази даних PostgreSQL на основі використання PgBouncer та pgPool, (у версіях Django від 1.6);
- рендеринг шаблонів за допомогою Jinja;

- використання `ultrajson` замість стандартної бібліотеки для серіалізації даних.

Застосування тих чи інших методів може суттєво вплинути на продуктивність додатків.

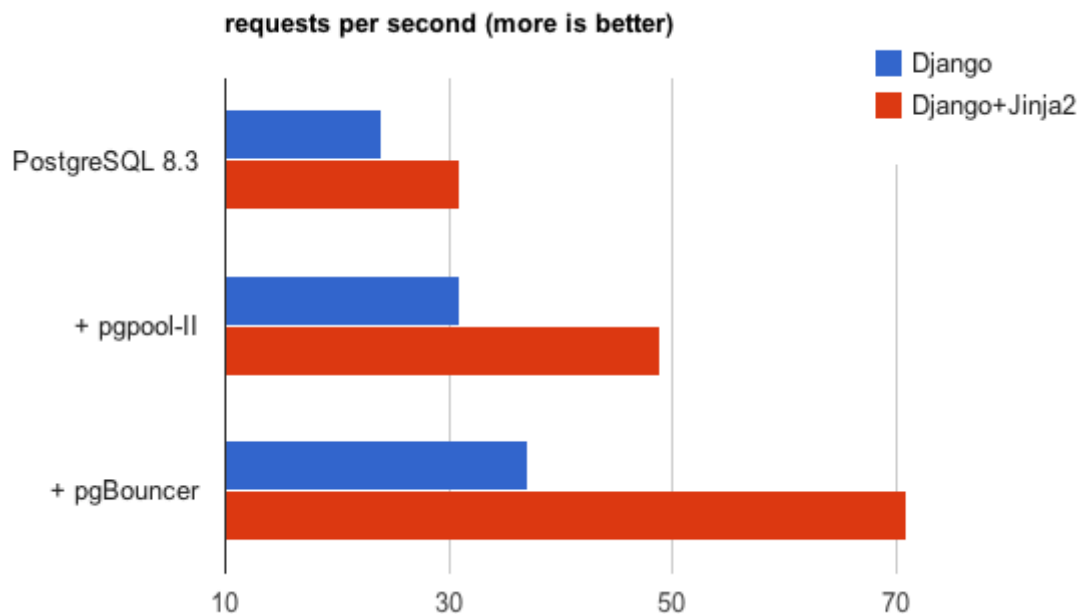


Рис 5.10 – Порівняння продуктивності бази даних PostgreSQL спільно з Django, з використанням `pgBouncer` + `pgPool` + `Jinja`, представлено в блозі `AskThePony` [18]

Вступ

У багатьох випадках API являє собою вузьке місце всієї IT-системи, високо завантаженого компонента, відповіді якого повинні бути швидкими і не справляти негативного враження на користувача від програми. Нижче наведено порівняння переваг та недоліків вибраних технологій REST, але спочатку вихідний код показує реалізацію додатку `WebService`.

`Flask`. Спеціальний клас `MethodView` розпізнає окремі типи запитів REST на основі загального способу відправлення.

Бібліотека RESTless

Бібліотека RESTless для Flask дозволяє легко додавати багато корисних функцій, зокрема:

1. Версія
2. Перевірка
3. Уточнення, які дані повинні бути доступними через API
4. Послідовність результатів
5. Попередня і завершальна обробка
6. Аутентифікація
7. Фільтрація

Таким чином, використання мікро-фреймворка, такого як Flask разом з бібліотекою RESTless, є хорошим рішенням, особливо, якщо API додається до існуючих програм.

Переваги Flask:

- дуже хороша продуктивність
- швидкий і простий прототип API
- простий і швидкий у навчанні навіть для людей без досвіду використання Flask
- може працювати на разі на одному сервері з Django
- за замовчуванням використовує SQLAlchemy, що надає більше параметрів, ніж Django.

Недоліки цього рішення:

- розсіяна документація, тому що у Flask є багато доповнень

- більше зусиль потрібно докласти до створення логічної структури програми
- неможливість використання методів Django і своїх власних бібліотек для обробки логіки додатка.

REST, доступні для Django

Є багато доступних бібліотек REST Django, на цей час однією з найбільш популярних є Django- rest-framework, тому що вона побудована на передових практиках, динамічно розвивається, має швидку і всебічну підтримку і поточну документацію. Django-rest-framework - це дуже просунутий інструмент, тоді як нижче представлена лише сама основна реалізація Webservice, що складається з наступних компонентів:

1. Визначення моделі даних
2. Впровадження серіалізатора - це клас, що описує, яким чином модель має виглядати в самому API
3. Відображення (вид)
4. URL

Переваги цього рішення:

- більш просунутий і розширюваний інструмент, ніж Flask
- впровадження, адаптоване до різноманітних потреб
- можливість використання класів та методів, реалізованих в Django та власному коді
- швидке автоматичне тестування (засноване на TestCase Django).

недоліки:

- □ недостатня підтримка вкладеного об'єкта, внаслідок чого для інтеграції, наприклад, з RestKit - допоміжною бібліотекою, яка підтримує REST на платформі iOS вимагає більше часу.
- □ менша продуктивність, ніж у Flask (але у відповідних блогах зустрічаються поради про те, як профільувати систему django-rest-для отримання кращих результатів).

Підсумки

Створення API-інтерфейсів на основі використаних рішень, зокрема архітектури django-rest, може бути дуже швидким, простим та принести багато переваг проекту. Структура програми, яка відповідає за API, є більш логічною, оскільки вона визначена моделями Django. Flask також є хорошим рішенням, особливо, коли API повинно бути сильно завантажено.

6 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

6.1 Опис ідеї проекту

Метою розділу є розроблення системи завадостійкої передачі даних із використанням віртуального послідовного порту. Розглянемо опис та зміст стартап-проекту табл. 6.1, основними складовими якої є:

- зміст ідеї
- можливі напрямки застосування;
- вигоди для користувача.

Таблиця 6.1 – Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
	1. Універсальність	Відсутність необхідності шукати одночасно і адаптер, і драйвер для нього
	2. Доступність	Низька собівартість дозволяє створити дешевий продукт
	3. Легкість у використанні	Кінцевий користувач не має мати спеціальних навичок для використання продукту

Дана система відрізняється від конкурентів наявністю повного набору інструментів в рамках себе (тобто це готовий до використання продукт, що не вимагає додаткових пристроїв або програмного забезпечення), низька собівартість компонентів, простота використання.

Основними конкурентами на ринку є TCP COM BRIDGE від AGG Software, USB to Serial Adapter від U.S. Converters LLC та Адаптер COM RS232. Вони значно відрізняються від нашого продукту. Докладний огляд сильних, слабких та нейтральних характеристик ідеї продукту у порівнянні з конкурентами розглянуто на табл. 6.2

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ n/n	Техніко- економічні характеристик и ідеї	<i>(потенційні) товари/концепції конкурентів</i>			
		<i>Мій проект</i>	TCP COM BRIDGE	USB to Serial Adapter	Адаптер COM RS232
1.	Економічні	\$20 за одиницю	\$30 за ліцензію на обмежений період	\$25 за одиницю	\$3 за одиницю
2.	Технологі- чність	Мультипла- тформа	Лише ПЗ	Фіксовані підклю- чення	Фіксовані підключення
3.	Сумісність з ОС	До Windows 10	До Windows 8	До Windows 7	До Windows XP

4.	Надійність	Відказо- стійкий	Відказо- стійкий	Відказо- стійкий	Відказо- стійкий
----	------------	---------------------	---------------------	---------------------	---------------------

Отже, до сильних сторін продукту можна віднести меншу ніж у більшості конкурентів ціну, сумісність із сучасною операційною системою, нейтральною є надійність, оскільки вона не відрізняється від конкурентів.

6.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 6.3):

Таблиця 6.3 – Технологічна здійсненність ідеї проекту

<i>№ n/n</i>	<i>Ідея проекту</i>	<i>Технології її реалізації</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
1	Апаратна частина	Вибір пристрою спроможного приймати сигнал і з COM пристрою, і з USB або Ethernet	Raspberry Pi (2 та вище), Orange Pi, Arduino	Доступна
2	Реалізація алгоритму для конвертації сигналів	Технологія для передачі інформації між COM та	Java	Доступна

		USB/Ethernet портами		
3	Налаштування програмно-апаратного комплексу через веб-сторінку	Забезпечення можливості користувачу вносити зміни в налаштування системи	Java	Доступна
<p>Обрана технологія реалізації ідеї проекту: Проект можливо реалізувати. Апаратна частина буде виконана на базі мікрокомп'ютера Raspberry Pi, алгоритми конвертації сигналу та можливість налаштування системи на базі Java.</p>				

Отже, за результатами аналізу таблиць можна зробити висновок, що технічна реалізація проекту можлива. В кінці таблиці було докладно розібрано технології, що будуть використані у реалізації проекту. Обрані технології є легкодоступними, Raspberry Pi поширена не дуже широко, але канал її поставок від виробника надійний.

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Для визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, побудуємо необхідні таблиці, які дозволять спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 6.4).

Таблиця 6.4 – Попередня характеристика потенційного ринку стартап-проекту

<i>№ n/n</i>	<i>Показники стану ринку (найменування)</i>	<i>Характеристика</i>
1	Кількість головних гравців, од	6-7
2	Загальний обсяг продаж, грн/ум.од	До \$30 за один адаптер або до \$30 за ліцензію на рік
3	Динаміка ринку (якісна оцінка)	Стагнація
4	Наявність обмежень для входу (вказати характер обмежень)	Важко знайти нових споживачі
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	До 20% за од. товару

Отже, ринок є малопривабливим для входження із власною реалізацією системи завадостійкої передачі даних із використанням віртуального послідовного порту, оскільки ринок насичений конкурентами, а нові споживачі навряд з'являться.

Далі визначимо потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи на таблиці 6.5.

Таблиця 6.5 – Характеристика потенційних клієнтів стартап-проекту

<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія (цільові сегменти ринку)</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
Створення програмно апаратного комплексу для передачі сигналу між пристроями з різними інтерфейсами	Підприємства, СТО, ентузіасти, які намагаються знайти використання для старих пристроїв	Фактори: Наявність необхідних технологій при реалізації платформ	Надійність, безперебійна робота, коректність відображення даних, сервісне обслуговування

Після визначення потенційних груп клієнтів проведемо аналіз ринкового середовища шляхом складання таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають на таблиці 6.6 та 6.6.

Таблиця 6.6 – Фактори загроз

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
1	Знецінення гривні	Попит зменшиться	Надання додаткових товарів або сервісів разом із продуктом
2	Ненадійна влада та непрацюючі загрози	Немає впевненості у правовій підтримці	Продовження існування
3	Частково введений військовий стан. Можливість посилення конфлікту	Частково введений військовий стан. Можливість посилення конфлікту	Припинення виробництва при розвитку конфлікту
4	Збільшення вартості розробки	Підвищення заробітної плати працівників	Зміна ціни продукції

Таблиця 6.7 – Фактори можливостей

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст можливості</i>	<i>Можлива реакція компанії</i>
1	Підвищення необхідності використовувати обладнання із СОМ портами	Збільшення попиту	Збільшення обсягів виробництва
2	Зменшення собівартості	Зменшення вартості	Покриття більшої частини ринку

Далі проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку табл. 6.8.

Таблиця 6.8 – Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
1. Тип конкуренції: чиста	Значна кількість виробників, переважно нецінова конкуренція	Реклама, гарантія якості продукції, підтримка покупця, додаткові сервіси для покупця
2. За рівнем конкурентної боротьби: світовий	Конкурентна боротьба в межах планети	Вихід на світовий ринок
3. За галузевою ознакою: внутрішньогалузева	Боротьба між товариствами однієї галузі	Реалізація інструментів, що покращать сприйняття споживачем та його комфорт використання продукту
4. Конкуренція за видами товарів: - товарно-видова	Види однієї категорії товарів, що можуть задовільнити конкретну потребу споживача	Створення додаткових функцій для конкретних потреб

Продовження таблиці 6.8

5. За характером конкурентних переваг - цінова	Боротьба за ринок за рахунок ціни-якості	Відповідна цінова політика
6. За інтенсивністю - не марочна	Відсутня прив'язаність до певної марки, можливість співпраці з будь-ким	Співпраця з різними марками

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі на таблиці 6.9.

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

	<i>Прямі конкуренти в галузі</i>	<i>Потенційні конкуренти</i>	<i>Постачальники</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
<i>Складові аналізу</i>	TCP COM BRIDGE, USB to Serial Adapter, Адаптер COM RS232	Відсутні	Відсутні	Контроль якості; Чутливість до зміни цін	Ціна; Змінні витрати; Доступність на ринку
Висновки:	Інтенсивність конкурентності низька	можливість входу на ринок є	Постачальники відсутні	Вимоги, в до якості та ціни	Обмеження у ціновій політиці.

Робота на ринку можлива, хоча існує достатньо конкурентів, орієнтація на сучасні операційні системи певно дасть можливість захопити частину ринку. Динаміка ринку знаходиться в стані стагнації, проте є можливість, що новий засіб змінить ситуацію. Перевагами системи є можливість працювати із сучасними операційними системами, простота у використанні та можливість роботи без додаткових інструментів.

На основі аналізу конкуренції, проведеного в таблиці 6.9, а також із урахуванням характеристик ідеї проекту (табл. 6.2), вимог споживачів до товару (таблиці 6.6) та факторів маркетингового середовища (таблиці 6.7-6.8) визначимо та обґрунтуємо перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 6.10

Таблиця 6.10 – Обґрунтування факторів конкурентоспроможності

<i>№ п/п</i>	<i>Фактор конкурентоспроможності</i>	<i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i>
1	Ціна	Споживач заінтересований в низькій ціні
2	Якість	Передача інформації без втрат
3	Довговічність	Відсутність потреби у заміні компонент системи
4	Легкість використання	Plug-n-Play

За визначеними факторами конкурентоспроможності (табл. 6.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 6.11).

Таблиця 6.11 – Порівняльний аналіз сильних та слабких сторін «назва проекту»

№ п/ п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні із TCP COM BRIDGE						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	18		+					
2	Якість	12		+					
3	Довговічність	14			+				
4	Легкість використання	15	+						

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 6.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 6.11).

Перелік ринкових загроз та ринкових можливостей складемо на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 6.12 – SWOT- аналіз стартап-проекту

Сильні сторони: Сумісність із сучасними системами, доступна ціна.	Слабкі сторони: Система вимагає дбайливого використання
Можливості: Збільшення попиту на роботу із СОМ портами, зменшення собівартості	Загрози: Поява нових конкурентів, зменшення попиту на товар

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. табл. 6.9, аналіз потенційних конкурентів).

Визначені альтернативи проаналізуємо з точки зору строків та ймовірності отримання ресурсів (табл. 6.13).

Таблиця 6.13 – Альтернативи ринкового впровадження стартап-проекту

<i>№ n/n</i>	<i>Альтернатива (орієнтовний комплекс заходів) ринкової поведінки</i>	<i>Ймовірність отримання ресурсів</i>	<i>Строки реалізації</i>
1	Зменшення собівартості	Низька	До вісімнадцяти місяців
2	Розроблення додаткових функцій	Середня	До чотирьох місяців
3	Підвищення продуктивності	Висока	До шести місяців
4	Запуск реклами	Середня	До одного місяця

6.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 6.14).

Таблиця 6.14 – Вибір цільових груп потенційних споживачів

<i>№ п/п</i>	<i>Опис профілю цільової групи потенційних клієнтів</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит в межах цільової групи (сегменту)</i>	<i>Інтенсивність конкуренції в сегменті</i>	<i>Простота входу у сегмент</i>
1	Підприємства	так	5%	Висока	Складна
2	СТО	так	10%	Висока	Складна
3	Ентузіасти	так	60%	Середня	Легка
У якості цільових груп обрано: підприємства, СТО, ентузіасти					

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку:

якщо компанія зосереджується на одному сегменті – вона обирає стратегію концентрованого маркетингу;

якщо працює із кількома сегментами, розробляючи для них окремо програми ринкового впливу – вона використовує стратегію диференційованого маркетингу;

якщо компанія працює із всім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги) – вона використовує масовий маркетинг.

Для роботи в обраних сегментах ринку необхідно сформувані базову стратегію розвитку (табл. 6.15).

У якості альтернативи розвитку обрано стратегію виклику лідера, стратегією охоплення ринку буде стратегія концентрованого маркетингу.

Таблиця 6.15 – Визначення базової стратегії розвитку

<i>Обрана альтернатива розвитку проекту</i>	<i>Стратегія охоплення ринку</i>	<i>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</i>	<i>Базова стратегія розвитку</i>
Стратегія виклику лідера	Стратегія концентрованого маркетингу	Індивідуальний підхід до клієнта; концентрація на необхідному функціоналі	Стратегія концентрованого маркетингу передбачає надання товару сильних, яскраво виділених властивостей

Наступним кроком виберемо стратегію конкурентної поведінки (табл. 6.16).

Таблиця 6.16 – Визначення базової стратегії конкурентної поведінки

<i>Чи є проект «першопрохідцем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>	<i>Стратегія конкурентної поведінки</i>
Ні	Так	Буде концентрація на меншому, але більш якісному наборі функцій	Стратегія виклику лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. табл. 6.6), а також в залежності від обраної базової стратегії розвитку (табл. 6.15) та стратегії конкурентної поведінки (табл. 6.16) розробляється стратегія позиціонування (табл. 6.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 6.17 – Визначення стратегії позиціонування

<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції власного стартап-проекту</i>	<i>Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)</i>
Низька ціна, надійність, функціональність	Стратегія концентрованого маркетингу-гу	Доступна ціна, підтримка сучасних операційних систем	Просто, якісно, доступно

Отже, у якості цільових груп було обрано підприємства, СТО та ентузіастів. У якості стратегії для охоплення ринку було обрано стратегію концентрованого маркетингу. Стратегія конкурентної поведінки була вибрана стратегія виклику лідера.

6.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 6.18 потрібно підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 6.18 – Визначення ключових переваг концепції потенційного товару

<i>№ n/n</i>	<i>Потреба</i>	<i>Вигода, яку пропонує товар</i>	<i>Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)</i>
1	Апаратна частина	Наявність додаткових функцій для віддаленого управління	Реалізовано сайт для віддаленого управління
2	Алгоритми конвертації сигналу	Просте додавання нових елементів системи	Забезпечена можливість додавання функцій, спеціально для кожного з клієнтів
3	Налаштування системи через вебсторінку	Надає можливість переглядати відео з камери	Відео можна переглядати на сайті для віддаленого управління
4	Низька ціна	Конкурентна ціна	Ціна нижча ніж у більшості конкурентів

Далі розробимо трирівневу маркетингову модель товару: уточнимо ідею продукту та/або послуги, її фізичні складові, особливості процесу її надання (табл. 6.19). В таблиці М/Нм – монотонні/немонотонні; Вр/Тх/Тл/Е/Ор – вартісні/ технічні/ технологічні/ ергономічні/ органолептичні;

Після формування маркетингової моделі товару було особливо відмічено – чим саме проект буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару.

Опис трьох рівнів моделі товару – один з найважливіших елементів розробки стартап проекту, оскільки описується загальна концепція товару.

Таблиця 6.19 – Опис трьох рівнів моделі товару

<i>Рівні товару</i>	<i>Сутність та складові</i>		
I.Товар за задумом	Система завадостійкої передачі даних із використанням віртуального послідовного порту		
	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1.Підтримка сучасними ОС	М	Тх/Тл/Е
	2.Простота використання	Нм	Тх/Тл
	3.Налаштування системи	М	Тх/Тл
	4. Якість переданого сигналу	М	Тх/Тл
	5. Термін дії	М	Вр/Тл
	Якість: відповідає нормам розробки програмного забезпечення.		
	Пакування: Інструкція до використання		
	Марка: ПП «AlexDevices»		
За рахунок чого потенційний товар буде захищено від копіювання: буде запатентовано ноу-хау			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 6.20). Аналіз проводиться експертним методом.

Таблиця 6.20 – Визначення меж встановлення ціни

<i>Рівень цін на товари-замінники</i>	<i>Рівень цін на товари-аналоги</i>	<i>Рівень доходів цільової групи споживачів</i>	<i>Верхня та нижня межі встановлення ціни на товар/послугу</i>
Відсутні	До \$30	\$200	\$15 – 27

Далі визначимо оптимальну систему збуту, в межах якого приймемо рішення (табл. 6.21):

проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);

вибір та remote control обґрунтування оптимальної глибини каналу збуту;

вибір та обґрунтування виду посередників.

Таблиця 6.21 – Формування системи збуту

<i>Специфіка закупівельної поведінки цільових клієнтів</i>	<i>Функції збуту, які має виконувати постачальник товару</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
Купування у роздріб	Доставка товару споживачеві, інструктаж	Виробник - споживач	Інтернет-магазин

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 6.22).

Концепція маркетингових комунікацій яскраво відображає специфіку поведінки цільових клієнтів, канали комунікації, якими вони користуються, завдання для рекламного повідомлення та концепцію рекламного звернення.

Таблиця 6.22 – Концепція маркетингових комунікацій

<i>Специфіка поведінки цільових клієнтів</i>	<i>Канали комунікацій, якими користуються цільові клієнти</i>	<i>Ключові позиції, обрані для позиціонування</i>	<i>Завдання рекламного повідомлення</i>	<i>Концепція рекламного звернення</i>
Ретельний вибір товару, здатного виконувати свої функції	Ділові зустрічі, телефони, електронна пошта, спеціальні форуми	Довгострокова підтримка споживачів	Переконання потенційних споживачів у необхідності придбання саме нашого товару	СОМ порт – дешево та сердито

Можна зробити висновок що хоча розроблена система має достатньо аналогічних функцій, проте яскраво виділені її особливості, що робить її бажанішою за конкурентів для вузького кола завдань. Технологічний аудит показав, що усі необхідні для виробництва технології наявні і легко доступні. Ринкові можливості запуску стартап-проекту показали доцільність виводу продукції по даній ідеї на ринок. Крім того, були розглянуті можливі ризики виходу на ринок та аналіз конкуренції. Була розроблена Попередня ринкова стратегія проекту, після чого, була

розроблена маркетингова програма. Тому можна зробити висновок, що система завадостійкої передачі даних із використанням віртуального послідовного порту є достатньо вигідною для впровадження у виробництво.

ВИСНОВКИ

В процесі виконання дисертації було проведено аналіз існуючих рішень для віртуалізації ПШ. На базі цього аналізу сформовано технологічні складові СЗПД та створено прототип, тобто мета досягнута.

При виконанні роботи було виконано наступні завдання:

- проведено аналіз існуючих рішень та програмних комплексів СЗПД на базі ВПП;
- розглянуто пристрої, що використовують подібну технологію;
- сформовано технологічні складові технології СЗПД;
- детально описано віртуальні реєстри ВПП, стек протоколу ВПП;
- Описано роботу диспетчера сеансів та процес віртуалізації;
- розглянуто теоретичні засади ЗСПД;
- наведені можливі втілення розглянутої технології;
- прототип успішно впроваджено в роботу підприємства.

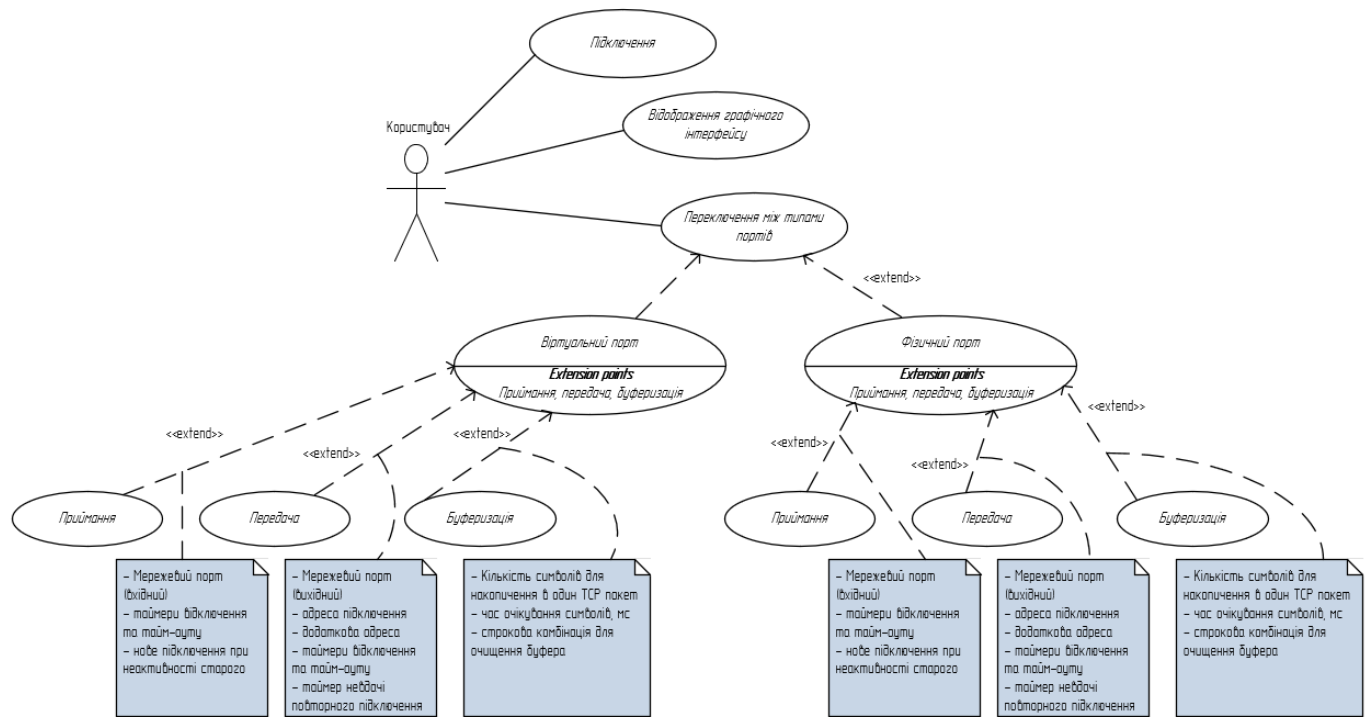
ПЕРЕЛІК ПОСИЛАНЬ

1. Електронний підручник з інформатики [Електронний ресурс] – режим доступу до ресурсу: <http://informatics.dp.ua/budova-personalnoho-kompyutera/>
2. Последовательный порт [Електронний ресурс] – режим доступу до ресурсу:http://www.wikiwand.com/ru/%D0%9F%D0%BE%D1%81%D0%B%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BE%D1%80%D1%82
3. Послідовний порт [Електронний ресурс] – режим доступу до ресурсу:http://znaimo.com.ua/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%BE%D1%80%D1%82
4. TCP COM Bridge [Електронний ресурс] – режим доступу до ресурсу: <https://www.aggsoft.ru/tcp-com/buy.htm>
5. USB to Serial Adapter [Електронний ресурс] – режим доступу до ресурсу: <http://www.usconverters.com/usb-serial-adapter-xs882?zenid=88dac63776afa4d676cf357591bb386b>
6. Адаптер COM RS232 [Електронний ресурс] – режим доступу до ресурсу: <https://www.olx.ua/obyavlenie/adapter-com-rs232-universalnyu-adapter-s-usb-na-com-IDAOh3U.html?sd=1#35e1363a4f>
7. Przemysłowe switche kolejowe z funkcja PoE [Електронний ресурс] – режим доступу до ресурсу: <https://www.dacpol.eu/ua/promislovi-en50155-komutatori-m12-garretcom-7099/product/przemyslowe-switche-kolejowe-zgodne-z-norma-en50155-12kx-10kt-6k32fc-6km-pes42>
8. Модуль RS232 ADAM-4570 [Електронний ресурс] – режим доступу до ресурсу: <https://www.proxis.ua/uk/product/ADAM-4570/>

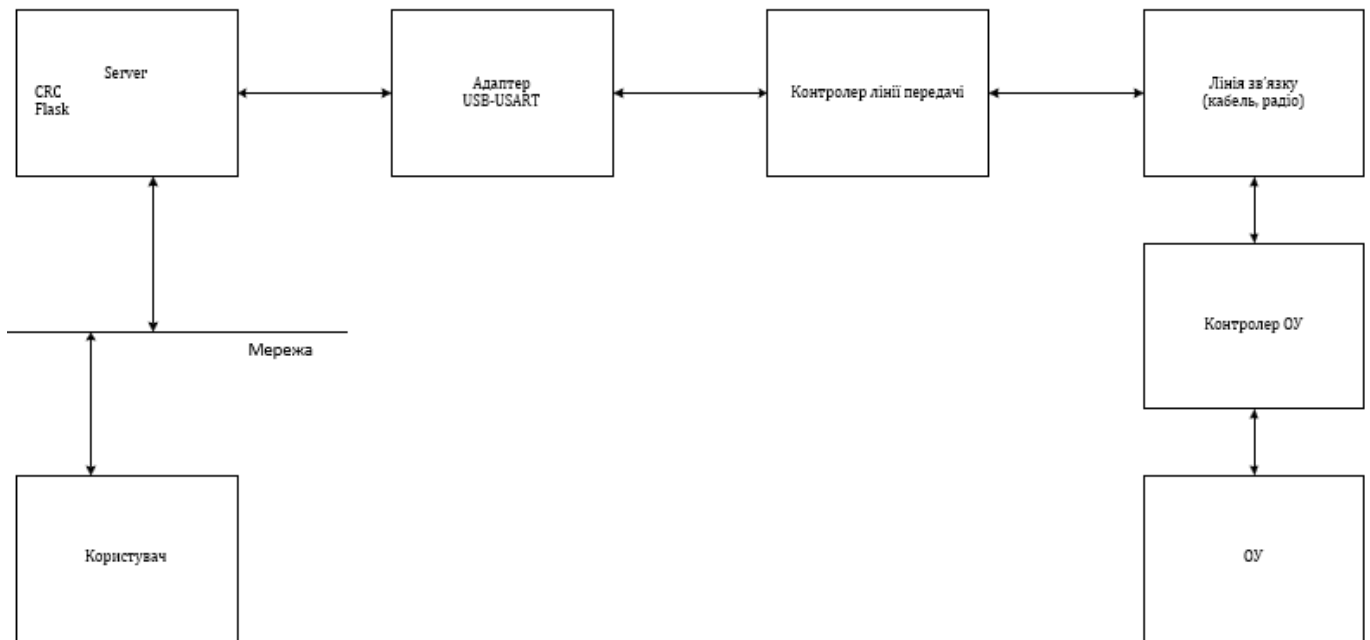
9. Serial UART Information [Электронный ресурс] – режим доступа до ресурсу: <https://www.lammertbies.nl/comm/info/serial-uart.html>
10. Jan Axelson "Serial Port Complete" - Lakeview Research LLC, Madison, 2007 - 379 pages
11. UART и с чем его едят [Электронный ресурс] – режим доступа до ресурсу: <https://habr.com/post/109395/>
12. Последовательный порт компьютера [Электронный ресурс] – режим доступа до ресурсу: <http://biosgid.ru/osnovy-ustrojstva-pk/com-port-ili-posledovatelnyj-port.html>
13. Магда Ю.С. «Программирование последовательных интерфейсов» – СПб.:БХВ-Петербург, 2009 – 304с.
14. Службы и сетевые порты в серверных системах Microsoft Windows [Электронный ресурс] – режим доступа до ресурсу: <http://www.rublin.org/article/sluzhby-i-setevye-porty-v-servernykh-sistemakh-microsoft-windows>
15. <http://www.realcoding.net/articles/rabota-s-posledovatelnyimi-portami-na-ibm-pc.html>
16. Жураковський Ю.П., Полторак В.П. Теорія інформації на кобуння: Підручник. - К.:Вища шк., 2001. - 255с
17. Работа с последовательными портами на IBM PC [Электронный ресурс] – режим доступа до ресурсу: https://uk.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BA%D0%BB%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D0%BD%D0%B0%D0%B4%D0%BB%D0%B8%D1%88%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BA%D0%BE%D0%B4
18. Flask/Django: Python RESTful API [Электронный ресурс] – режим доступа до ресурсу: <https://webmastah.pl/flaskdjango-python-restful-api/>

19. Serial-порт Mikrotik [Электронный ресурс] – режим доступа до ресурсу: <http://mikrotik-ukraine.blogspot.com/2015/12/serial-console-mikrotik.html>
20. Manual:System/Serial Console [Электронный ресурс] – режим доступа до ресурсу: https://wiki.mikrotik.com/wiki/Manual:System/Serial_Console

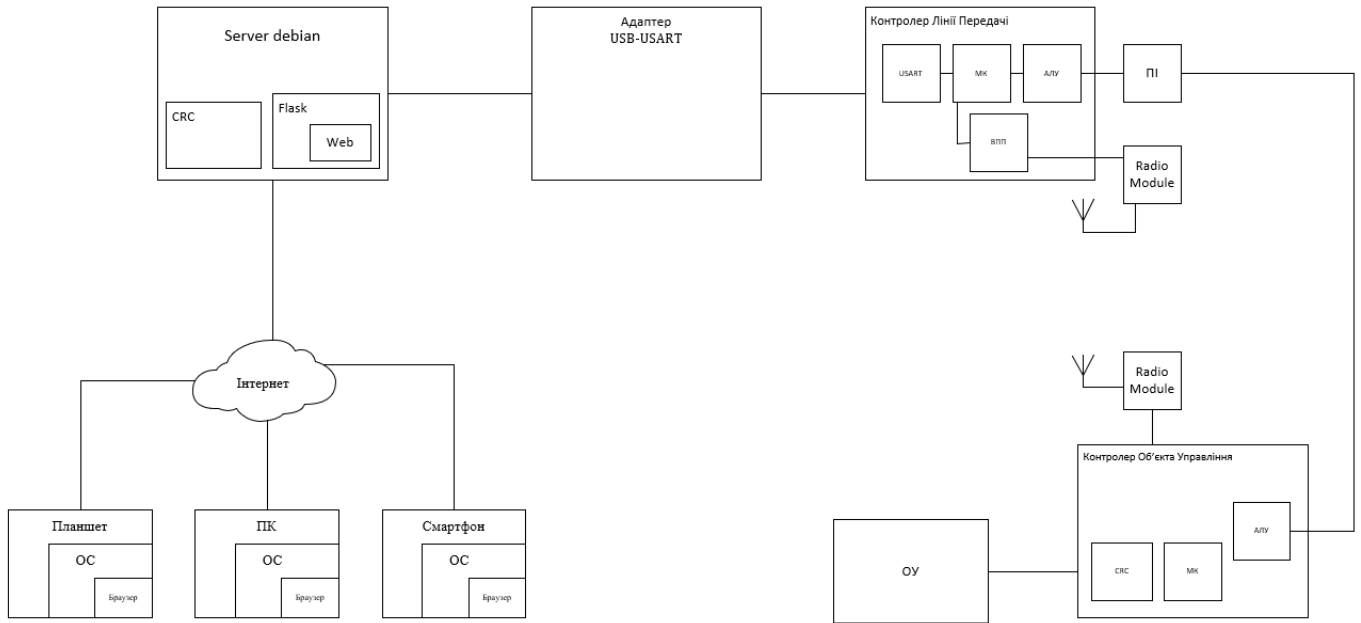
Додаток А Діаграма прецедентів користувача



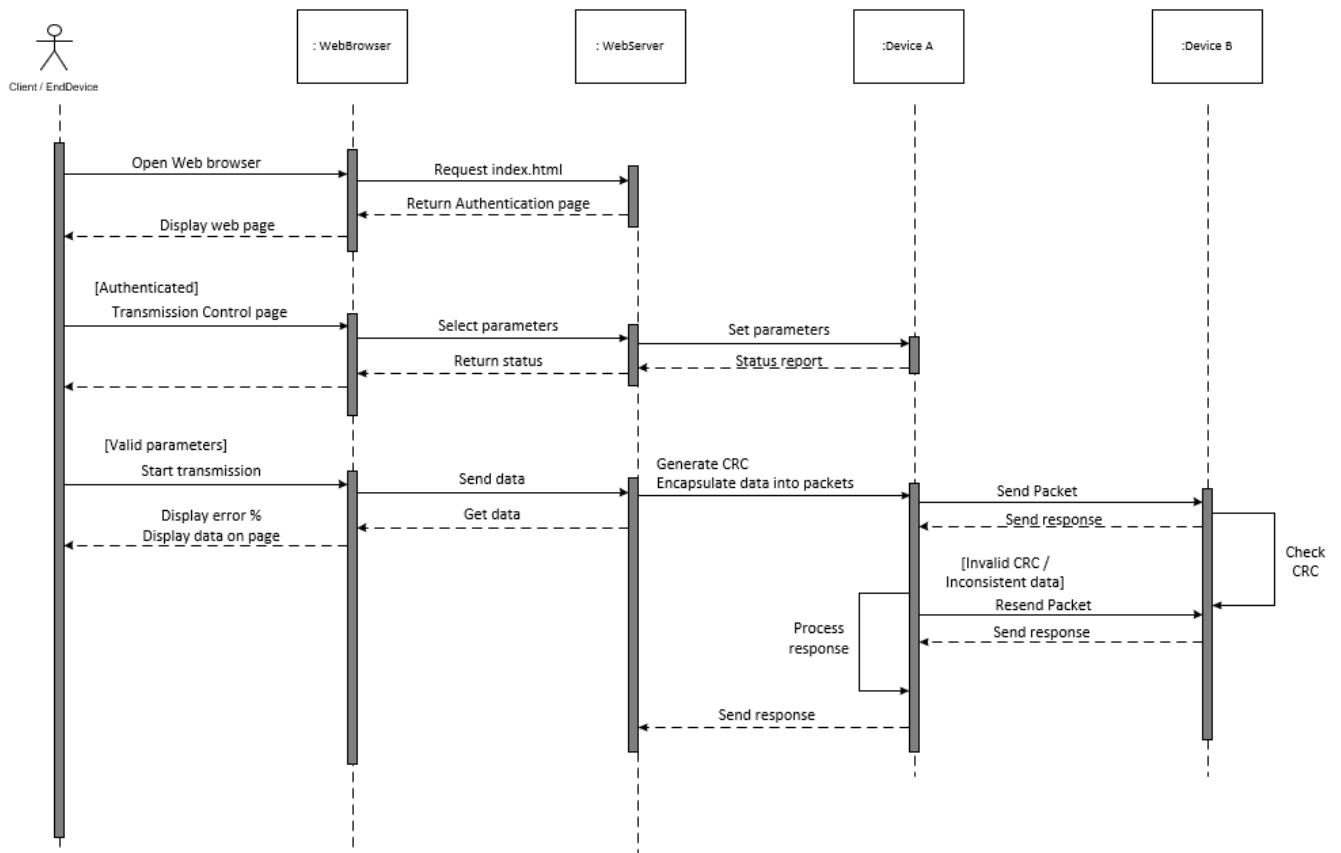
Додаток В Структурна схема



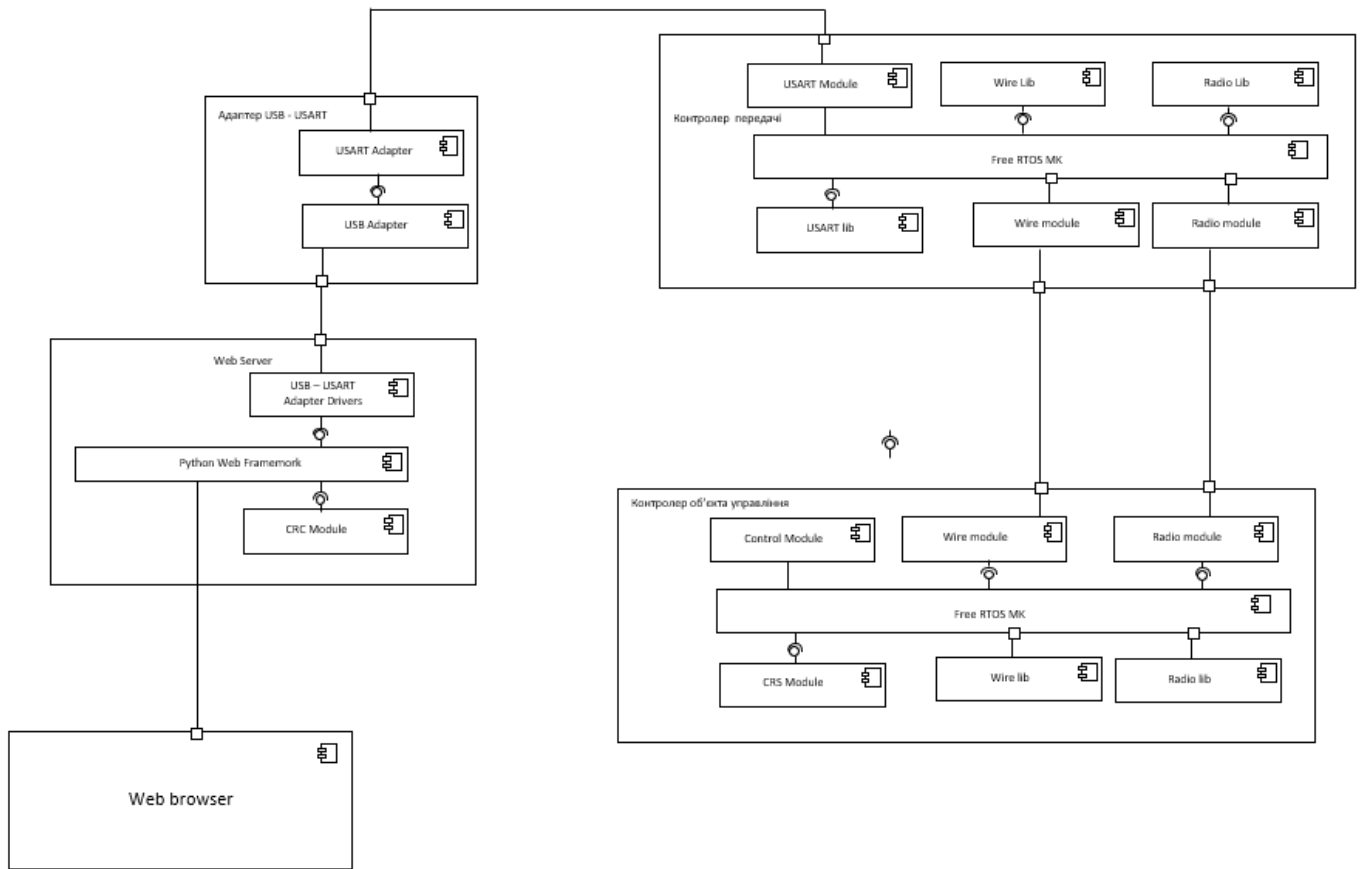
Додаток Г Функціональна схема



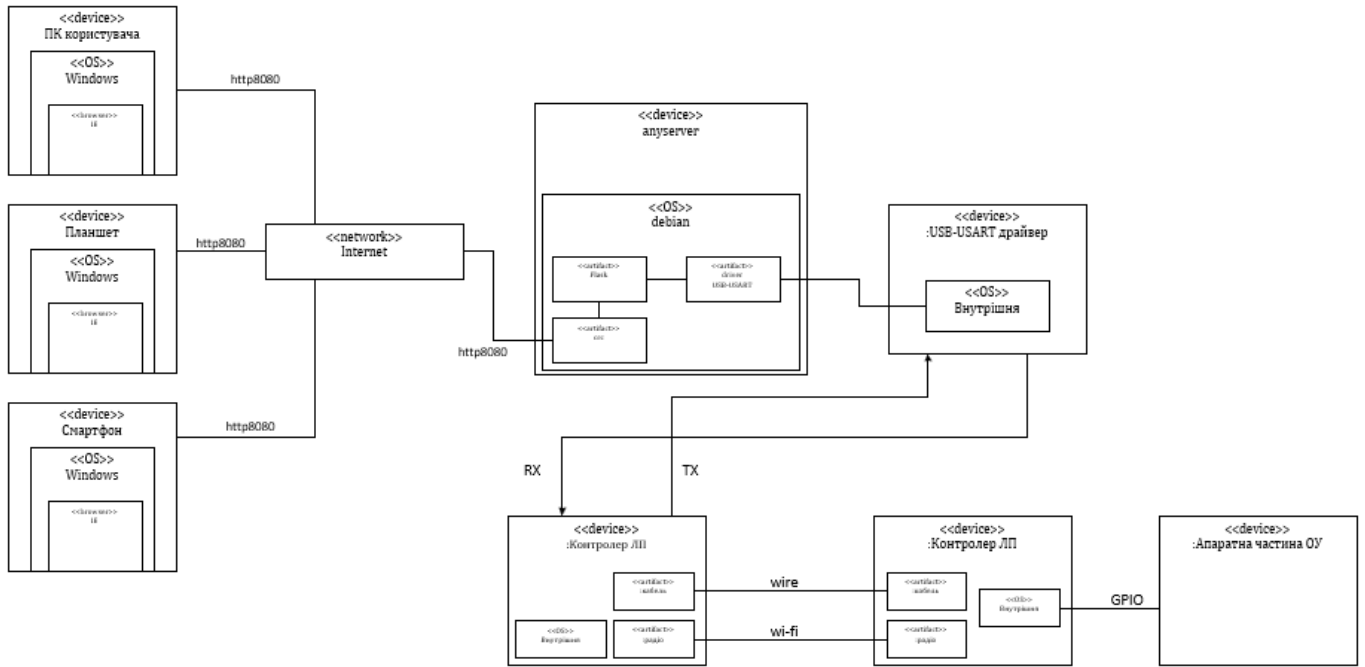
Додаток Д Діаграма послідовностей



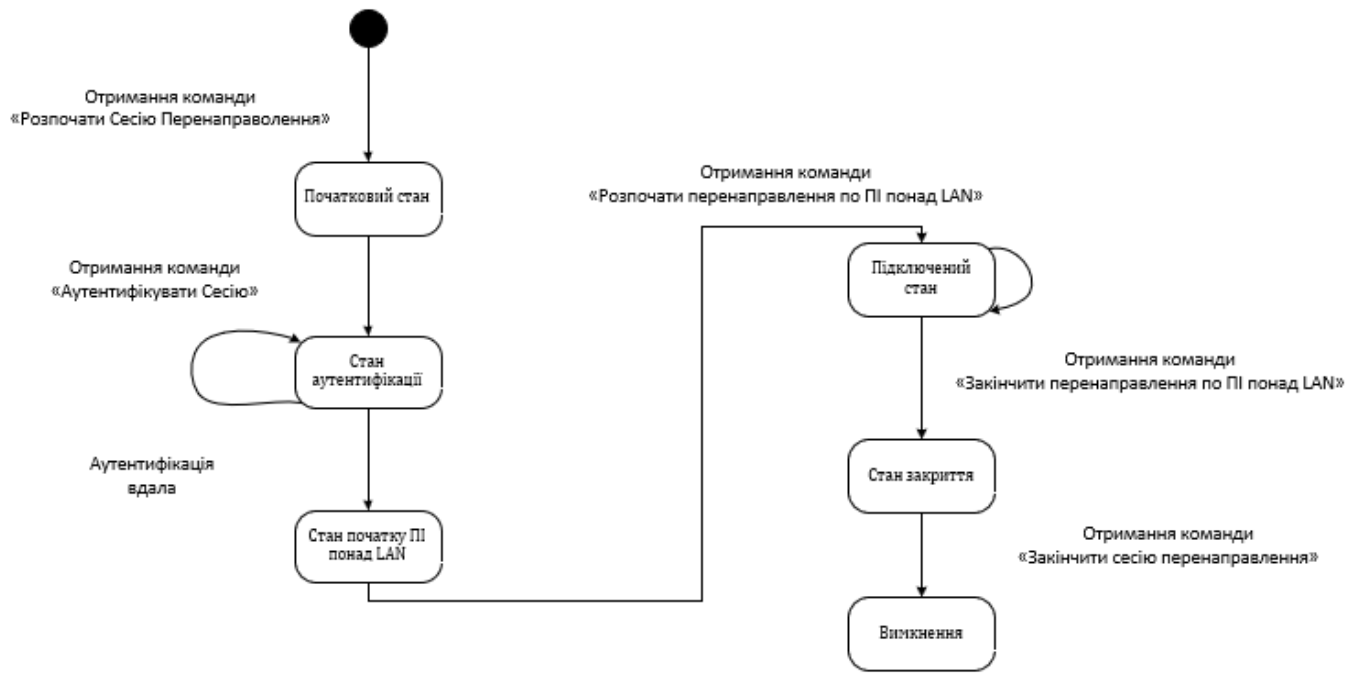
Додаток Е Діаграма компонентів



Додаток Ж Діаграма розгортання



Додаток 3 Діаграма робочого процесу пристрою



Додаток И Акт впровадження

infopulse

ТОВ «Інфопульс Україна»

Адреса: 03056, м. Київ, вул. Польова, 24

Тел./факс: (044) 457-88-56

26.11.2018

АКТ

ВПРОВАДЖЕННЯ РОЗРОБОК

ДИПЛОМНОЇ РОБОТИ В ДІЯЛЬНІСТЬ ПІДПРИЄМСТВА

Розроблена в дипломній роботі Поліщука Олексія Андрійовича «Система завадостійкої передачі даних із використанням віртуального послідовного порту» використовується за призначенням в компанії Інфопульс Україна.

Генеральний директор
ТОВ «Інфопульс Україна»



Сігов О.Б.