

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



**АВТОМАТИЗОВАНЕ
ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ
СИСТЕМ**

Лабораторна робота №2
“Створення клієнт-серверного
комунікатора”

Виконав:

студент гр. КІ-404
Попов М. Ю.

Прийняв:

аспірант
Федак П. Р.

Завдання:

SW <-> HW (FEF)

1. Create a simple communication schema **SW(client)** <-> **UART** <-> **HW(server)**.
2. The client should send a message to the server. The server should modify the message and send it back to the client.
3. Create **YML file** with next features:
 - a. build all binaries (create scripts in folder **ci/** if need);
 - b. run tests;
 - c. create artifacts with binaries and test reports;
4. Required steps.

Required steps for all tasks:

1. Add/modify UTs
2. Create/Modify the **ci** script to build, test and run project.
3. Create/Modify the **YML** file.
4. Commit changes to **feature/develop/<task number>**.
5. Update **README** with details:
 - how to build project;
 - how to run project;
 - Version number.
6. Create **GIT TAG**:

<PROJECT NAME>_<task number>_<VERSION>_ww<YYWWD>
 - YY – current year;
 - WW – work week;
 - D – current day number of weak.
7. Create pull request with name <task number> and submit **lecturer** as reviewer.
8. After the reviewer approved – merge into the develop branch.

Проста комунікаційна схема між клієнтською і серверною частинами програми:

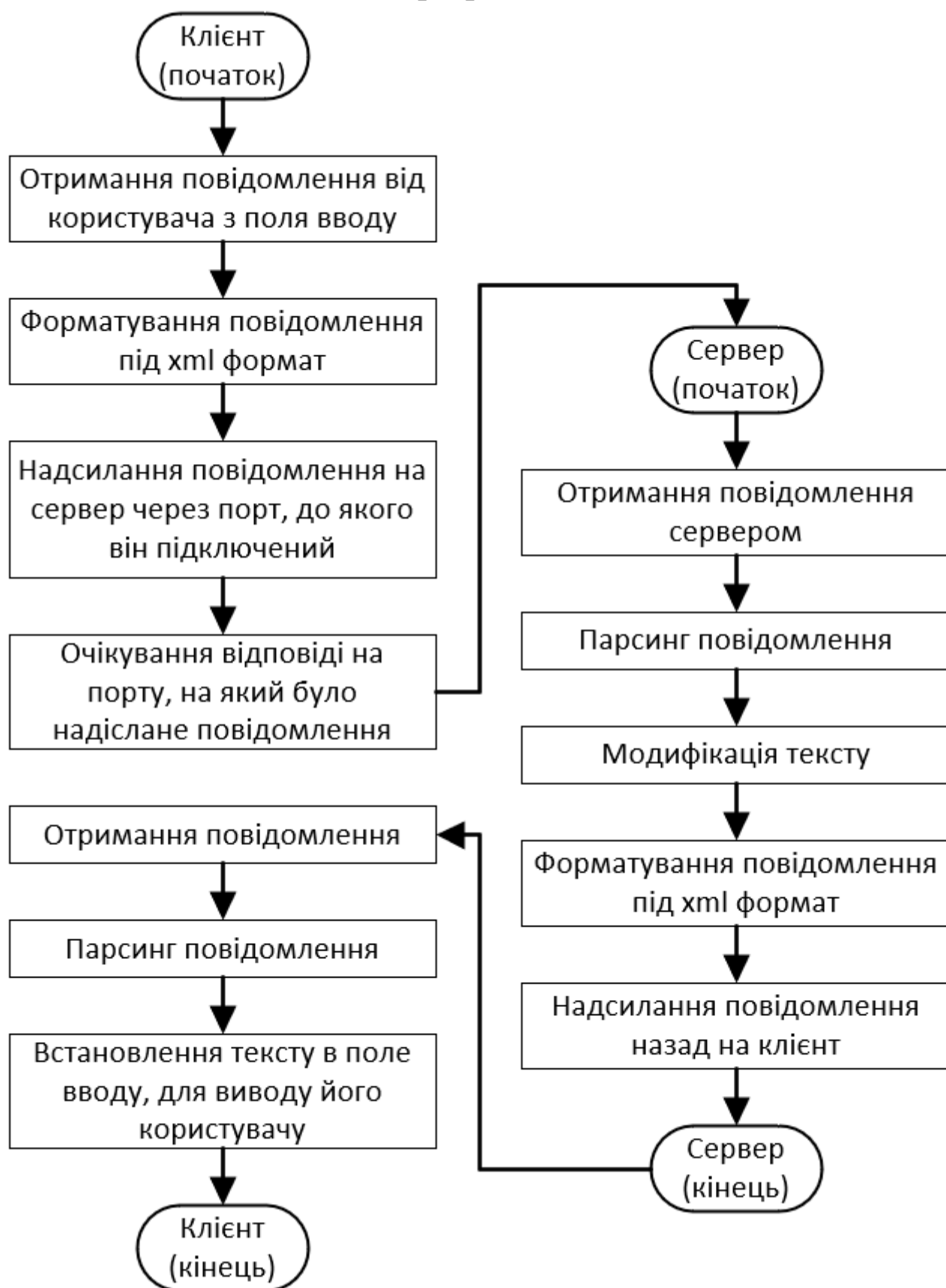


Рис. 1. Блок-схема комунікації між клієнтом і сервером

Виконання

1) Створення клієнтської частини програми:

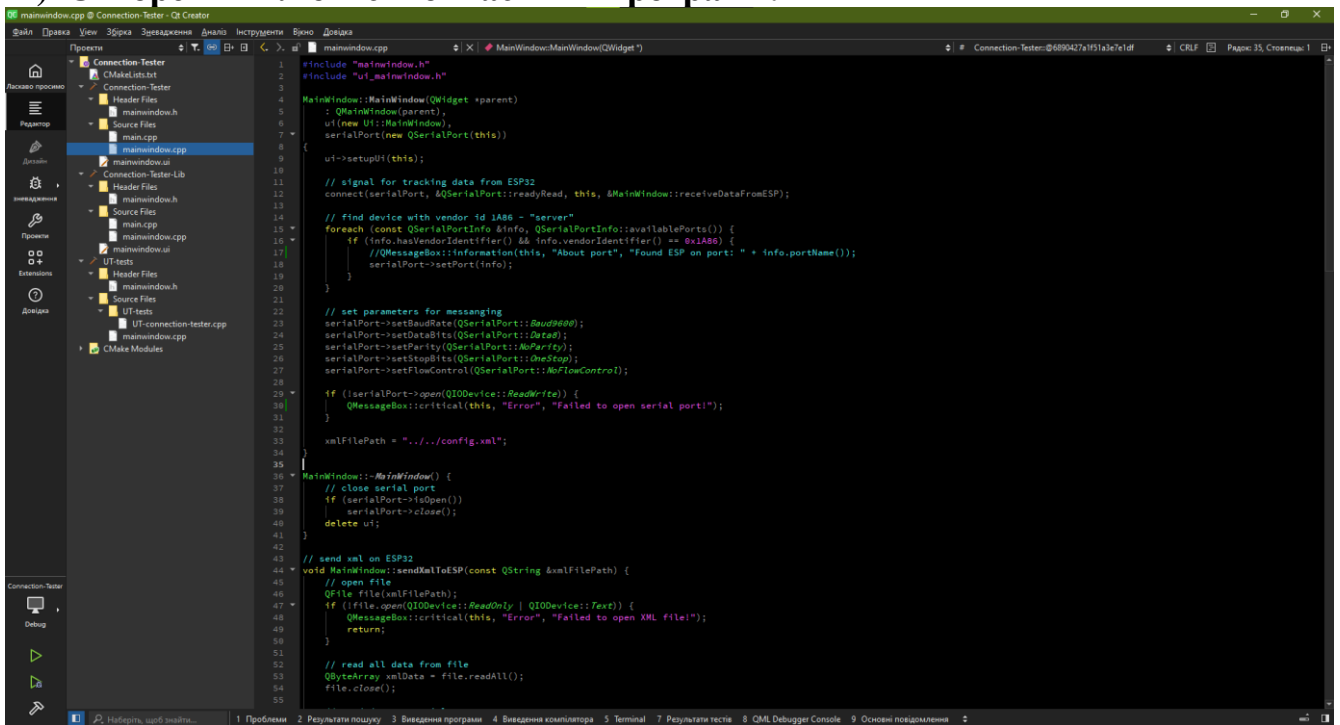


Рис. 2. Середовище розробки Qt creator з відкритим проектом

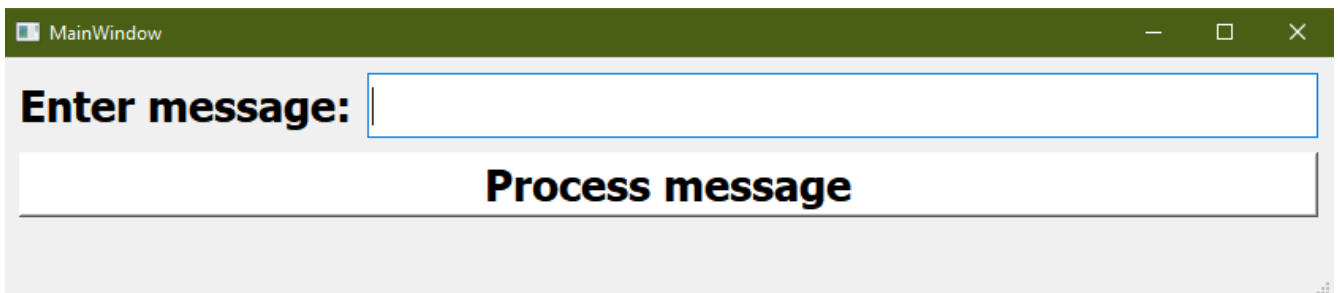


Рис. 3. Результат: графічна програма-клієнт

2) Створення прошивки серверної частини програми:

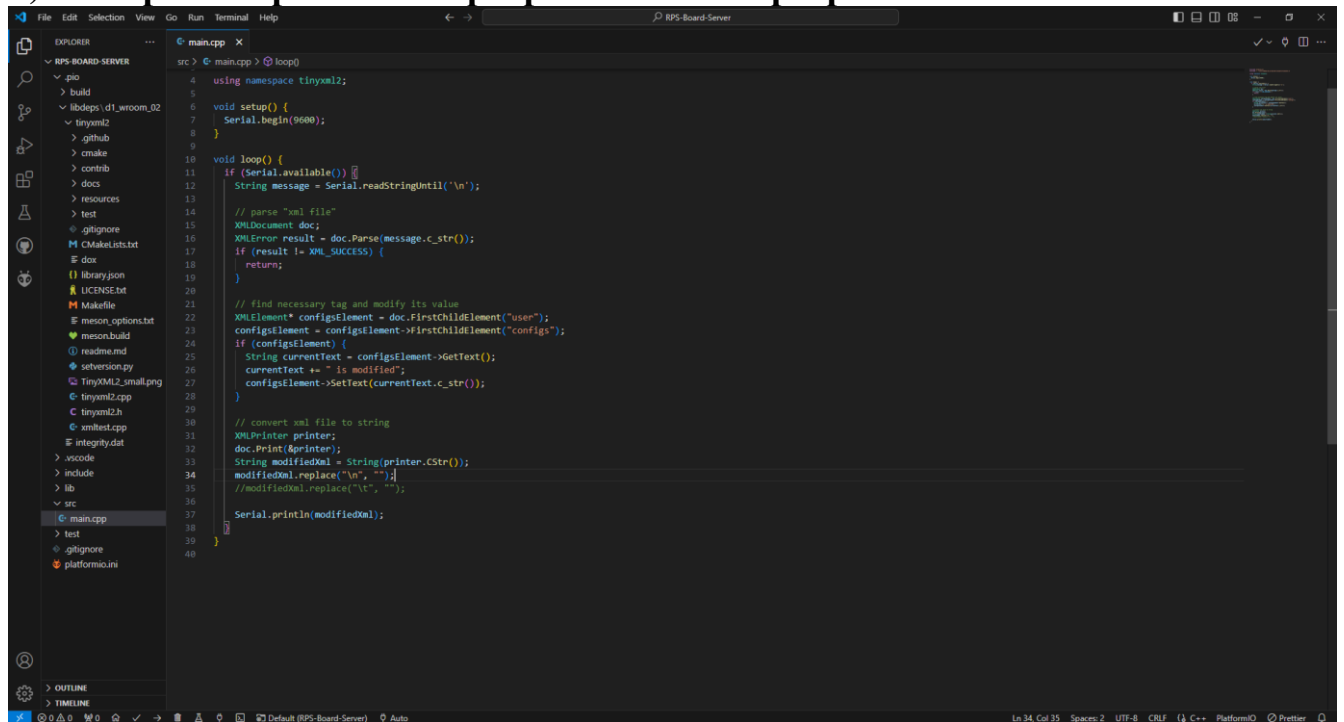


Рис. 4. Середовище розробки VS Code з плагіном PlatformIO та відкритим проектом

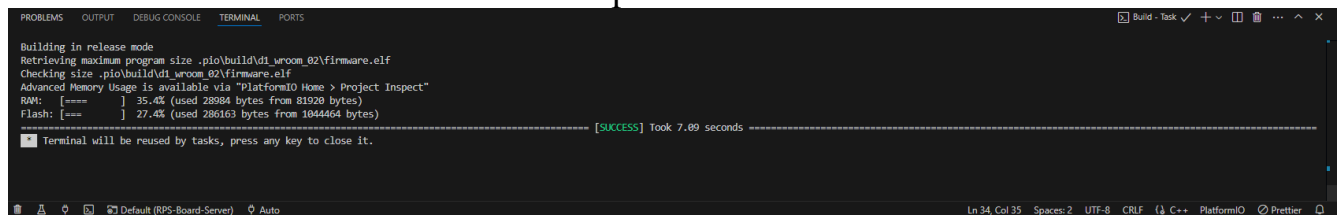


Рис. 5. Білд та завантаження програми на плату

3) Результат роботи програми:

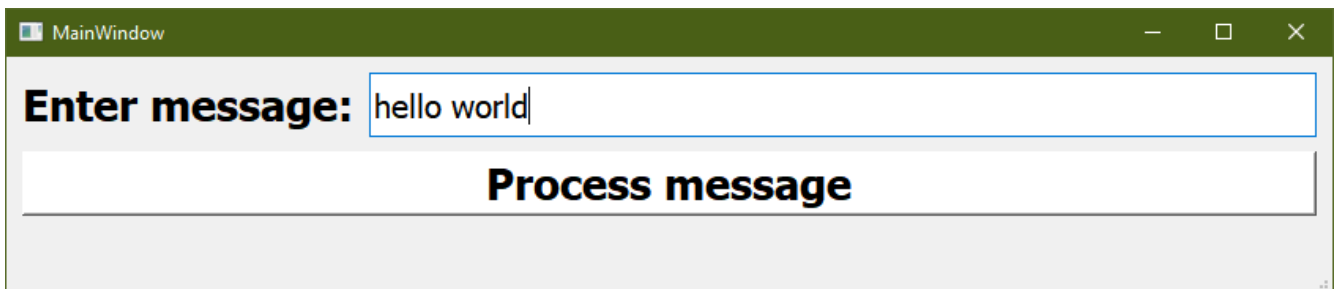


Рис. 6. Повідомлення до надсилання

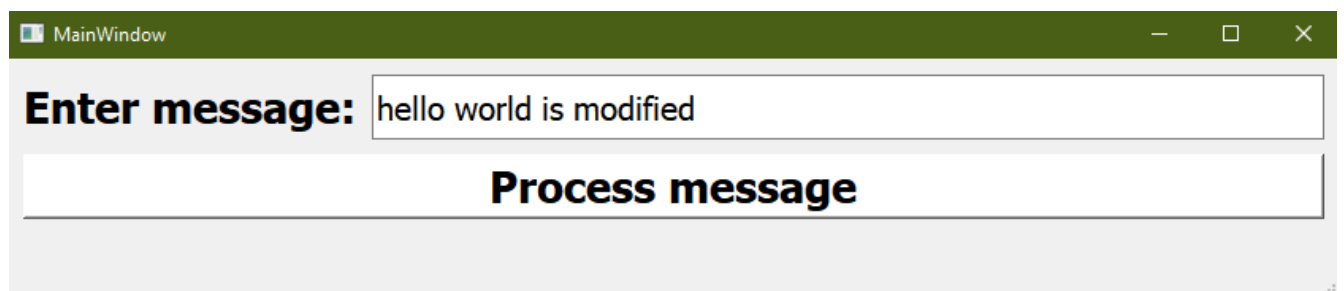


Рис. 7. Повідомлення після надсилання

4) Результат роботи тестів:

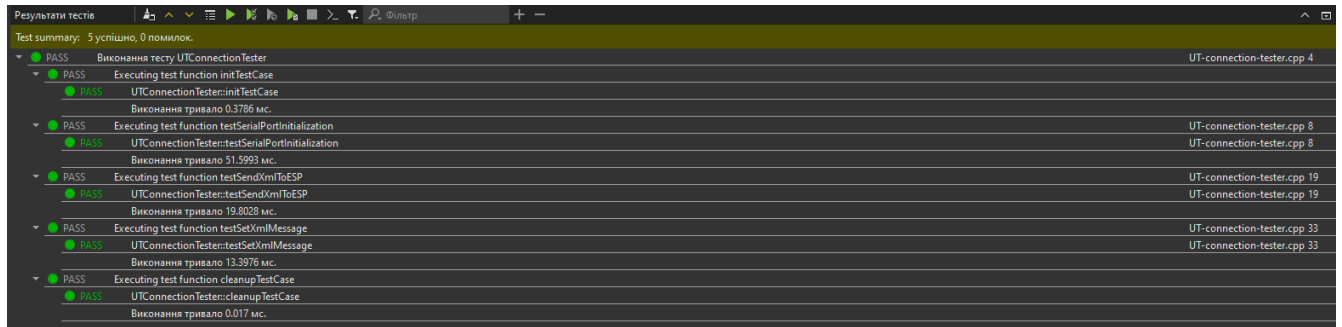


Рис. 8. Вікно, з успішним результатом тестів

5) Створення .yaml файла і результат роботи сі скриптів:

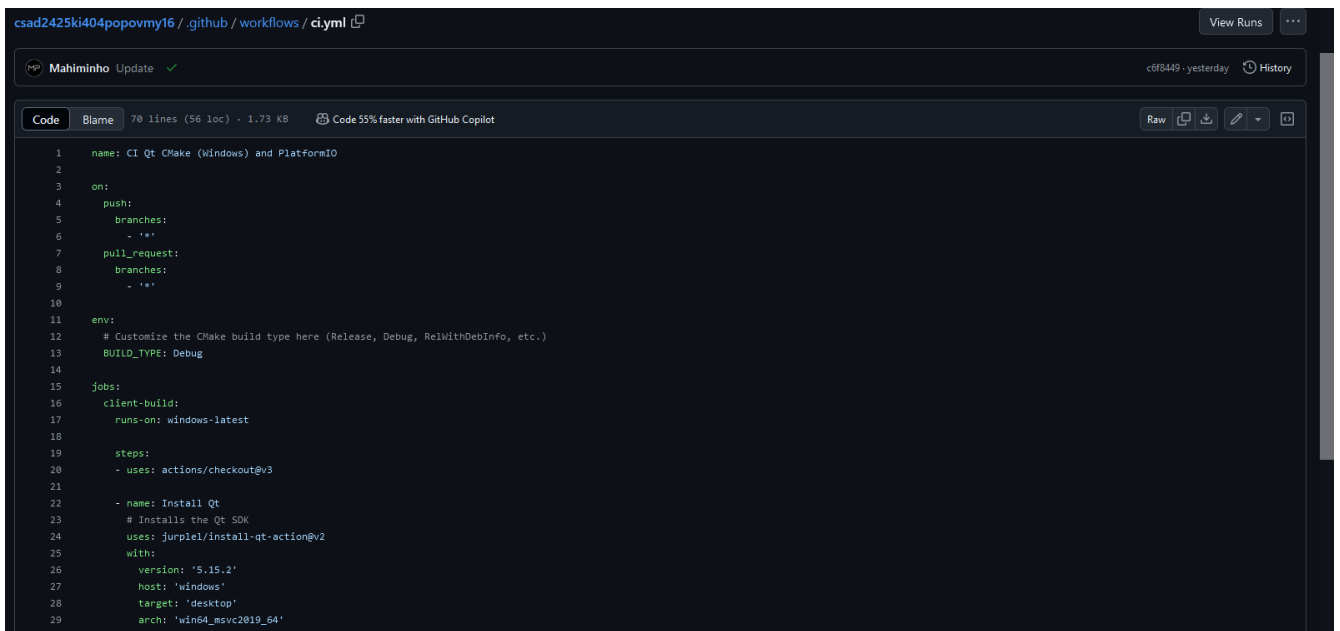


Рис. 9. Створений .yaml файл

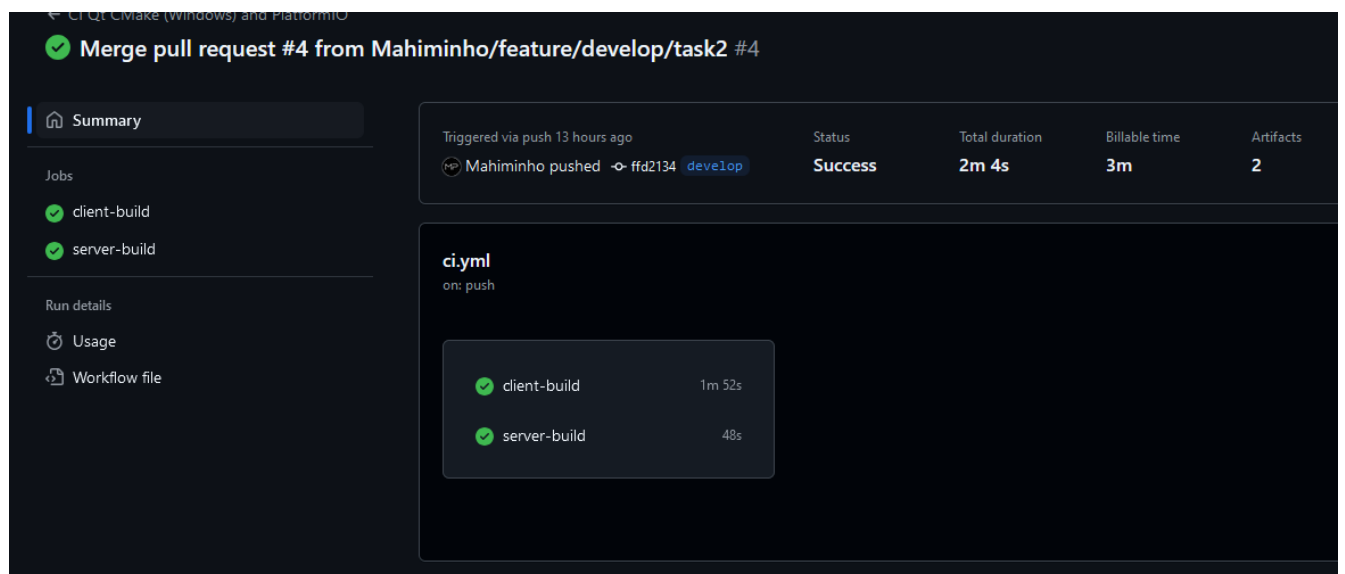


Рис. 10. Результат роботи github actions

6) Модифікація README.md:

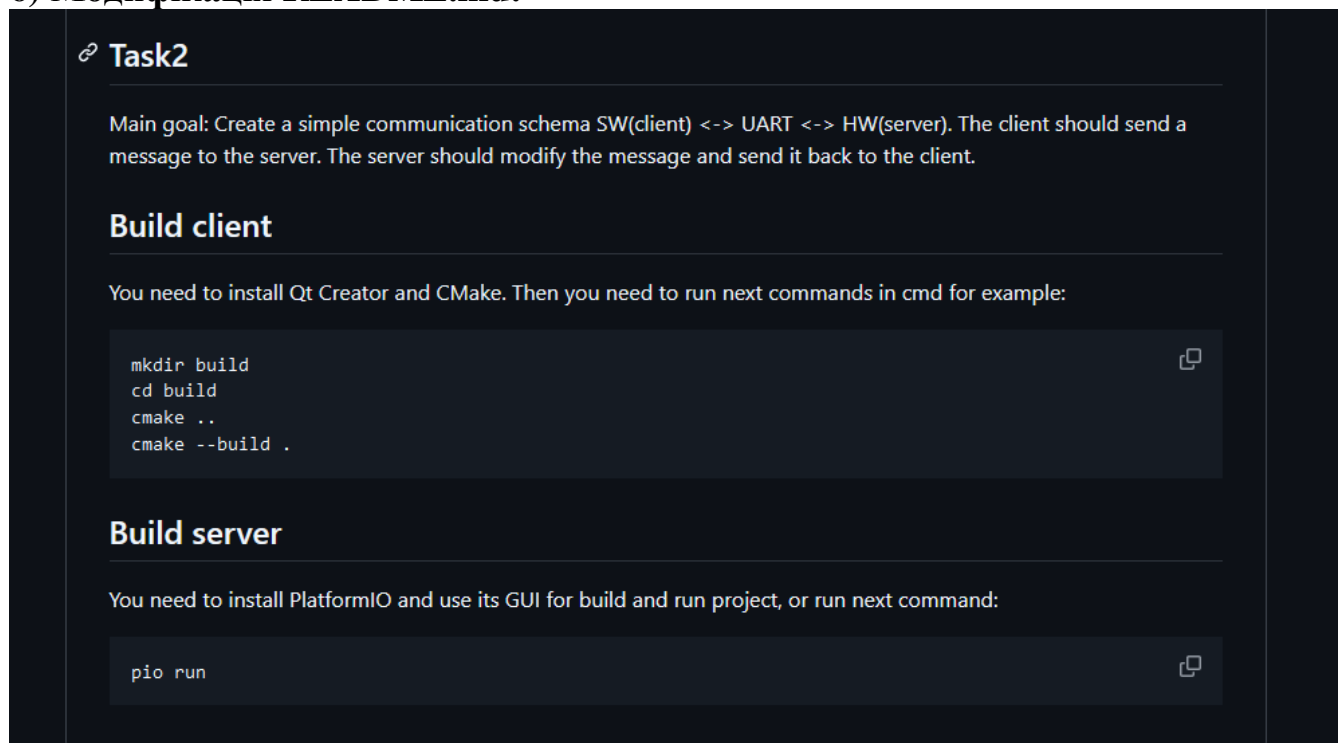


Рис. 11. Змінений README.md

7) Створення GIT TAG:

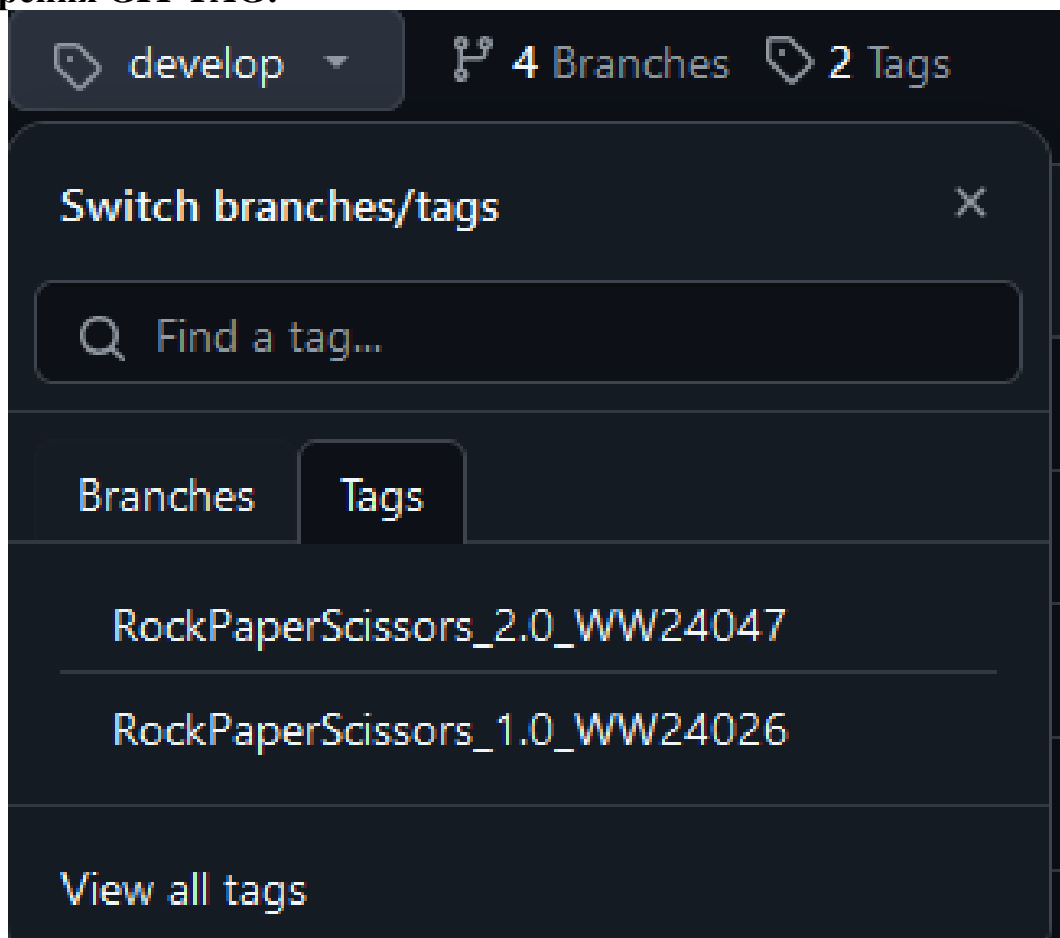


Рис. 12. Створений GIT TAG

7) Створення нової гілки:

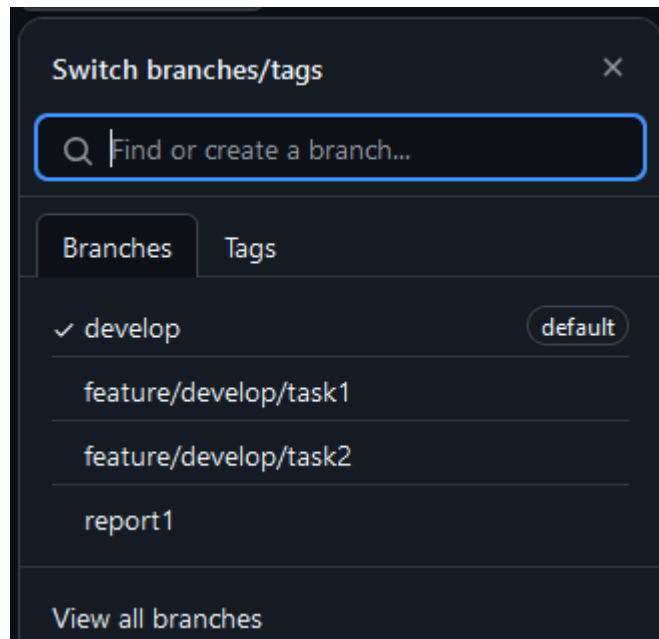


Рис. 13. Створена нова гілка

8) Створення пул реквесту, отримання апруву та мердж в головну гілку:

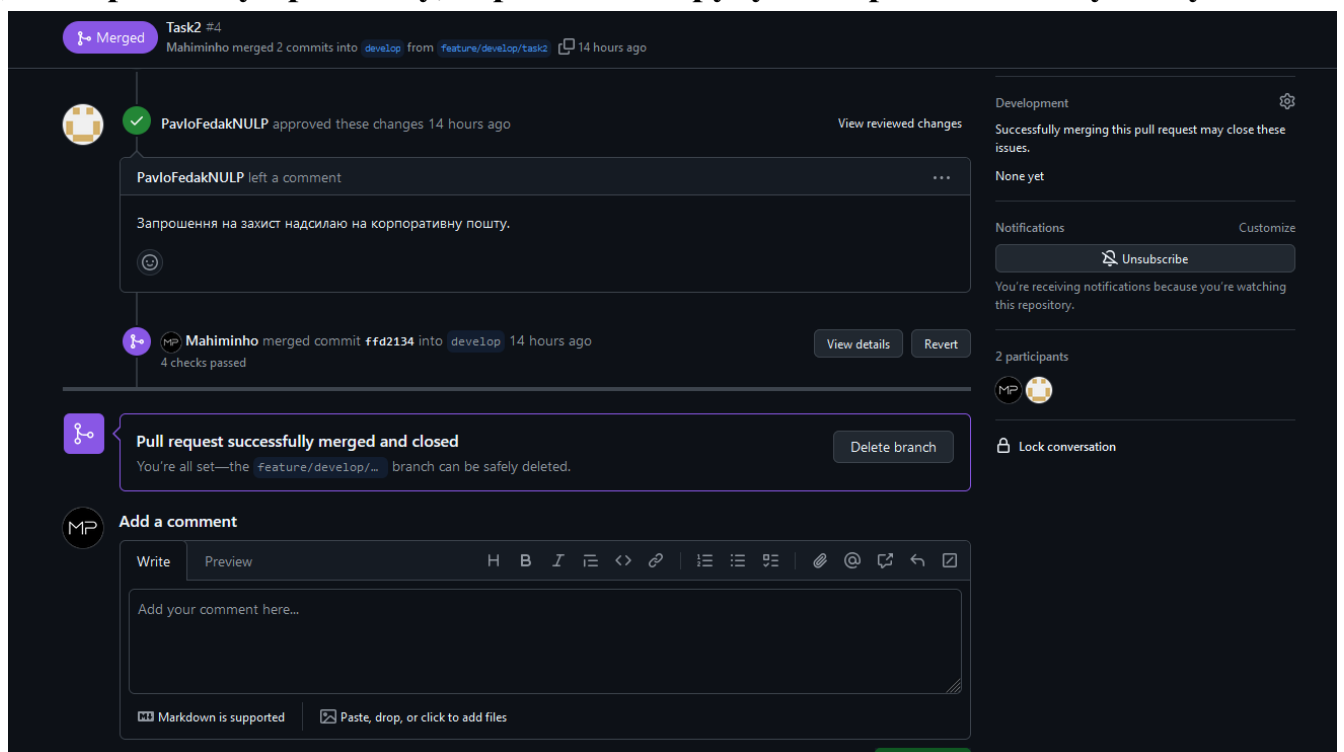


Рис. 14. Отриманий апрув в запит на пул реквест, та мердж в develop

Висновок:

Впід час виконання даної лабораторної роботи, я познайомився з YML, CI, Unit tests, навчився використовувати усі попередньозгадані технології, а також написав прості програми для комунікації між клієнтом і сервером.