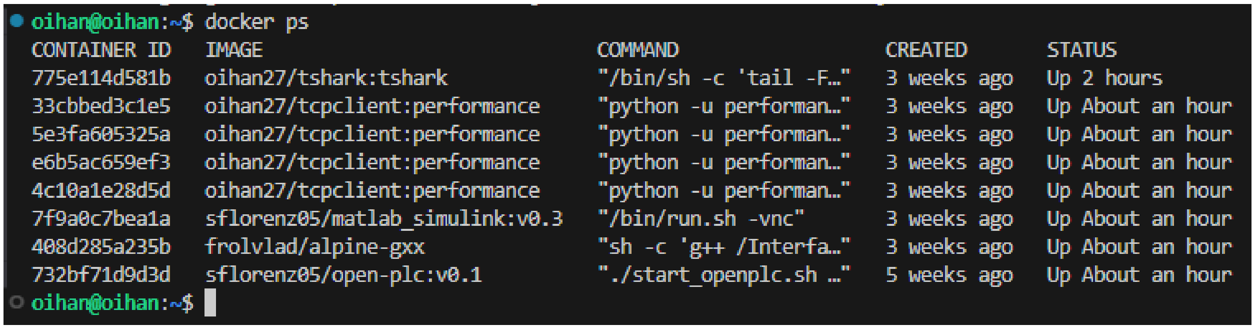
Interface Integration Tutorial

Written on: 22/07/24 by Isabella Scanlan
Revised on: 22/07/24 by Isabella Scanlan

This tutorial covers how to set up and run the interface used to facilitate the data exchange between the Omron CJ2M-CPU31 PLC and a Wastewater Treatment Plant (WWTP) in MATLAB Simulink. The interface consists of two files, server.py and plc.py. Server.py serves to create a virtual server that initially stores PLC data. As the WWTP simulation runs, server.py updates the values in the server according to the simulation. Plc.py, ran after server.py, writes the WWTP data stored in the virtual server created in server.py to the PLC.

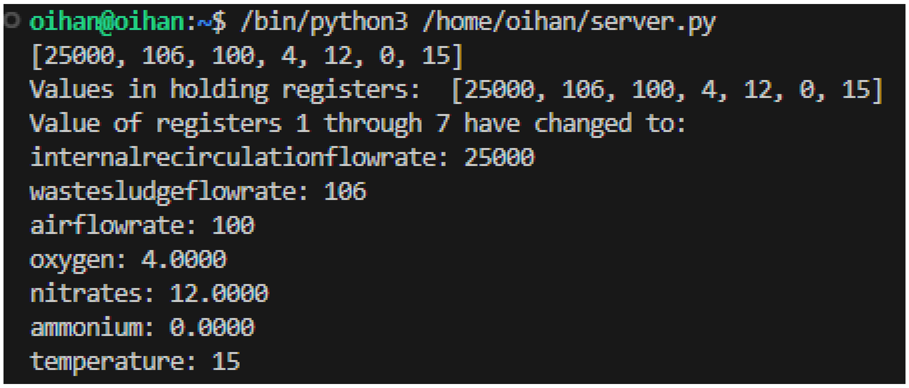
1. Establish an SSH connection to the appropriate server
   - 1.1. Open VS Code
   - 1.2. "Connect to remote development workspaces" on the welcome screen
   - 1.3. "Connect to Host"
   - 1.4. If 10.63.28.53 does not appear, "Add New SSH Host"
   - 1.5. Type "ssh oihan@10.63.28.53"
   - 1.6. Password: oihan
2. Start the virtual server in the interface
   - 2.1. Open server.py and plc.py on the screen, located under /home/oihan.
   - 2.2. Run "docker ps" in the command line to ensure proper containers are up, shown in **Figure 2.2.1**.



```
oihan@oihan:~$ docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED       STATUS
775e114d581b   oihan27/tshark:tshark              "/bin/sh -c 'tail -F…"   3 weeks ago   Up 2 hours
33cbbed3c1e5   oihan27/tcpclient:performance      "python -u performan…"   3 weeks ago   Up About an hour
5e3fa605325a   oihan27/tcpclient:performance      "python -u performan…"   3 weeks ago   Up About an hour
e6b5ac659ef3   oihan27/tcpclient:performance      "python -u performan…"   3 weeks ago   Up About an hour
4c10a1e28d5d   oihan27/tcpclient:performance      "python -u performan…"   3 weeks ago   Up About an hour
7f9a0c7bea1a   sflorenz05/matlab_simulink:v0.3    "/bin/run.sh -vnc"       3 weeks ago   Up About an hour
408d285a235b   frolvlad/alpine-gxx                "sh -c 'g++ /Interfa…"   3 weeks ago   Up About an hour
732bf71d9d3d   sflorenz05/open-plc:v0.1           "./start_openplc.sh …"   5 weeks ago   Up About an hour
oihan@oihan:~$
```

Figure 2.2.1*: Containers needed to run the interface*

   - 2.3. Run server.py by right clicking inside the file > Run Python > Run Python File in Terminal, or by typing in the command line "python3 server.py"
   - 2.4. If started successfully, outputs in the command line should appear similar to those in **Figure 2.4.1**. These values represent the most recent values written to the PLC.



```
oihan@oihan:~$ /bin/python3 /home/oihan/server.py
[25000, 106, 100, 4, 12, 0, 15]
Values in holding registers:  [25000, 106, 100, 4, 12, 0, 15]
Value of registers 1 through 7 have changed to:
internalrecirculationflowrate: 25000
wastesludgeflowrate: 106
airflowrate: 100
oxygen: 4.0000
nitrates: 12.0000
ammonium: 0.0000
temperature: 15
```

***Figure 2.4.1**: Initial outputs after running server.py*

3. Run the WWTP in MATLAB Simulink
    - 3.1. Open by going into the browser and typing “10.63.28.53:6080” in the search bar
    - 3.2. Once on the home page, double click on the MATLAB icon, shown in **Figure 3.2.1**.

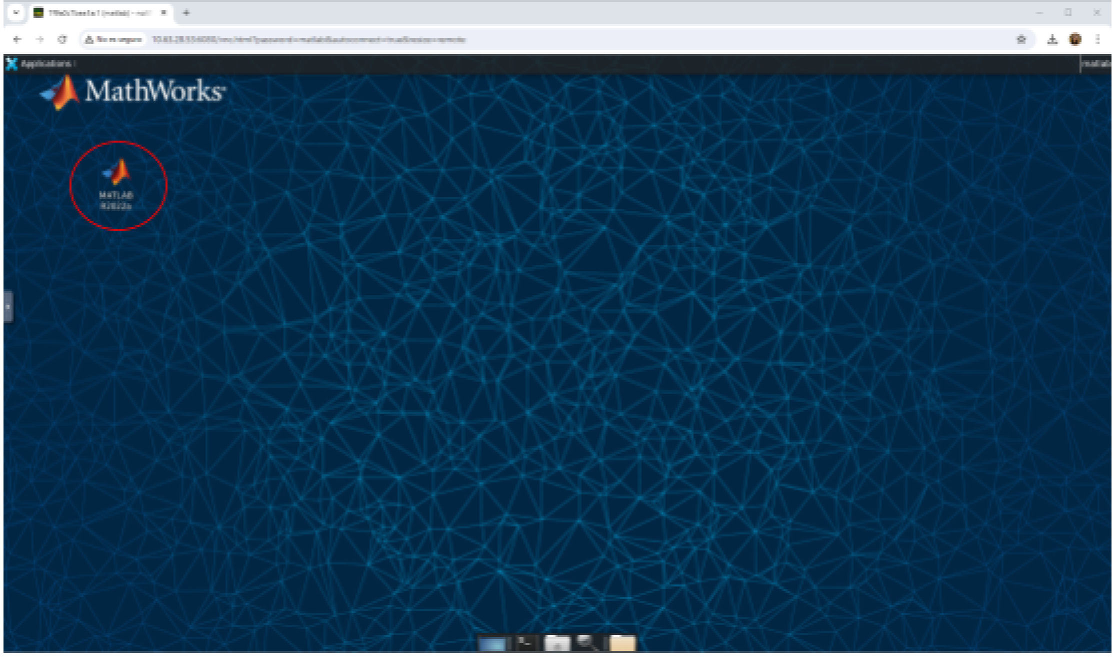


***Figure 3.2.1**: MATLAB icon on the home page*

- 3.3. Enter MATLAB login credentials if prompted.
- 3.4. Once in MATLAB, navigate to the files on the left of the screen. Right click on the “wwtp” folder > Add to Path > Selected Folders and Subfolders.
- 3.5. Double click on the file “wwtp3.slx” to open it. It may take a few minutes to open.
- 3.6. Once open, run by clicking the green play button under the “Simulation” tab at the top of the screen.
- 3.7. If running successfully, the values of oxygen, nitrates, and ammonium should be fluctuating decimals. After a few minutes, the numbers should settle similar to those shown in **Figure 3.7.1**.

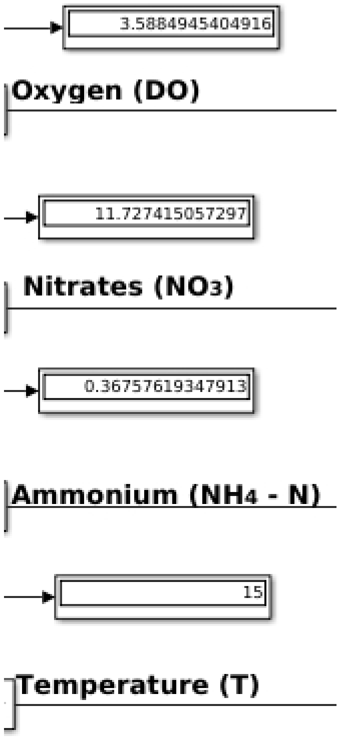
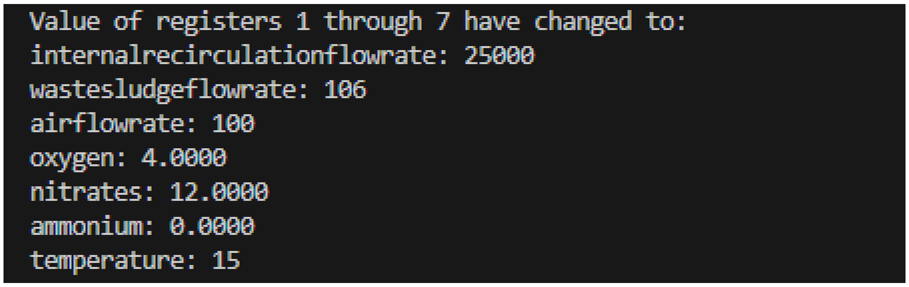


Figure 3.7.1*: Settled values for oxygen, nitrates, ammonium, and temperature*

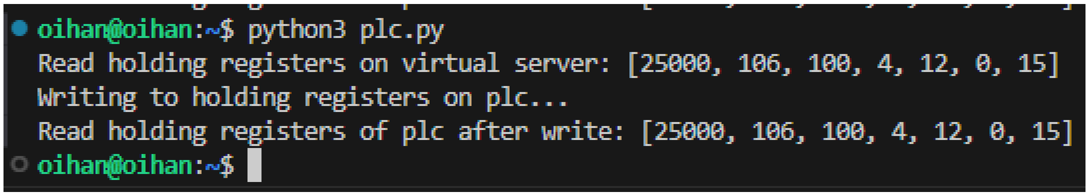
4. Monitor the data exchange between the Python interface and the WWTP
    4.1. If both programs are running successfully, the numbers in the command window of VS Code should be updating according to the numbers in the display boxes of the WWTP simulation, shown in **Figure 4.1.1**. These numbers are rounded to the nearest whole number due to rounding blocks in the WWTP simulation.



```
Value of registers 1 through 7 have changed to:
internalrecirculationflowrate: 25000
wastesludgeflowrate: 106
airflowrate: 100
oxygen: 4.0000
nitrates: 12.0000
ammonium: 0.0000
temperature: 15
```

Figure 4.1.1*: WWTP data being stored in the interface's virtual server*

5. Write the WWTP data to the PLC
    5.1. Run plc.py by typing "python3 plc.py" in a new terminal window.
    5.2. If ran successfully, this means that the program pulled the WWTP's data stored in the virtual server set up in server.py and wrote it to the PLC. Evidence of this should be printed in the command line, shown in **Figure 5.2.1**.



```
oihan@oihan:~$ python3 plc.py
Read holding registers on virtual server: [25000, 106, 100, 4, 12, 0, 15]
Writing to holding registers on plc...
Read holding registers of plc after write: [25000, 106, 100, 4, 12, 0, 15]
oihan@oihan:~$
```

Figure 5.2.1*: Virtual server data being written to the PLC*