


Міністерство освіти і науки України  
Державний заклад  
«Луганський національний університет імені Тараса Шевченка»


Навчально-науковий інститут математики та інформаційних технологій  
Кафедра інформаційних технологій та систем

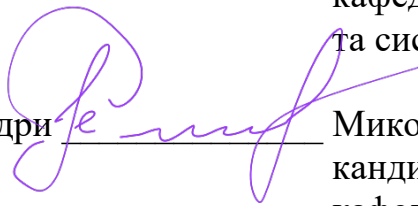
**Чамара Дмитро Іванович**

**Організація роботи навчальної лабораторії  
мікроконтролерів та IoT**

**кваліфікаційна робота  
здобувача вищої освіти другого (магістерського) рівня  
освітньої програми «Комп'ютерні мережі»  
за спеціальністю 123 Комп'ютерна інженерія**

Особистий підпис  Дмитро ЧАМАРА

Науковий керівник  Геннадій МОГИЛЬНИЙ,  
кандидат технічних наук, доцент  
кафедри інформаційних технологій  
та систем

Завідувач кафедри  Микола СЕМЕНОВ,  
кандидат педагогічних наук, доцент  
кафедри інформаційних технологій  
та систем

## **АНОТАЦІЯ**

**Чамара Дмитро Іванович**

**Тема: Організація роботи навчальної лабораторії мікроконтролерів та IoT.**

**Спеціальність:** 123 «Комп'ютерна інженерія»

**Установа:** ЛНУ імені Тараса Шевченка, 2026р.

**Об'єктом дослідження** є процес професійної підготовки фахівців з комп'ютерної інженерії в закладах вищої освіти при вивченні мікропроцесорних систем та технологій Інтернету речей.

**Предметом дослідження** є методи, засоби та технології організації навчальної лабораторії мікроконтролерів та IoT, включаючи апаратне забезпечення стендів та програмні механізми віддаленого доступу до них.

**Метою роботи** є розробка та впровадження комплексної системи організації роботи навчальної лабораторії мікроконтролерів та Інтернету речей, що забезпечує можливість проведення лабораторних занять як в очному, так і в дистанційному режимах з використанням реального обладнання.

### **Результати роботи.**

Розглято етапи еволюції IoT, стан лабораторної бази ЗВО України: проведено детальний огляд технічного оснащення провідних університетів, специфіку ОС для Raspberry Pi, крім того пропонується та технічно реалізується одини варіант моделі «Лабораторія як сервіс» (LaaS) на засадах Gnome-remote, яка дозволяє студентам отримувати віддалений доступ до реального обладнання (Raspberry Pi, датчики, камери) з будь-якої точки світу

**Ключові слова.** мікрокомп'ютер, операційна система, домен Microsoft, RASPBERRY PI, RDP, VPN, IOT (INTERNET OF THINGS), навчальний процес, навчальна лабораторія, віддалений режим.

## ABSTRACT

**Chamara Dmytro**

**Theme: Organization of the work of the microcontroller and IoT training laboratory.**

**Speciality:** 123 "Computer Engineering""

**Institution:** Luhansk Taras Shevchenko National University (LTSNU), 2026.

**The article of research** of the study is the process of professional training of computer engineering specialists in higher education institutions in the study of microprocessor systems and Internet of Things technologies.

**The subject of the research** is methods, tools and technologies for organizing a training laboratory of microcontrollers and IoT, including the hardware of the stands and software mechanisms for remote access to them.

**The purpose of the work** is to develop and implement a comprehensive system for organizing the work of a training laboratory of microcontrollers and the Internet of Things, which provides the possibility of conducting laboratory classes both in person and remotely using real equipment.

Results of the work.

The stages of IoT evolution, the state of the laboratory base of Ukrainian higher education institutions are considered: a detailed review of the technical equipment of leading universities, the specifics of the OS for Raspberry Pi, in addition, one version of the "Laboratory as a Service" (LaaS) model based on Gnome-remote is proposed and technically implemented, which allows students to receive remote access to real equipment (Raspberry Pi, sensors, cameras) from anywhere in the world.

**Keywords:** microcomputer, operating system, Microsoft domain, RASPBERRY PI, RDP, VPN, IOT (INTERNET OF THINGS), educational process, educational laboratory, remote mode.

Міністерство освіти і науки України  
Державний заклад «Луганський національний університет  
імені Тараса Шевченка»  
Факультет (інститут) Інститут математики та інформаційних технологій  
(повна назва)  
Кафедра Інформаційних технологій та систем  
(повна назва)  
Галузь знань 12, Інформаційні технології  
(код, назва)  
Напрямок підготовки (спеціальність) 123 «Комп'ютерна інженерія»  
(код, назва)

**ЗАВДАННЯ**  
**на кваліфікаційну роботу освітньо-кваліфікаційного рівня**  
**« магістр »**  
(назва рівня)

Студенту Чамара Дмитру Івановичу  
(прізвище, ім'я, по батькові)

Керівник кваліфікаційної роботи Могильний Геннадій Анатолійович  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

1. Тема роботи Організація роботи навчальної лабораторії мікроконтролерів та IoT

затверджена наказом по університету

2. Термін подання студентом закінченої роботи на кафедру

3. Вихідні дані до роботи У результаті виконання роботи необхідно дослідити історію розвитку IoT стан лабораторної бази в університетах; можливості підтримки змішаної форми навчання в період військового стану та запропонувати рішення; що дозволяє підтримати лабораторію IoT RaspberryPI у віддаленому режимі  
(визначаються кількісні або (та) якісні показники, яким повинен відповідати об'єкт розробки)

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)  
Вибір архітектури відааленого режиму , вибір технології приєднання аутентифікації, сервера приєднання, розробка параметрів приєднання налаштування параметрів віддаленого режиму

(визначаються назви розділів або (та) перелік питань, які повинні увійти до тексту ПЗ)

## 5. Індивідуальний план виконання кваліфікаційної роботи

№	Заходи	Термін виконання
1.	Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника.	До 30 жовтня 2023
2.	Аналіз літературних джерел за темою роботи. Розробка ТЗ. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи (пояснювальної записки) та плану експериментальних досліджень.	Другий тиждень жовтня 2024
3.	Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання керівником. Розробка методики тестування	До 1 грудня 2024
4.	Усунення зауважень, урахування рекомендацій керівника. Аналіз структури програмного забезпечення.	Перший тиждень грудня 2024
5.	Поетапний аналіз та обговорення результатів. Перевірка стану виконання роботи.	Перший тиждень грудня 2024
6.	Урахування рекомендацій керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Оформлення документації до проекту.	До 15 грудня 2024
7.	Попередній захист роботи на кафедрі.	За місяць до державної атестації
8.	Доопрацювання роботи з урахуванням рекомендацій після передзахисту. Розробка презентації. Підготовка графічних матеріалів. Перевірка на плагіат. Подання роботи науковому керівникові та рецензентові на підготовку відгуку та рецензії	За 10 днів до державної атестації
9.	Подання на кафедру остаточного варіанта роботи, з відгуком керівника і рецензена.	За 3 дні до державної атестації

## ЗМІСТ

Вступ.....	8
Розділ 1. Розвиток технології IoT .....	12
1.1 Історичний огляд та етапи розвитку .....	12
1.2 Огляд четвертого етапу еволюції Інтернету речей: Фаза платформ та екосистем (2010–2019 роки) .....	17
1.3 Огляд сучасного етапу розвитку IoT: Ера AIoT та Edge Computing .....	21
Розділ 2. Організація освітнього процесу у ЗВО України .....	26
2.1 Інфраструктура підготовки бакалаврів з комп'ютерної інженерії в Україні.....	26
2.1.1 Загальний огляд лабораторної та програмної база комп'ютерної інженерії в університетах України.....	29
2.1.2 Комп'ютерна інженерія в ЛНУ імені Тараса Шевченка: освітня стратегія .....	30
2.1.3 Огляд технічного оснащення університетів України.....	32
2.2 Змішана форма підготовки інженерів-комп'ютерників в умовах воєнного стану.....	44
2.2.1 Дії ЗВО для організації навчання в умовах воєнного стану та змішаного режиму навчання.....	47
2.2.2 Стратегії змішаного навчання в комп'ютерній інженерії та лабораторна база спеціальності.....	51
Розділ 3. Віддалений доступ до навчальної лабораторії мікроконтролерів та IoT .....	55
3.1 Огляд операційних систем для Raspberry pi .....	55
Raspbian / Raspberry Pi OS.....	55
Kali Linux .....	56
Windows 10 IoT Core.....	57
Core Ubuntu/ Ubuntu Desktop .....	57
RISC-OC .....	58
SARPi (Slackware ARM для Raspberry Pi) .....	58
Arch Linux ARM.....	59
FreeBSD.....	60

RetroPie.....	61
Home Assistant Operating System (HAOS).....	62
Аналіз рейтингів різноманітних ОС .....	64
3.2 Розгоратання Ubuntu.....	66
3.3 Вступ до віддаленого доступу .....	71
3.4 Інтеграція списку користувачів з інформаційною системою.....	72
3.5 Підключення по протоколу SSH .....	75
3.6 Підключення за допомогою Gnome-remote.....	76
3.7 Підключення зовнішніх користувачів через мережу Інтернет.....	80
3.7.1 Варіант1 – VPN .....	80
3.7.2 Варіант2 RDP SERVER .....	85
3.7.3 Варіант3 – прокидання портів роутера.....	86
3.8 Спільний доступ до зовнішніх приладів, підключення USB камери, та інші обмеження .....	88
Загальні висновки.....	94
Список літературних джерел .....	96
Додатки.....	103
Додаток А. Скрипт для підключення до навчальних стандартів на засадах RaspberryPI .....	103
Додаток Б. Модифікований скрипт для підключення до навчальних стандартів на засадах RaspberryPI .....	106

## ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімкою інтеграцією кіберфізичних систем у всі сфери людської діяльності. Концепція Інтернету речей (Internet of Things, IoT) трансформувалася з перспективного напрямку в буденну реальність, що охоплює промислову автоматизацію (Industry 4.0), розумні міста, агросектор, логістику та побутову електроніку. Ринок вбудованих систем та мікроконтролерів демонструє експоненціальне зростання, що своєю чергою формує гострий попит на кваліфікованих фахівців, здатних не лише програмувати готові рішення, але й проєктувати складні апаратно-програмні комплекси, налаштовувати взаємодію сенсорів з хмарними платформами та забезпечувати надійність мережевої інфраструктури.

У цьому контексті система вищої освіти стикається з серйозним викликом: як забезпечити якісну практичну підготовку інженерів в умовах динамічних змін технологічного стека та, що особливо актуально для сьогодення, в умовах обмеженого доступу до фізичних аудиторій. Традиційна модель організації навчальних лабораторій, що передбачає виключно очну присутність студентів біля апаратних стендів, стає недостатньо гнучкою. Карантинні обмеження останніх років та безпекова ситуація в Україні довели необхідність впровадження гібридних та дистанційних форм навчання, які не поступалися б якістю традиційним методам.

Проте, вивчення мікроконтролерів та IoT має свою специфіку, яку неможливо повністю перекрити віртуальними симуляторами (на кшталт Proteus, Tinkercad чи Cisco Packet Tracer). Симулятори, попри свою зручність, працюють в ідеалізованих умовах і не відтворюють реальних фізичних процесів, затримок у мережі, апаратних збоїв та нюансів роботи з реальними електричними сигналами. Тому виникає нагальна потреба в організації навчальних лабораторій нового типу, які поєднують фізичне обладнання з технологіями віддаленого доступу, дозволяючи студентам писати код,



завантажувати його в реальний мікроконтролер та спостерігати за результатом роботи (через веб-камери або телеметрію) з будь-якої точки світу.

Організація роботи такої лабораторії вимагає комплексного підходу, що включає не лише вибір сучасної елементної бази (мікроконтролерів, датчиків, виконавчих механізмів), але й розробку надійної програмної інфраструктури для керування доступом, забезпечення безпеки мережі та створення відповідного навчально-методичного забезпечення. Вирішення цих питань дозволить підвищити рівень компетентності випускників, наблизити навчальний процес до реальних виробничих умов та забезпечити безперервність освіти незалежно від зовнішніх обставин. Саме цим зумовлюється актуальність обраної теми магістерської кваліфікаційної роботи.

Кваліфікаційна робота виконана відповідно до плану науково-дослідних робіт кафедри інформаційних технологій та систем Луганського національного університету імені Тараса Шевченка в рамках підготовки фахівців за спеціальністю 123 «Комп'ютерна інженерія». Тематика дослідження відповідає пріоритетним напрямкам розвитку науки і техніки в Україні, зокрема у сфері інформаційно-комунікаційних технологій та впровадження новітніх засобів навчання.

**Об'єктом дослідження** є процес професійної підготовки фахівців з комп'ютерної інженерії в закладах вищої освіти при вивченні мікропроцесорних систем та технологій Інтернету речей.

**Предметом дослідження** є методи, засоби та технології організації навчальної лабораторії мікроконтролерів та IoT, включаючи апаратне забезпечення стендів та програмні механізми віддаленого доступу до них.

**Метою роботи** є розробка та впровадження комплексної системи організації роботи навчальної лабораторії мікроконтролерів та Інтернету речей, що забезпечує можливість проведення лабораторних занять як в

очному, так і в дистанційному режимах з використанням реального обладнання.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

1. Провести огляд історії розвитку IoT та сучасного стану розвитку Інтернету речей
2. Проаналізувати існуючі підходи до організації навчального процесу з вивчення мікропроцесорних систем та IoT у ЗВО, виявити переваги та складові частини існуючих рішень, що запроваджені у ЗВО, знайти недоліки та обґрунтувати загальні шляхи створення гібридної лабораторії.
3. Проаналізувати засоби віддаленого підключення до лабораторії RaspberryPI.
4. Спроектувати та реалізувати програмно-технічну систему віддаленого доступу до лабораторного обладнання, забезпечити механізми автентифікації користувачів та керування сесіями (зокрема, з використанням протоколів RDP та скриптової автоматизації).

Робота складається з трьох розділів.

Перший розділ присвячено історії розвитку IoT та розглянуто особливості сучасного стану технології IoT.

У другому розділі розглянуто особливості підготовки фахівців у галузі IoT та мікроконтролерів. Проведено порівняльний аналіз існуючих методів організації лабораторних робіт: від повних симуляторів до фізичних лабораторій. Обґрунтовано необхідність створення системи віддаленого доступу до реального обладнання, яка б поєднувала переваги фізичної взаємодії з гнучкістю змішаного навчання.

Третій розділ описує перелік складових для організації віддаленого доступу до навчальної лабораторії мікроконтролерів та IoT, наведено огляд

операційних систем для Raspberry pi, розгоратання Ubuntu, підключення зовнішніх користувачів через мережу Інтернет та наведено налаштування Спільного доступу до зовнішніх приладів, підключення USB камери, та інші обмеження.

# РОЗДІЛ 1. РОЗВИТОК ТЕХНОЛОГІЇ ІОТ

## 1.1 Історичний огляд та етапи розвитку

Концепція IoT пройшла шлях від поодиноких експериментів до глобальної мережі, що налічує десятки мільярдів пристроїв.

Ключові етапи (див.рис. 1.1):

- До-IoT ера (1980-ті – 1990-ті): Першим "розумним" пристроєм вважається автомат Coca-Cola в Університеті Карнегі-Меллона (1982), який передавав дані про наявність напоїв.
- Народження терміну (1999): Кевін Ештон (Kevin Ashton) вперше вжив термін "Internet of Things" під час презентації для Procter & Gamble, описуючи використання RFID-міток для управління ланцюгами поставок.
- Етап підключення (2000-ні): Масове поширення Wi-Fi та Ethernet. З'являються перші споживчі пристрої (наприклад, розумні лічильники).
- Етап платформ (2010-ті): Поява хмарних рішень (AWS IoT, Google Cloud IoT). IoT стає доступним для розробників завдяки Raspberry Pi та Arduino.
- Ера AIoT (2020 – сьогодні): Поєднання IoT зі штучним інтелектом (AI) та обробкою даних на межі (Edge Computing).



Рис. 1.1 Історичний огляд та етапи розвитку

Попередній аналіз показав, що складові IoT: Архітектура та компоненти. Ефективна IoT-система — це симбіоз «заліза», програмного забезпечення та мережевих протоколів(див.рис. 1.2).

1. Типова архітектура (трирівнева модель)
2. Рівень сприйняття (Perception Layer): Фізичні пристрої, сенсори та актуатори.
3. Мережевий рівень (Network Layer): Передача даних (Wi-Fi, Bluetooth, LoRaWAN, Zigbee, 5G).
4. Прикладний рівень (Application Layer): Хмарні платформи, мобільні додатки та аналітичні панелі (dashboards).

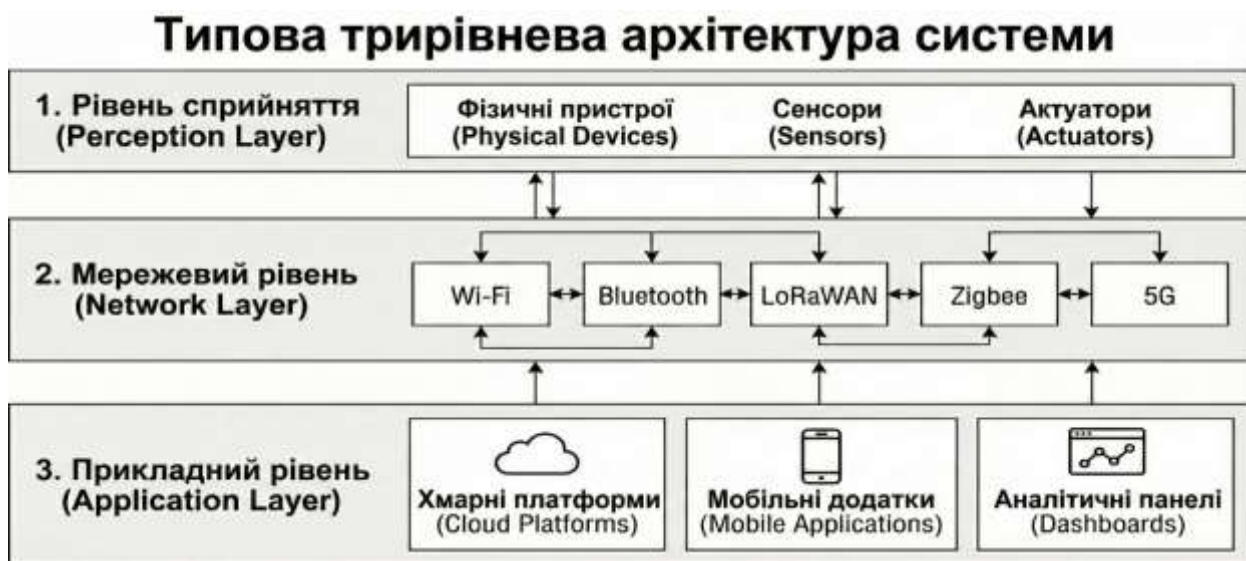


Рис. 1.2 Типова архітектура

Апаратна частина складається (див.рис. 1.3):

- Мікроконтролери (MCU): ESP32, STM32, Arduino (для простих задач).
- Одноплатні комп'ютери (SBC): Raspberry Pi, NVIDIA Jetson (для Edge AI).
- Сенсори: Датчики температури, вологості, освітлення, акселерометри.

- Модулі зв'язку: SIM-модулі, LoRa-трансивери.



Рис. 1.3 Типова апаратна частина

#### Програмна частина

- ОС реального часу (RTOS): FreeRTOS, Zephyr OS, RIOT.
- Протоколи передачі даних: MQTT (найпопулярніший завдяки легкості), CoAP, HTTP, AMQP.
- Прошивка (Firmware): Логіка роботи самого пристрою (написана переважно на C/C++ або MicroPython).

Аналіз сучасного стану та перспективні напрямки розвитку показав, що сьогодні IoT зміщує акцент з "простого збору даних" на "розумне прийняття рішень".

1. IIoT (Industrial IoT): Впровадження концепції "Індустрія 4.0", де заводи стають повністю автономними.
2. Edge Computing: Обробка даних безпосередньо на пристрої для зменшення затримки (latency) та економії трафіку.
3. Безпека: Залишається головним викликом. Більшість кібератак на IoT спрямовані на використання слабких паролів або вразливостей у застарілих прошивках.

4. Енергоефективність: Розвиток технологій LPWAN (NB-IoT), що дозволяють датчикам працювати до 10 років від однієї батарейки.



Рис. 1.4 Напрямки розвитку IoT

Для розробки IoT-рішень необхідний набір спеціалізованих інструментів (див. табл. 1.1):

Таблиця 1.1 Типи інструментів

Тип інструменту	Назва	Призначення
IDE (Розробка)	VS Code + PlatformIO	Універсальне середовище для прошивок.
Моделювання	Proteus / Wokwi	Симуляція схем та коду без реального заліза.
Прототипування	Node-RED	Візуальне програмування логіки та дашбордів.
Тестування MQTT	MQTT Explorer	Відлагодження потоків даних між пристроєм і сервером.
Аналітика	MATLAB/Simulink	Глибокий аналіз даних та побудова алгоритмів керування.

Провідні виробники (Hardware & Software) (див.рис. 1.5):

Апаратне забезпечення (Hardware)

- Espressif Systems: Відомі своїми чипами ESP8266 та ESP32 (стандарт для DIY та комерційних рішень).
- STMicroelectronics: Виробник мікроконтролерів STM32.



- Bosch Sensortec: Світовий лідер у виробництві високоточних MEMS-сенсорів.
- Raspberry Pi Foundation: Еталонні рішення для Edge-обчислень.



Рис. 1.5 Ключові виробники IoT

Попередній аналіз показав, що активно використовуються програмні та хмарні рішення (Software/Cloud – див. рис. 1.6)

1. Microsoft Azure IoT: Потужна екосистема для промислових масштабів.
2. Amazon Web Services (AWS) IoT Core: Гнучка платформа для мільйонів підключень.
3. Cisco: Рішення для мережевої інфраструктури IoT та безпеки.
4. PTC (ThingWorx): Провідна платформа для індустріального IoT (IIoT).



Рис. 1.6 Програмні та хмарні рішення IoT



Таким чином можна з стверджувати, що IoT сьогодні — це вже не просто технологія майбутнього, а критична інфраструктура сучасної економіки.

## **1.2 Огляд четвертого етапу еволюції Інтернету речей: Фаза платформ та екосистем (2010–2019 роки)**

Четверта фаза розвитку Інтернету речей (Internet of Things, IoT), що охоплює період 2010-х років, посідає ключове місце в історії технологій як епоха фундаментальної трансформації технологічної парадигми. Це період, коли IoT еволюціонував від сукупності ізольованих експериментальних рішень до створення глобально інтегрованих інтелектуальних систем. Саме в цей час відбулася дифузія технологій із вузькоспеціалізованих промислових лабораторій у площину масового споживання, що стало каталізатором глобальної цифровізації економіки та побуту.

Концептуальна характеристика етапу відзначилась переходом до хмароцентрированої моделі.

Визначальною ознакою 2010-х років став стратегічний перехід від локально замкнених мереж типу M2M (Machine-to-Machine) до архітектури хмарних обчислень (Cloud Computing). Потреба в обробці колосальних обсягів інформації зумовила появу спеціалізованих IoT-хабів від світових лідерів: AWS IoT, Google Cloud IoT та Microsoft Azure IoT [23], [28]. Ці платформи запропонували модель "IoT як послуга" (IoT-as-a-Service), надавши розробникам готові інструменти для:

- Масштабованої агрегації: можливості одночасного підключення мільйонів вузлів без втрати швидкості обробки.
- Глибокої аналітики: перетворення сирих даних із сенсорів у структуровану інформацію для прийняття бізнес-рішень.
- Економічної ефективності: перехід від капітальних витрат на власні

сервери до операційних витрат за використання хмарних ресурсів.

- В порівнянні з попереднім третім етапом (2000-ні) та для розуміння прогресу четвертого етапу необхідно зіставити його з попереднім етапом "масової зв'язності".

Таблиця 1.2. Порівняльна характеристика етапів

Критерій порівняння	Третій етап (Связність)	Четвертий етап (Платформи)
Основна мета	Підключити пристрій до мережі (IP-адресація).	Побудувати екосистему та проаналізувати дані.
Апаратна база	Промислові контролери, дорогі Wi-Fi модулі.	Raspberry Pi, ESP8266/ESP32, MEMS-сенсори.
Обробка даних	Локальна (на пристрої або локальному сервері).	Хмарна та туманна (Cloud/Edge computing).
Програмування	Низькорівневе (C/C++), закриті протоколи.	Високорівневе (Python, JS), відкриті стандарти API.
Масштабованість	Обмежена потужністю локального обладнання.	Майже безмежна завдяки хмарним ресурсам.

У порівнянні з третім етапом, де панували дорогі та складні в налаштуванні рішення, четвертий етап приніс в апаратній та програмній частинах:

1. Доступність Wi-Fi: Якщо в 2000-х підключення мікроконтролера до мережі потребувало шилдів за \$50-100, то з появою чипів ESP8266 вартість входу знизилася до \$2 [25].
2. Потужність обчислень: Поява одноплатних комп'ютерів (SBC) дозволила виконувати складну обробку зображень безпосередньо на місці збору даних [24].
3. Енергоефективність: Розвиток стандартів Bluetooth Low Energy (BLE) та LPWAN (LoRaWAN, NB-IoT) дозволив датчикам працювати роками від однієї батарейки.

Програмний підхід зазнав радикальних змін:

- Від мікрокоду до скриптів: Завдяки потужності нових чипів стало можливим використовувати Python (MicroPython) та JavaScript (Node.js) для програмування заліза.
- Стандартизація черг повідомлень: На зміну хаотичним протоколам прийшов стабільний MQTT, який став "мовою спілкування" для більшості пристроїв [26].
- No-Code та Low-Code: Поява таких інструментів, як Node-RED, дозволила створювати логіку взаємодії пристроїв графічним методом [29].

Функціонування IoT-систем четвертого етапу базувалося на ієрархічній чотирирівневій моделі [27] (рис1.8):

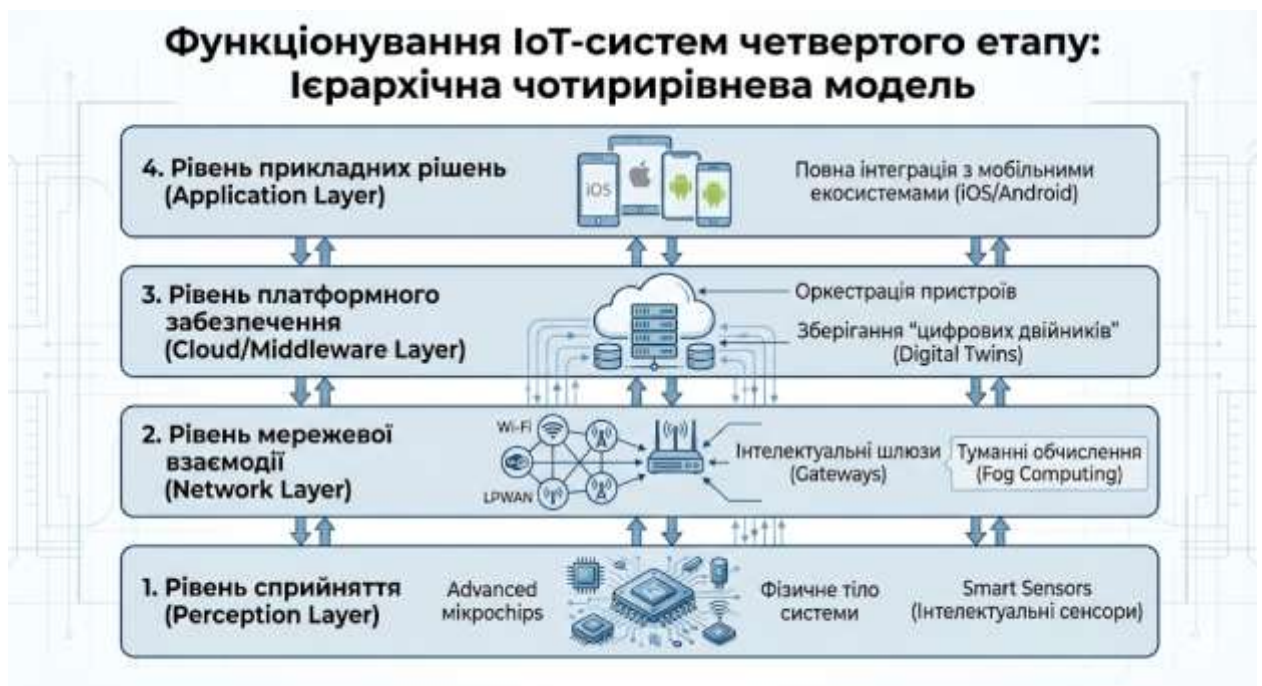


Рис. 1.7 Функціонування IoT-систем четвертого етапу

1. Рівень сприйняття (Perception Layer): Фізичне тіло системи. На відміну від 3-го етапу, сенсори стали інтелектуальними (Smart Sensors).
2. Рівень мережевої взаємодії (Network Layer): Впровадження

інтелектуальних шлюзів (Gateways), які виконують роль "туманних обчислень".

3. Рівень платформного забезпечення (Cloud/Middleware Layer): Тут здійснюється оркестрація пристроїв та зберігання "цифрових двійників".
4. Рівень прикладних рішень (Application Layer): Повна інтеграція з мобільними екосистемами (iOS/Android).

Технологічній основі апаратного забезпечення (Hardware) сприяло:

- Еволюція одноплатних комп'ютерів (SBC): Поява Raspberry Pi у 2012 році стала точкою відліку для недорогих IoT-шлюзів [24].
- Мікроконтролерні екосистеми: Прорив чипів ESP8266 та ESP32 від Espressif Systems [25].
- Прогрес сенсорики (MEMS): Масове виробництво дешевих акселерометрів та гіроскопів.

На цьому етапі була змінена програмна інфраструктура та еволюція протоколів:

- Спеціалізовані ОС: FreeRTOS та Zephyr для критичних завдань.
- MQTT як галузевий стандарт: Легкість та надійність передачі даних за стандартом OASIS [26].
- Концепція API-first: Легка інтеграція з будь-якими веб-сервісами через RESTful API.
- Кібербезпека: Перехід до PKI та TLS-шифрування для захисту каналів зв'язку.
- Цифрові двійники (Digital Twins): Моделювання фізичних об'єктів у цифровому просторі для предиктивного аналізу.

Приклади промислового застосування:

- Розумні міста (Smart City): Адаптивне вуличне освітлення та розумний збір сміття.
- Індустрія 4.0: Повний моніторинг конвеєрних ліній на базі хмарних платформ.

- Персоналізована медицина: Хмарний моніторинг здоров'я через фітнес-трекери.

Четвертий етап розвитку IoT якісно змінив спосіб нашої взаємодії з матеріальним світом. Якщо третій етап лише "підвів кабель" до речей, то четвертий — "навчив їх думати" та працювати в команді через хмарні платформи. Сформована екосистема стала фундаментом для сучасної епохи AIoT (Artificial Intelligence of Things).

### **1.3 Огляд сучасного етапу розвитку IoT: Ера AIoT та Edge Computing**

Сьогодні ми є свідками та учасниками п'ятого етапу еволюції Інтернету речей (IoT), який отримав назву AIoT (Artificial Intelligence of Things) [30]. Якщо попередні десятиліття були присвячені вирішенню фундаментальних проблем фізичного підключення пристроїв та накопиченню величезних обсягів "сирих" даних у централізованих сховищах, то сучасна ера фокусується на створенні інтелектуальної цінності безпосередньо у місці виникнення події [31].

AIoT — це не просто механічна сума IoT та штучного інтелекту; це глибокий технологічний симбіоз. У цій парадигмі мережі "німих" датчиків перетворюються на автономні, самонавчальні системи. Вони здатні самостійно аналізувати складні патерни, фільтрувати інформаційний шум та приймати критично важливі рішення в реальному часі. Цей перехід від пасивного збору даних до активного когнітивного аналізу радикально змінює ландшафт сучасної промисловості, медицини та міської інфраструктури [32].

З 2020 року домінуючим вектором став перехід до моделі Edge-first. Ця трансформація зумовлена експоненціальним зростанням кількості даних, які мережі вже не в змозі передавати в повному обсязі через обмежену пропускну здатність [31].

Ключові технологічні рішення (рис.1.8):

- Автономна інтелектуальність: Сучасні пристрої інтегрують у себе нейронні мережі. Це дозволяє камерам відеоспостереження самостійно ідентифікувати об'єкти, підвищуючи конфіденційність та розвантажуючи канали зв'язку [30].
- Граничні обчислення (Edge Computing): Обробка інформації на самому пристрої мінімізує затримку (latency) до часток мілісекунд, що є критичним для систем аварійного гальмування чи хірургічних роботів [31].
- Конвергенція 5G та IoT: Стандарт 5G забезпечує щільність підключень до 1 мільйона пристроїв на 1 км<sup>2</sup> та стабільність, необхідну для критичної інфраструктури [35].
- Swarm Intelligence (Ройовий інтелект): Взаємодія групи пристроїв (наприклад, складських роботів) без участі центрального контролера робить систему неймовірно стійкою [32].



Рис. 1.8 Ключові технологічні рішення

На сучасному етапі нові технології AIoT та Edge Computing використовують в перше чегу:

1. Автономний транспорт та V2X: Комунікація між авто (V2V) та інфраструктурою (V2I) дозволяє машинам "бачити" за рогом за допомогою даних від розумних світлофорів [33].
2. Smart City 2.0: Адаптивне управління енергомережами (Smart Grids), де AI прогнозує піки споживання на основі погодних умов [33].
3. Цифрові двійники (Digital Twins): Віртуальні копії заводів, що прогнозують знос деталей з точністю понад 99% за допомогою аналізу вібрацій [32].

Однак, і в сучасних умовах, використовують підходи 3-го та 4-го етапів Третій етап (Просте підключення) актуальний для:

- Сільського господарства: Датчики вологості ґрунту в полях, де достатньо передавати дані раз на кілька годин через мережі LoRaWAN. Тут важлива ціна та термін служби батареї до 10 років.
- Житлово-комунального сектору: Розумні лічильники води, де задача полягає лише у фіксації показників раз на місяць.
- Четвертий етап (Хмарні платформи) актуальний для:
- Глобальної логістики: Моніторинг автопарків, де дані збираються в хмару для подальшого аналізу логістом. Затримка у декілька секунд не впливає на результат.
- Аналітики Big Data: Збір статистики з тисяч точок для виявлення споживчих трендів, що вимагає величезних потужностей хмари [31].

**Результати впровадження на п'ятому етапі – це економічні та соціальні здобутки:**

- Predictive Maintenance: Промислові гіганти економлять до 30-40% на обслуговуванні, перейшовши до ремонтів "за потребою" [32].

- Персоналізована медицина (IoMT): Носимі кардіомонітори з AI здатні виявити ознаки серцевого нападу за кілька годин до симптомів [30].
- Екологічна стійкість: AIoT у системах зрошення економить до 50% прісної води за рахунок точного прогнозування [33].

Апаратне забезпечення (Hardware) та програмне забезпечення (Software):

- Нейроприскорювачі (NPU): Чипи на кшталт NVIDIA Jetson Orin, оптимізовані під нейромережі [31].
- Energy Harvesting: Датчики, що живляться за рахунок вібрації чи тепла [35].
- RISC-V: Відкриті архітектури для створення кастомних чіпів AIoT.
- Протокол Matter: Стандарт, що забезпечує локальну сумісність пристроїв від Apple, Google та Amazon [34].
- TinyML: Бібліотеки для запуску машинного навчання на мікроконтролерах [30].

Таблиця 13. Порівняльна характеристика етапів

Характеристика	Третій етап	Четвертий етап	П'ятий етап (AIoT)
Центральна концепція	Зв'язність.	Збір даних.	Автономний інтелект.
Місце рішень	Людина.	Хмарний алгоритм.	AI на пристрої.
Затримка	Висока.	Середня.	Ультранизька (<10 мс).

П'ятий етап розвитку IoT ознаменував фундаментальний перехід до створення когнітивних середовищ, де навколишній простір стає чутливим, адаптивним та інтелектуальним. У цій новій парадигмі AIoT фактично стає центральною нервовою системою глобальної цифровізації: кожен сенсор діє



як нервові закінчення, що дозволяє інфраструктурі реагувати на зміни фізичного світу за лічені мілісекунди. Головним ресурсом епохи стає вже не просто накопичення даних, а здатність систем миттєво перетворювати їх на контекстно-залежні дії та прогностичні рішення, що підвищують безпеку та ефективність. Ми знаходимося лише на початку цього масштабного шляху, де межа між нашою фізичною реальністю та її цифровим "двійником" остаточно розмивається, створюючи цілісну інтелектуальну екосистему.

## **РОЗДІЛ 2. ОРГАНІЗАЦІЯ ОСВІТНЬОГО ПРОЦЕСУ У ЗВО УКРАЇНИ**

### **2.1 Інфраструктура підготовки бакалаврів з комп'ютерної інженерії в Україні**

Детальний аналіз матеріально-технічного та програмного забезпечення, що використовується українськими університетами для підготовки бакалаврів за спеціальністю 123 «Комп'ютерна інженерія». Перелік посилань на інтернет-ресурси університетів, а також посилання на офіційні описи освітніх програм та ресурси університетів, що були проаналізовані наведено у списку використаних джерел [36-48] На основі наданих навчальних програм та описів лабораторій можна виділити ключові компоненти інфраструктури, необхідні для формування фахових компетентностей.

Загальний аналіз лабораторної бази та обладнання показав, що університети, які здійснюють підготовку за цією спеціальністю, спираються на розгалужену систему спеціалізованих лабораторій, які забезпечують практичну складову навчання:

1. Комп'ютерні класи та загальнотехнічні лабораторії: Оснащені потужною комп'ютерною технікою для моделювання обчислювальних процесів та розробки програмного забезпечення. Наприклад, у КПІ ім. Ігоря Сікорського функціонують сім навчально-наукових лабораторій та центр «Hewlett-Packard». У НТУ «Дніпровська політехніка» діють п'ять комп'ютерних лабораторій, оснащених 52 сучасними ПК.
2. Мережеві лабораторії: Багато ЗВО (Дніпровська політехніка, КПІ, КрНУ, ДУІКТ) використовують обладнання компанії Cisco для вивчення архітектури мереж, маршрутизації та комутації.
3. Лабораторії промислової автоматизації та кіберфізичних систем: Використовується обладнання провідних світових брендів, таких як Siemens та Festo Didactic. Це дозволяє студентам працювати з реальними

промисловими контролерами та стендами, що відповідають концепції «Індустрія 4.0».

4. Спеціалізовані лабораторії: Включають лабораторії VoIP-технологій, технічної діагностики та моделювання кіберфізичних систем (наприклад, у Чернігівській політехніці).

Програмне забезпечення у навчальному процесі спеціальності охоплює всі рівні взаємодії з апаратним забезпеченням:

- Операційні системи: Основний акцент робиться на Linux (як базовій платформі для системних розробників) та MS Windows.
- Системи автоматизованого проектування (САПР/CAD): Для проектування комп'ютерних систем та їх компонентів використовуються системи на базі ПЛІС (FPGA), засоби розробки друкованих плат (PCB-дизайн) та спеціалізовані середовища, такі як Altium Designer або аналоги від Aldec.
- Середовища розробки та мови програмування: Використовуються C/C++, Python (зокрема для IoT), Java, C# (.NET).
- Промислове ПЗ: Спеціалізована платформа TIA Portal від Siemens для програмування промислових систем.
- Інструменти моделювання: Cisco Packet Tracer для мереж, Matlab для обчислень та імітаційного моделювання.

Аналіз напрямку IoT, мікроконтролери та вбудовані системи показує, що цей напрямок є одним із найбільш пріоритетних та апаратно залежних. Для якісного навчання у цій сфері університети використовують наступне обладнання та технології:

Апаратне забезпечення:

- Мікроконтролерні платформи: Стенди на базі AVR (STK 300), комплекти Arduino (Ard Set) та Raspberry Pi (Rasp Set).

- Системи на кристалі (SoC) та ПЛІС: Використовуються комплекти розробника Terasic Development Kit та інше обладнання для роботи з архітектурою FPGA (Intel/Altera).
- Сенсори та виконавчі механізми: Мобільні сенсорні мережі та дидактичні комплекси для побудови кіберфізичних систем.
- Лабораторії «Smart Lab»: Спеціалізовані простори для інтеграції пристроїв у межах концепції Інтернету речей.

ЛНУ ім. Тараса Шевченка демонструє високий ступінь адаптації до сучасних умов, зокрема через залучення міжнародної підтримки:

- Ключові дисципліни: Програма ЛНУ включає курси «Мікрокомп'ютери та мікроконтролери» (ОК 18), «Комп'ютерна електроніка» (ОК 13) та «Комп'ютерна схемотехніка та логіка» (ОК 15).
- Компетентності: Студенти набувають здатності використовувати та адаптувати мінікомп'ютерні та мікропроцесорні системи (P17), а також експлуатувати віртуальні обчислювальні ресурси (P16).
- Дистанційне навчання: Освітній процес базується на системі eCampus (Moodle), яка інтегрує методичні матеріали, вебінари та інструменти для синхронної/асинхронної взаємодії.

Таким чином підготовка фахівців за спеціальністю 123 «Комп'ютерна інженерія» вимагає високої концентрації дороговартісного обладнання. Ключовим трендом є синергія класичної інженерної підготовки з сертифікатними програмами світових ІТ-гігантів (Cisco, Siemens, HP, AWS), що забезпечує студентам доступ до актуальних інструментів проектування вбудованих систем та IoT-інфраструктури. Використання ПЛІС, сучасних мікроконтролерів та спеціалізованого ПЗ для РСВ-дизайну є обов'язковим стандартом для навчання розробників програмно-апаратних комплексів.

### **2.1.1 Загальний огляд лабораторної та програмної база комп'ютерної інженерії в університетах України**

Аналіз підготовки фахівців за спеціальністю 123 «Комп'ютерна інженерія» в українських закладах вищої освіти свідчить про глибоку інтеграцію теоретичних знань із практичною роботою у спеціалізованих лабораторіях, оснащених сучасним апаратним та програмним забезпеченням. Для забезпечення навчального процесу університети використовують розгалужену мережу лабораторій, що охоплюють напрямки комп'ютерних мереж, мікропроцесорних систем, кіберфізичних систем та вбудованих рішень.

Встановлено, що лабораторна база університетів базується на поєднанні загальної обчислювальної техніки та вузькоспеціалізованих стендів:

- Мережеві лабораторії: Більшість університетів (НТУ «ДП», ДУІКТ, ЛНТУ, Чернігівська політехніка) використовують обладнання компанії Cisco для вивчення мережевих технологій, маршрутизації та комутації.
- Лабораторії вбудованих систем та мікроконтролерів: Використовуються спеціалізовані стенди на базі архітектури AVR (STK 300), комплекти Arduino (Ard Set) та Raspberry Pi (Rasp Set).
- Лабораторії ПЛІС та систем на кристалі (SoC): Для проєктування цифрових систем використовуються комплекти розробника Terasic Development Kit та інше обладнання для роботи з FPGA-технологіями.
- Промислова автоматизація та IoT: Використовується сучасне промислове обладнання Siemens та дидактичні комплекси Festo Didactic, що дозволяють моделювати кіберфізичні системи відповідно до стандартів «Індустрія 4.0».

- Спеціалізовані науково-дослідні простори: У Чернігівській політехніці функціонують лабораторії VoIP-технологій та моделювання кіберфізичних систем. В Одеській політехніці створено спеціалізовані лабораторії Smart Lab для роботи з Інтернетом речей.

Програмне забезпечення для IoT та вбудованих систем охоплює всі рівні розробки:

- Операційні системи: Основна увага приділяється Linux (зокрема курси NDG Linux Essentials) та MS Windows.
- Засоби проектування та моделювання: Використовуються САПР для розробки друкованих плат (PCB-дизайн), Cisco Packet Tracer для моделювання мереж, а також Matlab для імітаційного моделювання складних систем.
- Промислове та IoT ПЗ: Спеціалізована платформа Siemens TIA Portal для програмування промислових мереж, а також хмарні сервіси Amazon Web Services (AWS) для розробки IoT-рішень.
- Мови програмування: Системне та вбудоване програмування реалізується через вивчення C/C++, Python та Java.

### **2.1.2 Комп'ютерна інженерія в ЛНУ імені Тараса Шевченка: освітня стратегія**

Аналіз освітньої діяльності Луганського національного університету імені Тараса Шевченка (ЛНУ) за спеціальністю 123 «Комп'ютерна інженерія» свідчить про системний підхід до підготовки фахівців, зорієнтований на розробку та впровадження комп'ютерних і програмних систем різного призначення. В умовах воєнного стану університет проводить діяльність на базі Миргородської та Полтавської локацій, зберігаючи високу якість навчання через поєднання традиційних методів та сучасних віртуальних середовищ.

Лабораторна база та матеріально-технічне забезпечення.

Навчальний процес у ЛНУ базується на використанні як фізичних спеціалізованих лабораторій, так і віртуальних обчислювальних ресурсів:

- Спеціалізовані лабораторії кафедри: Використовуються для проведення лабораторних, практичних занять та науково-дослідних робіт. Лабораторії оснащені комп'ютерною технікою, мережевим та спеціалізованим устаткуванням.
- Інструменти та обладнання: Студенти вчаться застосовувати контрольно-вимірювальні прилади, програмно-технічні засоби автоматизації та системи автоматизації проектування (САПР).
- Віртуальне середовище: Програма передбачає розвиток здатності використовувати, адаптувати та експлуатувати віртуальні обчислювальні ресурси (компетентність Р16), що є критично важливим для дистанційної та змішаної форм навчання. Використовується система електронного навчання eCampus (на базі Moodle), яка забезпечує синхронний та асинхронний режими взаємодії.

У навчальному плані ЛНУ виділено окремий кластер дисциплін та результатів навчання, безпосередньо пов'язаних з Напрямком IoT, мікроконтролерами із вбудованими рішеннями та мікропроцесорною технікою:

- Ключові дисципліни: До обов'язкових компонентів циклу професійної підготовки включено курс «Мікрокомп'ютери та мікроконтролери» (ОК 18). Також важливими є курси «Комп'ютерна електроніка» (ОК 13) та «Комп'ютерна схемотехніка та логіка» (ОК 15).
- Апаратні навички: Програма формує здатність використовувати та адаптувати мінікомп'ютерні та мікропроцесорні системи (компетентність Р17). Студенти вчаться застосовувати ці системи у

практичній діяльності для вирішення конкретних інженерних задач (ПРН N18).

- Програмне забезпечення: Особливий акцент робиться на вмінні розробляти програмне забезпечення для вбудованих і розподілених застосунків, а також для мобільних і гібридних систем (ПРН N10).
- Проектування систем: Дисципліна «Проектування комп'ютерних систем» (ОК 19) готує студентів до створення комплексних програмно-апаратних рішень.

Програмне забезпечення, що вивчається, охоплює широкий спектр системних та прикладних засобів:

- • Операційні системи: Вивчаються принципи роботи та адміністрування ОС, зокрема в межах курсу «Операційні системи та системне програмування» (ОК 16).
- • Засоби розробки: Програма передбачає опанування сучасними мовами програмування для створення алгоритмічного та програмного забезпечення (компетентність P2).

### **2.1.3 Огляд технічного оснащення університетів України**

#### **Технічне оснащення лабораторій Smart Lab Одеської політехніки**

Проаналізувавши матеріально-технічну базу Національного університету «Одеська політехніка», який здійснює підготовку за спеціальністю 123 «Комп'ютерна інженерія» (освітня програма «Комп'ютерні системи та мережі»), можна виділити конкретний перелік мережевого та суміжного обладнання, що використовується в навчальному процесі,.

Лабораторна база університету для цього напряму зосереджена у чотирьох спеціалізованих лабораторіях Smart Lab та комп'ютерних класах,. Для вивчення мережевих технологій використовується наступне обладнання:



Лабораторії Smart Lab (яких в університеті налічується 4 одиниці) є ключовим елементом інфраструктури для опанування сучасних технологій у напрямках IoT, мікроконтролерів та вбудованих систем. На основі аналізу освітньо-професійної програми «Комп'ютерні системи та мережі» можна виділити наступний перелік технічних засобів та обладнання, що використовуються в цих та суміжних лабораторіях кафедри:

Мікроконтролерні платформи та вбудовані системи. Для вивчення архітектури та програмування мікропроцесорних пристроїв лабораторії забезпечені спеціалізованими стендами та індивідуальними наборами:

- • Стенди AVR Starter Kit STK 300: У розпорядженні студентів є 8 одиниць таких лабораторних стендів, які дозволяють вивчати роботу з мікроконтролерами архітектури AVR.
- • Комплекти Arduino (Ard Set 1): Наявні 8 комплектів, що є стандартом для розробки прототипів у сфері Інтернету речей (IoT).
- • Комплекти Raspberry Pi (Rasp Set 1): Використовується 8 комплектів для вивчення вбудованих систем на базі ОС Linux та реалізації складних обчислювальних вузлів IoT.

Проектування цифрових систем та ПЛІС (FPGA). Для низькорівневого проектування комп'ютерних систем та систем на кристалі використовуються професійні засоби розробки: Terasic Development Kit: Університет використовує 3 комплекти розробника для роботи з ПЛІС (FPGA), що дозволяє студентам проектувати цифрову логіку на апаратному рівні.

Мережеве обладнання та діагностика. Підготовка за напрямком комп'ютерних мереж (зокрема в межах курсів Cisco) підкріплена потужною апаратною базою:

- • Мережеве комунікаційне обладнання: Лабораторний фонд включає 20 одиниць мережесхемних пристроїв для побудови та налаштування локальних і корпоративних мереж.

- • Стенди для діагностики: Функціонують 14 лабораторних стендів, призначених для дослідження методів діагностики комп'ютерних систем та мереж.

Загальна інфраструктура та програмне забезпечення. Освітній процес у Smart Lab та комп'ютерних класах (яких всього 4) підтримується сучасними обчислювальними ресурсами:

- • Обчислювальна техніка: Робочі місця оснащені персональними комп'ютерами з безкоштовним доступом до мережі Internet.
- • Програмний стек: Лабораторії використовують спеціалізоване ПЗ для програмування, проєктування систем IoT та супроводу вбудованих рішень.

Такий набір технічних засобів дозволяє організовувати змішаний режим навчання, де студенти можуть працювати як із фізичними стендами безпосередньо в аудиторіях, так і використовувати індивідуальні набори (Arduino, Raspberry Pi) для виконання практичних завдань у дистанційному форматі.

### **Технологічна база підготовки інженерів у Дніпровській політехніці**

На основі аналізу освітньо-професійної програми та супутніх документів Національного технічного університету «Дніпровська політехніка», кафедра інформаційних технологій та комп'ютерної інженерії використовує потужну матеріально-технічну базу, що поєднує сучасне обчислювальне, мережеве та промислове обладнання,.

Для підготовки фахівців за спеціальністю 123 «Комп'ютерна інженерія» у Дніпрі використовується таке спеціалізоване обладнання:

- Мережеве обладнання Cisco: Кафедра має три комплекси мережного устаткування компанії Cisco. Це обладнання використовується для практичного освоєння технологій локальних, корпоративних і

глобальних мереж, а також для вивчення програмно-конфігурованих мережних середовищ,,.

- Промислове обладнання Siemens: Лабораторії оснащені сучасним промисловим обладнанням торговельної марки Siemens. Зокрема, використовуються промислові контролери для проектування та програмування складних систем передачі та обробки інформації,,.
- Дидактичні комплекси Festo Didactic: У навчальному процесі задіяні спеціалізовані стенди та комплекси Festo Didactic, які є провідним світовим рішенням для інженерної освіти у сфері автоматизації та кіберфізичних систем,,.
- Загальна обчислювальна база: Практична підготовка здійснюється на базі п'яти комп'ютерних лабораторій та спеціалізованих науково-дослідних лабораторій, які сумарно укомплектовані 52 сучасними комп'ютерами.

Програмно-апаратна інтеграція для IoT та вбудованих систем. Особливістю підготовки в НТУ «ДП» є глибока інтеграція апаратних засобів із професійним програмним забезпеченням, що критично важливо для напрямків мікроконтролерів та Інтернету речей:

1. Платформа TIA Portal (Siemens): Використовується як спеціалізоване ліцензійне середовище для проектування та програмування сучасних промислових систем і мереж.

2. Академія Cisco та хмарні сервіси: У програму імплементовано курси за міжнародними стандартами, зокрема «IoT Fundamentals: Connecting Things», що дозволяє студентам працювати з пристроями Інтернету речей,. Також використовуються ресурси академії Amazon Web Services (AWS) для роботи з хмарними обчисленнями,.

3. Системи розробки в GNU/Linux: У комп'ютерних класах розгорнуті open source системи розробки ПЗ для роботи в GNU/Linux-сумісних операційних системах, що є стандартом для розробників вбудованих систем,.

4. Моделювання та налаштування: Студенти використовують Cisco Packet Tracer для віртуального моделювання та налаштування мережних структур перед їх реалізацією на фізичному обладнанні,.

Така лабораторна база дозволяє моделювати процеси проєктування технічного та програмного забезпечення в реальних умовах, що відповідає сучасній концепції «Індустрія 4.0»,,.

### **Лабораторна база підготовки фахівців з комп'ютерної інженерії ДУІКТ**

На основі аналізу навчально-методичної документації Державного університету інформаційно-комунікаційних технологій (ДУІКТ), підготовка фахівців за спеціальністю 123 «Комп'ютерна інженерія» базується на використанні спеціалізованих лабораторій, що забезпечують практичну підготовку в галузі проєктування та обслуговування обчислювальних систем.

Для проведення практичних занять, спрямованих на формування фахових компетентностей, на кафедрі використовуються такі лабораторії:

- • Спеціалізована комп'ютерна лабораторія № 0-25, яка оснащена сучасною обчислювальною технікою та спеціалізованим обладнанням.
- • Спеціалізована комп'ютерна лабораторія № 404, де розгорнуто програмно-апаратні комплекси для вивчення мережних та системних технологій.

Аналіз обладнання та інструментарію. Лабораторна база кафедри орієнтована на опанування повного циклу розробки комп'ютерних систем — від низькорівневої електроніки до складних мережних структур. Основними технічними засобами є:

- Мережеві комплекси компанії Cisco: Лабораторії оснащені сучасними програмно-апаратними рішеннями Cisco, що дозволяє студентам отримувати практичні навички роботи з промисловим мережевим обладнанням.

- Контрольно-вимірювальні прилади: Використовуються для діагностики та налагодження електронних компонентів комп'ютерних систем.
- Системи автоматизації проєктування (САПР): Лабораторії забезпечені програмними засобами для автоматизованого та автоматичного проєктування апаратного забезпечення.

Обладнання для напрямків IoT, мікроконтролерів та вбудованих систем.

З позиції аналітика навчального процесу, важливо відзначити, що університет акцентує увагу на об'єктах діяльності, безпосередньо пов'язаних із вбудованими та кіберфізичними системами. Для цього в лабораторіях передбачено використання:

1. Програмно-технічних засобів автоматизації, що є базою для розробки систем Інтернету речей (IoT) та Industrial IoT.
2. Апаратної бази для вбудованих систем, що включає програмовані та реконфігуровні засоби, які використовуються в стаціонарних та мобільних комп'ютерних пристроях.
3. Інструментарію для розробки спеціалізованого ПЗ, зокрема для технологій мережних, мобільних та хмарних обчислень.

Високий рівень підготовки підтримується завдяки функціонуванню першого в Києві сертифікаційного центру Pearson VUE, де студенти мають можливість пройти сертифікацію від таких технологічних гігантів, як Cisco, Juniper, Huawei, ZTE, Apple, Microsoft та IBM. Це забезпечує доступ до актуальних апаратних стандартів, що використовуються у глобальній ІТ-індустрії для створення вбудованих рішень та мережевих інфраструктур.

Додатково освітній процес підтримується через систему дистанційного навчання Moodle, де розміщені навчально-методичні матеріали для проведення лабораторних занять та самостійної роботи здобувачів. 20% професійно-орієнтованих дисциплін викладаються англійською мовою, що

дозволяє студентам ефективно працювати з оригінальною технічною документацією до мікропроцесорної техніки та IoT-платформ.

### **Освітнє середовище підготовки комп'ютерних інженерів у Національному університеті «Чернігівська політехніка»**

Для організації якісного навчання, зокрема в дистанційному та змішаному режимах, університет використовує наступний комплекс інформаційних ресурсів:

#### **1. Платформи дистанційного навчання та комунікації**

- Система Moodle: Є основним хабом для розміщення навчально-методичного забезпечення. У системі представлені робочі програми, силабуси, конспекти лекцій, методичні вказівки для лабораторних і практичних занять, а також завдання для поточного та підсумкового контролю.
- Microsoft Teams: Активно використовується для проведення відкритих лекцій, гостьових вебінарів та семінарів у синхронному режимі.
- Zoom: Застосовується як альтернативний інструмент для проведення гостьових лекцій та онлайн-зустрічей.

#### **2. Бібліотечні та наукові ресурси**

- Наукова бібліотека університету: Забезпечує доступ до бібліотечно-інформаційної системи та повнотекстових баз даних навчальної і наукової літератури.
- Міжнародні бази даних: Університет надає доступ до провідних наукометричних баз, таких як Scopus та Web of Science, що критично важливо для дослідницької роботи студентів.

- Електронний репозитарій: Використовується для оприлюднення та зберігання кваліфікаційних робіт бакалаврів, що сприяє дотриманню принципів академічної доброчесності.

3. Лабораторна база та спеціалізоване ПЗ. Для підготовки в напрямках IoT, мікроконтролерів та вбудованих систем університет використовує спеціалізовані лабораторії, які оснащені відповідним програмним та апаратним забезпеченням:

- Лабораторія вбудованих систем: Оснащена обладнанням для вивчення мікроконтролерної техніки та розробки систем спеціального призначення.
- Лабораторія моделювання кіберфізичних систем: Використовується для дослідження складних інтегрованих систем, що є основою сучасного Інтернету речей (IoT).
- Лабораторія мережевих технологій та VoIP-технологій: Забезпечує практичну підготовку в галузі проектування та адміністрування комп'ютерних мереж.
- Спеціалізоване ПЗ: Програма передбачає використання сучасного програмного забезпечення для моделювання та проектування комп'ютерних засобів.

4. Додаткові освітні ресурси:

- Онлайн-курси та ресурси Інтернет: Студенти залучаються до використання зовнішніх освітніх платформ та ресурсів для самостійної роботи.
- Система менторства та тьюторства: Університет розвиває систему підтримки студентів («тьюторство та менторство»), що допомагає в адаптації до навчання та підготовці проєктів за завданнями від ІТ-галузі.

Завдяки інтеграції цих ресурсів, НУ «Чернігівська політехніка» забезпечує повний цикл підготовки інженерів, поєднуючи теоретичну базу в Moodle з практичними навичками у віртуальних та фізичних лабораторіях.

### **Освітнє середовище підготовки комп'ютерних інженерів у Національному університеті «КПІ»**

Центр Hewlett-Packard є важливою складовою лабораторної бази факультету інформатики та обчислювальної техніки і надає студентам низку стратегічних переваг для професійного становлення. Основні переваги та ресурси, які центр «Hewlett-Packard» надає здобувачам вищої освіти:

#### **1. Доступ до сучасного технічного та програмного забезпечення**

Центр оснащений передовим апаратним та програмним інструментарієм, що відповідає актуальним вимогам ІТ-індустрії. Це дозволяє студентам працювати не на застарілих тренажерах, а з промисловим обладнанням та софтом, що безпосередньо використовується провідними світовими компаніями для розв'язання складних інженерних завдань.

2. Забезпечення повного циклу проєктної діяльності. На базі центру реалізується наскрізний підхід до навчання, який включає такі етапи:

- Лабораторні заняття: Практичне відпрацювання теоретичних знань у межах навчальних дисциплін.
- Виконання курсових та дипломних проєктів: Можливість використовувати ресурси центру для розробки складних систем паралельно з вивченням відповідних курсів.
- Наукові дослідження: Центр надає базу для проведення експериментальних досліджень за професійною тематикою під керівництвом провідних фахівців кафедри.

3. Формування високого рівня професійної підготовки. Співпраця КПІ з такими гігантами, як Hewlett-Packard, гарантує високу якість навчання завдяки:



- Синхронізації з ринком праці: Високий рівень підготовки забезпечується наявністю договорів про співпрацю з провідними ІТ-компаніями, що дає студентам змогу опановувати компетенції, які реально затребувані сучасним ринком.
- Проектно-орієнтованому навчанню: Ресурси центру дозволяють моделювати процеси проектування технічного та програмного забезпечення в реальних умовах, що сприяє створенню інноваційних стартап-проектів.

4. Підтримка напрямків вбудованих систем та IoT. Хоча центр має загальноінженерну спрямованість, його база підтримує фундаментальні курси, критичні для напрямків мікроконтролерів та вбудованих систем:

- Студенти мають можливість відпрацьовувати навички у сфері Data Science, IoT та інтелектуальних інформаційних технологій, використовуючи обчислювальні потужності центру для аналізу великих обсягів даних.
- Інфраструктура центру інтегрована з освітніми компонентами, що стосуються архітектури комп'ютерів, мікропроцесорних засобів та цифрової схемотехніки.

5. Підтримка змішаного та дистанційного навчання. В умовах воєнного стану центр виступає як хаб, що інтегрується з платформою дистанційного навчання Sikorsky. Це дозволяє забезпечувати безперервність освітнього процесу, надаючи студентам можливість використовувати мережеві та обчислювальні ресурси навіть у віддаленому режимі роботи.

Таким чином, Центр «Hewlett-Packard» трансформує теоретичну підготовку студентів КІП у реальний інженерний досвід. Головною перевагою є можливість виконувати всі види практичних робіт — від лабораторних до наукових проєктів — на базі сучасної екосистеми, що робить випускників конкурентоздатними як в Україні, так і в міжнародних ІТ-проєктах.

Додатково розглянуто структуру та роль Науково-технічної бібліотеки (Scientific and Technical Library) КПІ ім. Ігоря Сікорського в межах підготовки фахівців за спеціальністю 123 «Комп'ютерна інженерія».

Згідно з проаналізованими даними, бібліотека є невід'ємною частиною інформаційного та навчально-методичного забезпечення університету, що відповідає встановленим технологічним вимогам до освітньої діяльності на бакалаврському рівні. Функціонально та інфраструктурно бібліотека займає наступне місце в структурі закладу:

- Інформаційний хаб та репозитарій: Бібліотека слугує базою для функціонування електронного архіву наукових та освітніх матеріалів Університету. Саме тут у вільному доступі розміщуються кваліфікаційні роботи бакалаврів після їх публічного захисту, що забезпечує прозорість навчання та дотримання принципів академічної доброчесності.
- Ресурсна база для ІТ-спеціальностей: Бібліотека забезпечує доступ до повнотекстових наукових ресурсів та баз даних, що є критичним для здобувачів освіти за програмою «Комп'ютерні системи та мережі». Це дозволяє студентам працювати з актуальними міжнародними стандартами та результатами сучасних досліджень у галузі комп'ютерної інженерії.
- Місце в структурі кампусу: У загальній структурі університету бібліотека входить до переліку основних підрозділів і сервісів, інформація про які (включаючи розташування на карті) міститься в розділі «Адреса та карта» офіційного порталу КПІ. Вона інтегрована в освітнє середовище разом із сімома навчально-науковими лабораторіями кафедри та центром «Hewlett-Packard», формуючи цілісний комплекс для виконання курсових та дипломних проєктів.

Для напрямку IoT, мікроконтролерів та вбудованих систем бібліотека відіграє роль методичного центру, надаючи доступ до документації з

проектування програмно-апаратних засобів та автоматизованих систем. Це підкріплює практичну підготовку фахівців, які використовують сучасну елементну базу, таку як ПЛІС (FPGA), для створення високопродуктивних комп'ютерних систем.

Таким чином, хоча фізичні координати (номер корпусу) у конкретних фрагментах навчальних планів не деталізуються, бібліотека територіально розташована в межах кампусу КПІ ім. Ігоря Сікорського та є ключовим складником інфраструктури підтримки якості вищої технічної освіти.

### **Цифрова екосистема дистанційного навчання СНУ імені Володимира Даля**

На засадах аналізу інформаційних ресурсів у мережі Інтернет зроблено огляд цифрової екосистеми, що забезпечує дистанційне навчання у Східноукраїнському національному університеті імені Володимира Даля (СНУ). Університет побудував комплексну систему, яка базується на поєднанні синхронних та асинхронних методів взаємодії. Основними платформами та ресурсами є:

1. Система електронного навчання eCampus: Це ключовий хаб дистанційної освіти університету, побудований на базі платформи Moodle. Вона забезпечує повний цикл навчання та документообігу, включаючи доступ до електронних курсів, силабусів та ресурсів для вибору індивідуальної траєкторії навчання. Платформа працює в режимі 24/7, надаючи доступ до навчально-методичних матеріалів з усіх дисциплін.

2. Засоби синхронної комунікації: Для проведення занять у режимі реального часу університет використовує технології відеоконференцзв'язку, вебінарів та чатів, які інтегровані в освітній процес. Це дозволяє не лише проводити традиційні лекції, а й залучати зовнішніх фахівців та науковців до онлайн-викладання у синхронному режимі з віддаленим підключенням.

3. Сайт Наукової бібліотеки СНУ: Забезпечує доступ до повнотекстових баз даних наукової та навчальної літератури, що є критично важливим для виконання самостійних та кваліфікаційних робіт.

4. Спеціалізовані лабораторії дистанційного навчання: Технічна база для онлайн-заходів була суттєво підсилена в межах проєкту USAID G-Kra-042. Це включає як самі лабораторії, так і індивідуальні комплекти обладнання, що видаються студентам для забезпечення практичної складової навчання в дистанційному форматі.

5. Офіційний та кафедральні веб-ресурси: Використовуються для оперативного інформування про правила прийому, графіки навчання та розміщення каталогів силабусів.

Завдяки активному використанню системи eCampus, університет успішно реалізує модель змішаного навчання, де асинхронне вивчення теорії гармонійно поєднується з інтерактивною практичною підготовкою.

## **2.2 Змішана форма підготовки інженерів-комп'ютерників в умовах воєнного стану**

Аналіз організації змішаного навчання та використання лабораторної бази для спеціальності 123 «Комп'ютерна інженерія» в умовах воєнного стану.

Лабораторії, що використовуються у змішаному режимі. Сучасна підготовка інженерів-комп'ютерників в змішаному форматі базується на поєднанні хмарних технологій, віртуальних симуляторів та віддаленого доступу до обладнання:

1. Мережеві віртуальні лабораторії: Більшість університетів (НТУ «ДП», ДУІКТ, КрНУ) використовують ресурси Cisco Networking Academy. Основним інструментом є Cisco Packet Tracer, який дозволяє студентам проєктувати та налаштовувати складні топології мереж у віртуальному середовищі.

2. Хмарні лабораторії IoT та обчислень: Використання платформи Amazon Web Services (AWS) дозволяє студентам працювати з реальними хмарними сервісами для Інтернету речей та розподілених обчислень без фізичного доступу до обладнання.
3. Лабораторії вбудованих систем та ПЛІС: Проектування здійснюється за допомогою спеціалізованого ПЗ (САПР), такого як середовища для роботи з FPGA (Intel/Altera) та інструменти розробки друкованих плат. Це дозволяє моделювати роботу вбудованих систем на комп'ютері студента перед реалізацією на «залізі».
4. Віртуальні обчислювальні ресурси та симулятори: Університети (наприклад, ЛНУ імені Тараса Шевченка та СХУ імені Володимира Даля) активно впроваджують компетентності з використання та адаптації віртуальних обчислювальних ресурсів, що дозволяє замінити частину фізичних стендів програмними аналогами. Використання Cisco Packet Tracer (ДУІКТ, НТУ «ДП», КрНУ) дозволяє відпрацьовувати навички налаштування мережевого обладнання без фізичного доступу до нього.
5. Спеціалізовані лабораторії дистанційного навчання: Створення окремих хабів, оснащених індивідуальними комплектами обладнання для віддаленої роботи, як це було зроблено у СХУ імені В. Даля за підтримки USAID.
6. Спеціалізоване ПЗ та САПР: Програми для проектування цифрових систем (VHDL/Verilog), систем на базі ПЛІС (FPGA) та розробки друкованих плат (PCB) можуть встановлюватися на персональні комп'ютери студентів або запускатися через віддалений робочий стіл (КПІ, ХНУРЕ).
7. Індивідуальні апаратні комплекти: Університети (Одеська політехніка, СХУ ім. В. Даля) використовують мобільні набори на базі Arduino (Ard

Set), Raspberry Pi (Rasp Set) та AVR Starter Kit STK 300, які студенти можуть використовувати вдома.

Змішаний режим дозволяє підтримувати високий теоретичний рівень та формувати навички програмування і моделювання. Однак, повна якісна підготовка інженера-комп'ютерника без фізичного контакту з промисловим обладнанням (контролери Siemens, стенди Festo Didactic) залишається проблематичною. Професійна якість забезпечується за рахунок інтеграції сертифікатних програм Cisco, HP та AWS, що робить випускників конкурентоздатними на міжнародному ринку. Однак, при цьому можна відокремити певні ризики та проблеми організації

- Безпекові: Проведення занять під час повітряних тривог.
- Кіберзагрози: Необхідність захисту інформаційних систем університетів, які використовуються для дистанційного доступу.
- Нерівність доступу: Студенти можуть мати різну потужність домашніх комп'ютерів або якість зв'язку.
- Вартість обладнання: Потреба в постійній модернізації лабораторної бази, що є дороговартісною.
- Методична адаптація: Необхідність перетворення традиційних лекцій та лабораторних на інтерактивні онлайн-курси в системах Moodle або eCampus.
- Організаційна невизначеність: Робота в умовах "неповної визначеності умов", що зафіксовано в освітніх стандартах.

Однак, аналіз документів та освітніх програм дозволяє стверджувати, що ряд ЗВО роблять певні кроки для покращення ситуації:

1. Розвиток LMS (Learning Management Systems): Впровадження потужних систем eCampus (ЛНУ, СХУ ім. В. Даля), Sikorsky (КПІ), Moodle (Чернігівська політехніка, ЛНТУ), що забезпечують цілодобовий доступ до матеріалів.

2. Залучення міжнародної допомоги: Реалізація проєктів на кшталт USAID G-Kra-042, завдяки якому СНУ ім. В. Даля отримав лабораторії дистанційного навчання та індивідуальні комплекти для студентів.
3. Релокація та адаптація: Університети з прифронтових зон (ЛНУ ім. Т. Шевченка, СНУ ім. В. Даля) відновили діяльність на безпечних локаціях, повністю зберігши освітній процес у віртуальному середовищі.
4. Військова підготовка: Інтеграція до програм дисциплін з базової загальновійськової підготовки (БЗВП) та цивільного захисту, що важливо для формування навичок виживання та оборони держави.
5. Партнерство з IT-гігантами: Створення спільних центрів (наприклад, Huawei Excellence Center у KAI або лабораторії HP у КПІ) для надання студентам доступу до найсучасніших технологій.
6. Застосування віртуальних ресурсів: Активно впроваджуються компетентності з використання та адаптації віртуальних обчислювальних ресурсів, що дозволяє замінити частину фізичних стендів програмними аналогами.

Ці кроки дозволяють українським технічним університетам не просто зберігати життєздатність, а й готувати фахівців, здатних розробляти рішення подвійного призначення, що є критичним для відновлення країни.

### **2.2.1 Дії ЗВО для організації навчання в умовах воєнного стану та змішаного режиму навчання**

Українські університети виконали комплекс заходів для забезпечення безперервності та безпеки освітнього процесу:

1. Розвиток цифрової інфраструктури:
  - Впровадження та вдосконалення систем електронного навчання: eCampus (на базі Moodle), Google Classroom, Microsoft Teams та платформа Sikorsky.

- Забезпечення синхронного (вебінари, онлайн-лекції) та асинхронного режимів навчання.

- Впровадження систем хмарних та віртуальних обчислень

## 2. Адаптація навчальних планів до реалій війни:

- Включення до програм обов'язкової дисципліни «Базова загальновійськова підготовка» (БЗВП), яка поєднує теоретичну підготовку в університеті та практичну на базі МО України.
- Фокус на технологіях подвійного призначення та захисті критичної інфраструктури.
- Перегляд змісту професійно-орієнтованих дисциплін з урахуванням потреб ринку праці в умовах воєнного стану.

## 3. Релокація та організація безпеки:

- Університети з прифронтових територій (ЛНУ, СЛУ імені В. Даля) організували роботу на базі нових локацій (Полтава, Миргород, Київ), зберігши науково-педагогічний потенціал.
- Розробка алгоритмів дій персоналу та студентів за сигналом «Повітряна тривога».

## 4. Соціально-психологічна та менторська підтримка:

- Впровадження програм психологічної стійкості («Турбота про себе», тренінги зі стресостійкості).
- Розвиток системи менторства та тьюторства для допомоги студентам у дистанційному режимі.

## 5. Інтернаціоналізація та партнерство:

- Активізація програм міжнародної академічної мобільності (Erasmus+), що дозволяє студентам продовжувати навчання в європейських університетах.
- Залучення фахівців провідних ІТ-компаній як гостей лекторів та менторів для підтримки актуальності знань.



Ці заходи дозволили спеціальності 123 залишатися однією з найбільш адаптованих до віддаленої роботи, оскільки інженерне проєктування значною мірою спирається на програмне моделювання, яке можна ефективно реалізувати через дистанційні платформи..

### **Трансформація лабораторій комп'ютерної інженерії в умовах війни**

Аналіз підготовки фахівців за спеціальністю 123 «Комп'ютерна інженерія» в умовах воєнного стану свідчить про глибоку трансформацію лабораторної бази. Основний акцент змістився на використання віртуальних обчислювальних ресурсів, хмарних сервісів та спеціалізованих індивідуальних комплектів обладнання для забезпечення безперервності навчання в дистанційному або змішаному режимах.

Нижче наведено конкретні дані щодо лабораторій, які організували або мають потенціал для дистанційного використання своїх ресурсів:

1. Використання віртуальних та хмарних платформ для лабораторних робіт  
Багато університетів замінили фізичну присутність у лабораторіях доступом до хмарних сервісів та симуляторів:

- Хмарні обчислення та IoT: Кременчуцький національний університет (КрНУ) та НТУ «Дніпровська політехніка» імплементували курси Amazon Web Services (AWS). Це дозволяє студентам використовувати ресурси віддалених центрів обробки даних для розробки IoT-рішень без необхідності фізичного доступу до серверного обладнання.
- Мережеві технології: Державний університет інформаційно-комунікаційних технологій (ДУІКТ), ЛНТУ та НТУ «Дніпровська політехніка» активно використовують віртуальні ресурси Cisco Academy, зокрема симулятор Cisco Packet Tracer. Це забезпечує можливість проєктування та налаштування складних мереж у віртуальному середовищі з будь-якого місця.

- Спеціалізовані платформи: КПІ ім. Ігоря Сікорського використовує власну платформу дистанційного навчання Sikorsky, яка інтегрує мережеві технології для проведення лекцій та лабораторних занять.

## 2. Лабораторії, адаптовані до змішаного та дистанційного режимів

Конкретні дані щодо організації лабораторної бази в університетах:

- СХУ імені Володимира Даля: За підтримки проєкту USAID (G-Kra-042) створено спеціалізовані лабораторії дистанційного навчання, що включають індивідуальні комплекти обладнання. Це дозволяє студентам працювати з апаратною частиною вдома, отримуючи менторську підтримку через систему eCampus (Moodle).
- Одеська політехніка: Університет використовує лабораторії Smart Lab (4 одиниці) та спеціалізовані комплекти:
  - ◦ Комплекти розробника Terasic Development Kit (FPGA) – 3 одиниці.
  - ◦ Лабораторні стенди AVR Starter Kit STK 300 – 8 одиниць.
  - ◦ Комплекти обладнання Arduino (Ard Set) та Raspberry Pi (Rasp Set) – по 8 одиниць. Ці засоби ідеально підходять для змішаного навчання: студент може отримати комплект для індивідуальної роботи або підключатися до стендів віддалено через інструменти Google та Microsoft.
- НУ «Чернігівська політехніка»: Лабораторні приміщення оснащені для дистанційного доступу до систем VoIP-технологій, мережевих технологій та моделювання кіберфізичних систем. Навчання підтримується через систему MOODLE.
- НТУ «Дніпровська політехніка»: Використовує платформу Siemens TIA Portal для дистанційного проєктування та програмування промислових систем і мереж. Це дозволяє моделювати роботу промислових контролерів без фізичної присутності в лабораторії промислової автоматизації.

### 3. Програмне забезпечення для дистанційного моделювання

Для реалізації напрямку вбудованих систем та мікроконтролерів у дистанційному режимі заклади вищої освіти використовують такий стек ПЗ:

- Системне програмування: Широке впровадження Linux (GNU/Linux-сумісних ОС) як середовища розробки з відкритим кодом, що легко розгортається на персональних комп'ютерах студентів.
- Проектування апаратури: Використання мов опису апаратури VHDL/Verilog та САПР для ПЛІС (FPGA), що дозволяє студентам КПІ, ХНУРЕ та НТУ «ДП» проектувати цифрові системи на логічному рівні перед їх фізичною реалізацією.
- Моделювання: Застосування Matlab (КрНУ, НТУ «ДП») для імітаційного моделювання складних процесів у кіберфізичних системах.

Висновок. В умовах воєнного стану університети успішно трансформували лабораторії з "місць фізичної присутності" на "центри надання ресурсів". Ключову роль відіграли індивідуальні набори (Arduino/Raspberry Pi), які студенти можуть використовувати вдома, та віртуалізація промислового обладнання (Cisco, Siemens). Особливо ефективним є кейс СНУ імені В. Даля з впровадженням мобільних комплектів обладнання за міжнародної підтримки, що є взірцем для організації змішаного навчання в технічних ЗВО.

#### **2.2.2 Стратегії змішаного навчання в комп'ютерній інженерії та лабораторна база спеціальності**

Напрями розвитку змішаного навчання та дистанційні лабораторії для забезпечення якісної підготовки за спеціальністю 123 у змішаному режимі, університети мають розвивати такі напрями:

1. Модель «Лабораторія як сервіс» (LaaS). Це передбачає перехід від фізичної присутності до віддаленого керування обладнанням. До першочергових кроків реалізації можна віднести впровадження систем віддаленого

доступу: Встановлення серверів для віддаленого підключення до робочих станцій, з'єднаних із ПЛІС (FPGA) та мікроконтролерами.

2. Візуалізація процесів: Оснащення стендів веб-камерами та зворотним зв'язком, щоб студент міг бачити результати виконання коду на реальному «залізі».
3. Використання хмарних симуляторів: Інтеграція з платформами типу Cisco Packet Tracer, Siemens TIA Portal та хмарами Amazon Web Services (AWS).
4. Забезпечення мобільними апаратними комплектами.

Для робіт, які неможливо віртуалізувати, студентам мають надаватися індивідуальні набори.

- Обладнання: Комплекти Arduino (Ard Set), Raspberry Pi (Rasp Set) та стенди на базі AVR (STK 300).
- Логістика: Створення фонду «переїзних» лабораторій, які видаються студентам на семестр.

На засадах проведеного аналізу встановлено, що частина університетів вже реалізував віддалений та змішаний доступ?

Згідно з аналізом джерел, низка університетів вже зробила успішні кроки:

- СХУ імені Володимира Даля: Реалізував проєкт USAID G-Kra-042, у межах якого сформовано лабораторії дистанційного навчання та закуплено індивідуальні комплекти обладнання для студентів.
- КПІ імені Ігоря Сікорського: Використовує платформу Sikorsky для інтеграції мережевих технологій та проведення лабораторних занять у віддаленому форматі.
- Одеська політехніка: Створила 4 спеціалізовані Smart Lab та забезпечила кафедру значною кількістю FPGA-комплектів (Terasic) та Arduino/Raspberry Pi (по 8 одиниць кожного типу) для гнучкої роботи.

- НТУ «Дніпровська політехніка»: Імплементувала ліцензійне ПЗ Siemens TIA Portal, що дозволяє дистанційно проєктувати промислові мережі.
- КрНУ: Активно використовує академії Cisco та AWS для надання доступу до хмарних обчислювальних ресурсів.

### **Основні напрямки удосконалення**

На засадах проведеного аналізу встановлено, що для досягнення повноцінної якості навчання в умовах воєнного стану та змішаного режиму необхідно:

1. Трансформація мережевого середовища. Організація віддаленого доступу до локальних обчислювальних ресурсів спеціалізованих лабораторій.
2. Трансформація системи електроживлення. Встановлення додаткових програмних та апаратних засобів атомічного переходу на резервне живлення, асиметричного вимкнення та включення обчислювальних приладів.
3. Організаційні заходи. Необхідність реорганізації графіків навчального процесу, розкладу занять та створення додаткових програмно-технічних застосунків, що дозволяють контролювати віддалений доступ до обладнання обчислювальних лабораторій , розробка методик використання віддаленого доступу.
4. Кіберзахист лабораторних мереж: Зі збільшенням віддалених підключень зростає ризик атак на інфраструктуру університетів. Необхідно впроваджувати захищені VPN-канали для студентів, налаштовувати брандмауери та мережеві екрани .
5. Спільне використання ресурсів (Resource Sharing): Створення міжвузівських хабів, де студенти з різних університетів (особливо

прифронтових) можуть віддалено використовувати унікальне дороге обладнання.

6. Гібридна атестація: Розробка методик захисту кваліфікаційних робіт, де апаратна частина демонструється в режимі реального часу через відеозв'язок, що вже практикується в ЛНУ та КрНУ.
7. Адаптація під технології подвійного призначення: Оскільки стандарт передбачає підготовку до захисту Вітчизни, лабораторії мають включати стенди з вивчення систем РЕБ, безпілотних технологій та захищеного зв'язку.

Ці кроки дозволять перетворити виклики змішаної форми освіти на перевагу, формуючи у студентів навички роботи з розподіленими системами, що є критичним для сучасної комп'ютерної інженерії.

## РОЗДІЛ 3. ВІДДАЛЕНИЙ ДОСТУП ДО НАВЧАЛЬНОЇ ЛАБОРАТОРІЇ MIIPOKONPOMEPІB TA IOY

### 3.1 Огляд операційних систем для Raspberry pi

#### Raspbian / Raspberry Pi OS

Raspbian [49]— це безкоштовна операційна система, заснована на дистрибутиві Linux Debian та оптимізована для апаратного забезпечення Raspberry Pi. Початкова версія порту Debian, ініційована головним чином розробниками Майком Томпсоном та Пітером Гріном, була випущена невдовзі після випуску Raspberry Pi у червні 2012 року. Відтоді Raspbian є офіційною операційною системою Raspberry Pi. На той час дистрибутив вже містив понад 35 000 програмних пакетів, доступних користувачеві після встановлення. Однак завдяки постійному розвитку обсяг репозиторію постійно зростає і тепер навіть включає офісні програми, такі як LibreOffice, та поштовий клієнт.

Щоб користуватися Raspbian, можете самостійно встановити систему на SD-карту за допомогою завантаженого RaspbianInstaller або завантажити готовий образ та скопіювати його на потрібну SD-карту. Для цього у вас є вибір між офіційними образами Raspberry Pi Foundation та образами, наданими членами дуже активної спільноти Raspbian. У першому випадку доступна як версія з настільним інтерфейсом (потрібна щонайменше одна SD-карта на 8 ГБ), так і slim-версія без графічного інтерфейсу (Lite) – кожна у вигляді ZIP-або торрент-файлу. У магазині Raspberry Pi також можете придбати карти з встановленими образами.

Таблиця 3.1 Перелік переваг та недоліків

Переваги	Недоліки
Дуже зручний для початківців	Затримки оновлень програмного забезпечення

Переваги	Недоліки
Відмінна підтримка	

## Kali Linux

Ще одним нащадком Debian, який можна налаштувати у версії ARM як дистрибутив Raspberry Pi, є Kali Linux [50]. Програма з відкритим кодом, розроблена Маті Ахоні та Девонем Кірнсом з американської компанії Offensive Security, була опублікована у своїй першій версії 13 березня 2013 року. Дистрибутив можна використовувати як звичайне програмне забезпечення операційної системи, але в основному він використовується для широкого тестування безпеки та проникнення комп'ютерних систем і мереж. Для цього програма містить понад 600 інструментів, включаючи мережевий сканер Nmap, фреймворк Metasploit (платформу для тестування експлойтів) або зломщик паролів John the Ripper. Kali Linux є офіційним наступником BackTrack, який базується на дистрибутиві Linux Ubuntu.

Kali Linux розкрито на сайті Offensive Security (<https://vimeo.com/offsec>) на Vimeo (<https://vimeo.com/>).

Оскільки Kali Linux зосереджений на темі безпеки, операційна система Raspberry Pi регулярно отримує оновлення безпеки та програмного забезпечення з репозиторію Debian. Це гарантує, що система завжди оновлена. За замовчуванням програмний пакет є досить мінімалістичним, щоб запропонувати найменшу можливу зону атаки. Додаткові програми можна додавати будь-коли за потреби. Велика спільнота активно бере участь у розробці та підтримці дистрибутива Linux через різні форуми, канали IRC та відкриту систему відстеження помилок. Щоб використовувати операційну систему для Raspberry Pi, просто завантажте відповідний файл образу Kali та скопіюйте його на SD-карту (мінімум 8 ГБ). Потім запусить образ на мінікомп'ютері за допомогою команди dd.

Система для пентестингу (тестування безпеки). На Pi 4 працює чудово завдяки підтримці моніторного режиму на вбудованому чипі Wi-Fi (з патчами)



або зовнішніх адаптерах. Ця система актуальна для спеціалістів ІБ. Звичайному користувачеві не потрібна.

Таблиця 3.2 Перелік переваг та недоліків

Переваги	Недоліки
Різні інтегровані засоби безпеки	Не підходить для початківців у Linux
Увімкнено тести безпеки комп'ютера/мережі	Порівняно високі вимоги до оперативної пам'яті

### **Windows 10 IoT Core**

У 2015 році Microsoft випустила свою першу операційну систему для пристроїв Інтернету речей, таких як Raspberry Pi (2 або 3), випустивши Windows 10 IoT Core (<https://developer.microsoft.com/en-us/windows/iot>).

Для ентузіастів та Raspberry Pi 4 ця система практично мертва. Microsoft змістила фокус уваги. Зараз спільнота запускає повноцінну Windows 10 або Windows 11 на ARM (проект Windows on Raspberry), яка надає справжній робочий стіл, чого IoT Core не вміла. Використовувати IoT Core зараз не має сенсу для більшості завдань. Зараз ентузіасти ставлять Windows 11 on ARM (повноцінну), але і вона працює на межі можливостей Pi 4

### **Core Ubuntu/ Ubuntu Desktop**

Ubuntu [51] роками вважається найпопулярнішим дистрибутивом Linux : Ubuntu однаково добре підходить як операційна система для домашніх комп'ютерів та серверів, а також використовується, наприклад, на Міжнародній космічній станції та є центральним операційним блоком марсохода BYOU. Програмне забезпечення, засноване на Debian та розроблене Canonical з 2004 року, характеризується головним чином своєю високою модифікацією та зручністю використання. Під назвою Ubuntu Core розробник опублікував варіант у 2014 році, який є мінімалістичною версією серверної версії, а також може використовуватися як операційна система Raspberry Pi. Подібний пакет вже був доступний з JeOS (Just Enough Operating System) починаючи з Ubuntu 8.04.

В цей час для Raspberry Pi 4 офіційно доступна повноцінна Ubuntu Desktop (версія з довгостроковою підтримкою — LTS). Більшість користувачів нададуть перевагу саме повній версії з інтерфейсом GNOME, а не урізаній версії Core, яка потрібна лише для промислових вбудованих рішень або серверів. Завантажити можна з ресурсу <https://ubuntu.com/download/raspberry-pi>

## RISC-OC

RISC OS [52], спочатку названа Arthur, — це операційна система, розроблена наприкінці 80-х років британською компанією Acorn для комп'ютера Archimedes на базі ARM. Відтоді RISC OS Open Limited (ROOL) відповідає за розробку програмного забезпечення, вихідний код якого знаходиться у вільному доступі з 2006 року. Власником є Castle Technology Ltd., і вони регулюють, серед іншого, продаж платної ліцензії на комерційне використання операційної системи. Оскільки вона була розроблена для використання на архітектурах ARM, RISC OS вже використовувалася на одноплатних комп'ютерах BeagleBoard та PandaBoard. З виходом Raspberry Pi вона швидко стала однією з найважливіших альтернатив Linux для роботи мінікомп'ютера.

Однак в цей час це швидше музейний експонат або хобі для гіків. Вона працює дуже швидко, але не підтримує сучасні стандарти (Wi-Fi та сучасні браузері працюють погано, немає багатопотоковості у звичному розумінні). Для реальної роботи на Raspberry Pi 4 не підходить.

## SARPi (Slackware ARM для Raspberry Pi)

Розроблений ще в 1993 році, Slackware є одним з найстаріших дистрибутивів Linux, який досі активно підтримується. Портування ARM, яке спочатку було доступне під назвою ARMedslack, а тепер відоме як **Slackware ARM**, також підходить як операційна система для Raspberry

Pi. Проект SARPİ ( **Slackware A RM** на **Raspberry Pi** ) допомагає зацікавленим сторонам встановлювати та налаштовувати систему.

Навіть після 20 років Slackware та його ARM-порт (з 2002 року) продовжують розвиватися. Встановлені компоненти забезпечують дистрибутиву, який регулярно розширюється новими пакетами, високий рівень **стабільності та безпеки**. Для встановлення файлу образу команда SARPİ рекомендує **SD-карту об'ємом щонайменше 16 ГБ**. Додаткову інформацію про системне програмне забезпечення, а також підтримку можна знайти в онлайн-посібниках Slackware.

Однак Slackware складна для новачка. На Raspberry Pi 4 існує безліч зручніших та сучасніших альтернатив. Використовується одиницями.

### **Arch Linux ARM**

У 2009 році команди розробників PlugApps та ArchMobile розпочали роботу над портом **мінімалістичного дистрибутива Linux Arch Linux** для процесорів ARM. Приблизно через рік було випущено перше видання для систем ARMv5, а потім версії для ARMv6 (2012), ARMv7 (2011) та ARMv8 (2015). З цієї причини рішення з відкритим кодом, відоме сьогодні під назвою Arch Linux ARM, підходить як операційна система для Raspberry Pi. Порт переносить основну філософію Arch Linux на одноплатний комп'ютер, де користувач є центром уваги та має **повний контроль над системою та відповідальність за неї**, спираючись на струнку базову структуру. Оновлення операційної системи виконуються за **циклом випуску**: замість випуску величезних оновлень у встановлені моменти часу, розробник випускає безперервні та регулярні менші пакети.

Arch Linux ARM [54] спирається на **програму-пакет Pacman**, яка була розроблена спеціально для оригінального дистрибутива. Починаючи з версії 4 (2011), менеджер програмного забезпечення також підтримує імпортовані пакети, щоб **можна було перевірити справжність завантажених пакетів**.

Офіційні нові пакети від команди Arch Linux отримують свій **цифровий підпис** під час процесу створення, перед тим як залишити безпечне середовище розробки, щоб запобігти втручанню та забезпечити максимальну безпеку. Використовуючи одну команду «`pacman-Syu`», переконайтеся, що операційна система Raspberry Pi має найактуальніше оновлення. **Репозиторій спільноти користувачів Arch Linux (AUR)**, який містить програми, створені користувачами, можна використовувати для розширення діапазону пакетів.

Щоб встановити Arch Linux ARM, просто завантажте відповідний файл `tar.gz` для вашої версії Raspberry, а потім розпакуйте його на відформатовану SD-карту (мінімум 2 ГБ).

Ця операційна система використовується тільки для просунутих користувачів. Однак на її базі будується Manjaro ARM, яка зараз навіть популярніша за чистий Arch на Pi 4 через зручність встановлення

## **FreeBSD**

Випущена в 1993 році як похідна від відомого дистрибутиву програмного забезпечення Berkeley (BSD), який вперше застосував підхід вільної операційної системи завдяки власній ліцензії BSD, FreeBSD (<https://www.freebsd.org/>) досі залишається одним з найважливіших проєктів з відкритим кодом. Близько 400 офіційно зареєстрованих розробників, а також тисячі інших учасників активно працюють над подальшим розвитком програмного забезпечення FreeBSD, яке характеризується своїми функціями безпеки та зберігання, а також першокласними мережевими можливостями. Завдяки підтримці архітектур ARMv6 та ARMv7, похідна BSD також може використовуватися як операційна система Raspberry Pi (1 та 2). У майбутньому третя версія мінікомп'ютера також має бути сумісною з поточною версією FreeBSD.

Завдяки своїм сильним сторонам у мережевій функціональності та стабільності, FreeBSD в основному використовується в серверному

середовищі. Як користувач, ви також отримаєте вигоду від швидкості операційної системи, яка в першу чергу зумовлена підсистемою зберігання даних, яка була переглянута у версії 10.0. Завдяки добре документованому API, операційну систему Raspberry Pi можна оптимально модифікувати відповідно до індивідуальних потреб, а також розширювати власними програмними компонентами. У разі виникнення проблем або питань, різні форуми та блоги спільноти FreeBSD надають підтримку від інших користувачів та розробників. Також існує постачальник, який пропонує комерційну підтримку системи. Щоб використовувати системне програмне забезпечення, необхідно створити завантажувальний образ та скопіювати його на SD-карту. Для цього доступний інструмент Crochet.[55]

Чудово працює на Pi 3, але екосистема Linux (драйвери входів-виходів загального призначення GPIO, камери, плати розширення) розвинена краще у інших ОС. Використовується переважно системними адміністраторами-ентузіастами.

Таблиця 3.3 Перелік переваг та недоліків

Переваги	Недоліки
Дуже швидкий, стабільний та ресурсоефективний	Поточна версія доступна лише для Raspberry Pi 3.
Величезна, активна спільнота	Відсутність драйверів

## RetroPie

Операційна система RetroPie [56], що за замовчуванням базується на Raspbian та різних інших програмних компонентах, перетворює ваш Raspberry Pi на ігрову консоль, на якій є можливість грати в улюблені консольні та класичні ігри для ПК. EmulationStation, написана на C++, дозволяє змінювати дизайн та макет інтерфейсу за допомогою готових тем. Програмне забезпечення RetroArch надає доступ до Libretro-API, за допомогою якого є можливість налаштувати керування різними емуляторами (доступно понад 50 систем) та додавати корисні функції

за потреби. Завдяки Kodi, дистрибутив Raspberry Pi також надає власний медіацентр, який дозволяє відтворювати фільми або музику на мінікомп'ютері.

Щоб встановити RetroPie на ваш Raspberry Pi, у вас є два варіанти: або встановити окремі компоненти вручну на попередньо встановленій Raspbian або іншій операційній системі Debian, або отримати доступ до запропонованих файлів образів, розпакувати їх та скопіювати на SD-карту. Детальний посібник з встановлення та налаштування, а також найважливіші посилання для завантаження можна знайти в офіційному каталозі GitHub (<https://github.com/retropie/retropie-setup/wiki/First-Installation>) операційної системи Raspberry Pi. Якщо виникнуть проблеми з встановленням або подальшим запуском системи, то форум RetroPie зазвичай пропонує рішення: Тут є можливість знайти не лише загальні обговорення та повідомлення про систему, але й розділ підтримки, де є можливість поспілкуватися з іншими користувачами RetroPie.

Таблиця 3.4 Перелік переваг та недоліків

Переваги	Недоліки
Швидкі реакції на нові релізи Raspberry Pi	Слабка документація окремих компонентів програмного забезпечення
Інтегрований медіацентр (Kodi)	Розширюваний варіант контролера

### **Home Assistant Operating System (HAOS)**

Статус: Спеціалізована ОС для автоматизації та розумного будинку

Home Assistant OS (HAOS) — це мінімалістична операційна система на базі Linux, розроблена виключно для запуску платформи Home Assistant. Вона перетворює Raspberry Pi на повноцінний хаб ("сервер") розумного будинку. Система керує всіма оновленнями, безпекою та дозволяє встановлювати додаткові програми (Add-ons) в один клік.

Процес встановлення на Raspberry Pi 4:

1. Завантажте та встановіть офіційну утиліту Raspberry Pi Imager на ваш комп'ютер.
2. Вставте карту пам'яті microSD (рекомендовано мінімум 32 ГБ, клас A2) або SSD-диск (для кращої швидкодії та надійності).
3. В Imager натисніть "CHOOSE OS" (Обрати ОС) -> "Other specific-purpose OS" (Інші спеціалізовані ОС) -> "Home assistants and home automation" -> "Home Assistant".
4. Оберіть версію для **Raspberry Pi 4**.
5. Натисніть "CHOOSE STORAGE" (Обрати сховище), виберіть вашу карту/диск і натисніть "WRITE" (Записати).
6. Вставте карту в Raspberry Pi, підключіть кабель Ethernet (важливо для першого запуску) і живлення.
7. Зачекайте 10-20 хвилин. Перейдіть у браузері за адресою `http://homeassistant.local:8123` для налаштування.

Посилання для завантаження ОС:

- Офіційна сторінка: Home Assistant Installation (<https://www.home-assistant.io/installation/raspberrypi/> )
- GitHub Releases: HAOS Releases (<https://github.com/home-assistant/operating-system/releases> ) найкраще використовувати Raspberry Pi Imager, він завантажує актуальну версію автоматично).

Слід відзначити певні переваги та недоліки цієї ОС

Переваги:

- Простота (Plug & Play): Не вимагає знання команд Linux.
- Магазин доповнень (Add-on Store): Дозволяє легко встановити MQTT брокер, Node-RED, File Editor, Google Drive Backup тощо.
- Керованість (Supervisor): Система сама стежить за оновленнями ядра та додатків.
- Повне рішення: Включає все необхідне "з коробки".

Недоліки:

- **Закритість:** Це система "чорна скринька". Ви не маєте повного доступу до терміналу Linux (тільки обмежений CLI), що може не сподобатися досвідченим адміністраторам.
- **Обмеження:** Складно запустити сторонній софт, якого немає в магазині Add-ons (потрібно створювати власні контейнери).

Аналіз складності викоричтання з описів у мережі Інтернет показує, що це найпростіший спосіб увійти у світ серйозної автоматизації. Встановлення легше, ніж інсталяція Windows. Весь інтерфейс графічний та перекладений українською мовою.

### Аналіз рейтингів різноманітних ОС

В цілому, комплексна оцінка різноманітних ОС наведена у таблиці 3.5. На засадах проведеного аналізу можна стверджувати (рис. 3.1, 3.2):

- до найбільш поаулярних Ос відносяться Raspbery PI OS та UBUNTU;
- для попереднього знаймства та початкового використання краще використовувати Raspbery PI OS;
- для швидкого знайомства с технологією розумного будинки – Home Assistant Operating System;
- для професійного використання – UBUNTU



Рис. 3.1 Рейтинг ОС



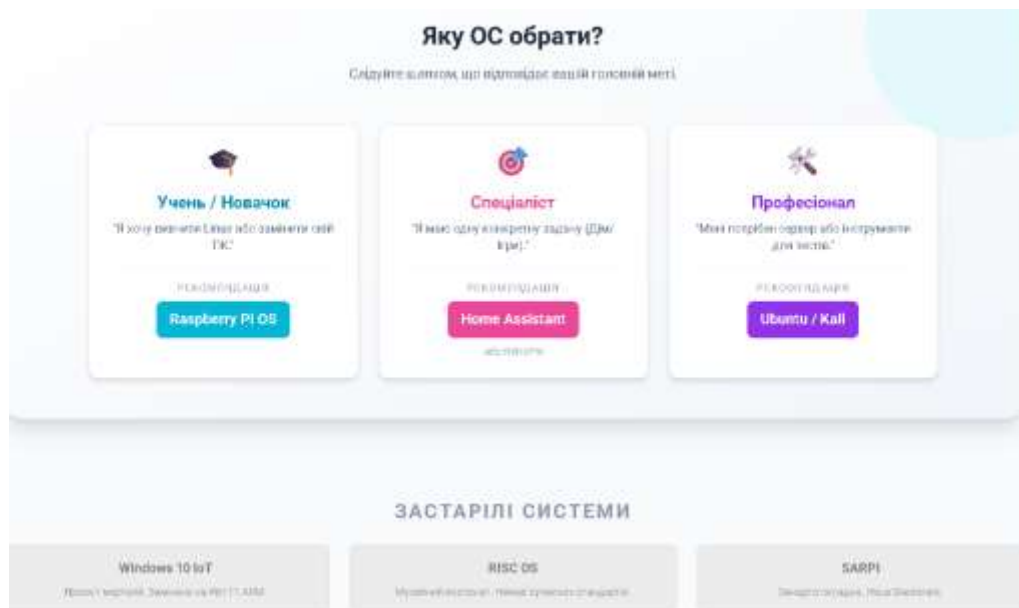


Рис. 3.2 Вибір ОС

Табл.3.5 Зведена інформація операційних систем

Назва ОС	Особливості ОС	Актуальність	Платна / Безкошт.	Зручність для навчання	Коментар
<b>Raspberry Pi OS</b>	Офіційна, стабільна, легка.	★★★★★	Безкоштовна	✓ Висока (вбудовані Python, Scratch)	Найкращий вибір для старту.
<b>Ubuntu</b>	Сучасна, популярна, 64-біт.	★★★★★	Безкоштовна	✓ Висока (для студентів IT/CS)	Чудова для сервера або як заміна ПК.
<b>Home Assistant OS</b>	Тільки для керування розумним будинком.	★★★★★ (ніша)	Безкоштовна	⚠ Тільки профільна (IoT/Автоматизація)	Стандарт індустрії DIY-автоматизації.
<b>RetroPie</b>	Ігровий комбайн (емулятори).	★★★★★ (ніша)	Безкоштовна	✗ Низька (розваги)	Найкраще для ретро-ігор.
<b>Kali Linux</b>	Інструменти для хакерів/безпеки.	★★★ (ніша)	Безкоштовна	⚠ Тільки профільна (Кібербезпека)	Звичайному користувачеві не потрібна.
<b>Arch Linux ARM</b>	Конструктор для профі.	★★	Безкоштовна	⚠ Тільки для поглибленого вивчення Linux	Складно, для досвідчених користувачів.
<b>Windows 10 IoT</b>	Урізана Win без робочого столу.	☠ Не актуальна	Безкоштовна (Core)	✗ Ні	Проект закритий, не має сенсу.
<b>RISC OS</b>	Дуже стара, незвичайна ОС.	☠ Не актуальна	Безкоштовна	✗ Ні (лише історія обчислювальної техніки)	Тільки як музейний експонат.

<b>Slackware / SARP i</b>	Найстаріший Linux, складний.	☠ Не актуальна	Безкоштовна	✗ Ні	Занадто складно.
---------------------------	------------------------------	----------------	-------------	------	------------------

### 3.2 Розгортання Ubuntu

Найпростіший спосіб – використовувати Raspberry Pi Imager [57], який дозволяє вибрати образ Ubuntu під час прошивки SD-карти.

Якщо користуєтеся Ubuntu, відкрийте термінал і виконайте команду:

```
sudo snap install rpi-imager
```

Альтернативним засобом є натискання клавіші «Shift» під час старту операційної системи Ubuntu



Рис. 3.3 Розгортання Raspberry PI

#### Ubuntu 24.04.2 LTS

Нова версія Ubuntu LTS для настільних ПК та ноутбуків. LTS означає довгострокову підтримку, що означає п'ять років безкоштовних оновлень безпеки та обслуговування, продовжених до 12 років з Ubuntu Pro.

#### Ubuntu 25.04

Найновіша версія операційної системи Ubuntu для настільних ПК та ноутбуків, Ubuntu 25.04, постачається з дев'ятьма місяцями оновлень безпеки та обслуговування, до січня 2026 року.

Core Ubuntu 24

Незмінна, суворо обмежена операційна система, розроблена для розгортання сценаріїв використання Інтернету речей (IoT) з акцентом на безпеку та спрощене обслуговування, що підтримується до 2036 року.

Підтримувані пристрої

Сертифіковані пристрої проходять тестування на надійність та продуктивність, що гарантує найкращий досвід роботи з Ubuntu одразу після встановлення. Версії Ubuntu LTS сертифіковані на вибраному обладнанні Raspberry Pi.

Таблиця 3.6 Пристрої Raspberry PI – сертифікація UBUNTU

		UBUNTU 24.04 LTS	UBUNTU 25.04	CORE UBUNTU 24
Raspberry Pi 5		Так, сертифіковано	Так	Так, сертифіковано
Raspberry Pi Zero 2 Вт		Так	Так	Так, сертифіковано
Raspberry Pi CM4		Так, сертифіковано	Так	Так, сертифіковано
Raspberry Pi 400		Так, сертифіковано	Так	Так, сертифіковано
Raspberry Pi 4		Так, сертифіковано	Так	Так, сертифіковано

Для встановлення нової версії UBUNTU – спочатку вставте microSD картку в комп'ютер. Включить живлення та натисніть клавішу «shift». З'явиться вікно завантаження. Оберіть пункт меню « CHOOSE OS».

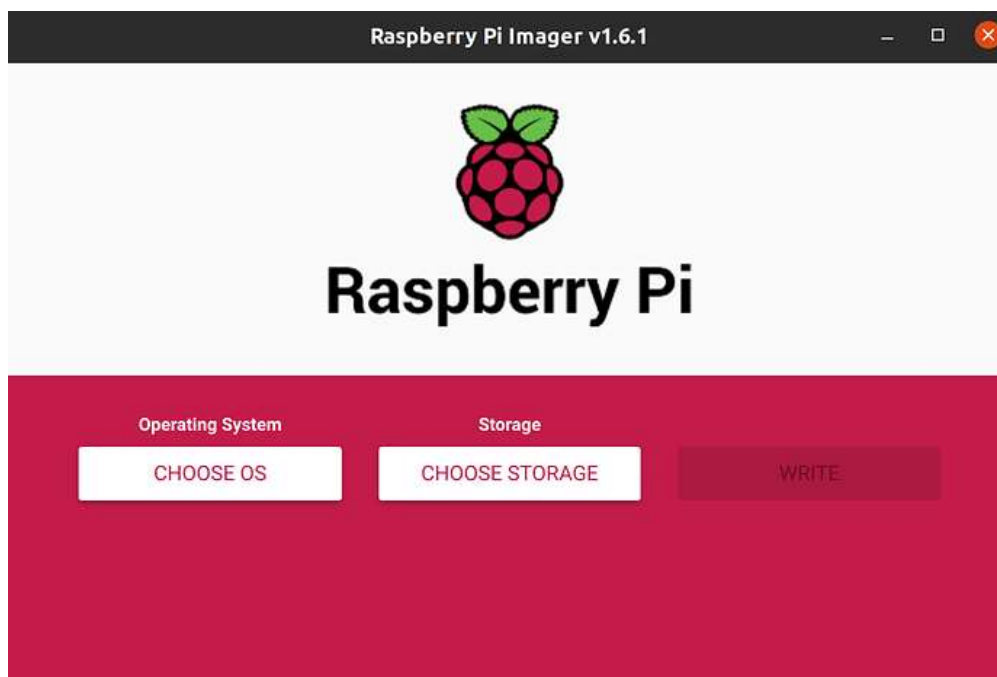


Рис. 3.4 Розгортання Raspberry Pi

Прокрутка вниз меню натисніть « Інше » загального призначення ОС ».



Рис. 3.5 Розгортання Ubuntu

Тоді буде можливість до переглянути список з Ubuntu завантажень до вибрати від. Виберіть « Ubuntu 20.10 Desktop (Raspberry Pi 4)». Як зазначено в той/та/те зображення це тільки працює для той/та/те the Raspberry Pi 4 з 4 ГБ або 8 ГБ оперативної пам'яті.

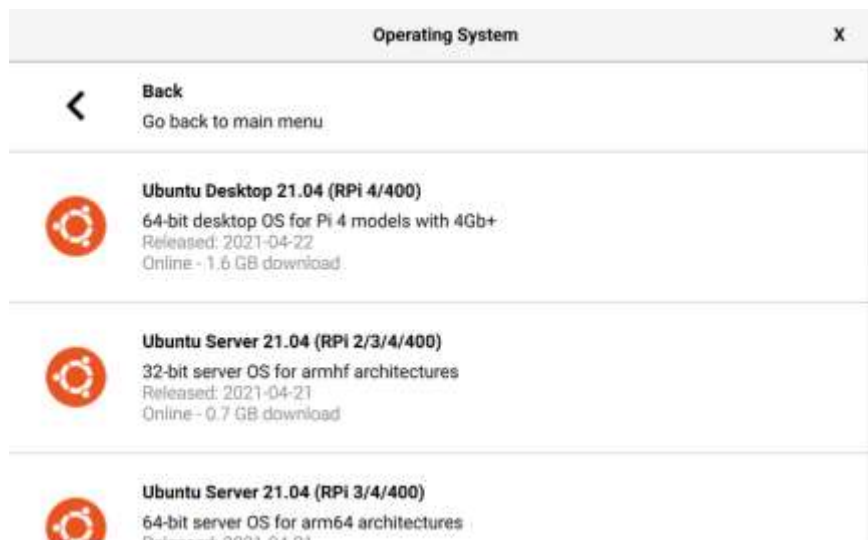


Рис. 3.6 Вибір версії Ubuntu

Виберіть Меню «SD Card ». Виберіть необхідний microSD картка, яку повинно бути вставленою, і натисніть кнопку « WRITE». Потім просто сидіти назад і чекати поки йде завантаження на налаштування та завантаження Робочого столу

Спочатку встановіть свою мову :



Рис. 3.7 Налаштування мови версії Ubuntu

Тоді виберіть макет клавіатури. Для Британський проти Американець клавіатури ти може використання клавіші '@' та '“' щоб перевірити що вони є в необхідному місті клавіатури.

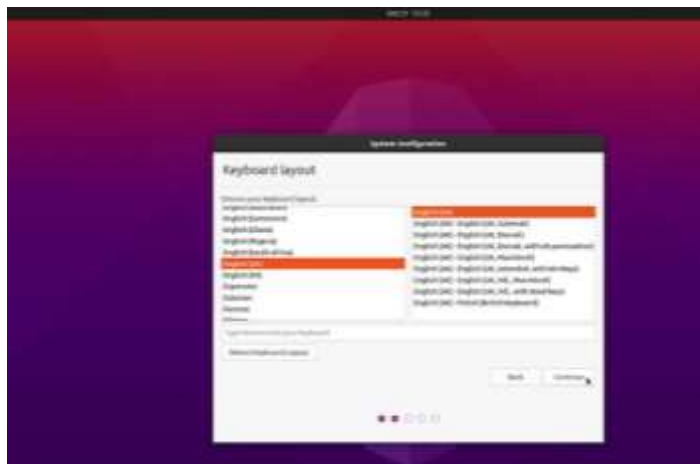


Рис. 3.8 Налаштування клавіатури Ubuntu

Оберіть собі часовий пояс та правильний час і також чи є зміна для періоду – літо:

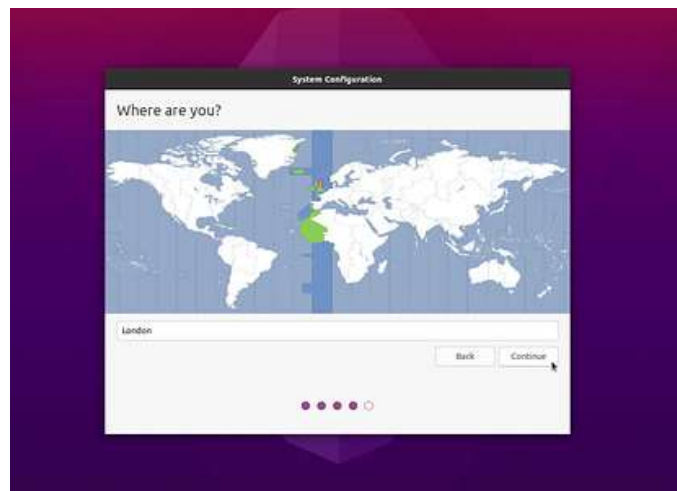


Рис. 3.9 Налаштування часу Ubuntu

Введіть набір для ROOT користувача і встановіть пароль

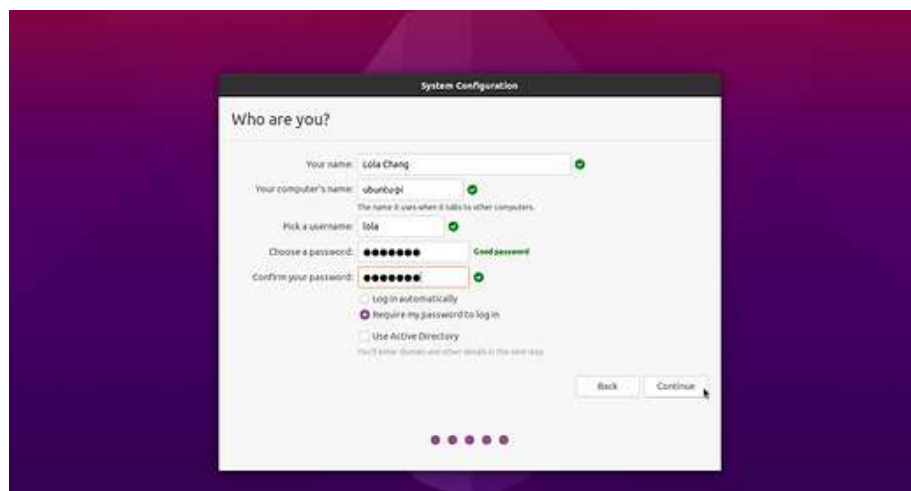


Рис. 3.10 Налаштування головного користувача Ubuntu

І нарешті чекайте завантаження та інсталювання компонентів Ubuntu  
Робочий стіл :

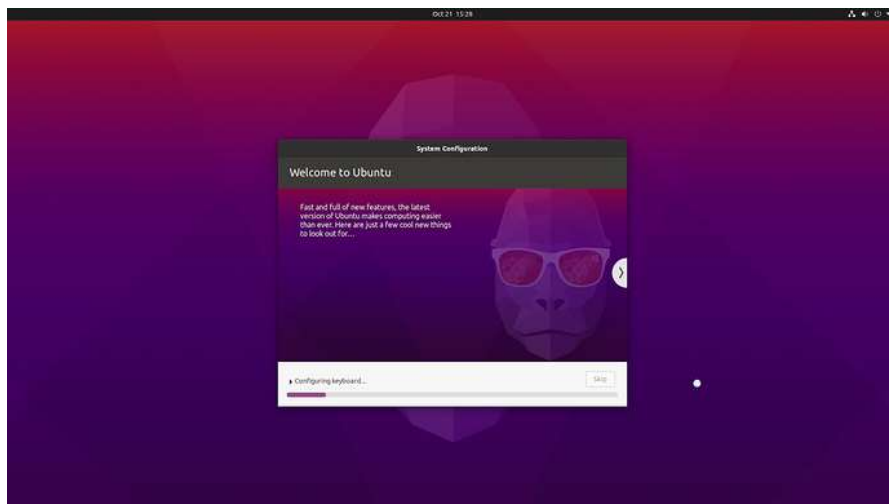


Рис. 3.11 Завершення інсталяції Ubuntu

### 3.3 Вступ до віддаленого доступу

Іноді потрібно отримати доступ до Raspberry Pi, не підключаючи його до монітора, клавіатури та миші. Можливо, Raspberry Pi вбудований у робота або встановлений у незручному місці. Або, можливо, у вас немає запасного монітора [58].

Щоб дистанційно керувати Raspberry Pi з іншого пристрою у вашій локальній мережі, скористайтесь типовою системною службою SSH.

Більш ґрунтовний аналіз наукової літератури та джерел Інтернет дозволив знайти інші додаткові програмні засоби:

- • Xrdp
- • NoMachine
- • **Gnome-remote**
- • X2go

Ці додаткові програмні засоби обкладають значною перевагою, але потребують окремого дослідження. Тому прийнято рішення, в межах цієї роботи розглянути тільки **Gnome-remote**, який стандартно є в налаштуваннях **UBUNTU**, можливості інших програмних засобів розглянути в межах наступного магістерського дослідження

SSH ( Secure Shell ) забезпечує безпечний доступ до термінального сеансу на вашому Raspberry Pi. VNC ( Virtual Network Computing ) забезпечує безпечний доступ до спільного доступу до екрана робочого столу на Raspberry Pi. Все, що потрібно, це інший комп'ютер, локальна мережа та локальна IP-адреса вашого Raspberry Pi. Raspberry Pi Connect безпечно надає доступ до екрана вашого Raspberry Pi без необхідності визначення вашої локальної IP-адреси.

### **3.4 Інтеграція списку користувачів з інформаційною системою**

Одна з багатьох можливостей отримати доступ до терміналу Raspberry Pi віддалено з іншого комп'ютера з виростанням єдиного, встановленого у інституті математики та інформаційних технологій імені користувача та відповідного паролю – це включення даного мікрокомп'ютеру до домену Microsoft AD

Встановлено, що для цього слід виконати декілька додаткових налаштувань. Слід відзначити, що у останніх версіях операційної системи Ubuntu (24 та 25) працює доволі стабільно. Всі налаштування необхідно виконати від імені користувача SUDO.

Перевести комп'ютер на статичну адресу (рис. 3.5 та 3.6)

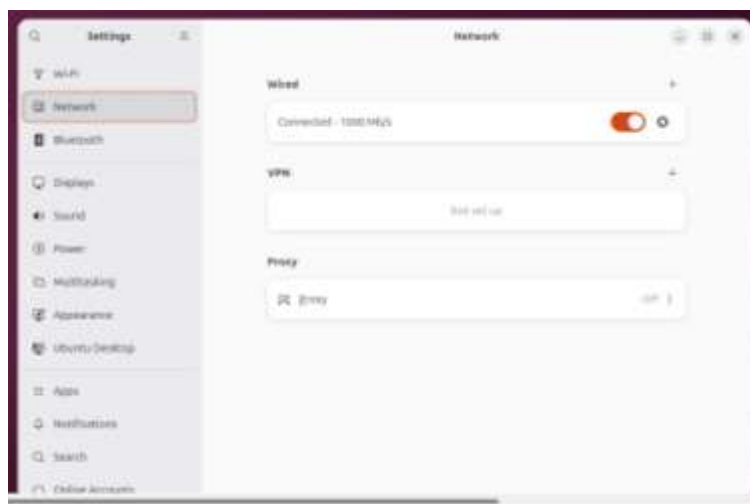


Рис. 3.12 Вікно налаштувань мережі



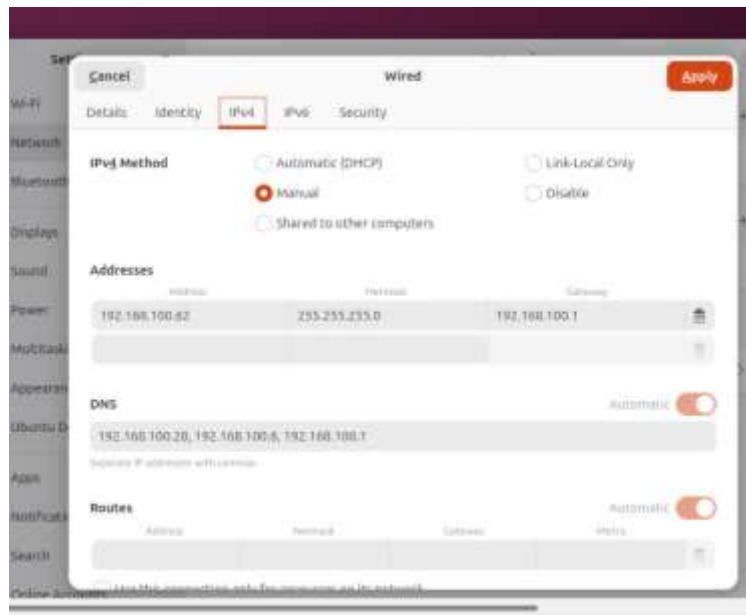


Рис. 3.13 Налаштування статичної IP адреси

Необхідно налаштувати ім'я комп'ютеру у файлі (рис.3.7)

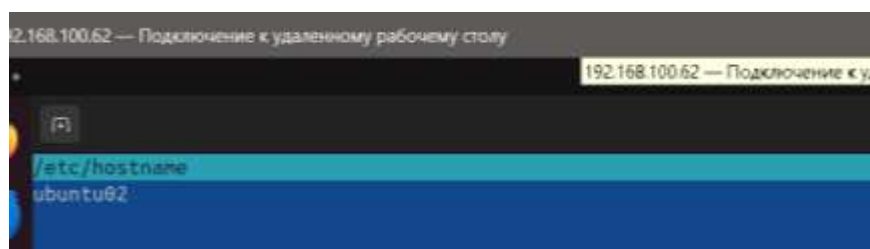


Рис. 3.14 Налаштування імені комп'ютеру

Додати DNS основних інформаційних ресурси до файлу (рис. 3.8)

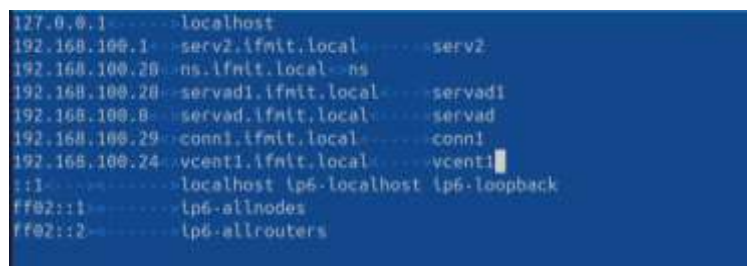


Рис. 3.15 Налаштування DNS для основних ресурсів

Встановити додаткові системні пакети до операційної системи за допомогою наступної команди

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install realmd sssd oddjob oddjob-mkhomedir adcli samba-common
```

Перевірити можливість інтеграції з доменом за допомогою команди

```
sudo realm -v discover <ім'я домену>
```

```
ubuntu@ubuntu02:~$ sudo realm -v discover ifmit.local
[sudo] password for ubuntu:
* Resolving: ldap._tcp.ifmit.local
* Performing LDAP DSE lookup on: 192.168.188.28
* Successfully discovered: ifmit.local
ifmit.local
type: kerberos
realm-name: IFMIT.LOCAL
domain-name: ifmit.local
configured: kerberos-member
server-software: active-directory
client-software: sssd
required-package: sssd-tools
required-package: sssd
required-package: libnss-sss
required-package: libpam-sss
required-package: adcli
required-package: samba-common-bin
login-formats: NtlmIfmit.local
login-policy: allow-realm-logins
ubuntu@ubuntu02:~$
```

Рис. 3.16 Перевірка налаштувань для підключення до домену

Провести інтеграцію до домену IFMIT

```
sudo realm join -v -U <ім'я_адміну_домену> ifmit.local
```

Скоригувати процедуру аутентифікації до комп'ютеру

```
sudo pam-auth-update
```

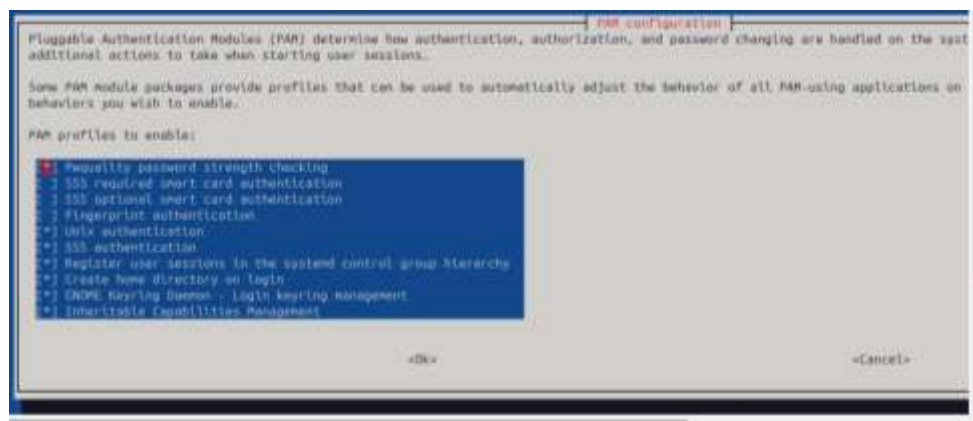


Рис. 3.17 PAM configurator

Перевірити, що операційна система «бачить» користувачів домену

```
getent passwd stud1@ifmit.local
```

```
ubuntu2@ubuntu02:~$ sudo pam-auth-update
ubuntu2@ubuntu02:~$ sudo getent passwd stud1@fntt.local
stud1@fntt.local:*:1400001122:1400000513:stud1:/home/stud1@fntt.local:/bin/bash
ubuntu2@ubuntu02:~$
```

Рис. 3.18 Перевірка результатів підключення до домену Ms AD

Перезавантажити комп'ютер за допомогою команди

```
sudo reboot
```

### 3.5 Підключення по протоколу SSH

Для встановлення SSH серверу на навчальні Raspberry PI з ОС Raspberry PI OS, UBUNTU необхідно відкрити вікно програми terminal еф подати наступні команди:

```
su -l <root_name>
;; ввести пароль для користувача root_name, що входить до групи sudoers
sudo apt-get install ssh
```

Аналіз використання ОЗУ показав наступне на ОС UBUNTU

- 1 користувач – 1.6 ГБ;
- 2 користувача – 2.3 ГБ;
- Кожен наступний – приблизно +0.6ГБ.

Для підключення до Raspberry PI 4 з використання протоколу SSH необхідно відкрити вікно терміналу на комп'ютері та введіть таку команду, замінивши <ip address> тимчасове поле IP-адресою Raspberry Pi, до якого намагаєтеся підключитися, та <username> своїм ім'ям користувача:

```
$ ssh <username>@<ip address>
```

Коли з'єднання запрацює, ви побачите попередження безпеки. Введіть код yes, щоб продовжити. Ви побачите це попередження лише під час першого підключення.

Введіть пароль свого облікового запису, коли буде запропоновано.

Тепер необхідно побачити командний рядок Raspberry Pi:

```
<username>@<hostname> ~ $
```

Тепер створено підключення до Raspberry Pi віддалено та можете виконувати команди.

### **3.6 Підключення за допомогою Gnome-remote**

Аналіз роботи на Raspbery PI4 при використанні ОС UBUNTU це швидкий спосіб використання віддалено доступу до навчальних Raspbery PI.

Слід зауважити, що ця можливість є тільки у останніх версіях ОС (рис. 3.19).

Треба увійти у налаштування (див. рис. 3.20-3.22), відкрити System потім закладку Remote Login та встановити ім'я користувача та пароль для віддалено підключення. Зробити це можна тільки за умови входу у систему від ROOT користувача (або скористатися кнопкою Unlock. Крім того змінники ці налаштування НЕ МОЖЛИВО у віддаленому режимі, тобто треба використовувати безпосередню за комп'ютером.



Рис. 3.19 Версія ОС UBUNTU



Рис. 3.20 Вхід до налаштувань



Рис.3.21 Властивості системи

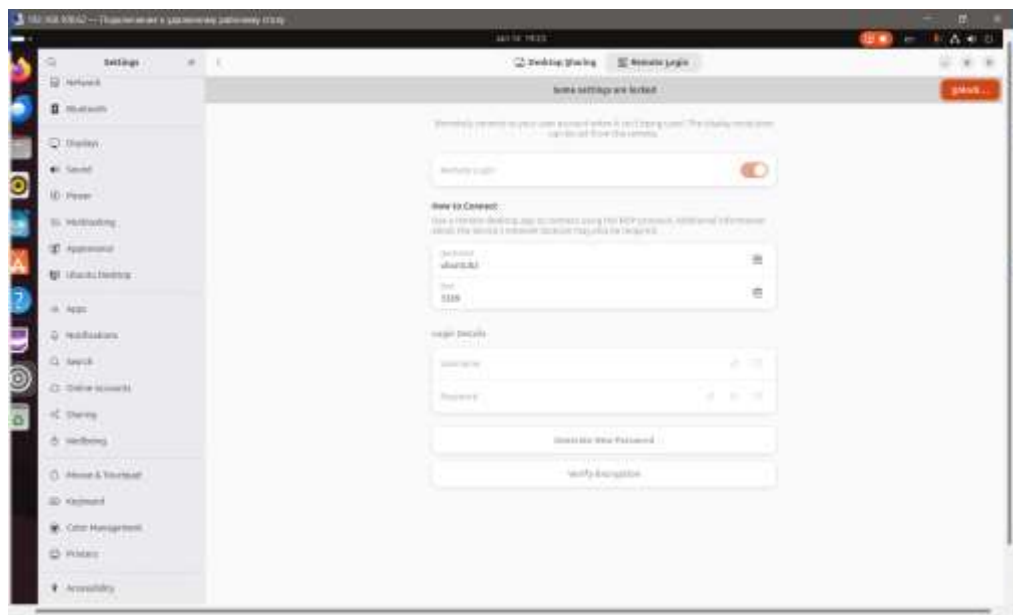


Рис. 3.22 Закладка Remote Login

Для підключення до цього режиму можна використовувати стандартного клієнта «Віддалений робочий стіл».

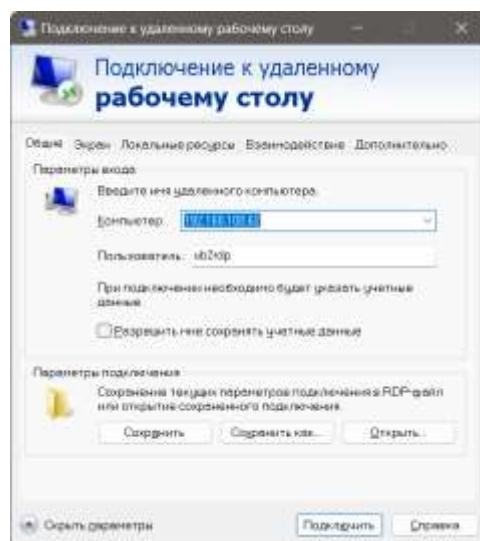


Рис.3.23 Клієнт Віддалено робочого столу Windows

Після підключення буде видано перелік користувачів (рис. 3.24), які останнім часом використовували цю Raspberry PI, треба обрати певного користувача (або обрати – не має у переліку) потім ввести відповідний пароль.

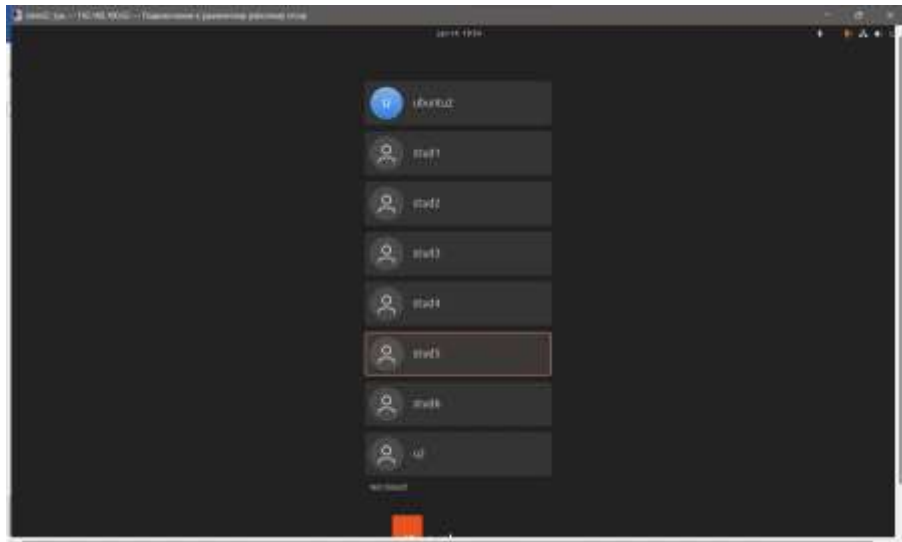


Рис. 3.24 Перелік користувачів

В процесі використання встановлені певні особливості використання клієнту «Віддалений робочий стіл»:

1. У закладці «Екран» треба ретельно підбирати параметри роздільної здатності екрану. В деяких випадках, а особливо у режимі «На весь екран» підключення не відбувається.
2. Треба відредагувати службовий файл, що створюється за допомогою кнопки «Зберегти» знайти строку: `use redirection server name:i:0` та змінити на `use redirection server name:i:1` (рис. 3.25).



Рис. 3.25 Зміна параметрів підключення

### **3.7 Підключення зовнішніх користувачів через мережу**

#### **Інтернет**

У цьому розділі розглянемо основні шляхи, які необхідно виконати для забезпечення роботи навчальної лабораторії при виконанні лабораторних робіт студентами при змішаній формі навчання. Основна мета цих налаштувань – забезпечення виконання завдань у домашньому режимі.

Аналіз різноманітних варіантів дозволив виділити наступні шляхи :

1. Організація доступу до локальної мережі шляхом використання VPN підключення на мережевому, пороговому роутері та використання додаткового командного файлу
2. Використання додатково RDP підключення на засадах ОС Microsoft Windows Server та використання того же додаткового командного файлу.
3. Прокидання певних портів на IP адреси навчальних стендів лабораторії та створення модифікації додаткового командного файлу.

Розглянемо певні особливості кожного з перелічених шляхів.

#### **3.7.1 Варіант1 – VPN**

У такому випадку, на зовнішньому, пороговому роутері активується VPN сервер. VPN сервер через систему RADIUS перевіряє користувача та приєднує до локальної мережі навчального закладу. Студент запускає додатковий файл автоматизації підключення до певних RaspberryPI (лабораторних стендів), який йому попередньо надається( рис.3.26).

Існує багато варіантів VPN підключень в межах цієї роботи розглянемо варіант створення VPN типу L2TP на засадах роутеру MikroTik з ОС · Route OS v.6.49.10.





Рис. 3.26 Схема підключення через VPN

В межах цієї роботи **НЕ РОЗГЛЯДАЄМО** додаткові налаштування та створення доменної служби AD Microsoft, а також додаткової служби «Політика мережі».

Налаштування роутеру виконаємо за допомогою програми Wibox. Для цього запускаємо Windox, підключаємося до роутеру. Першим кроком створюємо пул адрес для майбутнього VPN підключення (рис. 3.27, 3.28), переходимо до меню IP->Pool.



Рис.3.27 Перехід до меню Pool

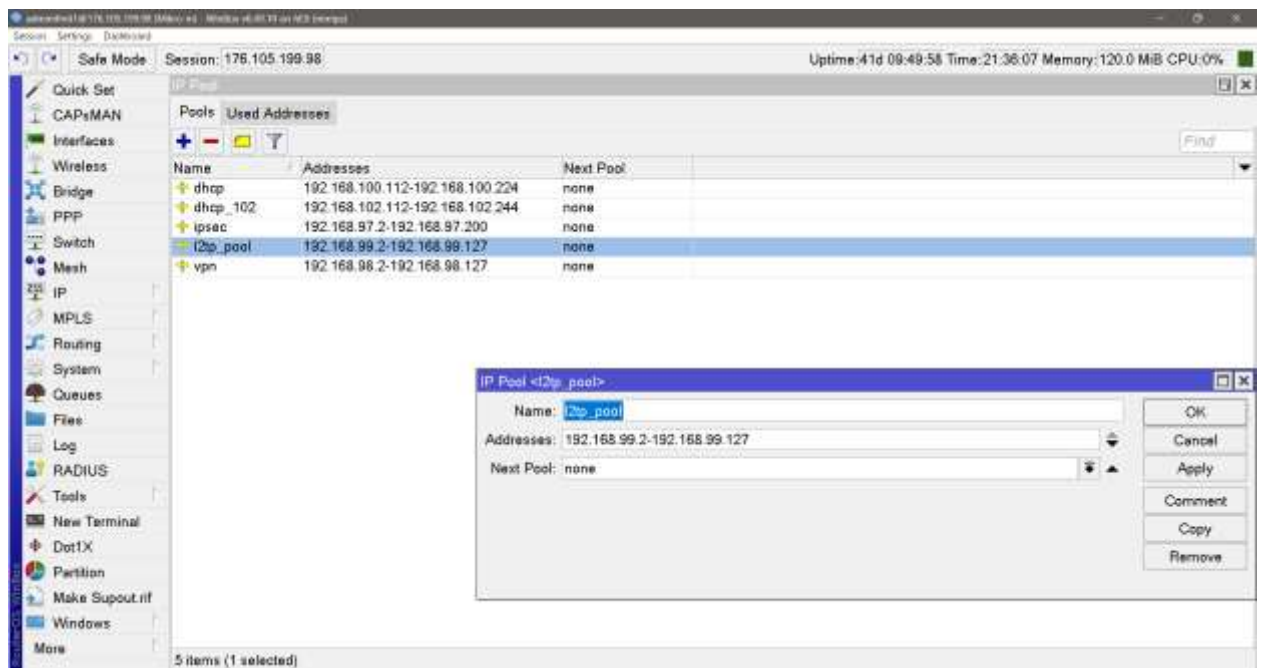


Рис.3.28 Створення пуду адрес для VPN

Переходимо до меню PPP -> Profiles (рис.3.29) та створюємо профіль підключення.

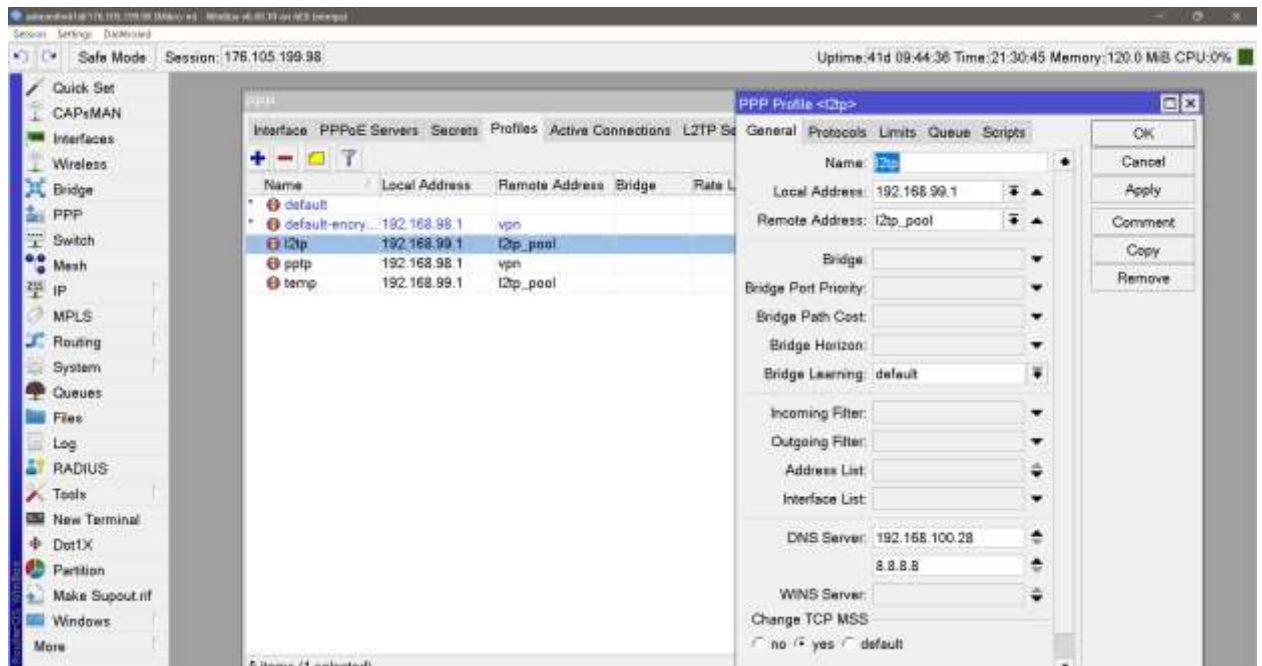


Рис. 3.29 Створення Profile для VPN підключення

Переходимо до закладки Interface та натискаємо кнопку L2TP Server (рис. 3.30), де налаштовуємо параметри VPN серверу типу L2TP.

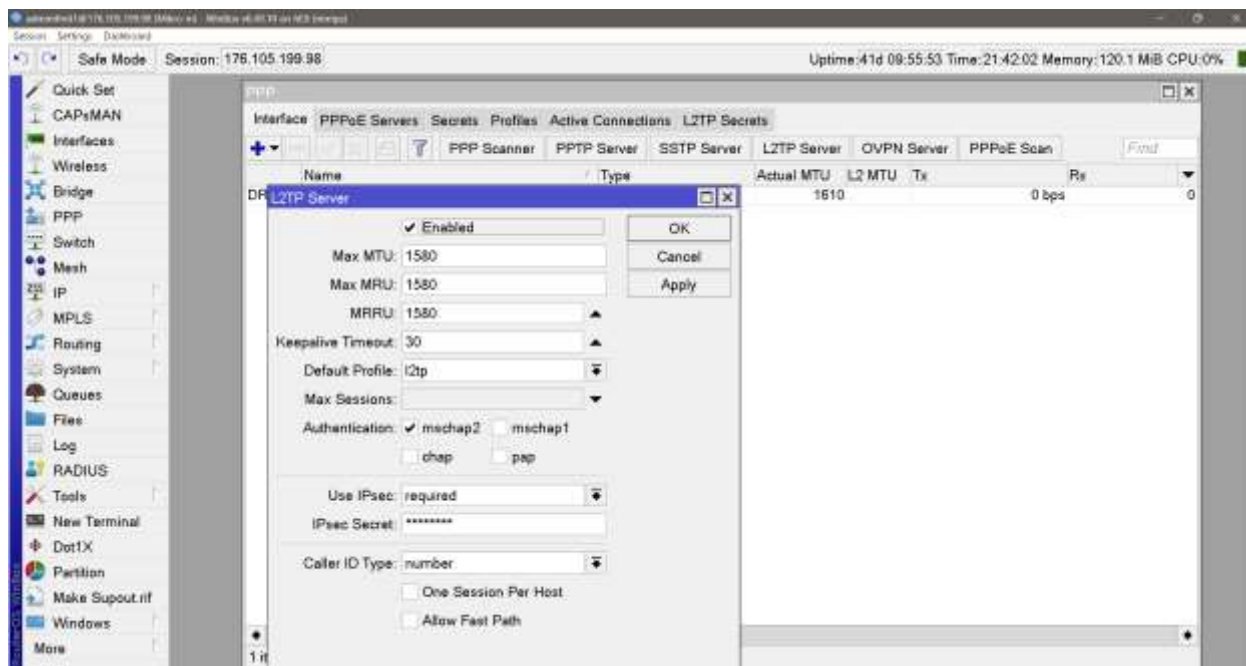


Рис. 3.30 Налаштовуємо параметри L2TP серверу

Переходимо у закладку Secrets та активуємо PPP Authentication&Accounting (Рис.3.31)

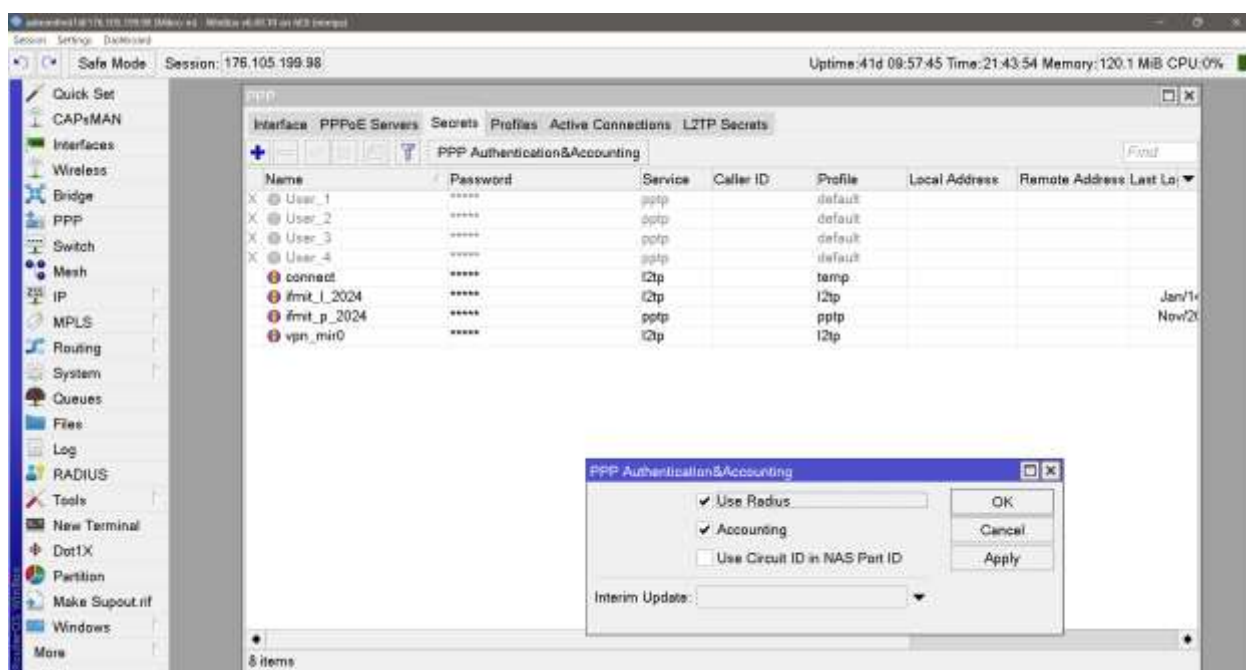


Рис.3.31 Активізація віддаленої аутентифікації

Обираємо меню RADIUS (рис.3.32) та додаємо параметри Radius серверу

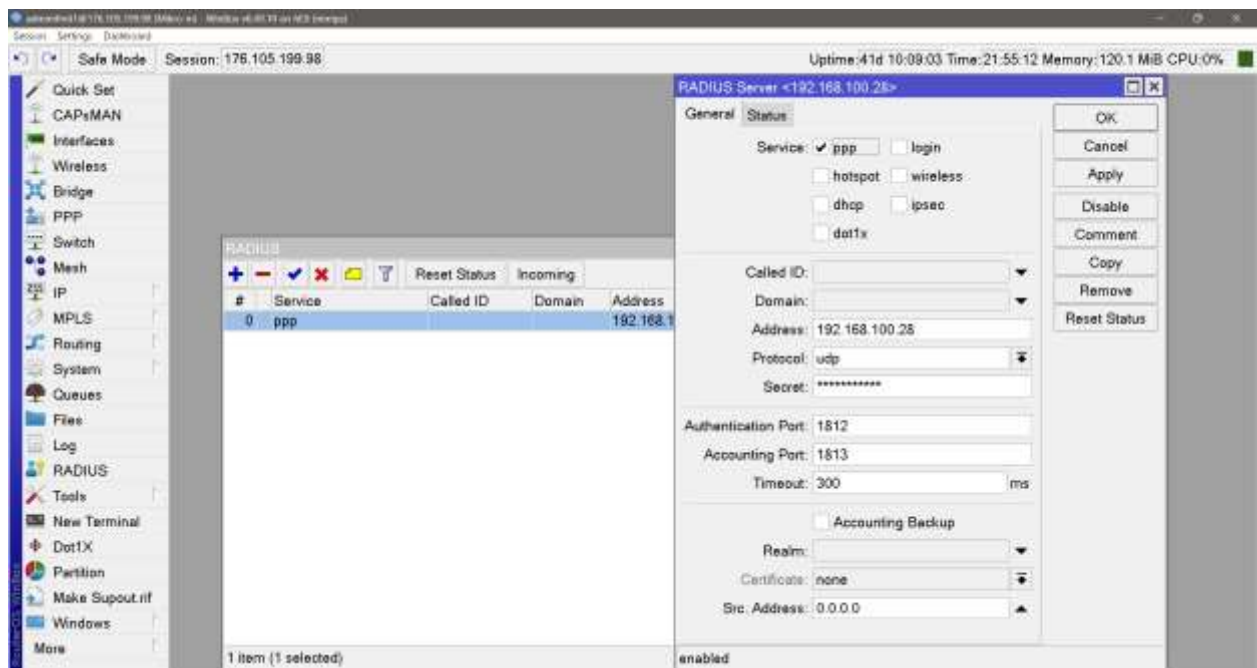


Рис. 3.32 Налаштування RADIUS підключення

Таким чином, після такого налаштування та налаштування служби «Сервер політики мережі» на домені Microsoft AD.

Однак, для повної автоматизації процесу використання RaspberryPI (лабораторних стендів IoT) зовнішніми користувачами необхідно розробити додатковий файл автоматизації підключення до певних RaspberryPI (лабораторних стендів).

Загальна структура такого файлу повинна включати (див. Додаток А):

- налаштування IP адрес та параметрів підключення,
- вивід на екран переліку RaspberryPI (навчальних стендів) та обирання навчального стенду,
- команд для підключення до певного стенду,
- завершення з ліквідацією паролів.

Приклад роботи цього скрипту (bat файлу) показано на рис. 3.33 та 3.34.

Суттєвим недоліком цього варіанту є те, що викладач розповсюджує файл автоматизації підключення і здобувачі освіти його випадково пошкоджують.

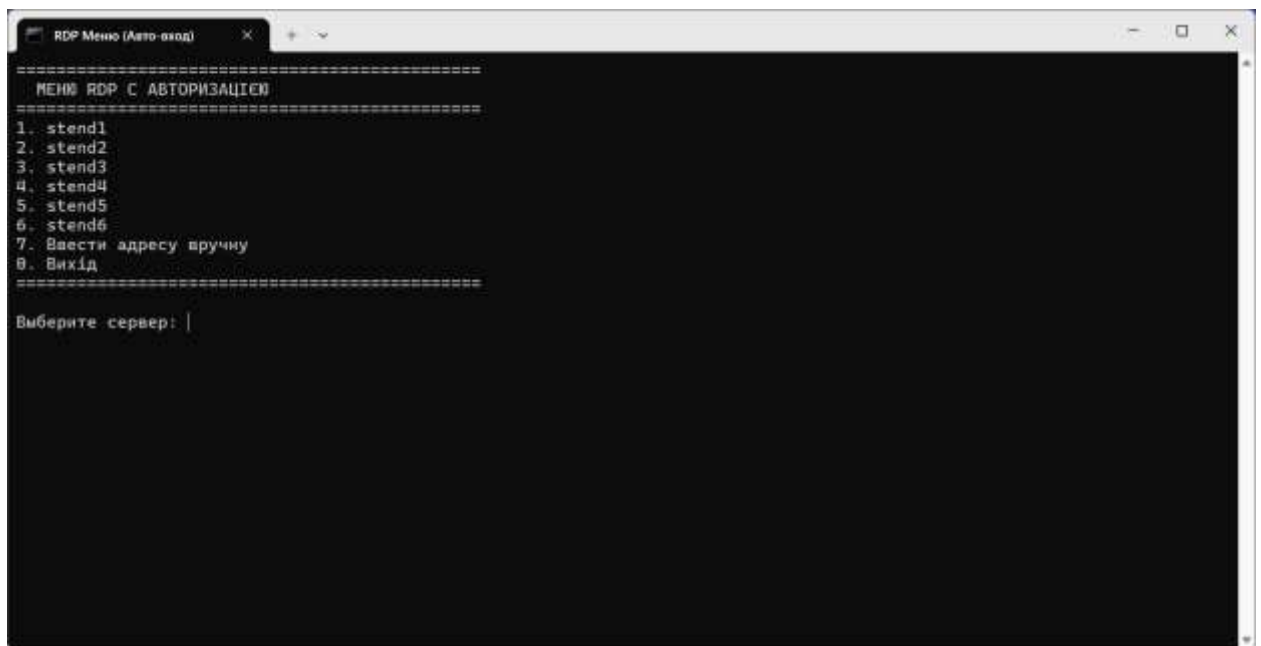


Рис. 3.33 Приклад роботи скрипту

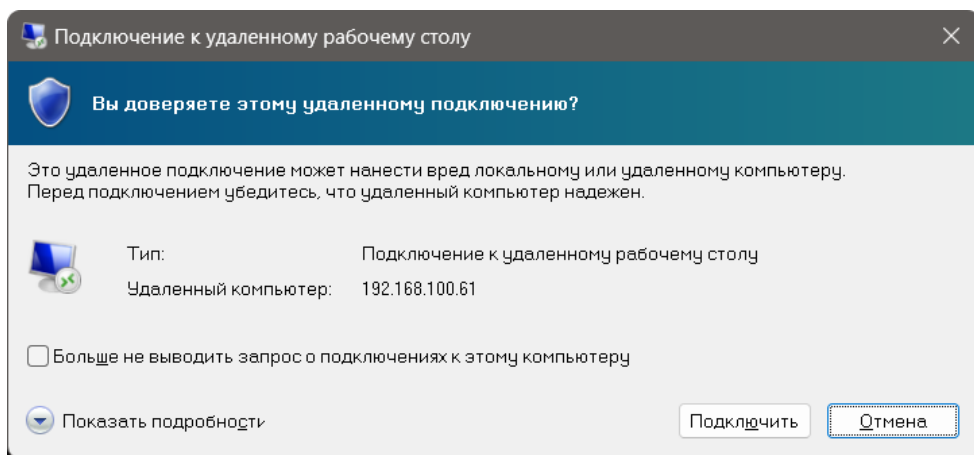


Рис. 3.34 Приклад роботи другої частини скрипту

### 3.7.2 Варіант2 RDP SERVER

Загальна схема такого використання виглядає наступним чином. Зовнішній користувач (здобувач освіти) приєднується до RDP серверу. Вважаємо, що в інформаційній системі є Microsoft Windows Server з активованою службою віддалених робочих столів, крім того налаштовано зовнішній доступ студентів (здобувачів освіти) до цього серверу, крім того, всі необхідні налаштування на пороговому роутері та брандмауері RDP серверу вже налаштовані. На сервері розташовано додатковий файл автоматизації підключення до певних RaspberryPI (лабораторних стендів), який студент запускає для підключення к певному лабораторному стенду (рис. 3.35).





Рис. 3.35 Схема підключення через RDP

Слід зауважити, що додатковий файл автоматизації підключення до певних RaspberryPI (лабораторних стендів) можна розташувати у певному каталозі, що доступен здобувачам освіти тільки на читання, наприклад: `c:\Users\Public\Desktop`, що унеможливить його випадкове пошкодження.

При цьому, не потрібно використовувати VPN підключення.

### 3.7.3 Варіант3 – прокидання портів роутера

Для такого варіанту підключення, на зовнішньому, пороговому роутері за допомогою `dst-nat` перенаправляються запити на певні порти (наприклад з 13391 по 13397 – для семи стендів). `Dst-nat` перенаправляє їх на певні локальні адреси (наприклад з 192.168.100.61 по 192.168.100.67) на порт служби `Gnome-remote RaspberryPI` (лабораторних стендів) – 3389. Студентам надаємо ім'я користувача та пароль (наприклад однакові до всіх стендів). Таке рішення потребує додаткових заходів безпеки на зовнішньому, пороговому роутері (рис. 3.36)



Рис.3.36 Підключення через прокидання портів

Для створення такого приєднання на роутері Mikrotik необхідно подати команду (для першого стенду):

```
/ip firewall nat add chain=dstnat action=dst-nat \
protocol=tcp dst-address=176.105.199.98 dst-port=13391 \
to-addresses=192.168.100.61 to-ports=3389 \
comment="RDP forward: 1391 -> 3389"
```

Однак даний варіант має суттєві недоліки з точки зору безпеки. Тому пропонується більш жорстке правило за допомогою команди, яка контролює перелік адрес, з яких підключаються користувачі до першого стенду:

```
/ip firewall nat add chain=dstnat action=dst-nat \
protocol=tcp dst-address=176.105.199.98 dst-port=13391 \
src-address-list=iotstend \
to-addresses=192.168.100.61 to-ports=3389 \
comment="RDP forward: 13391 -> 192.168.100.61:3389"
```

У цьому правилі додано параметр `src-address-list=iotstend` – де `iotstend` це перелік адрес студентів. Щоб його створити викладач на вступному інструктажі просить студентів приєднатися до окремого порту, наприклад 13390. При цьому він тимчасово створює правило за допомогою команди:

```
/ip firewall filter add chain=input protocol=tcp dst-port=13390 \
```

```
action=add-src-to-address-list address-list=iotstend \  
comment="create iotstend list when requesting port 13390 for IoT stands"
```

Для підключення студентам (здобувачам освіти) надається модифікований файл автоматизації підключення до певних RaspberryPI (лабораторних стендів) (див. Додаток Б).

### **3.8 Спільний доступ до зовнішніх приладів, підключення USB камери, та інші обмеження**

В процесі використання різноманітних варіантів віддалено доступу з'ясувалося, що користувачі не мають права використовувати зовнішні порти.

Для вирішення цього питання було проведено додатковий аналіз літературних джерел.

Встановлено, що є можливість скористатися спеціальним керуванням за допомогою компоненту UDEV [59]

За допомогою спеціальних файлів є можливість призначити певний доступ до зовнішніх приладів/портів.

Необхідно створити додаткові правила.

Правила udev зберігаються в папці `/etc/udev/rules.d` . Файл правил обов'язково повинен мати розширення `.rules` . Зазвичай у цій папці вже є кілька файлів udev rules, але їх чіпати не рекомендується, для своїх правил краще створити окремий файл, наприклад:

```
touch /etc/udev/rules.d/10-local.rules
```

Правило udev складається з декількох пар ключ - значення , розділених комою. Одні ключі використовуються для перевірки відповідності пристрою певному правилу. У таких ключах використовується знак `==` для поділу пари, наприклад: `SUBSYSTEM == "block"` . Це означає, що правило буде застосовано тільки якщо значення ключа `SUBSYSTEM` для цього пристрою дорівнює `block` . Інші ключі використовуються для вказівки, якщо всі умови



відповідності виконуються. Для розділення пар у таких ключах використовується знак дорівнює "=", наприклад, NAME = "mydisk" . Ну і цілком правило:

```
SUBSYSTEM=="block", ATTR(size)=="1343153213", NAME="mydisk"
```

- SUBSYSTEM – підсистема пристрою;
- KERNEL - ім'я, яке видається пристрою ядром;
- DRIVER - драйвер, який обслуговує пристрій;
- ATTR – sysfs-атрибут пристрою;
- SUBSYSTEMS – підсистема батьківського пристрою.

Пристрій може мати батьківські пристрої, наприклад, жорсткий диск має батьківський пристрій SSCI , який, у свою чергу, має батьківський пристрій - шину BUS . Іноді потрібно отримати інформацію від батьківського пристрою. Для цього використовуються ключі SUBSYSTEMS, KERNELS, DRIVERS, ATTRS відповідно.

Для дій використовуються ключі:

- NAME – встановити ім'я файлу пристрою;
- SYMLINK – альтернативне ім'я пристрою;
- RUN – виконати скрипт при підключенні пристрою;
- GROUP - група, яка має доступ до файлу;
- OWNER – власник файлу пристрою;
- MODE – маска прав доступу.

В результаті проведені налаштування для використання ВЕБ камери та зовнішнього мікроконтролера – файл Arduino.rules

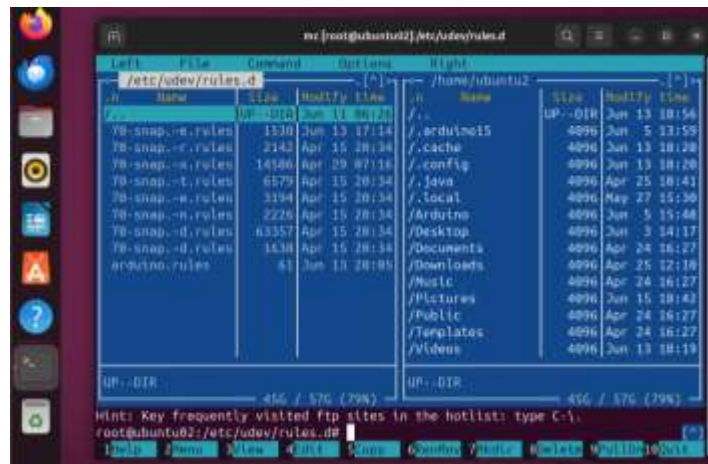


Рис. 3.37 Перелік правил UDEV

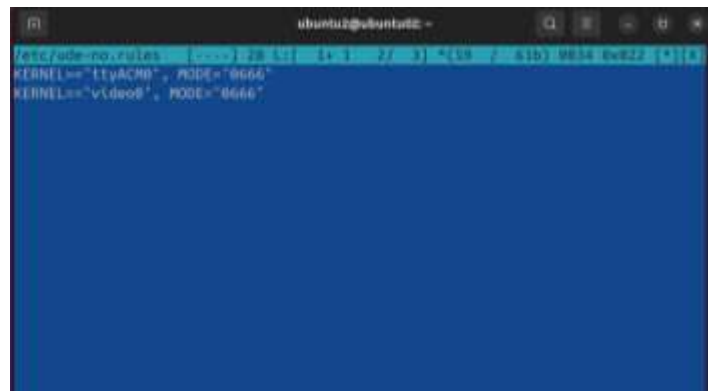


Рис. 3.38 Приклад правил UDEV

Для підключення зовнішньої USB камери необхідно додати файл \*.desktop у каталог

/etc/xdg/autostart/

На рис. 3.39 Показано перелік файлів цього каталогу

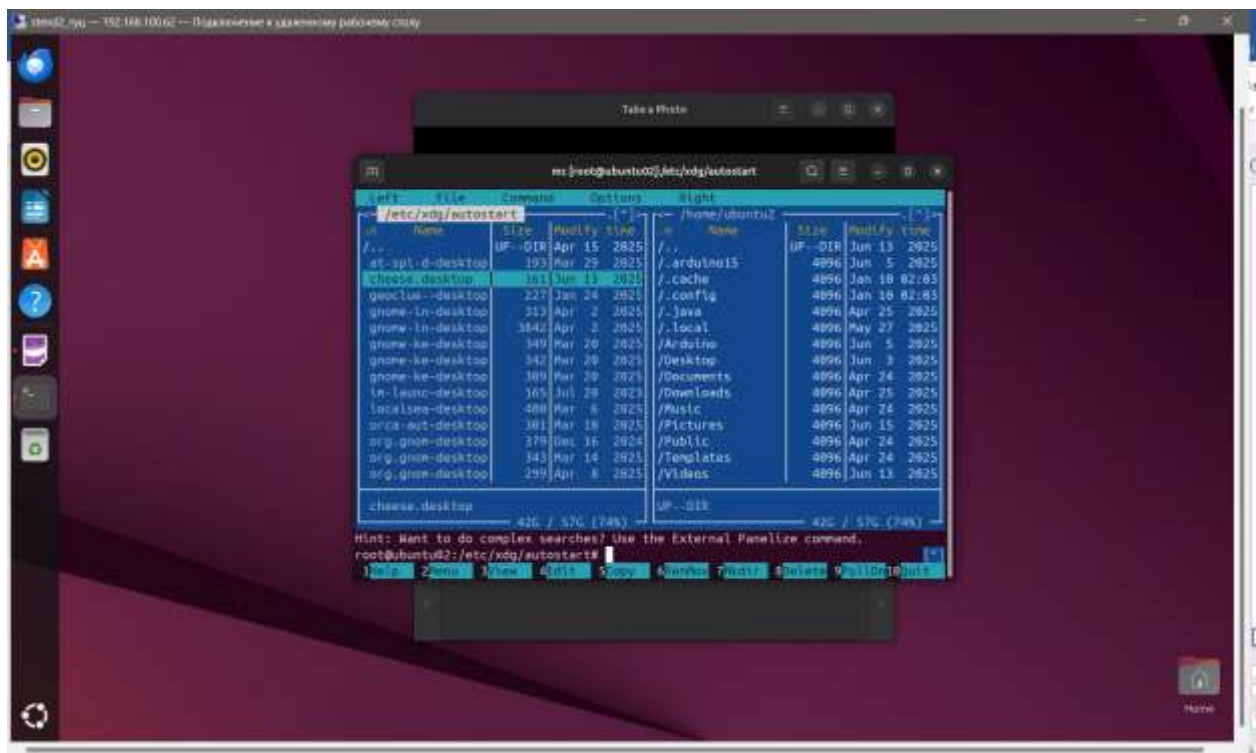


Рис. 3.39 Автозавантаження додатків

Приклад налаштувань додатку наведено на рис. 3.40

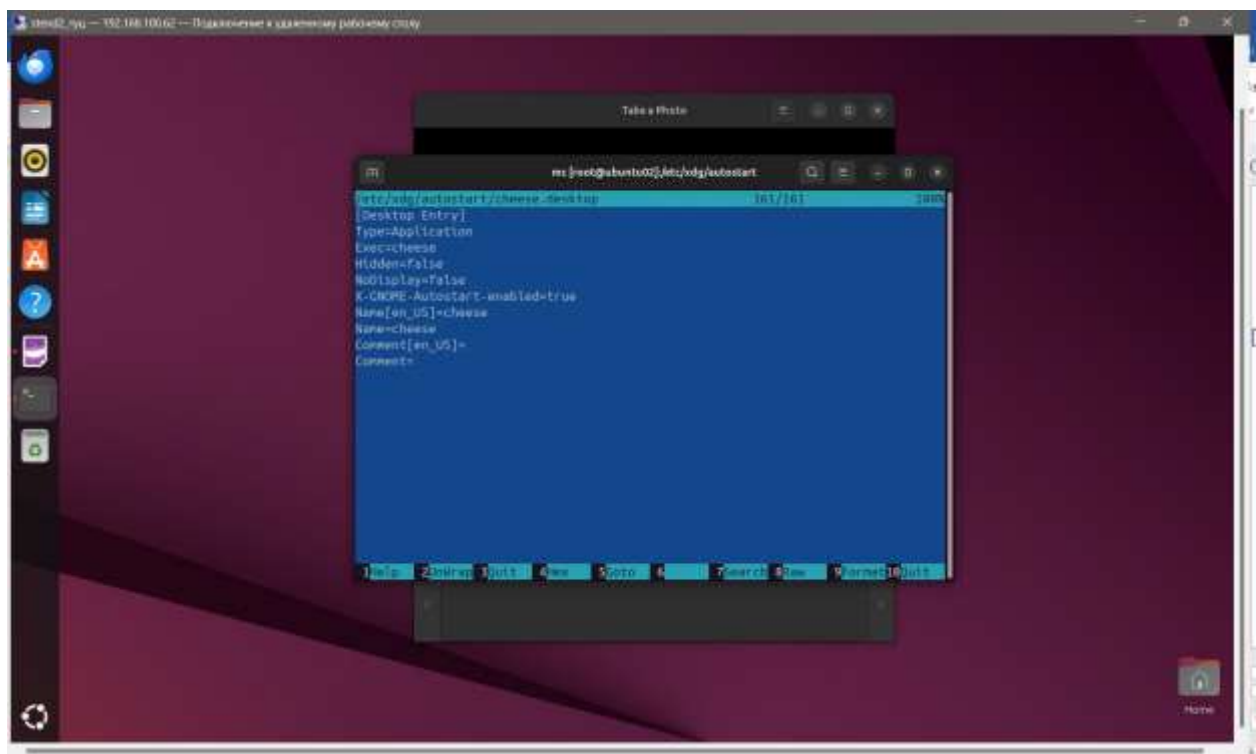


Рис. 3.40 Налаштування додатку для підтримки камери

Для автоматичного перезавантаження системи та примусового завершення роботи зі стендом скористуйтеся командою:

```
sleep 5400 && systemctl --user logout
## параметр -r - це час у секундах через який перезавантажити систему

#Скористатися командою at дозволяє запланувати виконання команди
# інсталивати пакет at
sudo apt update && sudo apt install at
sudo systemctl enable atd
sudo systemctl start atd

# команда для планування
echo "systemctl --user exit" | at now + 90 minutes
```

Для автоматичного налаштування необхідно створити файл \*.desktop з цією командою аналогічно налаштуванню для камери. Цей файл необхідно розташувати у наступному каталозі

/etc/xdg/autostart/

В процесі використання віддаленого доступу встановлено, що необхідно **обмежити кількість одночасних підключень**. Встановлено, що пакет Gnome-remote не має варіантів налаштувань для цього обмеження. Єдиним шляхом може бути використання пакету iptables Для цього необхідно:

```
# перевірити версію та наявність пакету
iptables -version

# встановити додатковий пакет - для збереження налаштування після
#перезавантаження системи
sudo apt update
sudo apt install iptables-persistent

# перевірити конфлікт зі службою ufw та відключити її
sudo ufw disable

#обмеження одночасних підключень
sudo iptables -I INPUT -p tcp --dport 3389 -m connlimit --connlimit-above 1
--connlimit-mask 0 -j REJECT
```

При цьому основні параметр:

--dport 3389 – стежить за портом RDP (перевірте, що у вас 3389, це стандарт).

--connlimit-above 1 — спрацьовує, якщо активних підключень вже більше 1.

--connlimit-mask 0 – найважливіший параметр. Він каже вважати підключення з усіх IP-адрес разом. Без цього параметра ліміт працював би "1 підключення на одну людину/IP", а двоє різних людей змогли б зайти.

-j REJECT - відкидає зайвих.

В процесі використання віддаленого доступу додатково встановлено, що необхідно **автоматично перезавантажувати систему** після роботи студента з певним стендом. В іншому випадку в системі можуть залишатася налаштування певних приладів/портів, з якими працював здобувач освіти. Для цього необхідно дозволити певним користувачам перезавантажувати систему без використання пароля SUDO. відредагувати файл sudoers.

#Відкрийте файл /etc/sudoers за допомогою visudo:  
sudo visudo /etc/sudoers

Додайте правило для вашої групи AD:

Вам потрібно використовувати ім'я групи AD із правильним синтаксисом. Часто потрібно екранувати пробіли або використовувати повне доменне ім'я.

Для групи IT Admins у домені MYDOMAIN:

```
# Синтаксис SSSD/Winbind з екрануванням пробілу:  
%"IOT\ USERS"@MYDOMAIN ALL=(ALL) NOPASSWD: /sbin/reboot  
%"IOT\ USERS"@MYDOMAIN ALL=(ALL) NOPASSWD: /sbin/shutdown
```

У такому випадку є можливість створити файл \*.desktop в каталозі /etc/xdg/autostart/, наприклад reb.desktop, який налаштувати на команду

```
#/etc/xdg/autostart/reboot.desktop  
echo "sudo reboot" | at now + 90 minutes  
#або  
sleep 5400 && sudo reboot
```

```
#або  
sudo shutdown -r +90 "The system will reboot and shut down in 90 min."
```

## ЗАГАЛЬНІ ВИСНОВКИ

У роботі обґрунтовано, що повноцінна підготовка фахівців за спеціальністю 123 «Комп'ютерна інженерія» неможлива лише за допомогою віртуальних симуляторів, оскільки вони не відтворюють реальних фізичних процесів та апаратних нюансів. В цілому можна казати, що в роботі пропонується та технічно реалізується модель «Лабораторія як сервіс» (LaaS), яка дозволяє студентам отримувати віддалений доступ до реального обладнання (Raspberry Pi, датчики, камери) з будь-якої точки світу.

У ході дослідження було виявлено та проаналізовано:

- Етапи еволюції IoT: від перших експериментів (автомат Coca-Cola) до сучасних епох AIoT (штучний інтелект речей) та Edge Computing (граничні обчислення).
- Стан лабораторної бази ЗВО України: проведено детальний огляд технічного оснащення провідних університетів (КПІ, НТУ «ДП», Одеська політехніка та ін.), що підтвердило необхідність модернізації методів доступу до обладнання.
- Специфіку ОС для Raspberry Pi: проаналізовано понад 10 операційних систем (Raspberry Pi OS, Ubuntu, Kali Linux, RetroPie тощо) та визначено, що Ubuntu є найбільш оптимальною для професійної підготовки та віддаленої роботи.
- Проблему обмеженого доступу: виявлено, що стандартні налаштування віддаленого доступу часто обмежують використання зовнішніх пристроїв (USB-камер, мікроконтролерів Arduino), що потребує спеціального налаштування правил доступу.

Для вирішення поставлених завдань у роботі пропонуються наступні технічні рішення:

1. Програмно-технічна система віддаленого доступу: використання протоколів SSH та RDP (через Gnome-remote) для керування лабораторними стендами.

2. Інтеграція з Microsoft Active Directory: запропоновано механізм єдиної аутентифікації студентів через доменну службу, що спрощує адміністрування.

3. Сценарії автоматизації: розроблено спеціальні скрипти (bat-файли), які автоматизують процес підключення студентів до стендів, включаючи додавання та подальше видалення облікових даних для безпеки.

4. Налаштування мережевої інфраструктури: запропоновано три варіанти зовнішнього підключення — через VPN (на базі MikroTik), RDP-сервер або прокидання портів (NAT) з жорстким контролем IP-адрес студентів.

5. Керування апаратними ресурсами: використання правил UDEV для надання студентам прав на використання USB-портів та підключених периферійних пристроїв віддалено.

Однак, запропоновані рішення мають певні недоліки та ризики:

- Залежність від зв'язку: якість навчання критично залежить від стабільності інтернет-з'єднання як в університеті, так і у студента.
- Технічна складність: налаштування системи (Linux, VPN, AD) вимагає високої кваліфікації технічного персоналу лабораторії.
- Апаратні обмеження: запуск повноцінних графічних оболонок (Ubuntu Desktop) на мікрокомп'ютерах створює значне навантаження на оперативну пам'ять, особливо при декількох сесіях.
- В роботі не розглянути можливості інших спеціалізованих пакетів.
- Кіберзагрози: відкриття віддаленого доступу до внутрішньої мережі ЗВО створює додаткові вектори для хакерських атак.

## СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. The "Only" Coke Machine on the Internet. [Електронний ресурс] URL: <https://www.cs.cmu.edu/~coke/> (дата звернення: 03.01.2026)
2. A brief history of the Internet of Things - IBM Think. [Електронний ресурс] URL: <https://www.ibm.com/think/topics/iot-first-device> (дата звернення: 03.01.2026)
3. The Coke Machine - Carnegie Mellon University News. [Електронний ресурс] URL: <https://www.google.com/search?q=https://www.cmu.edu/home/news/history/coke-machine.html> (дата звернення: 03.01.2026)
4. Living Internet: Internet Toaster. [Електронний ресурс] URL: [https://www.livinginternet.com/i/ia\\_myths\\_toast.htm](https://www.livinginternet.com/i/ia_myths_toast.htm) (дата звернення: 03.01.2026)
5. The Internet of Things: A History - Wired BrandLab. [Електронний ресурс] URL: <https://www.google.com/search?q=https://www.wired.com/brandlab/2016/10/the-internet-of-things-a-history/> (дата звернення: 03.01.2026)
6. John Romkey: Internet Pioneer and the First Smart Device - Avast Blog. [Електронний ресурс] URL: <https://blog.avast.com/the-internets-first-smart-device> (дата звернення: 03.01.2026)
7. Carnegie Mellon's Vending Machine - Vertical Innovations. [Електронний ресурс] URL: <https://www.vertical-innovations.com/coca-cola-vending-machine/> (дата звернення: 03.01.2026)
8. Origins of the Internet of Things - ResearchGate. [Електронний ресурс] URL: [https://www.researchgate.net/publication/381689333\\_Internet\\_Of\\_Things\\_IOT\\_Origins\\_Embedded\\_Technologies\\_Smart\\_Applications\\_and\\_its\\_Gro](https://www.researchgate.net/publication/381689333_Internet_Of_Things_IOT_Origins_Embedded_Technologies_Smart_Applications_and_its_Gro)



wth\_in\_the\_Last\_Decade (дата звернення: 03.01.2026)

9. IEEE Xplore: Analysis of early network-connected appliances.  
[Електронний ресурс] URL:  
<https://www.google.com/search?q=https://ieeexplore.ieee.org/document/297298> (дата звернення: 03.01.2026)
10. RFC 1157 - A Simple Network Management Protocol (SNMP).  
[Електронний ресурс] URL: <https://datatracker.ietf.org/doc/html/rfc1157>  
(дата звернення: 03.01.2026)
11. Ashton, K. That 'Internet of Things' Thing. [Електронний ресурс] URL:  
<https://www.rfidjournal.com/that-internet-of-things-thing> (дата звернення: 03.01.2026).
12. History of the Auto-ID Center. [Електронний ресурс] URL:  
<https://autoid.mit.edu/about-us> (дата звернення: 03.01.2026).
13. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. [Електронний ресурс] URL:  
[https://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf) (дата звернення: 07.01.2026).
14. The smart meter: a history of Enel's innovation. [Електронний ресурс] URL: <https://www.enel.com/company/stories/articles/2021/04/smart-meter-history> (дата звернення: 07.01.2026).
15. Internet of Things (IoT) History. [Електронний ресурс] URL:  
<https://www.postscapes.com/internet-of-things-history/> (дата звернення: 07.01.2026).
16. RFC 4944 — Transmission of IPv6 Packets over IEEE 802.15.4 Networks.  
[Електронний ресурс] URL: <https://datatracker.ietf.org/doc/html/rfc4944>  
(дата звернення: 07.01.2026).
17. White Paper: Cortex-M for Beginners. [Електронний ресурс] URL:  
<https://community.arm.com/arm-community-blogs/b/architectures-and->

- [processors-blog/posts/white-paper-cortex-m-for-beginners](https://processors-blog/posts/white-paper-cortex-m-for-beginners) (дата  
звернення: 07.01.2026).
- 18.MQTT Essentials: Part 1 – Introducing MQTT. [Електронний ресурс]  
URL: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/> (дата звернення: 07.01.2026).
- 19.Nabaztag — Wikipedia. [Електронний ресурс] URL:  
<https://en.wikipedia.org/wiki/Nabaztag> (дата звернення: 07.01.2026).
- 20.IEEE 802.15.4-2003 — IEEE Standard for Low-Rate Wireless Personal  
Area Networks. [Електронний ресурс] URL:  
<https://standards.ieee.org/ieee/802.15.4/3387/> (дата звернення:  
07.01.2026).
- 21.Adam Dunkels — Protothreads: Simplifying Event-Driven Programming.  
[Електронний ресурс] URL: <https://dunkels.com/adam/pt/> (дата  
звернення: 07.01.2026).
- 22.GS1 EPCglobal — The Overview and Standards. [Електронний ресурс]  
URL: <https://www.gs1.org/standards/epcglobal> (дата звернення:  
07.01.2026).
- 23.AWS IoT Core – Amazon Web Services. [Електронний ресурс] URL:  
<https://aws.amazon.com/iot-core/> (дата звернення: 05.01.2026).
- 24.Raspberry Pi – About us. [Електронний ресурс] URL:  
<https://www.raspberrypi.com/about/> (дата звернення: 05.01.2026).
- 25.Espressif Systems – ESP32 Series. [Електронний ресурс] URL:  
<https://www.espressif.com/en/products/socs/esp32> (дата звернення:  
05.01.2026).
- 26.MQTT.org – Documentation. [Електронний ресурс] URL:  
<https://www.google.com/search?q=https://mqtt.org/documentation/> (дата  
звернення: 05.01.2026).
- 27.Arduino – What is Arduino? [Електронний ресурс] URL:  
<https://www.google.com/search?q=https://www.arduino.cc/en/Guide/Intro>

uction (дата звернення: 05.01.2026).

28.Azure IoT Services – Microsoft. [Електронний ресурс] URL: <https://azure.microsoft.com/en-us/solutions/iot/> (дата звернення: 05.01.2026).

29.Node-RED – Low-code programming for event-driven applications. [Електронний ресурс] URL: <https://nodered.org/> (дата звернення: 05.01.2026).

30.AIoT: The Next Evolution of the Internet of Things. [Електронний ресурс] URL: <https://www.google.com/search?q=https://www.techtarget.com/iotagenda/definition/AIoT-Artificial-Intelligence-of-Things> (дата звернення: 05.01.2026).

31.What is Edge Computing? [Електронний ресурс] URL: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-edge-computing/> (дата звернення: 05.01.2026).

32.Industry 5.0: Towards a sustainable, human-centric and resilient European industry. [Електронний ресурс] URL: [https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50\\_en](https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50_en) (дата звернення: 05.01.2026).

33.Smart Cities: The Future of Urban Environments. [Електронний ресурс] URL: <https://www.google.com/search?q=https://www.ieee.org/about/news/2024/smart-cities-iot.html> (дата звернення: 05.01.2026).

34.Matter: The Foundation for Connected Things. [Електронний ресурс] URL: <https://csa-iot.org/all-solutions/matter/> (дата звернення: 05.01.2026).

35.5G IoT: The foundation for an intelligent world. [Електронний ресурс] URL: <https://www.google.com/search?q=https://www.ericsson.com/en/internet-of-things/5g-iot> (дата звернення: 05.01.2026) (дата звернення:

05.10.2025).

36. КІП ім. Ігоря Сікорського. Програма «Комп'ютерні системи та мережі» [Електронний ресурс] URL: [https://osvita.kpi.ua/F7\\_OPPB\\_KSM](https://osvita.kpi.ua/F7_OPPB_KSM) (дата звернення: 05.10.2025).
37. КІП ім. Ігоря Сікорського. Програма «Системне програмування та спеціалізовані комп'ютерні системи» [Електронний ресурс] URL: [https://osvita.kpi.ua/F7\\_OPPB\\_SPSKS](https://osvita.kpi.ua/F7_OPPB_SPSKS) (дата звернення: 05.10.2025).
38. Національний авіаційний університет (Київський авіаційний інститут).  
◦ Офіційний сайт та ресурси кафедри [Електронний ресурс] URL: <https://nau.edu.ua/>, <http://ccs.nau.edu.ua/> (дата звернення: 05.10.2025).
39. Харківський національний університет радіоелектроніки (ХНУРЕ).  
◦ Опис спеціальності 123: <https://nure.ua/abituriyentam/spetsialnosti-ta-spetsializatsiyi/spetsialnist-f7-komp-iuterna-inzheneriia> (дата звернення: 05.10.2025).
40. НТУ «Дніпровська політехніка». ◦ Освітні плани та програми кафедри [Електронний ресурс] URL: [https://it.nmu.org.ua/ua/edu\\_ped\\_work/OKX\\_OPP\\_edu\\_plans.php](https://it.nmu.org.ua/ua/edu_ped_work/OKX_OPP_edu_plans.php) (дата звернення: 05.10.2025).
41. Луцький національний технічний університет (ЛНТУ). Сторінка освітньої програми [Електронний ресурс] URL: <https://lntu.edu.ua/uk/studentu-0/navchannya/osvitniy-programi#5115> (дата звернення: 05.10.2025).
42. НУ «Чернігівська політехніка». Описи освітніх програм [Електронний ресурс] URL: [https://op.stu.cn.ua/view/total\\_view.php](https://op.stu.cn.ua/view/total_view.php) (дата звернення: 05.10.2025).
43. Державний університет інформаційно-комунікаційних технологій (ДУІКТ). Освітні програми кафедри комп'ютерної інженерії [Електронний ресурс] URL: <https://duikt.edu.ua/ua/1824-osvitni-programi-kafedra-kompyuternoї-inzheneriї> (дата звернення: 05.10.2025).

44. Кременчуцький національний університет ім. М. Остроградського (КрНУ) [Електронний ресурс] URL: ° Сайт кафедри комп'ютерної інженерії та електроніки: <http://cee.kdu.edu.ua/> (дата звернення: 05.10.2025).
45. ДЗ „Луганський національний університет імені Тараса Шевченка”. ОПП. Бакалаврський рівень. 123 Комп'ютерна інженерія / F7 Комп'ютерна інженерія [Електронний ресурс] URL: [https://luguniv.edu.ua/?page\\_id=61087](https://luguniv.edu.ua/?page_id=61087) (дата звернення: 05.10.2025).
46. Криворізький національний університет (КНУ). ° Сайт кафедри комп'ютерних систем та мереж [Електронний ресурс] URL: <http://www.ksm.knu.edu.ua/> (дата звернення: 05.10.2025).
47. Східноукраїнський національний університет імені Володимира Даля. ° Навчальні ресурси у системі Moodle [Електронний ресурс] URL: <http://moodle2.snu.edu.ua/course/view.php?id=3953>. (дата звернення: 05.10.2025).
48. Одеська політехніка. ° Опис програми «Комп'ютерні системи та мережі» [Електронний ресурс] URL: <https://op.edu.ua/education/programs/bac-123-2>. (дата звернення: 05.11.2025).
49. Raspbian. [Електронний ресурс] - <https://www.raspbian.org/> (дата звернення: 05.11.2025).
50. Kali Linux. [Електронний ресурс] - <https://www.kali.org/> (дата звернення: 05.11.2025).
51. Ubuntu for desktops. [Електронний ресурс] - <https://ubuntu.com/desktop> (дата звернення: 05.11.2025).
52. RISC OS. [Електронний ресурс] - <https://www.riscosopen.org/content/> (дата звернення: 05.11.2025).
53. SARPİ. [Електронний ресурс] - <https://www.slackbuilds.org/repository/14.2/> (дата звернення: 05.11.2025).

54. Arch Linux. [Електронний ресурс] <https://archlinux.org/download/> (дата звернення: 05.05.2025).
55. Installing FreeBSD for Raspberry Pi. [Електронний ресурс] - <https://freebsdoundation.org/freebsd-project/resourcesold/installing-freebsd-for-raspberry-pi/> (дата звернення: 05.12.2025).
56. RetroPie. [Електронний ресурс] <https://retropie.org.uk/> (дата звернення: 05.05.2025).
57. Raspberry Pi OS. [Електронний ресурс] - <https://www.raspberrypi.com/software/> (дата звернення: 05.12.2025).
58. Introduction to remote access. [Електронний ресурс] - <https://www.raspberrypi.com/documentation/computers/remote-access.html#introduction-to-remote-access/> (дата звернення: 05.12.2025).
59. Налаштування udev rules в Linux. [Електронний ресурс] - <https://losst.pro/nastrojka-udev-rules-v-linux> (дата звернення: 05.12.2025).

## ДОДАТКИ

### Додаток А. Скрипт для підключення до навчальних стендів на засадах RaspberryPI

```
@echo off
rem ssh ubuntu2@192.168.100.62
chcp 65001 >nul
title RDP Меню (Авто-вход)

:: --- налаштування стендів ---
:: налаштування IP адрес та параметрів підключення

set IP1=192.168.100.61
set USER1=ub2rdp
set PASS1=ub2rdp

set IP2=192.168.100.62
set USER2=ub2rdp
set PASS2=ub2rdp

set IP3=192.168.100.63
set USER3=ub2rdp
set PASS3=ub2rdp

set IP4=192.168.100.64
set USER4=ub2rdp
set PASS4=ub2rdp

set IP5=192.168.100.65
set USER5=ub2rdp
set PASS5=ub2rdp

set IP6=192.168.100.66
set USER6=ub2rdp
set PASS6=ub2rdp

:: вивід на екран переліку (стендів) та обирання навчального стенду

:MENU
cls
echo =====
echo  МЕНЮ СТЕНДІВ С АВТОРИЗАЦІЄЮ
echo =====
echo 1. stend1
echo 2. stend2
echo 3. stend3
echo 4. stend4
```

```
echo 5. stend5
echo 6. stend6
echo 7. Ввести адресу вручну
echo 0. Вихід
echo =====
echo.
```

```
set /p choice=Виберите сервер:
```

```
:: перехід на певний стенд
```

```
if "%choice%"=="1" goto CONNECT1
if "%choice%"=="2" goto CONNECT2
if "%choice%"=="3" goto CONNECT3
if "%choice%"=="4" goto CONNECT4
if "%choice%"=="5" goto CONNECT5
if "%choice%"=="6" goto CONNECT6
if "%choice%"=="7" goto MANUAL
if "%choice%"=="0" exit
goto MENU
```

```
:: команд для підключення до певного стенду
```

```
:CONNECT1
```

```
echo.
echo Додаємо учетні дані для stend1.
:: Додаємо пароль в хранилище
cmdkey /generic:TERMSRV/%IP1% /user:%USER1% /pass:%PASS1%
echo Запускаємо підключення...
start mstsc /v:%IP1%
goto CLEANUP
```

```
:CONNECT2
```

```
echo.
echo Додаємо учетні дані для stend2....
cmdkey /generic:TERMSRV/%IP2% /user:%USER2% /pass:%PASS2%
echo Запускаємо підключення...
start mstsc /v:%IP2%
goto CLEANUP
```

```
:CONNECT3
```

```
echo.
echo Додаємо учетні дані для stend3....
cmdkey /generic:TERMSRV/%IP3% /user:%USER3% /pass:%PASS3%
echo Запускаємо підключення...
start mstsc /v:%IP3%
goto CLEANUP
```

```
:CONNECT4
```



```
echo.
echo Додаємо учетні дані для stend4....
cmdkey /generic:TERMSRV/%IP4% /user:%USER4% /pass:%PASS4%
echo Запускаємо підключення...
start mstsc /v:%IP4%
goto CLEANUP

:CONNECT5
echo.
echo Додаємо учетні дані для stend5....
cmdkey /generic:TERMSRV/%IP5% /user:%USER5% /pass:%PASS5%
echo Запускаємо підключення...
start mstsc /v:%IP5%
goto CLEANUP

:CONNECT6
echo.
echo Додаємо учетні дані для stend6....
cmdkey /generic:TERMSRV/%IP6% /user:%USER6% /pass:%PASS6%
echo Запускаємо підключення...
start mstsc /v:%IP6%
goto CLEANUP

:: завершення з ліквідацією паролів

:CLEANUP
:: Чекаємо 3 сек, чтобы mstsc успел прочитать пароль
timeout /t 3 /nobreak >nul
echo.
echo Удаляем збережений пароль з системи безпеки...
:: Удаляем пароль (безопасность)
if "%choice%"=="1" cmdkey /delete:TERMSRV/%IP1%
if "%choice%"=="2" cmdkey /delete:TERMSRV/%IP2%
if "%choice%"=="3" cmdkey /delete:TERMSRV/%IP3%
if "%choice%"=="4" cmdkey /delete:TERMSRV/%IP4%
if "%choice%"=="5" cmdkey /delete:TERMSRV/%IP5%
if "%choice%"=="6" cmdkey /delete:TERMSRV/%IP6%
echo Готово. Вікно зачиняється.
timeout /t 2 >nul
exit

:MANUAL
cls
echo.
set /p manual_ip=Введіть IP адресу:
start mstsc /v:%manual_ip%
exit
```

## **Додаток Б. Модифікований скрипт для підключення до навчальних стендів на засадах RaspberryPI**

```
@echo off
rem ssh ubuntu2@192.168.100.62
chcp 65001 >nul
title RDP Меню (Авто-вход)

:: --- налаштування стендів ---
:: налаштування IP адрес та параметрів підключення

set IPG=192.168.100.61

set IP1=%IPG%:13391
set USER1=ub2rdp
set PASS1=ub2rdp

set IP2=%IPG%:13392
set USER2=ub2rdp
set PASS2=ub2rdp

set IP3=%IPG%:13393
set USER3=ub2rdp
set PASS3=ub2rdp

set IP4=%IPG%:13394
set USER4=ub2rdp
set PASS4=ub2rdp

set IP5=%IPG%:13395
set USER5=ub2rdp
set PASS5=ub2rdp

set IP6=%IPG%:13396
set USER6=ub2rdp
set PASS6=ub2rdp

:: вивід на екран переліку (стендів) та обирання навчального стенду

:MENU
cls
echo =====
echo    МЕНЮ СТЕНДІВ С АВТОРИЗАЦІЄЮ
echo =====
echo 1. stend1
echo 2. stend2
echo 3. stend3
echo 4. stend4
```

```
echo 5. stend5
echo 6. stend6
echo 7. Ввести адресу вручну
echo 0. Вихід
echo =====
echo.
```

```
set /p choice=Виберите сервер:
```

```
:: перехід на певний стенд
```

```
if "%choice%"=="1" goto CONNECT1
if "%choice%"=="2" goto CONNECT2
if "%choice%"=="3" goto CONNECT3
if "%choice%"=="4" goto CONNECT4
if "%choice%"=="5" goto CONNECT5
if "%choice%"=="6" goto CONNECT6
if "%choice%"=="7" goto MANUAL
if "%choice%"=="0" exit
goto MENU
```

```
:: команд для підключення до певного стенду
```

```
:CONNECT1
```

```
echo.
echo Додаємо учетні дані для stend1.
:: Додаємо пароль в хранилище
cmdkey /generic:TERMSRV/%IP1% /user:%USER1% /pass:%PASS1%
echo Запускаємо підключення...
start mstsc /v:%IP1%
goto CLEANUP
```

```
:CONNECT2
```

```
echo.
echo Додаємо учетні дані для stend2....
cmdkey /generic:TERMSRV/%IP2% /user:%USER2% /pass:%PASS2%
echo Запускаємо підключення...
start mstsc /v:%IP2%
goto CLEANUP
```

```
:CONNECT3
```

```
echo.
echo Додаємо учетні дані для stend3....
cmdkey /generic:TERMSRV/%IP3% /user:%USER3% /pass:%PASS3%
echo Запускаємо підключення...
start mstsc /v:%IP3%
goto CLEANUP
```

```
:CONNECT4
```

```
echo.
echo Додаємо учетні дані для stend4....
cmdkey /generic:TERMSRV/%IP4% /user:%USER4% /pass:%PASS4%
echo Запускаємо підключення...
start mstsc /v:%IP4%
goto CLEANUP

:CONNECT5
echo.
echo Додаємо учетні дані для stend5....
cmdkey /generic:TERMSRV/%IP5% /user:%USER5% /pass:%PASS5%
echo Запускаємо підключення...
start mstsc /v:%IP5%
goto CLEANUP

:CONNECT6
echo.
echo Додаємо учетні дані для stend6....
cmdkey /generic:TERMSRV/%IP6% /user:%USER6% /pass:%PASS6%
echo Запускаємо підключення...
start mstsc /v:%IP6%
goto CLEANUP

:: завершення з ліквідацією паролів

:CLEANUP
:: Чекаємо 3 сек, чтобы mstsc успел прочитать пароль
timeout /t 3 /nobreak >nul
echo.
echo Удаляем збережений пароль з системи безпеки...
:: Удаляем пароль (безопасность)
if "%choice%"=="1" cmdkey /delete:TERMSRV/%IP1%
if "%choice%"=="2" cmdkey /delete:TERMSRV/%IP2%
if "%choice%"=="3" cmdkey /delete:TERMSRV/%IP3%
if "%choice%"=="4" cmdkey /delete:TERMSRV/%IP4%
if "%choice%"=="5" cmdkey /delete:TERMSRV/%IP5%
if "%choice%"=="6" cmdkey /delete:TERMSRV/%IP6%
echo Готово. Вікно зачиняється.
timeout /t 2 >nul
exit

:MANUAL
cls
echo.
set /p manual_ip=Введіть IP адресу:
start mstsc /v:%manual_ip%
exit
```

