

LAB 1: HOMEWORK

Estimated Time: 40 minutes

Objective: The objective for this Lab is to allow the students to explore the virtual SCADA environment and make changes to the current program running on the PLC.

Purpose: By modifying the current PLC program, the student will understand more about the programming language for PLCs and will explore how bad programming practices can turn into vulnerabilities that will be exploited by cyber-attacks.

Lab Setup and Requirements: The machines in the computer lab at S208 were loaded with the software required for this exercise. Please use one of the machines in the lab to complete this homework.

EXERCISE #1 – START THE SCADA LAB ENVIRONMENT

Section 1: Import the virtual machine into virtual box.

1. Open [Virtual Box](#) on your machine.
2. In the Virtual Box Manager, select File>Import Appliance
3. Click "Choose a virtual appliance file to import" icon and browse to the scadalab.ova. Click "Import".

Section 2: Login to virtual machine and run Water Tank Docker containers.

1. To start the virtual machine in the VirtualBox Manager, select the scadalab VM, right-click and select Start>Normal Start. Login using the following credentials - username:ccre, password:ccre.
2. Open Terminal by clicking on Applications -> Terminal Emulator
3. Navigate to the scripts folder with the command:

```
cd /home/ccre/scadalab/scripts
```

4. Run the script to configure the network with the command:

```
./netstart.sh
```

If it asks for a password, type: ccre

Obs: while writing the password, the characters won't be shown.



5. Start the Water Tank simulation with the command:

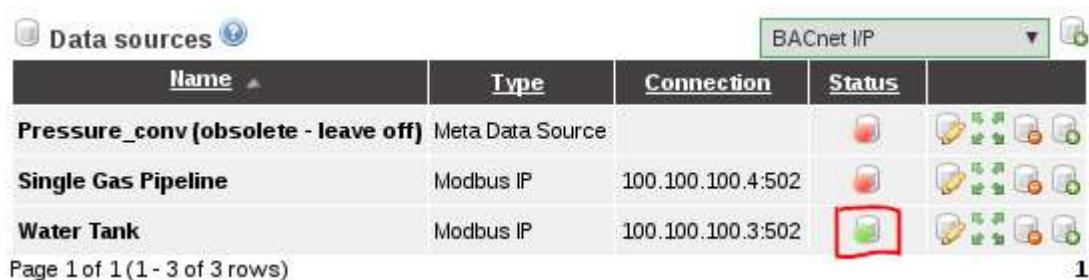
```
./watertank.sh
```

6. Launch the water tank HMI by opening the internet browser (Applications > Web Browser) and navigate to:

100.100.100.2:8080/ScadaBR

Login to ScadaBR using username:admin, password:admin

7. Click on Data Sources on the top menu and then enable water tank data sources to allow ScadaBR to pull data from OpenPLC.



8. Click on Graphical Views, select the Water Tank HMI from the drop down menu.



9. Verify the levels are changing on the Water Tank HMI.

EXERCISE #2 – MODIFYING LADDER LOGIC

This exercise will introduce basic Ladder Logic and familiarize students with how Ladder Logic can be created using the PLCopen Editor. The PLCopen Editor is a software that enables you to write PLC programs, which contains the logic that the PLC must execute.

1. Open a terminal (Applications -> Terminal Emulator) and navigate to PLCopen Editor folder:

```
cd /home/ccre/scadalab/lab1/editor/
```

2. Run PLCopen Editor with the command:

```
python PLCOpenEditor.py
```

3. In the PLCopen Editor, select File>Open. Navigate to ccre/scadalab/lab1 folder (you can find the ccre folder on the left sidebar). Double click the Water_Tower.xml file in the Lab 1 Directory.
4. Double-click on "My Program" and scroll to the bottom of the drawing area to add in the new logic for the alarm.
5. Using PLCopen Editor, write the required logic to turn on the Low_Alarm whenever the water level in the tank is below 20%.
6. Save your project by clicking on File -> Save as... and choose a filename with the .xml extension. You will be required to submit this file as your assignment, therefore it is recommended that you save it on a place that will be easy for you to find later, i.e. Desktop.

EXERCISE #3 – TESTING THE MODIFIED PROGRAM

This exercise will guide you through the steps required to upload your program to the PLC and test if it is working properly.

1. Inside PLCopen Editor, select File>Generate Program and save the file as watertower_low_alarm.st in the folder ccre/scadalab/lab1
2. Launch the internet browser (Applications > Web Browser) and go to 100.100.100.3:8080
3. Click "Choose File" and grab the watertower_low_alarm.st file created. Click Upload Program. You should receive a "Program compiled without errors" message. If you don't get this message, there is an error in your program. Go back to Exercise #2 and try to fix it.
4. Open ScadaBR again. If closed, launch the internet browser on the virtual machine and navigate to 100.100.100.2:8080/ScadaBR and login to



ScadaBR (username:admin, password:admin). Make sure you only have one window (or tab) of your browser on ScadaBR. If you have multiple windows with ScadaBR opened, you might face some weird graphical glitches.

5. Test the alarm by observing it coming on when the level is less than 20%. (Setpoints may need to be changed to allow level to reach 20%). If it doesn't come on when the level is below 20% something might be wrong with your program. Go back to Exercise #2 and try to fix it.
6. If your program is working correctly you may proceed and upload it to Canvas. You only need to upload your project file (.xml file saved in your Desktop) as your assignment.

ACKNOWLEDGEMENTS

This lab was developed at the University of Alabama in Huntsville by Stefanie Smith, Ben McGee, Thiago Alves, Joseph Lee, and Tommy Morris.

OpenPLC is a completely open programmable logic controller with development environment, human machine interface, programmable logic controller source code, and reference hardware available at <http://www.openplcproject.com/>. The OpenPLC project was founded by Thiago Alves of the University of Alabama in Huntsville.

The Simulink models, human machine interface implementation, and ladder logic program for the gas pipeline and water storage tank test beds used on this laboratory exercise are copyrighted property of the University of Alabama in Huntsville.

