**TCP Client–Server Application Documentation**

**Overview**

This application implements a simple TCP-based command protocol between a client and a server on windows.

Client sends three commands:

1. SELECT <OPEN/CLOSE> <duration>
2. EXECUTE
3. READ

Server simulates opening or closing a valve for the specified duration, maintains a valve-position indicator, and replies with confirmations or error messages.

**System Requirements**

Windows 10 or Windows 11 (I have run on windows 11) 64-bit

No complier or IDE required on target machine (as I have already compiled on my machine)

Build-in Winsock2 library (Ws2\_32.dll)

**Distribution Package**

server.exe

client.exe

README.txt

**server.exe** – the TCP server

**client.exe** – the TCP client

**README.txt** – run instructions

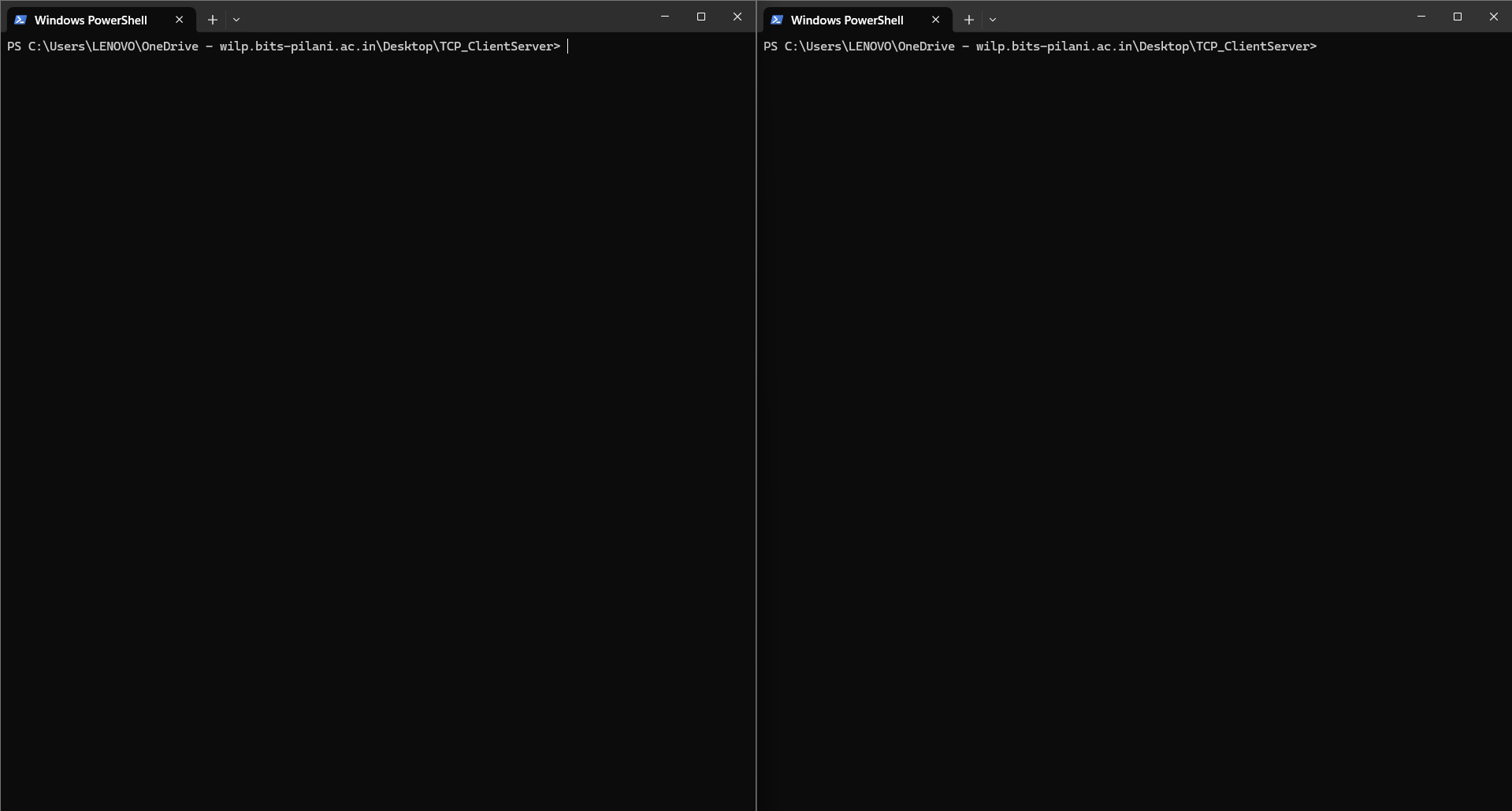
STEPS TO RUN THE APPLICATION ////

TCP Client–Server Application

=============================

1. Unzip TCP\_ClientServer.zip anywhere.

2. Open TWO Command Prompt windows:



Window 1 (Server):

cd C:\path\to\unzipped

RUN >> server.exe

[Server starts listening on port 5000…]

A screen shot of a computer

AI-generated content may be incorrect.

Window 2 (Client):

cd C:\path\to\unzipped

RUN>> client.exe

You should see [Client connects to 127.0.0.1:5000…]

A screen shot of a computer

AI-generated content may be incorrect.

3. In the CLIENT window, enter commands:

SELECT OPEN 5

EXECUTE

READ

A screenshot of a computer

AI-generated content may be incorrect.

4. To exit:

• In client: type exit

• In server: Ctrl+C

Troubleshooting:

- “Connection refused” → make sure server.exe is running.

- Windows Firewall popup → click “Allow access.”

Command Reference

Command >> SELECT <SOL> <secs> , Response on Success >> SELECT OK , Error Conditions >> Invalid syntax; bad solenoid; non-positive duration

Command >> EXECUTE , Response on Success >> EXECUTE OK , ERROR Conditions >> No Prior SELECT; solenoid already executing

READ >> Valve is OPEN / Value is CLOSE >>   
Any other inputs >> ERROR: Unknown command >>

Code Quality & Error Handling

Initialization Checks

Verifies WSAStartup , sockets(), bind(), listen(), and accept() return valid result.

Input validation

Strips CR/LF before parsing.

Validates SELECT has exactly two arguments: a valid solenoid name (OPEN or CLOSE) and a positive integer duration.

Ensure EXECUTE only runs after a successful SELECT

Error Reporting

Sends clear error messages back to client(eg., ERROR: SELECT must precede EXECUTE).

Prints detailed WSAGetLastError() codes on the server console for socket failures.

State management

Resets the selected solenoid after execution after execution to prevent accidental re-execution.

Maintains and reports the current valve position correctly.

Next Steps

We can create GUI front-end for enhanced logging

Autometed test scripts

Configurable ports

Multi-client support.