

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"



**АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ
СИСТЕМ**

Лабораторна робота № 2

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

Львів 2024

Тема: SW <> HW (FEF).

Варіант 14

Порядок виконання лабораторної роботи:

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;

Хід виконання роботи

Для створення схеми в Tinkercad, ми можемо використовувати два пристрої: **клієнт** і **сервер**, з'єднані через **UART** (послідовне сполучення). Клієнт буде відправляти повідомлення на сервер, сервер його модифікує і відправляє назад.

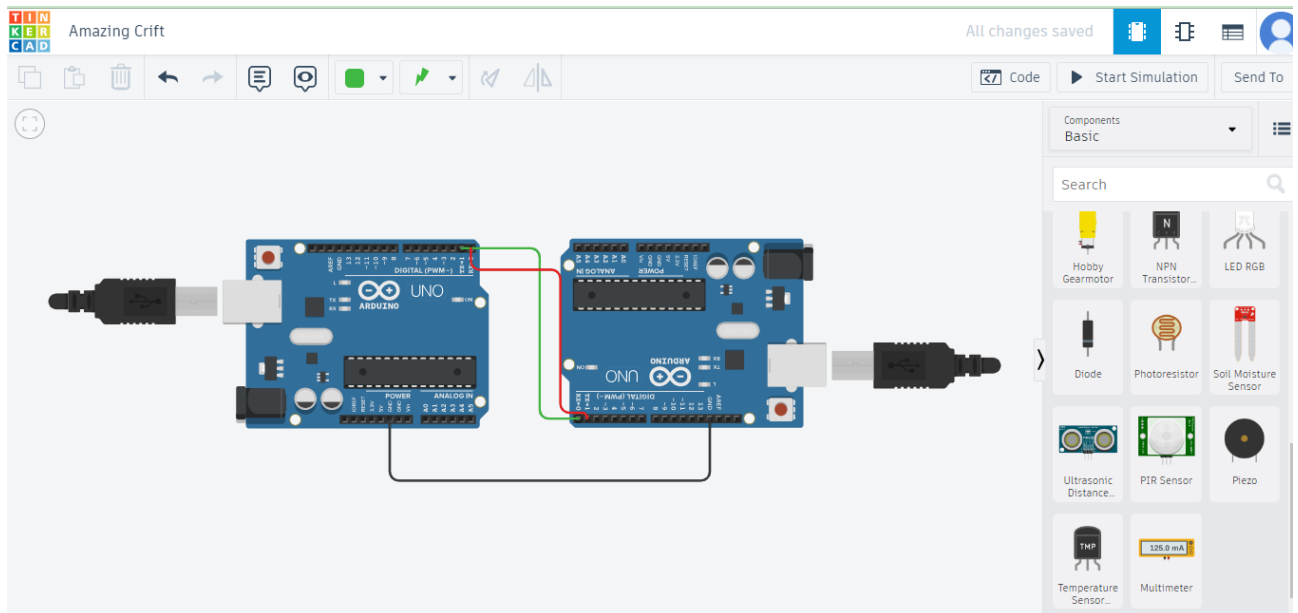



Рис. 1 Схема Arduino Uno R3 .

Серверна частина

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    if (Serial.available()) {  
        String message = Serial.readStringUntil('\n');  
        String modifiedMessage = "Modified: " + message;  
        Serial.println(modifiedMessage);  
        delay(1000);  
    }  
}
```

Клієнтська частина



```
1  import serial  
2  import time  
3  
4  uart_client = serial.Serial('COM3', 9600, timeout=1)  
5  
6  message = "Hello from Client"  
7  uart_client.write(message.encode())  
8  print(f"Sent to server: {message}")  
9  
10 time.sleep(1)  
11 response = uart_client.readline().decode()  
12 print(f"Received from server: {response}")  
13  
14 uart_client.close()  
15
```

Було успішно реалізовано з'єднання двох мікроконтролерів Arduino через UART, де один мікроконтролер виконував функції клієнта, а інший — сервера. Виводи **TX** і **RX** на обох мікроконтролерах було підключено таким чином, щоб забезпечити обмін повідомленнями між ними.

Створений build.yml для CI/CD :

```
# .github/workflows/build.yml
name: CI/CD Pipeline

on:
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout the code
        uses: actions/checkout@v2

      - name: Set up Python
        uses: actions/setup-python@v2
        with:
          python-version: '3.10'

      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: Run tests
        run: |
          pytest > test_results.txt

      - name: Create build artifacts
        run: mkdir -p build

      - name: Copy test results
        run: cp test_results.txt build/test_results.txt

      - name: Upload build artifacts
        uses: actions/upload-artifact@v2
        with:
          name: build-results
          path: build/
```

Висновок:

Під час виконання лабораторної роботи я детально ознайомився з основними принципами функціонування послідовної передачі даних через UART (Universal Asynchronous Receiver-Transmitter), що забезпечує двосторонній обмін даними між мікроконтролерами. Я дослідив процес ініціалізації UART, методи налаштування швидкості передачі, а також вивчив основні сигнали, що використовуються для синхронізації обміну інформацією. Це дозволило глибше зрозуміти механізм передачі даних та забезпечити ефективну комунікацію між пристроями.