

Міністерство освіти і науки України
Український державний університет науки і технологій

Комп'ютерних технологій і систем
(назва факультету)

Електронні обчислювальні машини
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи
бакалавра
(ступінь вищої освіти)

на тему: Розробка навчального тренажера з дисципліни «Проектування мікропроцесорних систем». Мережева частина.

за освітньою програмою Комп'ютерна інженерія

зі спеціальності: 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Виконав: студент групи: КС1811

K. Oley
(підпис студента)

/ Кирило ОЛІЙНИК /
(Ім'я ПРІЗВИЩЕ)

Керівник:

В. С. С.
(підпис)

/ ст. викладач, Володимир ДЗЮБА /
(посада, ім'я ПРІЗВИЩЕ)

Нормоконтролер:

В. С. С.
(підпис)

/ ст. викладач Володимир ДЗЮБА /
(посада, ім'я ПРІЗВИЩЕ)

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

Студент

K. Oley
(підпис)

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Computer Technologies and Systems
(faculty)

Electronic Computers
(department)

Explanatory Note

to Bachelor's Thesis
(higher education degree)

on the topic: Development of a training simulator in the discipline "Design of microprocessor systems". Network part.

according to educational curriculum Computer Engineering

in the Speciality: 123 Computer Engineering

(speciality and its code)

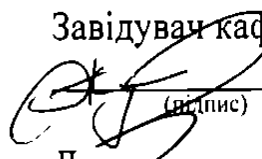
Done by the student of the group: K.Oley / Kyrylo Oliinyk /
(name, surname)

Scientific Supervisor: BSY / Sr. Lecturer, Volodymyr Dziuba /
(position, name, surname)

Normative controller: BSY / Sr. Lecturer, Volodymyr Dziuba /
(position, name, surname)

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи
Кафедра: Електронні обчислювальні системи
Рівень вищої освіти: Перший (бакалаврський)
Освітня програма: Комп'ютерна інженерія
Спеціальність: 123 Комп'ютерна інженерія
(шифр та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри ЕОМ
 Ігор ЖУКОВИЦЬКИЙ
(вiдпис) (Ім'я ПРІЗВИЩЕ)
Дата _____

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра
(ступінь вищої освіти)

студенту Олійнику Кирилу Олеговичу
(Прізвище, Ім'я По батькові)

1. Тема роботи: Розробка навчального тренажеру з дисципліни «Проектування мікропроцесорних систем». Мережева частина

Керівник роботи: Дзюба Володимир Володимирович
(Прізвище, Ім'я По батькові, науковий ступінь, вчене звання)

затвердені наказом від "07" 12 2021 р. № 68ст

2. Строк подання студентом роботи: 17.06.2022 р.

3. Вихідні дані до роботи: 8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE, 1985, 120с; Методичні вказівки з дисципліни «Проектування мікропроцесорних систем».

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Основна частина:

Паралельний периферійний адаптер Intel 8225A

Програмне забезпечення для дистанційного навчання

Розробка програмного забезпечення мережевої частини тренажеру

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	През
1	Вступ та постановка завдання	15.04.2022	50
2	Паралельний периферійний адаптер Intel 8225A	27.04.2022	20
3	Програмне забезпечення для дистанційного навчання	10.05.2022	30
4	Розробка програмного забезпечення мережевої частини тренажеру	30.05.2022	50
5	Висновки	03.06.2022	50
6	Подання кваліфікаційної роботи до кафедри	17.06.2022	
7	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	22.06.2022	

Студент

K. Oley
(підпис)

Кирило ОЛІЙНИК
(підпис)

Керівник роботи

В. Дзюба
(підпис)

Володимир ДЗЮБА
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ та постановка завдання	15.04.2022	5%
2	Паралельний периферійний адаптер Intel 8225A	27.04.2022	20%
3	Програмне забезпечення для дистанційного навчання	10.05.2022	20%
4	Розробка програмного забезпечення мережевої частини тренажеру	30.05.2022	50%
5	Висновки	03.06.2022	5%
6	Подання кваліфікаційної роботи до кафедри	17.06.2022	
7	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	22.06.2022	

Студент

_____ (підпис)

Кирило ОЛІЙНИК

_____ (Ім'я ПРІЗВИЩЕ)

Керівник роботи

_____ (підпис)

Володимир ДЗЮБА

_____ (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавру:
46 с, 35 рис., 3 табл., 2 додатки, 10 джерел.

Об'єкт розробки – мережева частина навчального тренажера з дисципліни «Проектування мікропроцесорних систем».

Мета роботи – розробити алгоритм мережевої частини. Розробити протокол взаємодії між серверною та клієнтською частиною тренажера для реалізації мережевих можливостей. Розробити інтерфейс користувача серверної та клієнтської частини. Програмно реалізувати мережеву частину навчального тренажера з дисципліни «Проектування мікропроцесорних систем».

Методи дослідження – аналіз існуючого подібного програмного забезпечення та його порівняння, тестування програмної реалізації.

Одержані результати – зроблено аналіз програмного забезпечення для дистанційного навчання. Розроблений алгоритм мережевої частини, протокол взаємодії між клієнтом і сервером. Реалізована мережева частина навчального тренажера. Програмна реалізація виконана в середовищі Lazarus за допомогою Object pascal, з використанням мережевої бібліотеки Indy.

Ключові слова: НАВЧАЛЬНИЙ ТРЕНАЖЕР, ДИСТАНЦІЙНЕ НАВЧАННЯ, СЕРВЕР, КЛІЄНТ, INTEL 8255A.

ЗМІСТ

ВСТУП ТА ПОСТАНОВКА ЗАВДАННЯ	6
1 ПАРАЛЕЛЬНИЙ ПЕРИФЕРІЙНИЙ АДАПТЕР INTEL 8255A.....	7
1.1 Загальні характеристики, відомі аналоги	7
1.2 Внутрішня структура пристрою та налаштування	8
1.3 Особливості використання тренажеру ППА в дистанційному навчальному процесі.....	12
1.4 Висновки	13
2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ	15
2.1 Zoom	15
2.2 Lider	16
2.3 Telegram	18
2.4 Порівняння програм для дистанційного навчання.....	20
2.5 Висновки	22
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МЕРЕЖЕВОЇ ЧАСТИНИ ТРЕНАЖЕРА	23
3.1 Загальна структура тренажера та його особливості	23
3.2 Мережева частина тренажера, мережеві частини інтерфейсу програми	25
3.3 Алгоритм обміну інформацією між окремими частинами тренажера ...	29
3.4 Розробка мережевої частини ПЗ тренажера	32
3.5 Пункти довідки, присвячені мережевій частині.....	37
3.6 Висновки.....	39
ВИСНОВКИ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
ДОДАТОК А.....	ПОМИЛКА! ЗАКЛАДКУ НЕ ВИЗНАЧЕНО.
ДОДАТОК Б	ПОМИЛКА! ЗАКЛАДКУ НЕ ВИЗНАЧЕНО.

ВСТУП ТА ПОСТАНОВКА ЗАВДАННЯ

Метою кваліфікаційною роботи є розробка мережевої частини навчального тренажеру по дисципліні «Проектування мікропроцесорних систем», завдяки якій, викладач зможе видавати завдання студентом, та перевіряти їх знання. Для досягнення мети, в ході роботи необхідно виконати такі завдання:

- ознайомитись з матеріалом по паралельному периферійному адаптеру;
- проаналізувати використовуване програмне забезпечення для дистанційного навчання;
- розробити алгоритм мережевої частини тренажеру;
- програмно реалізувати мережеву частину тренажеру;
- перевірити працездатність розробленої частини тренажеру;
- описати процес розробки мережевої частини навчального тренажеру.

Мережева частина програми-тренажеру повинна мати:

- клієнтську частину;
- серверну частину;
- можливість передавати інформацію з клієнта на сервер;
- можливість передавати завдання з серверу на клієнт.

Для розробки мережевої частини тренажеру було обрано мову object pascal та середовища Lazarus. Для реалізації серверу та клієнту було обрано мережеву бібліотеку Indy.

1 Паралельний периферійний адаптер INTEL 8255A

1.1 Загальні характеристики, відомі аналоги

Intel 8255A – програмований пристрій вводу/виводу паралельної інформації, застосовується в якості елемента вводу/виводу загального призначення, який сполучає різні види периферійних пристроїв з магістраллю даних систем обробки інформації. Радянський аналог – КР580ВВ55. На рисунку 1.1 зображено мікросхему Intel 8225A у корпусі DIP40.[1]



Рисунок 1.1 – Intel 8225A у корпусі DIP40 [2]

ППА працює в наступних чотирьох режимах:

- Нульовий (Синхронний) – три канали вводу/виводу працюють незалежно один від одного, є можливість запрограмувати на ввід, чи на вивід інформації;
- Перший (Для А – асинхронний-однонаправлений, для В - асинхронний) – в цьому режимі є можливість передавати інформацію по каналам А та В, а окремі біти порту С використовуються для керування обміном даних;
- Другий (Для А – асинхронний-двонаправлений) – канал А отримує властивість двонаправленої шини, також використовує окремі біти для порту С;
- Режим роботи з окремими бітами каналу С – дозволяє змінювати значення або скидати окремі біти у каналі С.

На рисунку 1.2 зображено розпіновку паралельного периферійного адаптеру Intel 8225A.

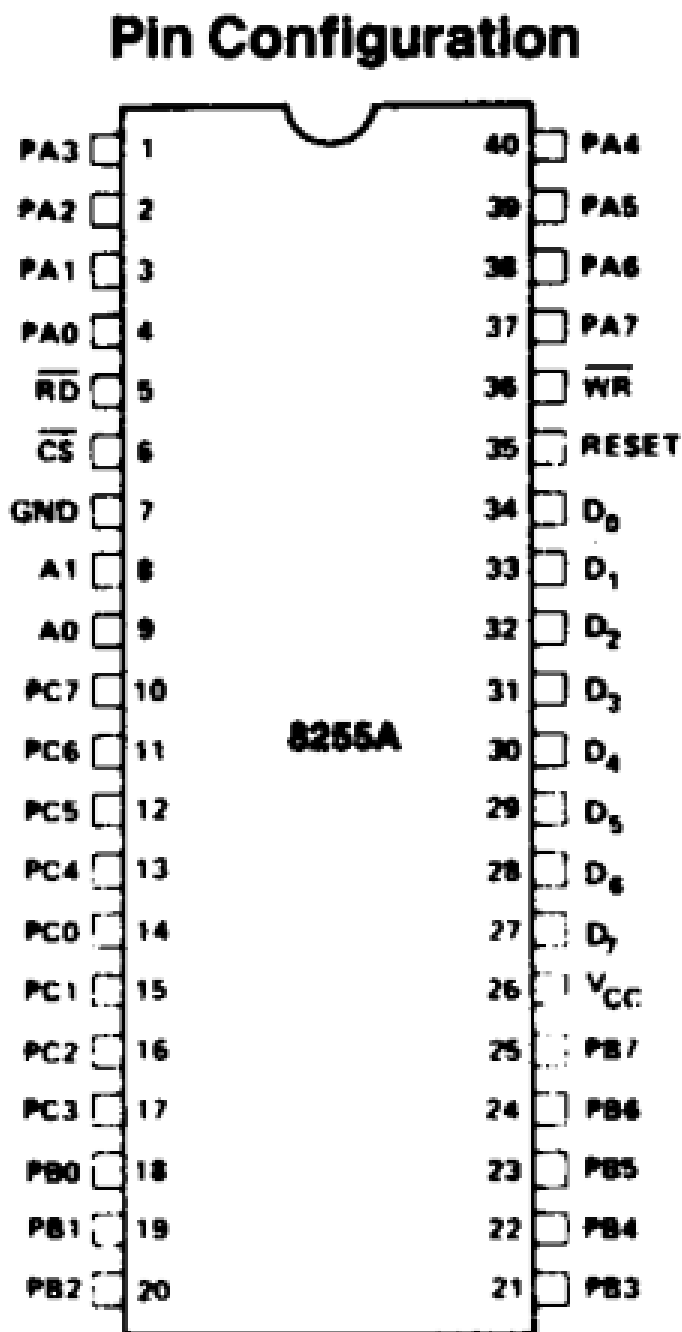


Рисунок 1.2 – Розпіновка Intel 8255A [5]

Основні особливості мікросхеми: працює у чотирьох режимах; має 3 канали вводу/виводу; окреме програмування кожного каналу.

1.2 Внутрішня структура пристрою та налаштування

На рисунку 1.3 зображена структурна схема паралельного периферійного адаптеру Intel 8225A.

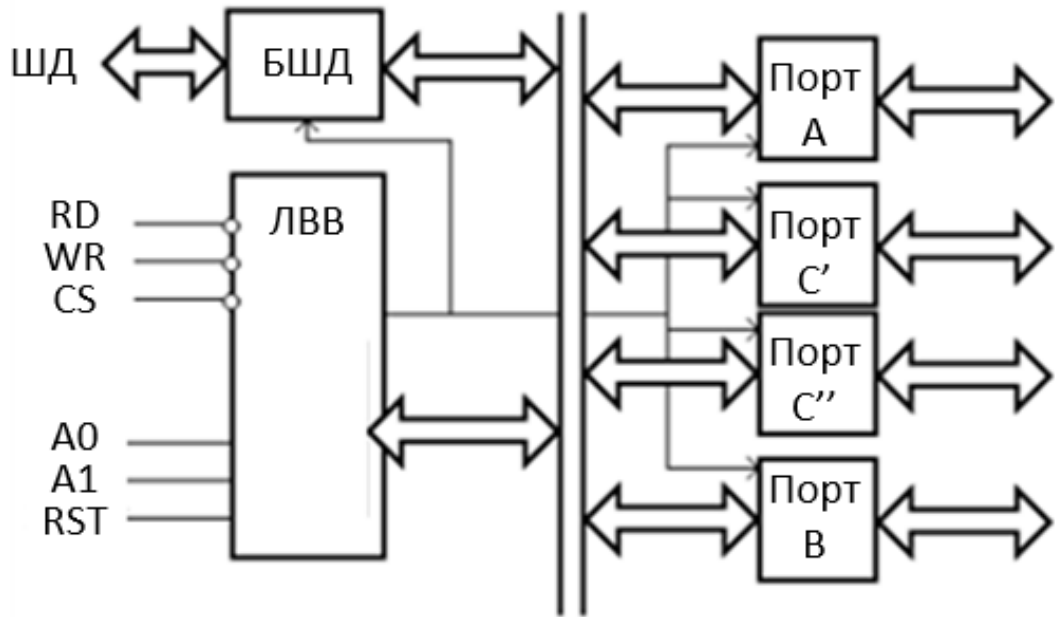


Рисунок 1.3 – Структурна схема ППА I8255A [3]

На схемі позначені основні складові периферійного адаптеру:

- БШД – буфер шини даних, відповідає за ввід та вивід даних із шини даних.
- ЛВВ – логіка вводу/виводу;
- Порт А, Порт В, Порт С – три 8-розрядні канали вводу/виводу (порт С – дозволяє побітове керування);
- RD – сигнал читання;
- WR – сигнал запису;
- CS – сигнал вибору мікросхеми;
- RESET – сигнал скидання;
- A0, A1 – сигнали адресації .

У таблиці 1.1 наведено використання ніжок A0 та A1 для адресації.

Таблиця 1.1 – Використання ніжок A0 та A1 для адресації [2]

A0	A1	Обраний регістр
0	0	Порт А
0	1	Порт В
1	0	Порт С
1	1	Регістр керуючого слова

ППА структурно розділена на дві групи: група А та група В. Група А включає в себе канал А, та старшу тетраду каналу С. Група В включає в себе канал В, та молодшу тетраду каналу С.

Кожний з трьох каналів може виконувати функції синхронного порту вводу чи виводу (режим 0). В асинхронному режимі (режим 1) може працювати лише канал А і канал В, так як для використання цих портів в цьому режимі беруться додаткові 3 лінії (для кожного з каналів) з порту С. Дві з них відповідають за сигнал рукостискання і один з них за запит переривання. Також є випадок асинхронної передачі даних- асинхронний двоспрямований (Режим 2). В цьому режимі може працювати лише порт А, так як з порту С використовуються 5 ліній для сигналів синхронізації.

Вибір режиму для ППА виконується за допомогою керуючого слова, яке вводиться в однойменний регістр. На рисунку 1.4 зображено формат керуючого слова.

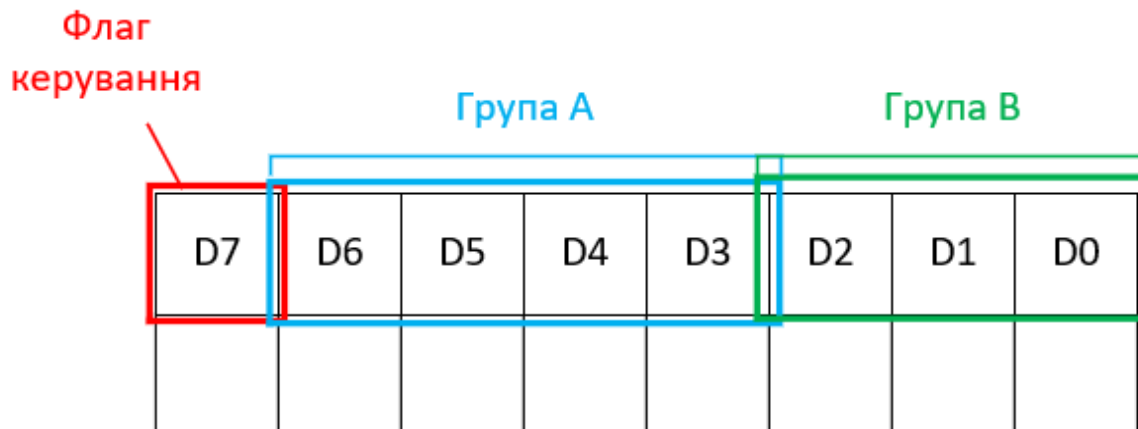


Рисунок 1.4 – Формат керуючого слова

- Старший розряд D7 при значенні 1 налаштований на вибір режиму, при значенні 0 – на керування портом С;
- D6,D5 – відповідає за режим порту А, при значенні 00, синхронний режим роботи, при 01, асинхронний односпрямований, при 10, асинхронний двоспрямований;
- D4 – відповідає за напрям порту А, при 0, на вивід, при 1, на ввід;
- D3 – відповідає за напрям старшої тетради порту С, при 0, на вивід, при 1 на ввід;
- D2 – відповідає за режим порту В, при 00, синхронний, при 01, асинхронний
- D1 – відповідає за напрям порту В, при 0, на вивід, при 1, на ввід;
- D0 – відповідає за напрям молодшої тетради порту С, при 0, на вивід, при 1 на ввід;

На рисунку 1.5 зображено слово управління розрядом порту С.



Рисунок 1.5 – Слово управління розрядом порту C

Якщо сигнали адресації обрані для запису у регістр керуючого слова ($A0 = 1$, $A1 = 1$), та сьомий біт регістра шини даних дорівнює 0, значення молодшого розряду шини даних заноситься у розряд порту C, який визначений у 1,2 і 3 розряді вхідного слова.

1.3 Особливості використання тренажеру ППА в дистанційному навчальному процесі

У наш час дистанційне навчання по зрозумілим причинам майже повністю замінило денне. У дистанційного навчального процесу є і переваги, і недоліки. Серед переваг можна виділити:

- Доступність та інтерактивність навчальних матеріалів;
- Зосередженість та залученість в навчальний процес (при бажанні);
- Зручність.

Навчальний процес в основному проходить у Zoom – хмарній платформі для проведення відеоконференцій. За розкладом викладачі запускають конференцію та студенти приєднуються і слухають лекцію, виконують лабораторні та практичні роботи. Але деякі речі дуже складно пояснити словами, для повного розуміння теми студенту необхідно власноруч практикуватися з тими чи іншими речами. Наприклад, у випадку з програмуванням студенту немає складності завантажити потрібну середу

розробку та почати програмувати, брати інформацію з методичних вказівок та лекцій. Але у деяких дисциплінах немає можливості дистанційно зрозуміти весь учбовий матеріал, саме через відсутність нормальної практики. Наприклад при дистанційному проходженні курсу «Проектування мікропроцесорних систем» виникали складнощі з розумінням роботи деяких мікросхем. При денному навчанні у викладачів була можливість видати студентам потрібні мікросхеми та продемонструвати як вони працюють, дати студентом самостійно розібратися, та здобути практичні навички. А при віддаленому учбовому процесі цієї можливості у студентів та викладачів немає. Саме тому і було прийнято рішення розробити навчальний тренажер.

Навчальний тренажер, який емулює роботу паралельного периферійного адаптеру Intel 8225A, допоможе студентам розібратися з пристроєм мікросхеми. Студенти зможуть, в умовах дистанційного навчання, власноруч програмувати ППА, розібратися з усіма його можливостями та недоліками. Мережева частина тренажера дозволить студентам зі свого домашнього комп'ютера виконувати задані викладачем задачі. Розроблена програма має підвищити якість дистанційного учбового процесу, по дисципліні «Проектування мікропроцесорних систем» у декілька разів.

1.4 Висновки

В цьому розділі був представлений паралельний периферійний адаптер Intel 8225A. Були визначені його основні характеристики та відомі аналоги. ППА має три канали вводу/виводу, чотири режими роботи, та є можливість програмувати кожний канал окремо.

Структурно мікросхема складається з наступних елементів:

- БШД – буфер шини даних, відповідає за ввід та вивід даних із шини даних.
- ЛВВ – логіка вводу/виводу;
- Порт А, Порт В, Порт С – три 8-розрядні канали вводу/виводу (порт С – дозволяє побітове керування).

Були визначені основні особливості використання тренажеру у дистанційному навчальному процесі.

2 Програмне забезпечення для дистанційного навчання

2.1 Zoom

Zoom – це частково безкоштовна хмарна платформа для проведення запланованих відеоконференцій. Дуже важлива програма для дистанційного учбового процесу. Завдяки цій програмі студенти мають можливість за розкладом слухати лекції, пояснення для лабораторних та практичних робіт, або заходити на консультації. На рисунку 2.1 зображено логотип Zoom.



Рисунок 2.1 – Логотип Zoom [7]

У кожного користувача є певний ідентифікатор, по якому, якщо запущена конференція, можуть приєднатися інші люди. У безоплатній підписці є можливість одночасного розміщення до 100 користувачів в одній кімнаті. Функціонал Zoom дозволяє:

1. Спілкуватися за допомогою мікрофону;
2. Транслювати картинку з камери;
3. Дивитися кількість учасників та їх імена;
4. Спілкувати та пересилати файли у чаті;
5. Транслювати картинку з камери;
6. Є можливість запису свого екрану.

На рисунку 2.2 Зображений інтерфейс Zoom з позначенням основних елементів.

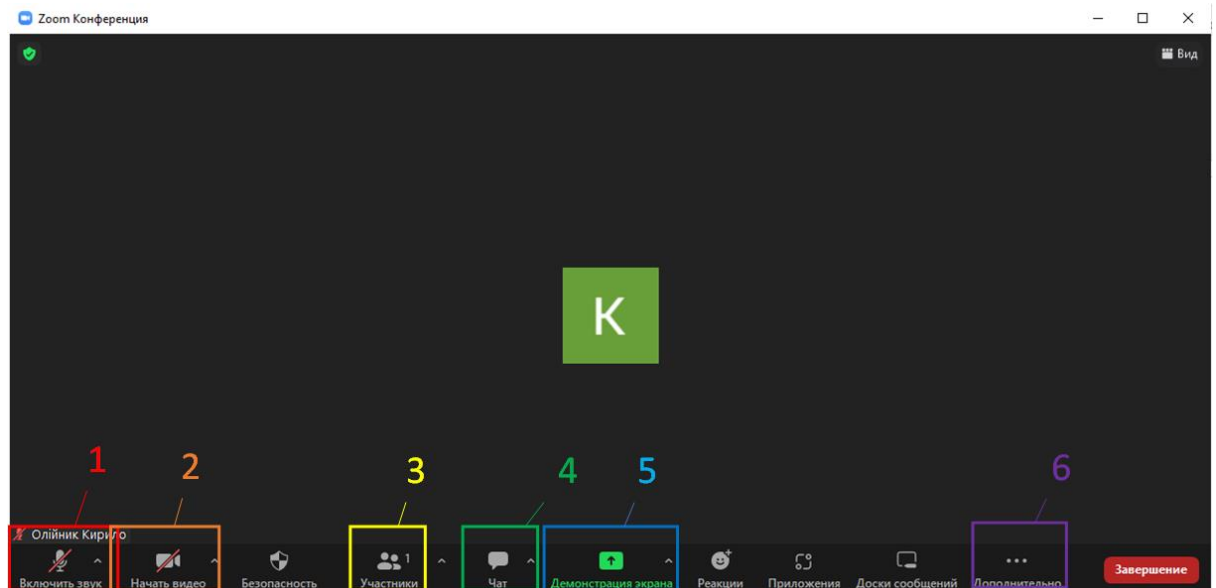


Рисунок 2.2 – Інтерфейс конференції Zoom

Особливостями даного програмного забезпечення є наступне:

- доступність (стандартної підписки вистачає як і для викладачів так і для студентів);
- можливість проведення лекцій та практичних занять;
- можливість передачі файлів.
- Але є у Zoom певні недоліки:
- під час демонстрації екрану дуже часто картинка відображається нестабільно, особливо якщо у лектора, або у студента повільний інтернет;
- немає можливості відстежувати відвідуваність занять;
- проблеми з безпекою.

Zoom – є невід’ємною частиною дистанційного навчального процесу. Дуже зручна, та головне, безкоштовна програма.

2.2 Lider

Lider.diit.edu.ua – це система керування курсами, розроблена на основі Moodle. Дуже зручна інтернет платформа, на якій викладачі розміщують завдання, тести, літературу, лекції і інше. Є необхідною складовою навчального процесу. На рисунку 2.3 зображений логотип Lider.



Рисунок 2.3 – Логотип інтернет-ресурсу Lider [6]

Функціонал лідеру дозволяє викладачу наступне:

1. Створювати курси;
2. Викладати лекції;
3. Проводити тестування;
4. Проводити екзамени;
5. Публікувати об'явлення;
6. Викладати файли (наприклад методичні вказівки або літературу);
7. Написати повідомлення;
8. Продивитись оцінки студента по курсу.

Lider систематизує учбовий процес, студент з легкістю може дізнатись які дисципліни в нього доступні, скільки лекцій він має пройти та скільки лабораторних робіт йому необхідно здати. Якщо виникають якісь питання то студент може написати повідомлення викладачу. На рисунку 2.4 зображена початкова сторінка студента.

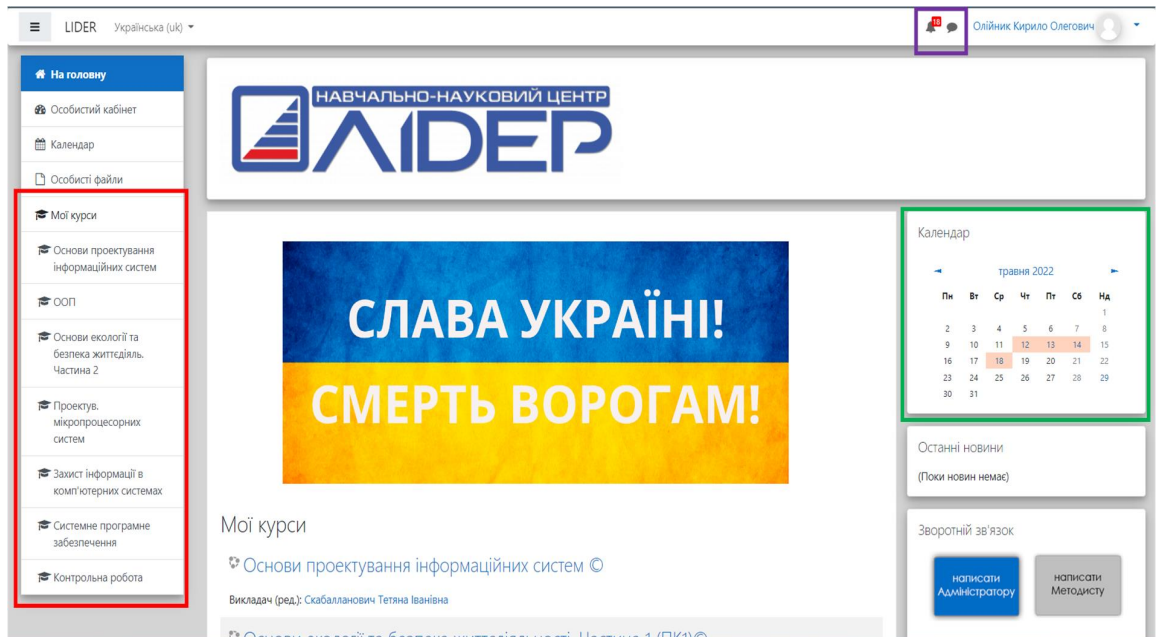


Рисунок 2.4 – Стартова сторінка студента

Червоним кольором позначені доступні студенту курси. Зеленим кольором зображений дуже зручний календар, по якому можна дізнатись коли і по яким предметам у студента дедлайни. Фіолетовим кольором відмічені сповіщення та повідомлення. У сповіщеннях можна подивитись чи перевірів викладач здані лабораторні чи ще ні.

Серед особливостей даного сайту можна зазначити наступне:

- систематизація усіх курсів, лекцій та завдань;
- можливість здачі екзаменів та різних тестів, лабораторних робіт;
- є можливість зв'язатись з викладачем.

Інтернет платформа Lider є основним засобом для дистанційного учбового процесу. Дуже зручна система керування курсами.

2.3 Telegram

Telegram – це система миттєвого обміну повідомленнями, яка дозволяє обмінюватися текстовими, аудіо та відеоповідомленнями, картинками та файлами багатьох форматів. На рисунку 2.5 зображений логотип Telegram.



Рисунок 2.5 – Логотип Telegram

Серед корисних функцій месенджеру можна позначити наступне:

1. Можливість підтримувати зв'язок студента з викладачем;
2. Можливість створювати групи;
3. Можливість передавати файли.

На рисунку 2.6 зображений інтерфейс Telegram.

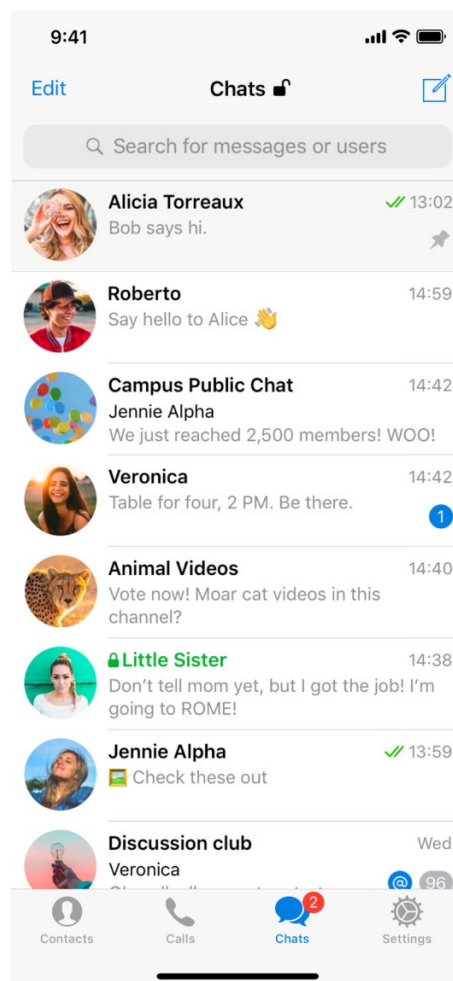


Рисунок 2.6 – Інтерфейс Telegram. [8]

У дистанційному навчанні даний додаток є дуже ефективним інструментом зв'язку між викладачем та студентом. Викладач може створити групу по своїй дисципліні, та додати туди всіх учнів, ділитися посиланням на конференцію у Zoom, або робити якісь оголошення.

Серед особливостей даного програмного забезпечення можна виділити наступне:

- висока зручність;
- висока швидкість передачі інформації;
- безпечність.

Telegram – дуже зручний месенджер, який покращує дистанційний учбовий процес завдяки своєму функціоналу та особливостям.

2.4 Порівняння програм для дистанційного навчання

Основними програмами для дистанційного навчання є:

- Zoom;
- Lider;
- Telegram.

Усі вони необхідні для ефективного та зручного учбового процесу. Кожна програма має свої особливості та переваги, саме тому мета використання у кожної з них різна. У таблиці 2.1 наведено порівняння використовуваного програмного забезпечення для дистанційного навчання.

Таблиця 2.1 – Порівняння функціоналу програмного забезпечення для дистанційного навчання

Програмний продукт	Zoom	Lider	Telegram
Проводження лекції та практичних занять	+	-	-

Продовження таблиці 2.1

Відправка повідомлень	+	+	+
Передача файлів	+	+	+
Публікація лекцій та вказівок для лабораторних робіт	-	+	+
Можливість створювати та проходити тести, здавати екзамени	-	+	-
Створення груп з студентами	-	-	+
Трансляція екрану	+	-	-

Для мережевої роботи навчального тренажеру, необхідно мати можливість транслювати свій екран. Ця функція потрібна для перевірки знань студента, або для того, щоб викладач міг продемонструвати як потрібно працювати з тренажером.

Zoom необхідний для проведення онлайн лекцій, консультацій, лабораторних та практичних занять, де потрібне живе спілкування. Але через його недоліки, а саме, через погану стабільність передачі картинки екрану, використання Zoom для демонстрації роботи студента з тренажером не є доцільним.

Lider дуже зручна система, необхідна для систематизації усіх курсів, для публікації літератури, лекцій, методичних вказівок, або оголошень, які необхідні для освітнього процесу, проходження тестів та здачі екзаменів. Але у нього немає можливості транслювати екран.

Telegram дозволяє швидко зв'язатись з викладачем або з студентом, задати питання або попросити провести консультацію, викладач може створити групу з потоком студентів по своїй дисципліні та обмінюватися корисною інформацією. Функція демонстрації екрану також відсутня.

Серед проаналізованого програмного забезпечення можливість передачі зображення з екрану є тільки в Zoom, але через описані у пункті 2.1 недоліки, використання цієї програми не є ефективним. Саме через це необхідно створити власну мережеву частину тренажеру, яка буде передавати дані з клієнту до серверу байтами, що забезпечить стабільну роботу.

2.5 Висновки

Кожна з використовуваних програм необхідна для дистанційного навчального процесу, у кожній з них є власні унікальні функції, але жодна з них не підходить для використання замість мережевої частини навчального тренажеру. Весь функціонал та особливості розглянутого програмного забезпечення, а також його недоліки, будуть враховані при розробці програми.

3 Розробка програмного забезпечення мережевої частини тренажера

3.1 Загальна структура тренажера та його особливості

У наш час неможливо уявити навчальний процес без можливості дистанційної роботи. Саме для цього було розроблено навчальний тренажер, який емулює роботу паралельного периферійного адаптеру, з функцією дистанційної перевірки знань.

Розроблена програма складається за наступних елементів:

- Елементи керування;
- Елементи виводу інформації;
- Серверна частина тренажера;
- Клієнтська частина.

На рисунку 3.1 зображена структура навчального тренажера.

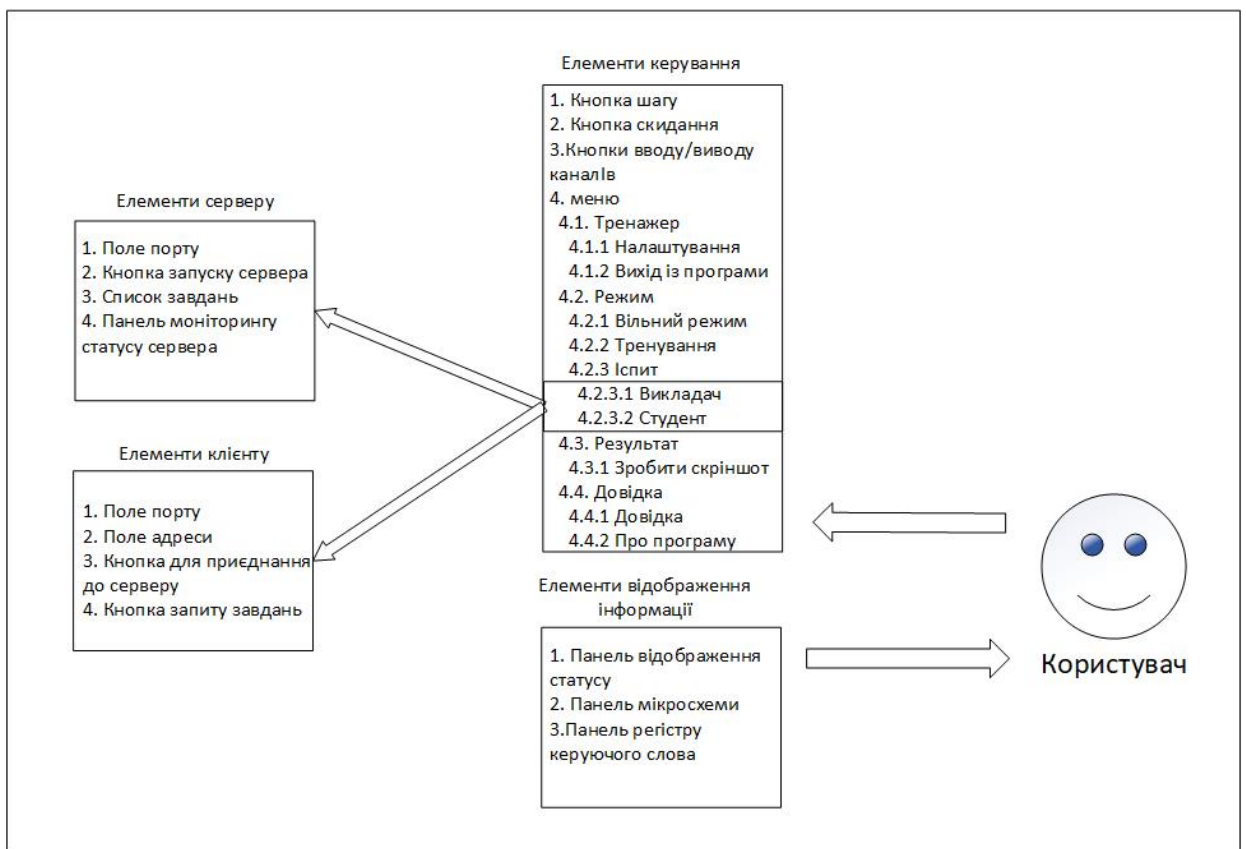


Рисунок 3.1 – Структура тренажера

Елементи керування необхідні для вводу або виводу інформації (кнопки вводу/виводу каналів), вибору режиму роботи адаптеру, або скидання до початкового стану (кнопка скидання). Робота тренажера виконується покроково, для переходу на наступний крок була додана кнопка кроку.

Вся необхідна інформація виводиться на елементи відображення. Панель статусу показує на який режим налаштований адаптер (синхронний, асинхронний для порту В, синхронний, асинхронний однонаправлений, асинхронний двонаправлений для порту А), та напрям (ввід чи вивід). На панелі реєстра керуючого слова можна дізнатись його зміст побайтно.

У меню можна закрити програму, обрати режим роботи (вільний, тренування, іспит), зберегти скріншот результату у bmp форматі, відкрити довідку та отримати інформацію про програму.

Для запуску серверу необхідно обрати режим «Іспит», пункт «Викладач». Відкриється панель серверу, на якій можна ввести порт, з яким буде працювати сервер, запустити сервер, та відправити завдання студенту.

При виборі режиму «Іспит», пункт «Студент» відкривається клієнтська частина тренажера, за допомогою якої можна ввести необхідну адресу та порт серверу, приєднатися, та запросити завдання у викладача.

На рисунку 3.2 зображена структура взаємодії клієнта та сервера.

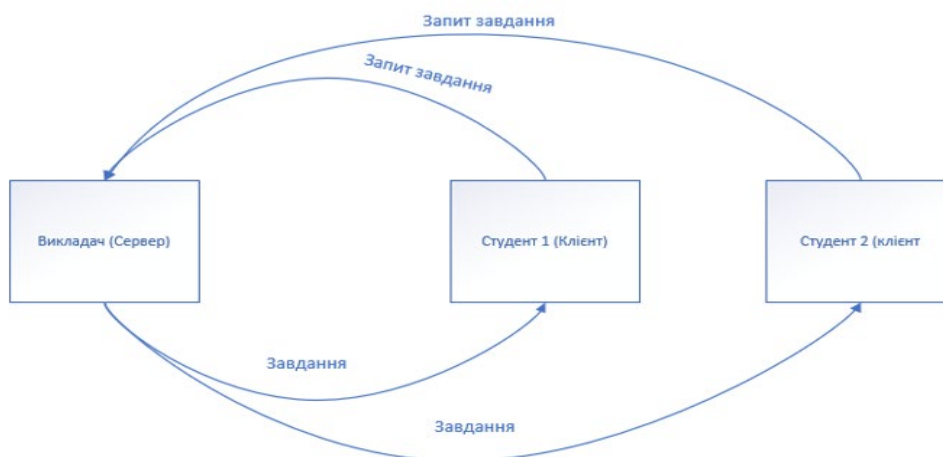


Рис 3.2 – Структура взаємодії клієнта та сервера

Розроблена програма працює на будь якій операційній системі, працездатність була перевірена на Windows, Mac і Linux. Тренажер складається з файлу формату .exe та файлу довідки формату .chm. Розмір файлу програми – 93 МБ, а розмір довідки – 12 КБ .

Для того, щоб працювати з мережевою частиною, а саме, запустити сервер, необхідно щоб був відкритий порт від 1000 до 10000.

3.2 Мережева частина тренажера, мережеві частини інтерфейсу програми

Для дистанційної роботи програми було розроблено клієнтську та серверну частину. Студент має можливість запросити завдання у викладача. Викладач може відправити завдання студенту, у реальному часі спостерігати процес виконання обраного завдання, та отримувати звіт про виконання завдання студентом.

На рисунку 3.3 зображена серверна частина тренажера.

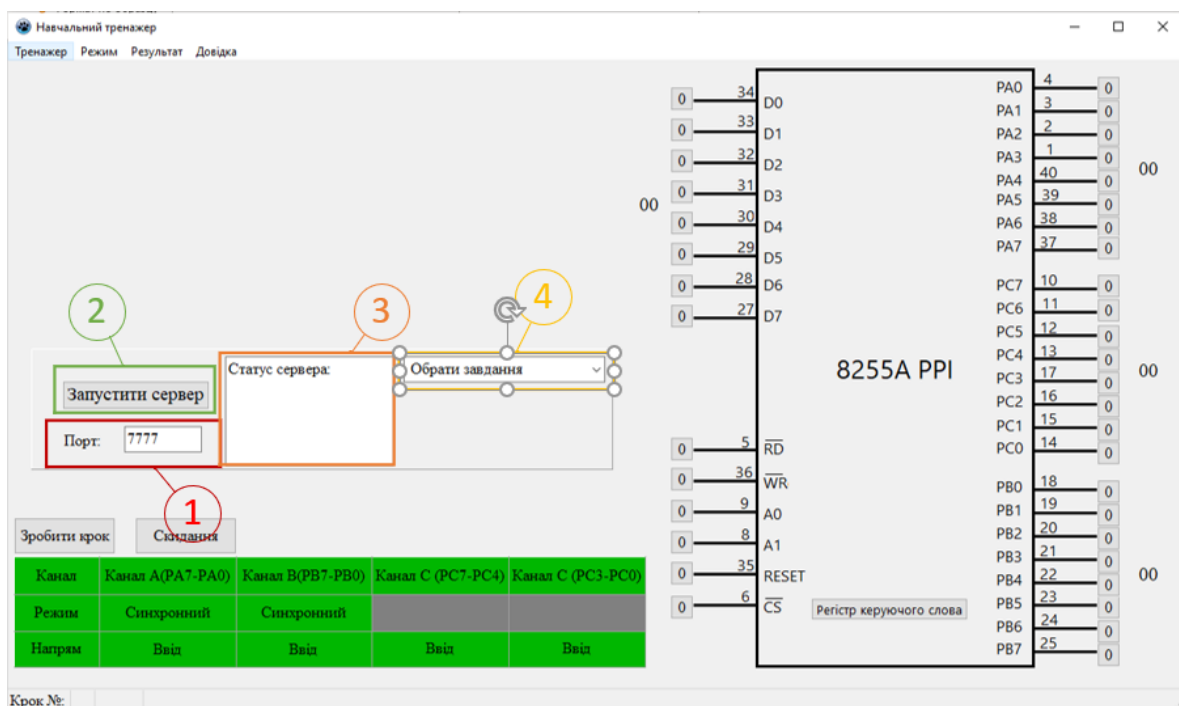


Рисунок 3.3 – Серверна частина тренажера

Сервер має наступні елементи інтерфейсу:

1. Поле для вводу порту;
2. Кнопку «запустити сервер»;
3. Вікно для відображення статусу сервера;
4. Список завдань, які можна відправити студенту (клієнту).

Для запуску серверу викладач має зайти у пункт меню «режими», обрати «іспит», та «викладач». Після цього відкривається серверна частина тренажеру. Викладач має вписати номер порту, та натиснути «запустити сервер». У полі статусу сервера прийде повідомлення «Сервер створено». Після цього студент може приєднатись до серверу, та отримати завдання. На рисунку 3.4 зображено вигляд повідомлення про створення серверу.

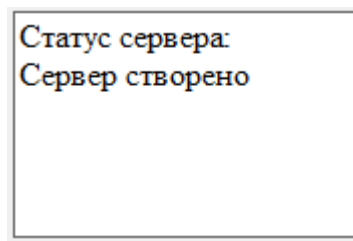


Рисунок 3.4 – Повідомлення про створення серверу

Клієнт має наступні елементи інтерфейсу:

1. Поле для вводу адреси серверу та порту;
2. Кнопку «приєднатися» до серверу;
3. Вікно для спостереження статусу взаємодії клієнта та сервера;
4. Кнопку «отримати завдання», яка відправляє запит до викладача(сервера).

На рисунку 3.5 зображено клієнтську частину тренажеру.

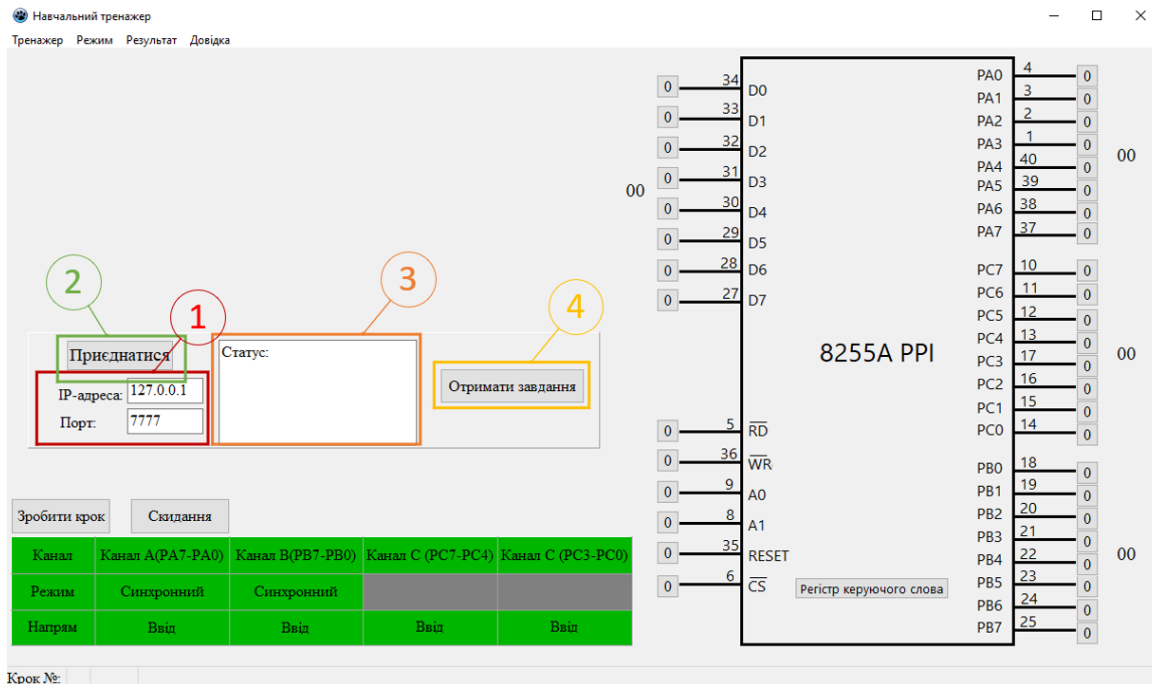


Рисунок 3.5 – Клієнтська частина тренажера

Коли викладач створив сервер, студенту необхідно ввести IP-адресу, та порт у відповідні поля. Після цього він може приєднатися до серверу натиснувши кнопку «приєднатися». У текстове поле зі статусом прийде повідомлення про успішне приєднання, викладачу також прийде повідомлення. На рисунку 3.6 та 3.7 зображені повідомлення про приєднання до серверу у студента та викладача.

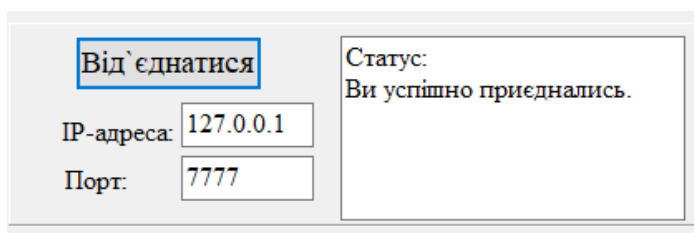


Рисунок 3.6 – Повідомлення про приєднання до серверу (студент)

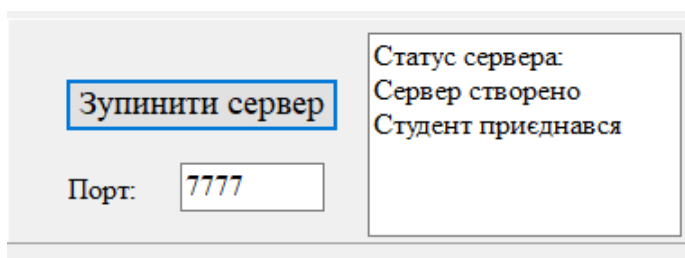


Рисунок 3.7 – Повідомлення про приєднання до серверу (викладач)

Після успішного приєднання, студент може відправити запит на завдання, для цього потрібно натиснути кнопку «Запросити завдання». Викладач отримає повідомлення, обере зі списку потрібний варіант, та відправить його студенту.

На рисунку 3.8 та 3.9 зображено вибір завдання викладачем, та отримання студентом.

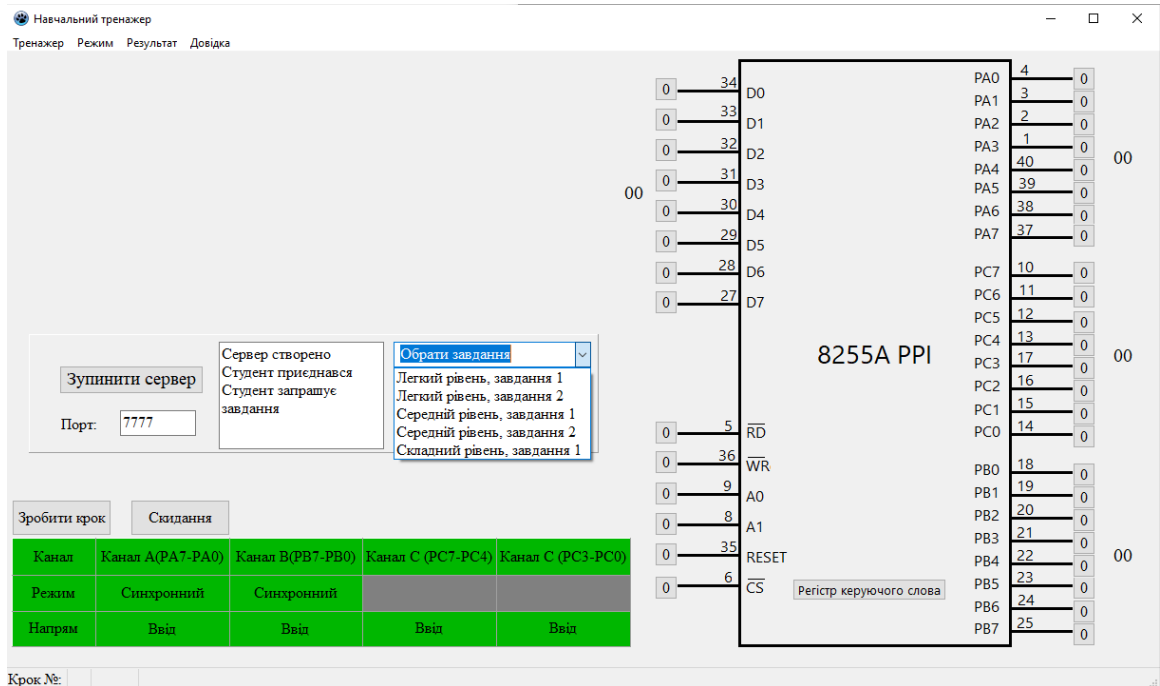


Рисунок 3.8 – Вибір варіанту завдання для студента (сервер)

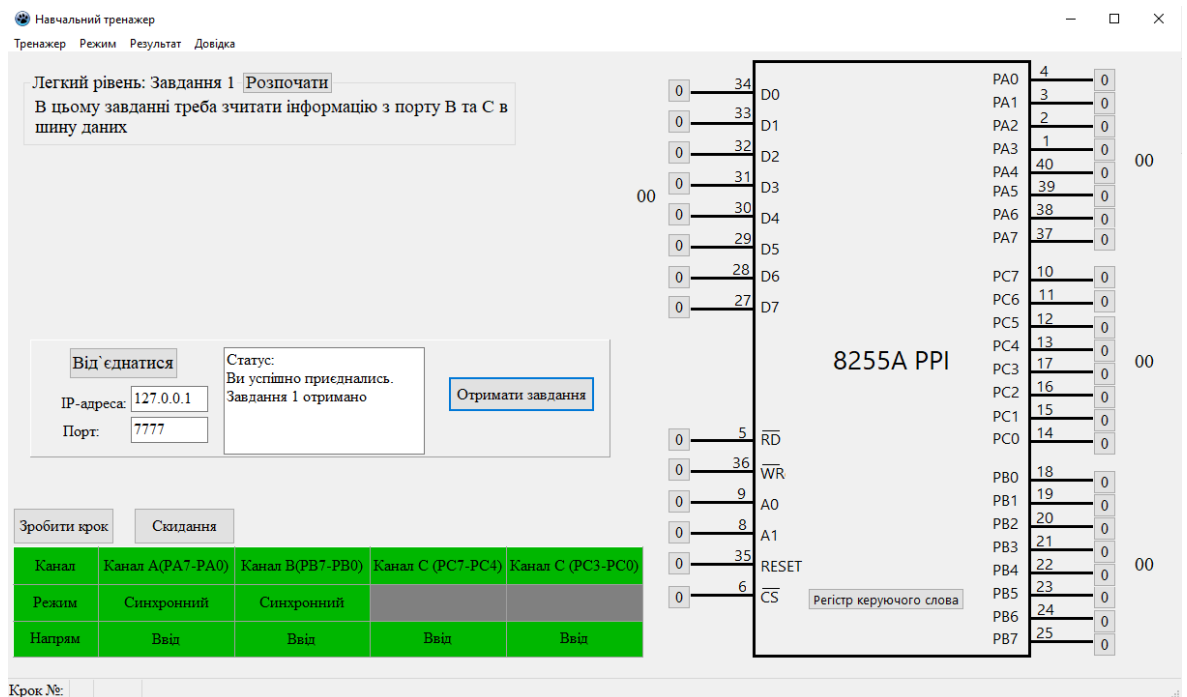


Рисунок 3.9 – Отримання завдання студентом (клієнт)

Після успішного виконання завдання, викладачу прийде повідомлення з вашим результатом. На рисунку 3.10 зображено виконання студентом завдання, та на рисунку 3.11 зображено повідомлення про виконання завдання на сервері.

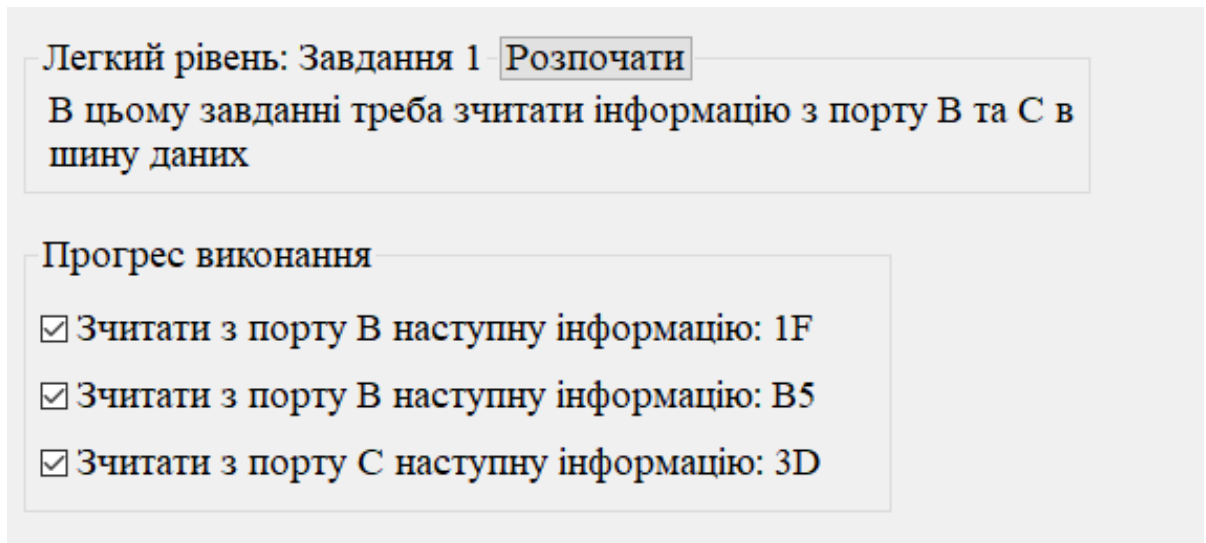


Рисунок 3.10 – Студент виконав завдання

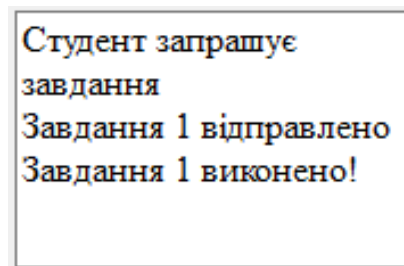


Рисунок 3.11 – Повідомлення на сервері

3.3 Алгоритм обміну інформацією між окремими частинами тренажера

На рисунку 3.12 зображений спрощений алгоритм обміну інформацією.

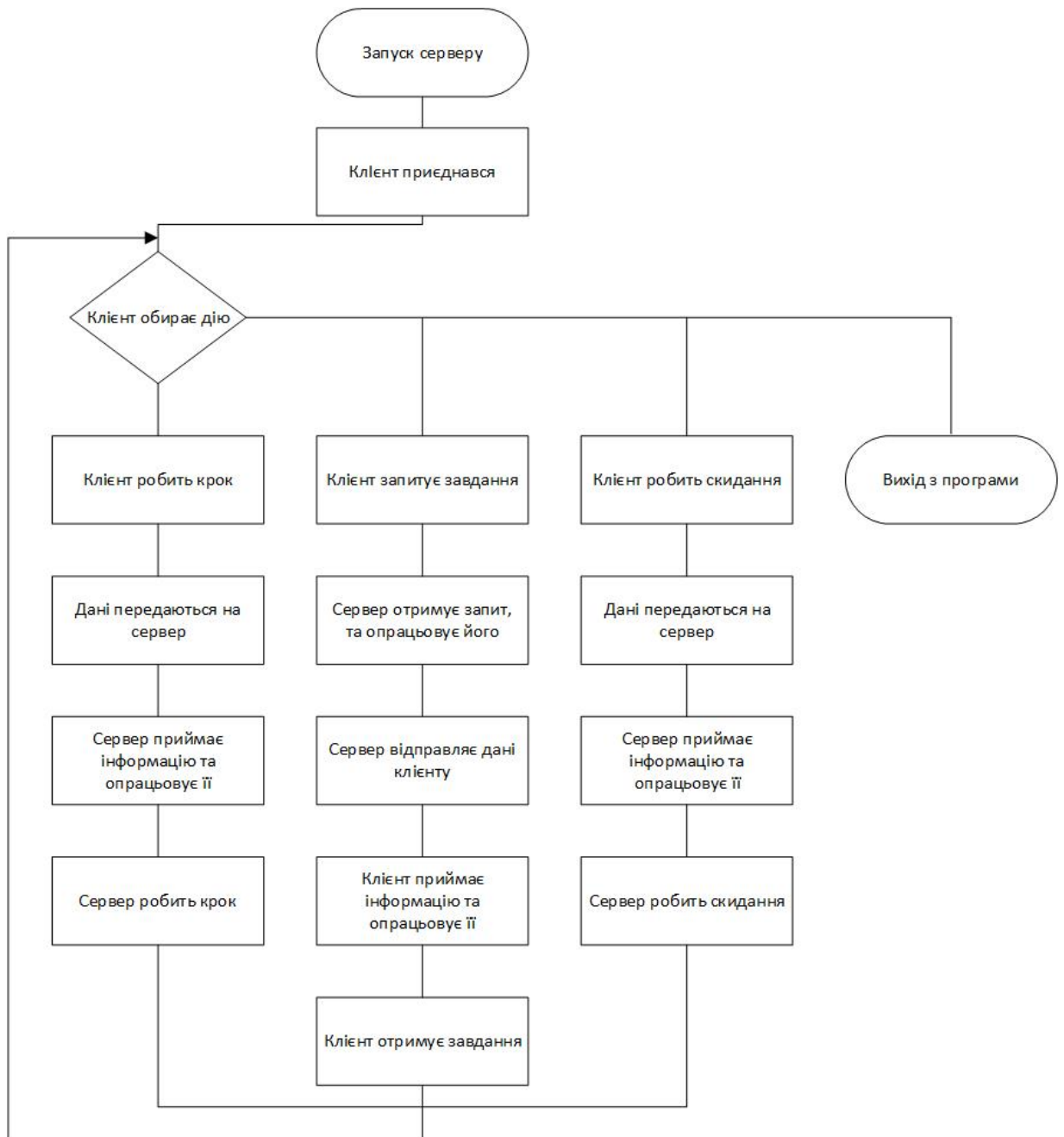


Рисунок 3.12 – Спрощений алгоритм обміну інформацією

Викладач вводить потрібний порт, та запускає сервер. Студент вводить IP-адресу і порт, та приєднується до серверу. Після успішного приєднання усі маніпуляції студента з паралельним периферійним адаптером будуть відображатися і у викладача. Якщо клієнт вводить необхідні йому дані у поля каналів, та натискає кнопку «наступний крок», викладач отримує повідомлення, у якому описані всі дії які виконав студент, відповідно змінює поля ППА, та

також робить крок. Завдяки цьому викладач завжди може знати що саме зараз робить студент.

Якщо студент натискає кнопку «отримати завдання», то від клієнта до серверу відсилається відповідний запит. Викладач отримує повідомлення про те, що студент бажає отримати завдання, та обирає зі списку потрібний варіант. Після цього сервер надсилає до клієнту повідомлення, в якому вказано номер завдання, яке відкриється у студента. При правильному виконанні завдання, викладач отримує повідомлення з результатом.

Якщо у клієнтській частині була натиснута кнопка скидання, клієнт передає повідомлення з інформацією про це серверу. Сервер опрацьовує цю інформацію, після чого робить скидання до початкового стану і у себе.

Вся інформація з клієнту до серверу надходить у вигляді строки з 41 символу, де кожний елемент відповідає конкретній дії або полю, яке необхідно змінити. На рисунку 3.13 зображена структура строки, яку клієнт надсилає серверу.

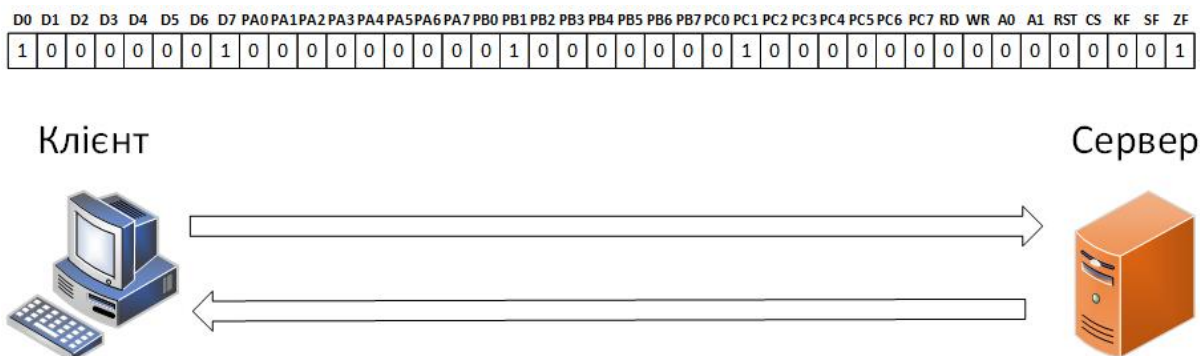


Рисунок 3.13 – Структура строки, яку клієнт надсилає серверу

Перші 32 символи описують склад полів вводу/виводу інформації з портів. Якщо у студента D0 дорівнює одиниці, то і у сервера D0 буде дорівнювати одиниці. Наступні 6 символів відповідають за налаштування адаптеру (RD, WR, A0, A1, RST, CS). Для того щоб сервер розумів яку дію виконав студент, було добавлено 3 прапорці, які передають серверу необхідну інформацію.

- KF – прапорець, який відповідає за «наступний крок». Якщо студент зробив крок, значення KF змінюється на 1, це дає змогу серверу зрозуміти що був зроблений крок, та виконати цю дію у себе.
- SF – прапорець, який відповідає за «скидання». Коли студент робить скидання до початкового стану, значення SF змінюється на 1, та серверна частина також виконує операцію скидання.
- ZF – прапорець, який відповідає за функцію «отримати завдання». При натисканні однойменної кнопки значення ZF змінюється на 1, викладачу приходить повідомлення що студент запитав завдання. На рисунку 3.12 продемонстровано зміст строки в той момент, коли студент просить видати завдання.

У випадку, коли студент надсилає запит на завдання, та $ZF = 1$, серверу необхідно надіслати відповідь до клієнта. У серверній частині, після отримання повідомлення про запит завдання, викладачу необхідно обрати потрібний варіант зі списку. При виборі варіанту сервер надсилає до клієнта сигнал, у числовому значення, яке означає номер завдання, яке було обрано. Клієнт кожні 100 мс перевіряє чи прийшла відповідь, та якщо прийшла, відповідно до номеру який він отримав, виводиться завдання.

3.4 Розробка мережевої частини ПЗ тренажера

Для розробки програми було використано відкриту середу розробки на мові програмування Object Pascal – Lazarus, та бібліотеку мережевих компонентів – Internet direct (Indy).

Для створення клієнт-серверної частини програми був обраний мережевий протокол TCP. Мережева модель передачі даних TCP є більш надійною ніж UDP, у таблиці 3.1 наведено порівняння.

Таблиця 3.1 – Порівняння протоколів TCP та UDP [9]

	Протокол TCP	Протокол UDP
Тип з'єднання	Орієнтований на з'єднання	Без прив'язки до з'єднання
Установка послідовності	Так	Ні
Де використовують	Електронна пошта, завантаження, спільне використання файлів.	Потокова передача аудіо та відеофайлів

Для реалізації серверу був використаний компонент Indy IdTCPServer, який зображений на рисунку 3.14. На рисунку 3.15 зображені властивості цього компоненту.

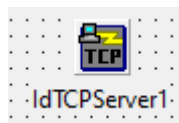


Рисунок 3.14 – Компонент IdTCPServer

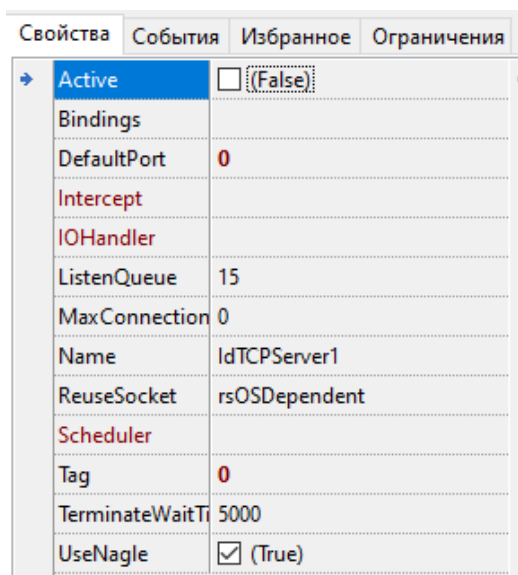


Рисунок 3.15 – Властивості компоненту IdTCPServer

Основні властивості:

- Active, відповідає за запуск серверу, при значенні True – сервер працює, False – сервер зупинений;
- Bindings, відповідає за порт та IP-адресу, з якими сервер буде працювати.

На рисунку 3.16 зображена властивість серверу Bindings.

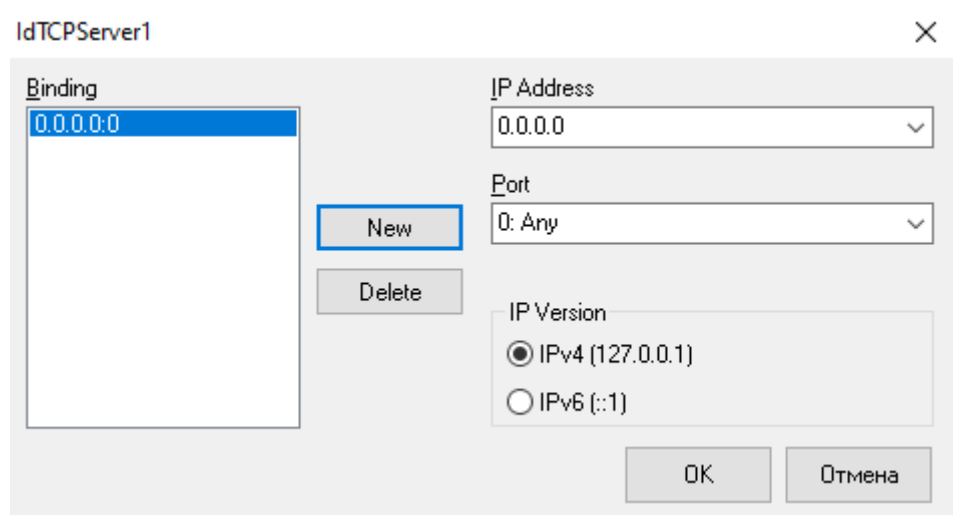


Рисунок 3.16 – IdTCPServer, властивість Bindings

Для вводу порту було використано компонент edit, який дозволяє вписати значення, та кнопка, завдяки якій сервер починає роботу. При натисненні властивість Active стає True, та у Bindings вписується значення поля edit (порт). IP- адреса за замовченням 0.0.0.0, це значить що до сервера може приєднатися клієнт з будь-яким IP.

Описати логіку роботи сервера потрібно у події OnExecute. Була написана процедура, яка приймає всі дані та опрацьовує їх. Також були використані подія OnConnect (спрацьовує при підключенні клієнту до серверу) та OnDisconnect (спрацьовує при відключення клієнту до серверу) для виводу повідомлення про успішне з'єднання з клієнтом. На рисунку 3.17 зображені події компоненту сервера.

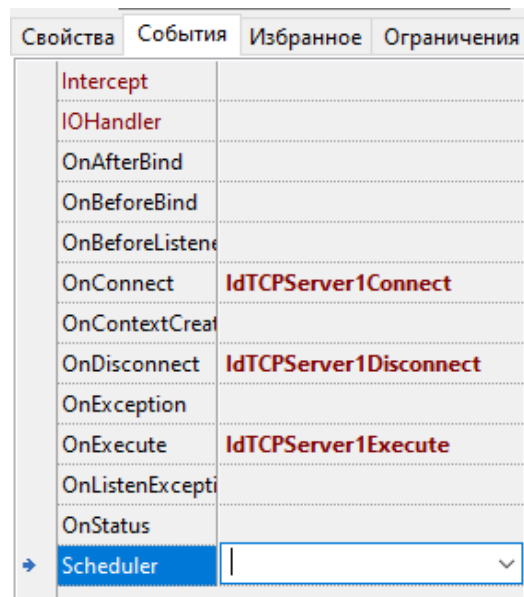


Рисунок 3.17 – Події компонента IdTCPServer

Для реалізації клієнту був використаний компонент Indy IdTCPClient, який зображений на рисунку 3.18. На рисунку 3.19 зображені властивості цього компонента.

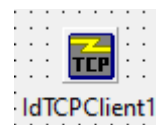


Рисунок 3.18 – Компонент IdTCPClient

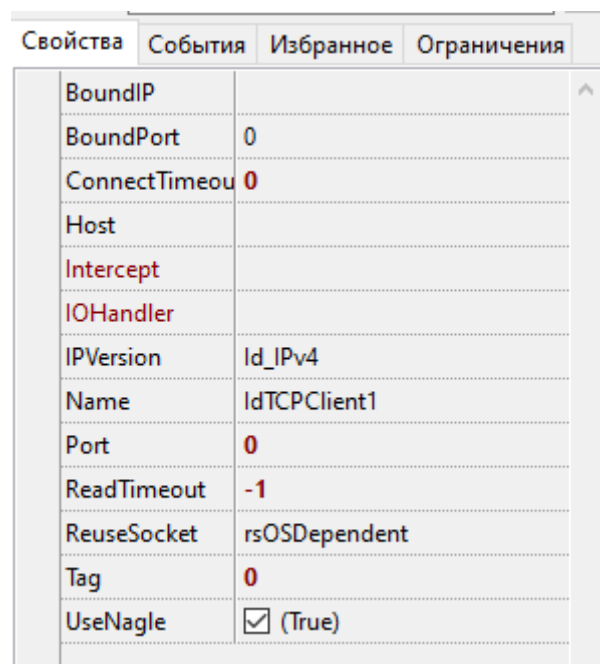


Рисунок 3.19 – Властивості компоненту IdTCPClient

Основні властивості:

- Host – відповідає за IP-адресу сервера, до якого потрібно підключитися;
- Port – відповідає за порт серверу.

Для вводу порту та адреси було використано компоненти edit, в які користувач може ввести необхідні значення. За підключення до серверу відповідає кнопка «Приєднатися». При натисканні спрацьовує функція `Indy IdTCPClient.connect`, та клієнт успішно з'єднується з сервером.

Для реалізації прийому номера завдання від сервера був добавлений компонент Lazarus – Timer, який зображений на рисунку 3.20. На рисунках 3.21 та 3.22 зображені властивості та події компоненту Timer.

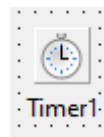


Рисунок 3.20 – компонент Timer

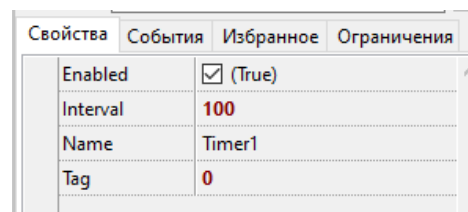


Рисунок 3.21 – Властивості компоненту Timer

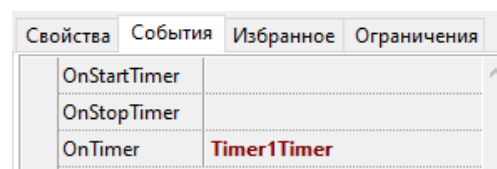


Рисунок 3.22 – Події компоненту Timer

Коли `Enabled = true`, таймер активний. Кожні 100 мс спрацьовує подія `OnTimer`. Написана процедура, яка кожній тік таймеру перевіряє чи надійшло з серверу повідомлення, та якщо надійшло, опрацьовує його.

3.5 Пункти довідки, присвячені мережевій частині

3.5.1 Розробка довідки

За допомогою мови HTML було розроблено довідку. Безкоштовна програма HTML HELP WORKSHOP від Microsoft Corporation дала змогу провести конвертацію із html формату у довідниковий файл з розширенням chm. Файл довідки було включено у тренажер. Відкрити його можна у пункті меню, довідка. На рисунку 3.23 зображено процес конвертації файлу довідки.

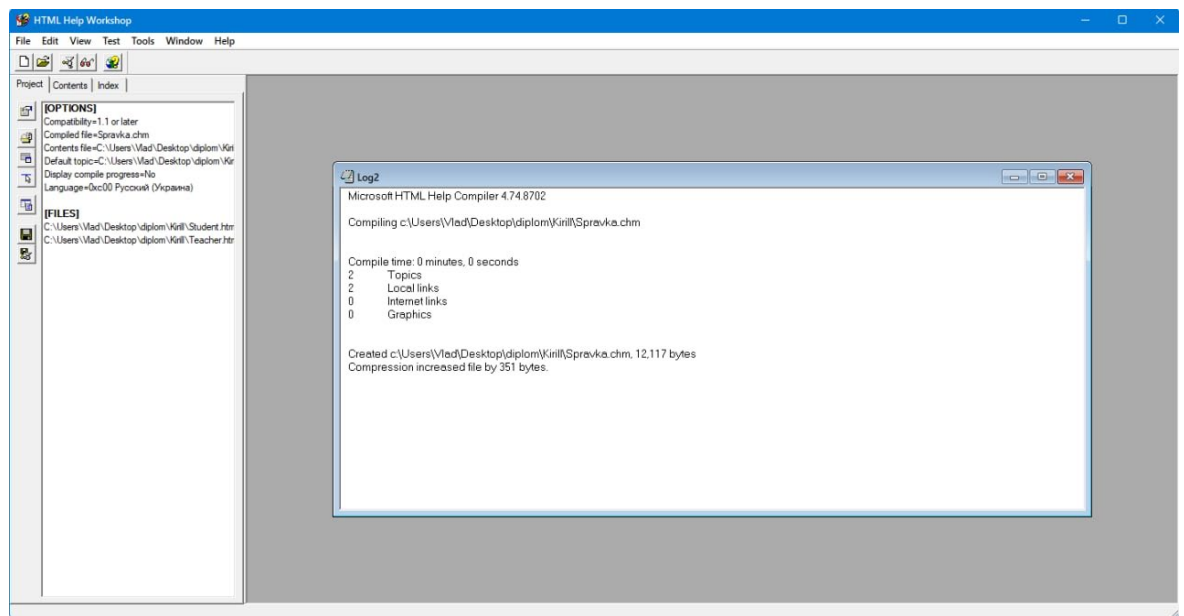


Рисунок. 3.23 – Процес конвертації файлу довідки

Довідка включає в себе наступні розділи:

- Інструкція для викладача;
- Інструкція для студента.

На рисунку 3.24 зображений загальний вигляд довідки.

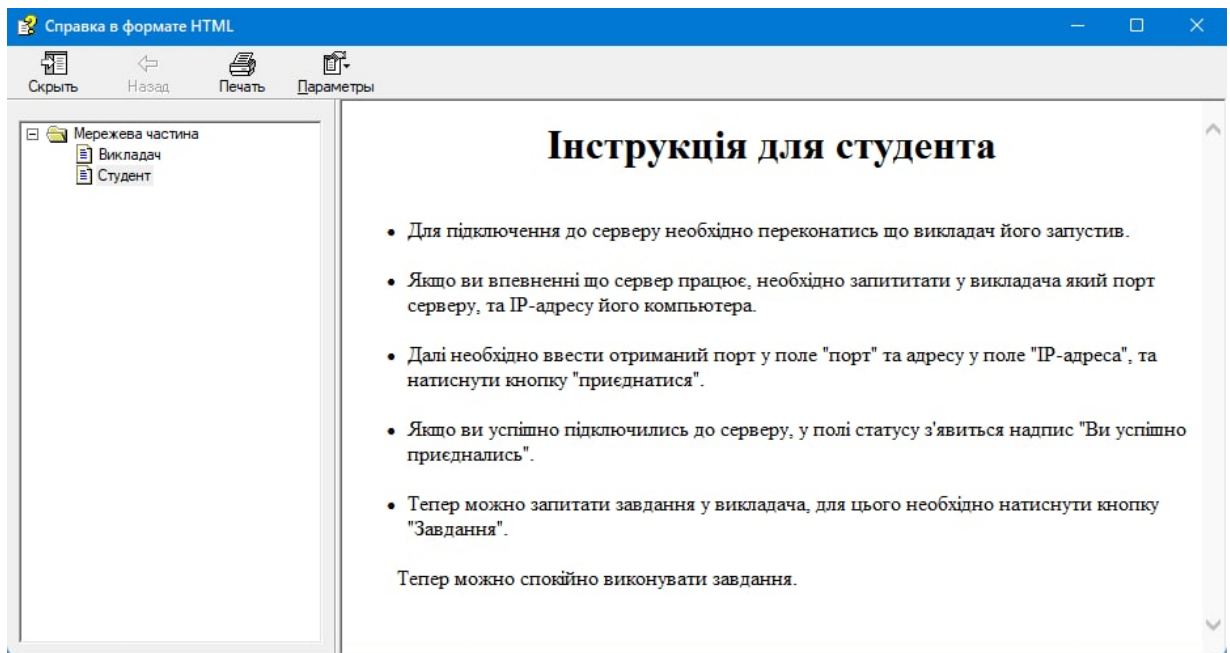


Рисунок. 3.24 – Загальний вигляд файлу довідки

Довідка допоможе розібратись як працювати з тренажером, як створити або приєднатись до серверу, та допоможе краще ознайомитись з паралельним периферійним адаптером I8225.

3.5.2 Текстова частина довідки

3.5.2.1 Запуск тренажера у мережевому режимі у якості сервера.

Для здійснення запуску серверу, викладачу необхідно в меню обрати пункт «Режими», «Іспит» та «Викладач». Програма відкриє серверний інтерфейс. Після цього необхідно ввести та порт, за яким буде працювати сервер та натиснути кнопку «запустити сервер». Якщо сервер запущено успішно, то в полі статусу сервера прийде повідомлення «Сервер запущено».

3.5.2.2 Запуск тренажера у мережевому режимі в якості клієнта.

Для того, щоб приєднатись до серверу, необхідно обрати пункт меню «Режими», «Іспит» та «Студент». Після того як викладач запустив сервер, студенту необхідно ввести IP-адресу та порт серверу і натиснути кнопку «Підключитися». Якщо операція виконана успішно, то в поле статусу прийде повідомлення «Ви успішно приєдналися до серверу».

3.6 Висновки

В цьому розділі було описано розробку мережевої частини тренажеру. Спочатку була повністю продумана структура програми та створений алгоритм, по якому і відбувалась реалізація проекту.

Програма складається з наступних елементів:

- Елементи керування;
- Елементи виводу інформації;
- Серверна частина тренажеру;
- Клієнтська частина.

Проект було розроблено у відкритій середі розробки на мові програмування Object Pascal – Lazarus, мережеву частину було зроблено завдяки бібліотеці мережевих компонентів – Internet direct (Indy).

З тренажером можна працювати як і в офлайн режимі, так і в онлайн режимі. Була повністю реалізована клієнтська частина (для студента) та серверна частина (для викладача) які взаємодіють між собою за допомогою протоколу передачі даних TSP. Передача даних виконується у вигляді строки – де кожен символ відповідає якійсь операції.

ВИСНОВКИ

В результаті виконання бакалаврської роботи було оглянуто теоретичний матеріал по паралельному периферійному адаптеру Intel 8225A, Було проаналізовано використовуване програмне забезпечення для дистанційного навчання, було розроблено алгоритм, протокол взаємодії між клієнтом і сервером. Розроблено мережеву частини. Було виконано всі поставлені задачі, а саме:

1. Було розроблено клієнтську частину програми;
2. Розроблено серверну частину;
3. Було програмно реалізовано можливість передавати інформацію з клієнта на сервер;
4. Реалізовано передачу завдання з серверу на клієнт.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Периферийные устройства микропроцессорной электроники: методические рекомендации по выполнению лабораторных работ для студентов направления подготовки 200100.62 «Приборостроение» / Е.В. Сыпин, Е.С. Повернов, О.Ю. Якушева; Алт. гос. техн. ун-т, БТИ. – Бийск: Изд-во Алт. гос. техн. ун-та, 2013. – 52 с.
2. Intel 8255 [Электронный ресурс] – Режим доступа: https://uk.wikipedia.org/wiki/Intel_8255
3. Архітектура комп'ютерів. Проектування інформаційно-керуючих систем на базі однокристальних ЕОМ.[Текст] : навчальний посібник у 2 ч. Ч.1/ Ю.І.Хмарський; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна.- Днепропетровськ, 2015.-100 с.
4. Программируемый периферийный адаптер 8255А (KP580BB55A) [Электронный ресурс] – Режим доступа: <http://www.cnc-lab.com/bidi8255.htm>
5. 8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE, 1985, 120с
6. НАВЧАЛЬНО-НАУКОВИЙ ЦЕНТР ЛІДЕР [Електронний ресурс] – Режим доступа: <https://lider.diiit.edu.ua/>
7. ZOOM [Электронный ресурс] – Режим доступа: <https://zoom.us/>
8. Telegram [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Telegram>
9. Чем отличается протокол TCP от UDP [Электронный ресурс] – Режим доступа: <http://pyatilistnik.org/chem-otlichaetsya-protokol-tcp-ot-udp/>
10. Медвідь В.Р. Проектування мікропроцесорних систем керування : навчальний посібник, перевидання / Медвідь В.Р., Пісціо В.П., Козбур І.Р. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 360 с