**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**



**АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ КОМП’ЮТЕРНИХ СИСТЕМ**

Лабораторна робота № 3

Виконав: ст. гр. КІ-401

Данилюк. М.

Прийняв: Шпіцер А.С

Львів 2024

**Тема:** Створення клієнтської та серверної частини на Arduino Uno R3 гри “Rock paper scissors” з допомогою мови програмування С++ та збереження користувацьких налаштувань у XML.

**Порядок виконання лабораторної роботи:**

**Варіант 4**

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 Code, 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.

Клієнтська частина на C++.

#include <iostream>

#include <string>

#include <windows.h>

HANDLE serialHandle;

bool initializeSerialPort(const std::string& portName) {

    std::wstring widePortName(portName.begin(), portName.end());

    serialHandle = CreateFile(

        widePortName.c\_str(), GENERIC\_READ | GENERIC\_WRITE, 0, NULL,

        OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

    if (serialHandle == INVALID\_HANDLE\_VALUE) {

        std::cerr << "Failed to open serial port: " << portName << std::endl;

        return false;

    }

    DCB serialParams = { 0 };

    serialParams.DCBlength = sizeof(serialParams);

    GetCommState(serialHandle, &serialParams);

    serialParams.BaudRate = CBR\_9600;

    serialParams.ByteSize = 8;

    serialParams.StopBits = ONESTOPBIT;

    serialParams.Parity = NOPARITY;

    SetCommState(serialHandle, &serialParams);

    COMMTIMEOUTS timeout = { 0 };

    timeout.ReadIntervalTimeout = 50;

    timeout.ReadTotalTimeoutConstant = 50;

    timeout.ReadTotalTimeoutMultiplier = 10;

    timeout.WriteTotalTimeoutConstant = 50;

    timeout.WriteTotalTimeoutMultiplier = 10;

    SetCommTimeouts(serialHandle, &timeout);

    return true;

}

void sendMove(const std::string& move) {

    DWORD bytesWritten;

    WriteFile(serialHandle, move.c\_str(), move.length(), &bytesWritten, NULL);

    WriteFile(serialHandle, "\n", 1, &bytesWritten, NULL);

}

std::string receiveResponse() {

    char buffer[256];

    DWORD bytesRead;

    if (ReadFile(serialHandle, buffer, sizeof(buffer), &bytesRead, NULL)) {

        return std::string(buffer, bytesRead);

    }

    return "";

}

int main() {

    std::string portName = "COM3";

    if (!initializeSerialPort(portName)) {

        return 1;

    }

    std::cout << "Connected to Arduino on " << portName << std::endl;

    while (true) {

        std::cout << "Enter your move (Rock, Paper, Scissors) or type EXIT to quit: ";

        std::string clientMove;

        std::getline(std::cin, clientMove);

        if (clientMove == "EXIT") {

            break;

        }

        sendMove(clientMove);

        std::string clientResponse = receiveResponse();

        std::cout << "Server response: " << clientResponse << std::endl;

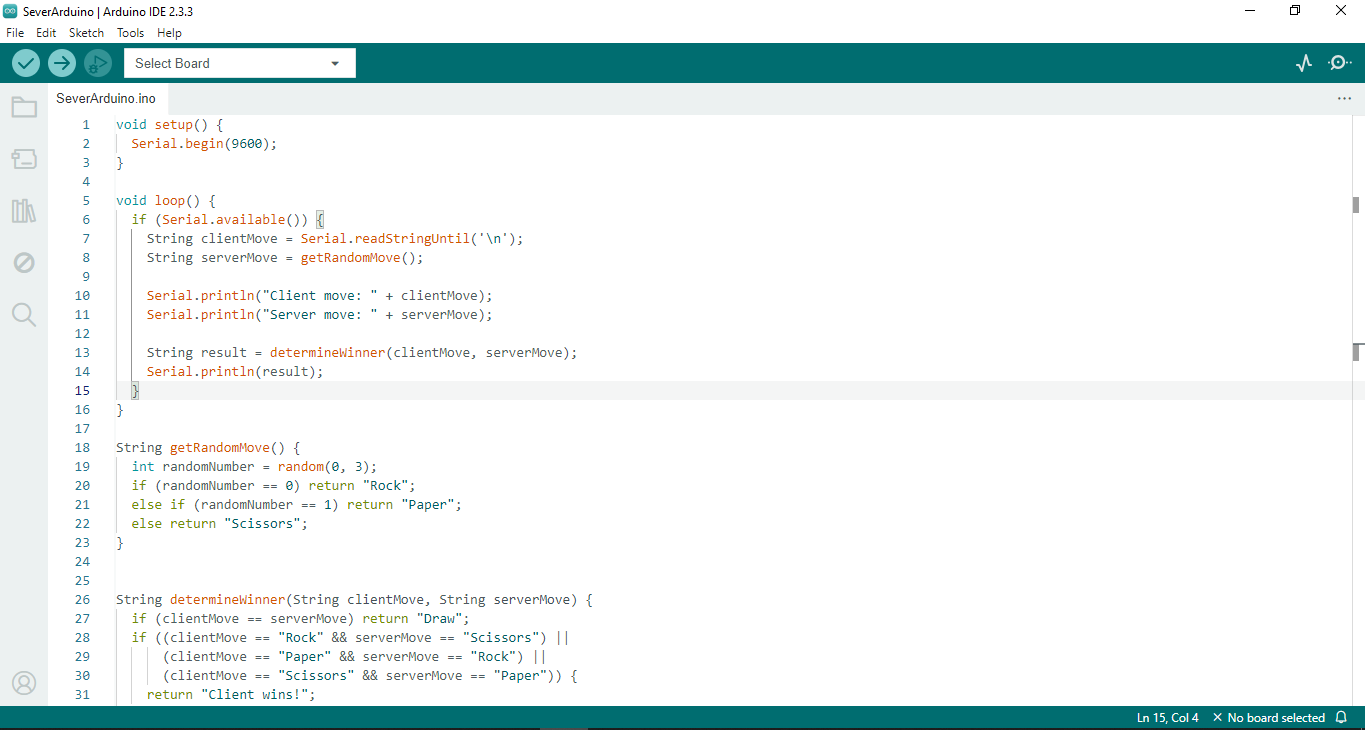
    }

    CloseHandle(serialHandle);

    return 0;

}

**Файл Arduino.ino (server):**





**Висновки :**

У ході виконання лабораторної роботи було реалізовано гру "Камінь, ножиці, папір" із використанням платформи Arduino для серверної частини та мови програмування C++ у середовищі Visual Studio Code для клієнтської частини.

На сервері (Arduino) створено програму, яка отримує команди від клієнта через послідовний інтерфейс (Serial), обробляє їх і виконує відповідну логіку гри. У ході роботи сервер також здійснював управління світлодіодом залежно від отриманих команд, що дозволило перевірити правильність обміну даними між клієнтською і серверною частинами.

На стороні клієнта (C++) реалізовано програму, що встановлює з’єднання з сервером через COM-порт, надсилає команди та отримує відповіді від сервера. За допомогою функцій передачі даних через послідовний порт було забезпечено інтерактивність гри між двома системами, що дозволило користувачу вводити команди з клієнтського додатка, а сервер реагував відповідно.

Ця лабораторна робота допомогла глибше зрозуміти основи серійної комунікації між Arduino та ПК, а також принципи побудови клієнт-серверної архітектури на основі простих апаратних і програмних рішень. Це сприяло вдосконаленню навичок програмування як у середовищі Arduino IDE, так і у Visual Studio Code з використанням мови C++.