

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ЛЬВІВСЬКА ПОЛІТЕХНІКА**



Автоматизоване проектування комп'ютерних систем

**Task 2. Implementing UART Communication**

Виконав ст. гр КІ-401

Гербей О.М.

Прийняв:

Федак П.Р.

## Опис завдання

Для виконання завдання №2 потрібно виконати наступні задачі:

- 1) Створити просту схему комунікації між клієнтом та сервером використовуючи UART.
- 2) Клієнт повинен відправляти повідомлення серверу, після чого сервер повинен його модифікувати та відправляти клієнту.
- 3) Створити YML файл який буде включати наступний функціонал:
  - Створення бінарних файлів проєкту
  - Запуск тестування проєкту
  - Створення артефактів з бінарними файлами та результатами тестування.

## Теоретичні відомості

UART (Universal Asynchronous Receiver-Transmitter) — пристрій для асинхронної передачі даних по послідовній лінії без тактового сигналу, з використанням стартового і стопового бітів. Застосовується у мікроконтролерах та комп'ютерах.

YML (YAML) — формат текстових файлів для структурованих даних з простим синтаксисом. Використовує відступи для структури, пари "ключ: значення" і тире для масивів.

Arduino — відкрита платформа для створення електронних проєктів, яка поєднує мікроконтролерні плати та просте

середовище програмування. Ідеальна для початківців і хобістів, підтримує багато бібліотек та має активну спільноту.

## Виконання завдання

1. Написав просту схему комунікації між клієнтом та сервером:

main.cpp

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <windows.h> // Для роботи із серійним портом
5  #include <cstdlib>    // Для getenv
6
7  HANDLE hSerial; // Дескриптор серійного порту
8  std::ofstream mockSerialOut; // Файл для емуляції запису
9  std::ifstream mockSerialIn;  // Файл для емуляції читання
10
11 // Перевіряємо, чи запускається програма в CI-середовищі
12 bool isCIEnvironment() {
13     const char* ci = std::getenv("CI");
14     return ci != nullptr;
15 }
16
17 // Функція для відкриття серійного порту або файлу
18 bool openSerialPort(const char* portName) {
19     if (isCIEnvironment()) {
20         // У CI-середовищі використовуємо файл як "серійний порт"
21         mockSerialOut.open("mock_serial_out.txt");
22         mockSerialIn.open("mock_serial_in.txt");
23         if (!mockSerialOut.is_open() || !mockSerialIn.is_open()) {
24             std::cerr << "Error: Unable to open mock serial port files!" << std::endl;
25             return false;
26         }
27         std::cout << "Mock serial port initialized for CI environment." << std::endl;
28         return true;
29     }
30
31     // У звичайному середовищі працюємо з реальним серійним портом
32     hSerial = CreateFileA(portName, GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
33     if (hSerial == INVALID_HANDLE_VALUE) {

```

```

34         std::cerr << "Error: Unable to open serial port!" << std::endl;
35         return false;
36     }
37
38     DCB dcbSerialParams = { 0 };
39     dcbSerialParams.DCBlength = sizeof(dcbSerialParams);
40     if (!GetCommState(hSerial, &dcbSerialParams)) {
41         std::cerr << "Error: Unable to get serial port state!" << std::endl;
42         CloseHandle(hSerial);
43         return false;
44     }
45
46     dcbSerialParams.BaudRate = CBR_9600;
47     dcbSerialParams.ByteSize = 8;
48     dcbSerialParams.StopBits = ONESTOPBIT;
49     dcbSerialParams.Parity = NOPARITY;
50
51     if (!SetCommState(hSerial, &dcbSerialParams)) {
52         std::cerr << "Error: Unable to set serial port state!" << std::endl;
53         CloseHandle(hSerial);
54         return false;
55     }
56
57     return true;
58 }
59
60 // Функція для запису в серійний порт або файл
61 void writeSerialPort(const std::string& data) {
62     if (isCIEnvironment()) {
63         if (mockSerialOut.is_open()) {
64             mockSerialOut << data << std::endl;

```

```
63 void writeSerialPort(const std::string& data) {
64     mockSerialOut << data << std::endl;
65     std::cout << "Mock Sent: " << data << std::endl;
66 }
67 } else {
68     DWORD bytesWritten;
69     if (!WriteFile(hSerial, data.c_str(), data.size(), &bytesWritten, NULL)) {
70         std::cerr << "Error: Failed to write to serial port!" << std::endl;
71     } else {
72         std::cout << "Sent to Arduino: " << data << std::endl;
73     }
74 }
75 }
76
77 // Функція для читання із серійного порту або файлу
78 std::string readSerialPort() {
79     if (isCIEEnvironment()) {
80         if (mockSerialIn.is_open()) {
81             std::string line;
82             if (std::getline(mockSerialIn, line)) {
83                 return line;
84             }
85         }
86         return "";
87     }
88
89     char buffer[256] = { 0 };
90     DWORD bytesRead;
91     if (ReadFile(hSerial, buffer, sizeof(buffer) - 1, &bytesRead, NULL) && bytesRead > 0) {
92         buffer[bytesRead] = '\0';
93         return std::string(buffer);
94     }
95     return "";
96 }
97
98 // Основна функція
99 int main() {
100     const char* portName = "COM7"; // Змініть на ваш COM порт
101
102     // Відкриваємо серійний порт
103     if (!openSerialPort(portName)) {
104         return 1;
105     }
106
107     int attempts = 5; // Ліміт кількості ітерацій
108     while (attempts-- > 0) {
109         // Відправляємо повідомлення
110         std::string message = "Hmm... I'll make this move.\n";
111         writeSerialPort(message);
112
113         // Читаємо відповідь
114         std::string response = readSerialPort();
115         if (!response.empty()) {
116             std::cout << "Server says: " << response << std::endl;
117         }
118     }
119
120     // Закриваємо серійний порт
121     CloseHandle(hSerial);
122     return 0;
123 }
```

2. Створив .ino файл.

```
Code Blame 10 lines (9 loc) · 202 Bytes

1 void setup() {
2     Serial.begin(9600);
3 }
4
5 void loop() {
6     if (Serial.available() > 0) {
7         String receivedMessage = Serial.readStringUntil('\n');
8         Serial.println("modified: " + receivedMessage);
9     }
10 }
```

### 3. Створив YML файл




```
1 name: Build and Test
2
3 on:
4   push:
5     branches:
6       - main
7   pull_request:
8     branches:
9       - main
10
11 jobs:
12   build:
13     runs-on: windows-latest # Використання Windows для підтримки MSYS2
14
15     steps:
16     - name: Checkout code
17       uses: actions/checkout@v3
18
19     - name: Set up MSYS2
20       uses: msys2/setup-msys2@v2
21       with:
22         update: true
23
24     - name: Create Mock Serial Input
25       run: echo "Mock response from server" > mock_serial_in.txt
26
27     - name: Build Project
28       run: |
29         mkdir -p build
30         g++ -o build/client lib/client/main.cpp -std=c++17
31
32     - name: Test Application
33       run: |
```

```

30      g++ -o build/client lib/client/main.cpp -std=c++17
31
32      - name: Test Application
33      run: |
34          set CI=true # Установлюємо змінну середовища для емуляції
35          ./build/client
36
37      - name: Upload binaries and test reports
38      uses: actions/upload-artifact@v3
39      with:
40          name: build-artifacts
41          path: |
42              build/client.exe
43              mock_serial_in.txt
44
45      - name: Clean up Mock Files
46      if: always() # Завжди виконуємо, навіть якщо тест провалився
47      run: |
48          del mock_serial_in.txt
49          del mock_serial_out.txt

```

4. Створив артефакти з бінарними файлами та звітом тестів:

Artifacts		
Produced during runtime		
Name	Size	
 build-artifacts	76.9 KB	 

```

PS D:\scad\csad2425Ki401HerbeiOleksandr03> .\client.exe
Sent to Arduino: Hmm... I'll make this move.

Server says: modified: Hmm... I'll make this move.

Do you want to send another message? (y/n): Y
Sent to Arduino: Hmm... I'll make this move.

Do you want to send another message? (y/n): Y
Sent to Arduino: Hmm... I'll make this move.

Do you want to send another message? (y/n): 

```

## Висновок

Під час виконання завдання №2 було розроблено просту схему комунікації між клієнтом та сервером, а також скрипти для перевірки цілісності проєкту.



