

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"



**АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ
СИСТЕМ**

Лабораторна робота № 3

Виконав:
ст. гр. КІ-401
Медвідь Д.С
Прийняв:
Федак П.Р.

Львів 2024

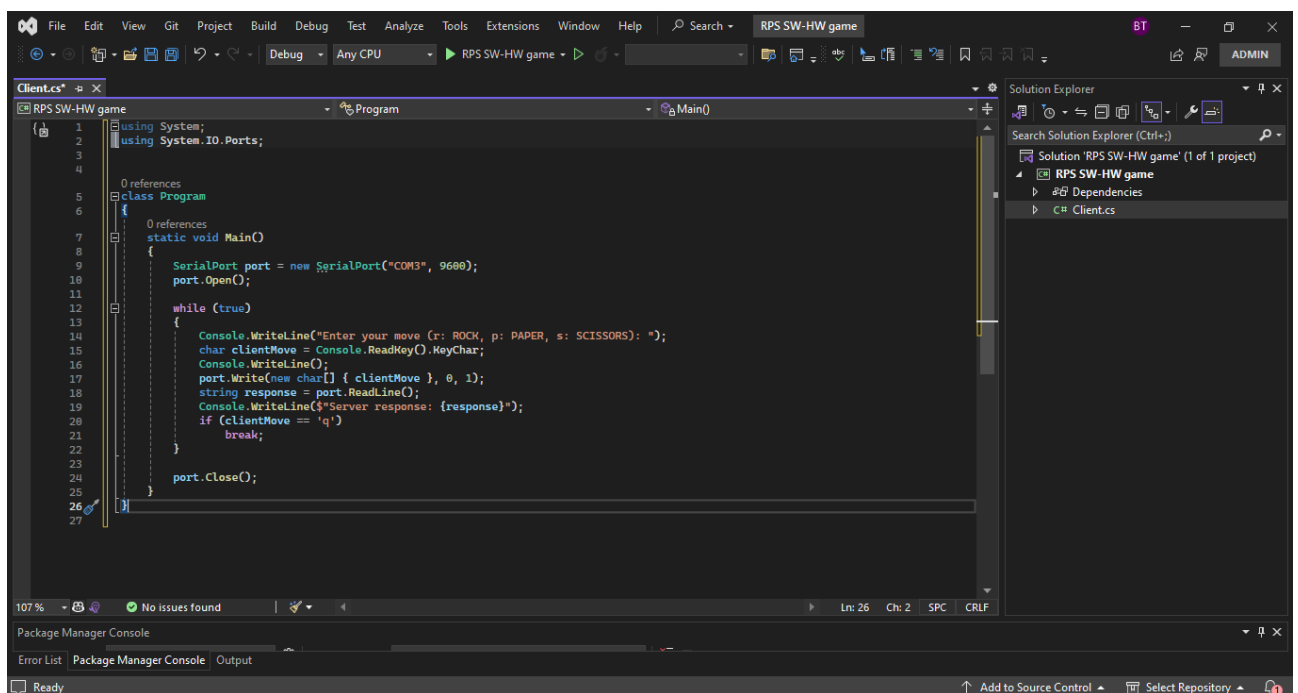
Тема: Створення клієнтської та серверної частини на Arduino Uno R3 гри “Rock paper scissors” з допомогою мови програмування C#.

Порядок виконання лабораторної роботи:

Варіант 14

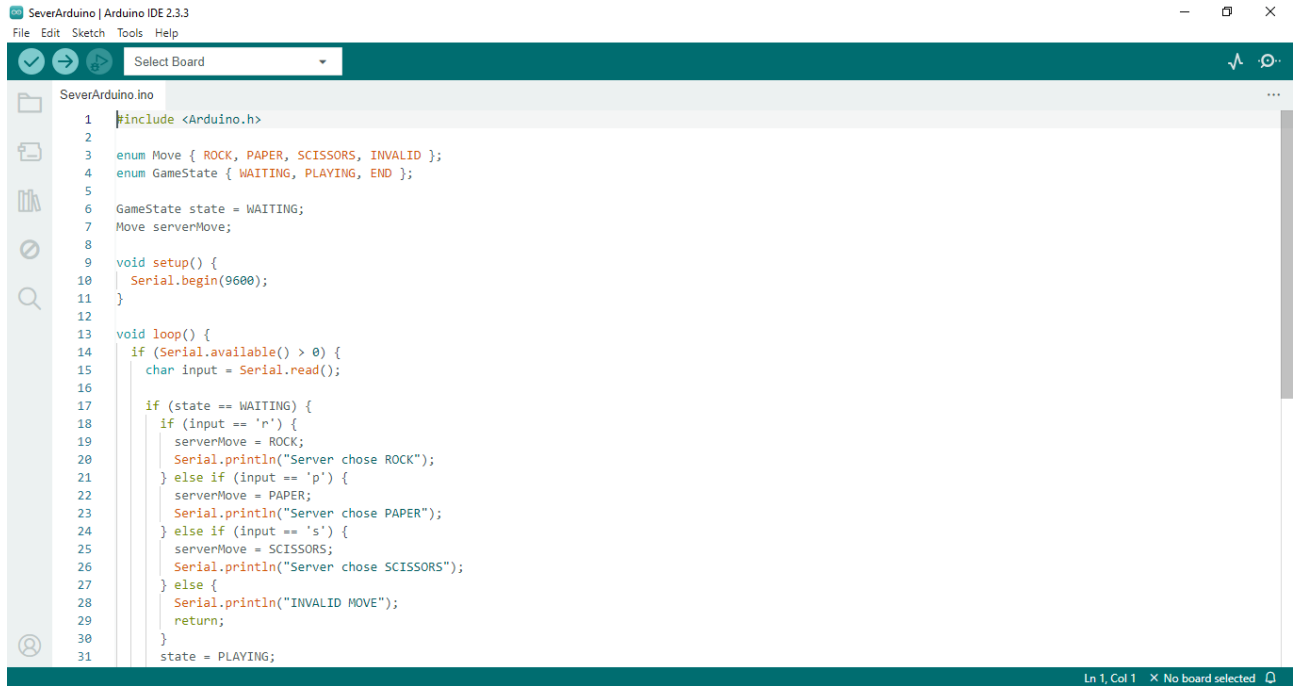
Для реалізації гри «Камінь, ножиці , папір» , я використав Arduino IDE для написання коду для сервера на Arduino, а для клієнта на C# вибрав Visual Studio, написав код для Arduino, який приймає команди від клієнта і випадковим чином вибирає (камінь, ножиці або папір). Сервер обробляє ці команди, визначає переможця і відправляє результат назад клієнту.

Файл Client.cs (client):



```
1 using System;
2 using System.IO.Ports;
3
4
5 0 references
6 class Program
7 {
8     0 references
9     static void Main()
10     {
11         SerialPort port = new SerialPort("COM3", 9600);
12         port.Open();
13
14         while (true)
15         {
16             Console.WriteLine("Enter your move (r: ROCK, p: PAPER, s: SCISSORS): ");
17             char clientMove = Console.ReadKey().KeyChar;
18             Console.WriteLine();
19             port.Write(new char[] { clientMove }, 0, 1);
20             string response = port.ReadLine();
21             Console.WriteLine($"Server response: {response}");
22             if (clientMove == 'q')
23                 break;
24         }
25         port.Close();
26     }
27 }
```

Файл Arduino.ino (server):



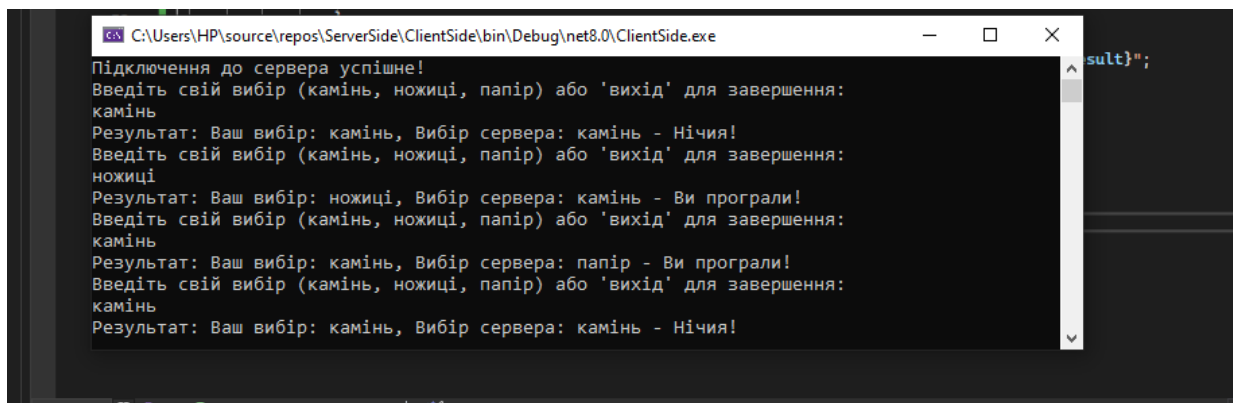
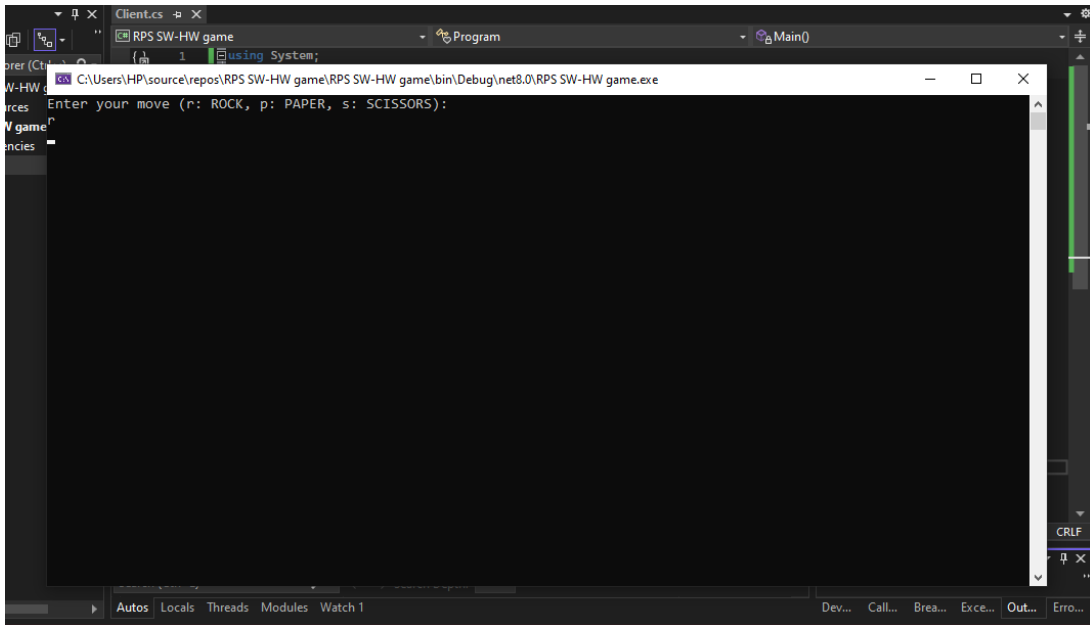
```
SeverArduino.ino
1  #include <Arduino.h>
2
3  enum Move { ROCK, PAPER, SCISSORS, INVALID };
4  enum GameState { WAITING, PLAYING, END };
5
6  GameState state = WAITING;
7  Move serverMove;
8
9  void setup() {
10   Serial.begin(9600);
11 }
12
13 void loop() {
14   if (Serial.available() > 0) {
15     char input = Serial.read();
16
17     if (state == WAITING) {
18       if (input == 'r') {
19         serverMove = ROCK;
20         Serial.println("Server chose ROCK");
21       } else if (input == 'p') {
22         serverMove = PAPER;
23         Serial.println("Server chose PAPER");
24       } else if (input == 's') {
25         serverMove = SCISSORS;
26         Serial.println("Server chose SCISSORS");
27       } else {
28         Serial.println("INVALID MOVE");
29         return;
30       }
31       state = PLAYING;
```



```
30   }
31   state = PLAYING;
32   Serial.println("Server move completed. Please enter your move (r, p, s):");
33 }
34
35 else if (state == PLAYING) {
36   Move clientMove = INVALID;
37
38   if (input == 'r') clientMove = ROCK;
39   else if (input == 'p') clientMove = PAPER;
40   else if (input == 's') clientMove = SCISSORS;
41
42   if (clientMove != INVALID) {
43     Serial.print("Client chose: ");
44     Serial.println(clientMove);
45
46     if (serverMove == clientMove) {
47       Serial.println("It's a draw!");
48     } else if ((serverMove == ROCK && clientMove == SCISSORS) ||
49              (serverMove == PAPER && clientMove == ROCK) ||
50              (serverMove == SCISSORS && clientMove == PAPER)) {
51       Serial.println("Server wins!");
52     } else {
53       Serial.println("Client wins!");
54     }
55     state = WAITING;
56   }
57 }
58 }
59 }
60 }
```

Результати виконання:

За допомогою паралельного виконання двох консольних проектів отримав інтерактивну гру між сервером та клієнтом(тобто між комп'ютером та людиною).



Висновки :

В результаті проведення лабораторної роботи з розробки гри "Камінь, Ножиці, Папір" між сервером на Arduino R3 і клієнтом на C# було успішно налаштовано комунікацію між ними за допомогою серійного зв'язку. Для реалізації проекту використовувалося середовище Arduino IDE для написання коду сервера, який генерує випадковий вибір і визначає переможця, а також Visual Studio для розробки клієнтського додатку на C#. У рамках роботи було встановлено необхідні пакети для роботи з серійними портами, що забезпечило можливість обміну даними між Arduino і C#. Клієнт надсилав введення гравця (камінь, ножиці або папір) на сервер, який у відповідь відправляв результат гри. Код був

протестований, і результати підтвердили правильність роботи обох компонентів системи.