Exercise 1: Getting Started with SCADA

# Objective

This exercise has two parts.

1. Install software to use OpenPLC. You will install and configure:
   1. Two virtual machines with Ubuntu Desktop LTS.
   2. Moba Xterm.
   3. OpenPLC runtime.
   4. OpenPLC Editor.
   5. Arduino IDE.
   6. SCADABr Human Machine Interface (HMI) software.
2. Design ladder logic and implement logic circuitry for your first SCADA system.

# Part 1: Installing Software

## Virtual Box VMs

1. Download and install the Oracle Virtual Box.  
   Link: <https://www.virtualbox.org/wiki/Downloads>  
   Select the link that matches your operating system (Windows, MAC, Linux).  
   Run the downloaded executable and accept all defaults.  
   You may need to reboot your computer.
2. Download and install the extension pack.

Link: <https://www.virtualbox.org/wiki/Downloads>

Click the link under “VirtualBox 6.1.16 Oracle VM VirtualBox Extension Pack

Clink on the downloaded file, in Explorer or in your browser, and follow directions to install.

## Install Ubuntu Desktop LTS

1. Download an Ubuntu Desktop LTS .iso file.

Link: <https://ubuntu.com/download/desktop>  
Click the Download link to download. The file is quite large (2GB+). It will take a while to download. Coffee break 😊

1. Install the .iso within VirtualBox.
2. Open the Oracle VirtualBox VM Manager.
3. Press *New*
4. Name the new VM *PLC*
5. Change the type to Linux
6. Select Defaults for Memory Size, Hard Disk, Hard disk file type, Storage on physical hard disk, File location and size.
7. Your VM is now created, but we still must install the OS on the VM.
8. Within the Oracle VirtualBox VM Manager, select the PLC VM, and then click Settings.
9. Click Storage.
10. Under Controller IDE select the DVD symbol.
11. On the far right click the DVD symbol and select *Choose a disk file*.
12. Select the .iso file downloaded from the Ubuntu website.
13. Start the VM and follow install instructions within the Ubuntu OS.
14. Remove the .iso from the VM DVD drive.
15. Reboot your PLC VM (this new VM).
16. Configure to turn off the desktop (GUI) environment.  
    We will run with command line to conserve computer resources.

Link: <https://linuxconfig.org/how-to-disable-enable-gui-on-boot-in-ubuntu-20-04-focal-fossa-linux-desktop>

Open a terminal.

Enter command *sudo systemctl set-default multi-user*

Enter command *shutdown -r* to reboot.

## Clone the PLC Linux to create a second VM for your HMI.

In the Oracle VM VirtualBox Manager

Select and highlight the PLC VM

Right click and select Clone

Set the name to *HMI* then click Next

Select Linked clone then click Clone

## Install OpenPLC runtime in the PLC VM

Link: <https://www.openplcproject.com/runtime/linux/>

git clone https://github.com/thiagoralves/OpenPLC\_v3.git

cd OpenPLC\_v3

./install.sh linux

Use command *ip a* to note your VM’s IP address.

The OpenPLC runtime starts automatically whenever your VM boots.

Create a network proxy to access the OpenPLC runtime from your host’s browser.

In the Oracle VM VirtualBox Manager

Select the PLC VM

Click settings and select Network

Under Adapter 1, Advanced, click port forwarding.

Add a rule to forward port 8080 from the PLC VM to port 8000 on the host.

Then in your browser navigate to <http://localhost:8000> (This webpage is the control interface for your PLC)

*Note: you may need to reboot the VM for the proxy to take effect.*

## Install MobaXterm

Link: <https://mobaxterm.mobatek.net/download.html>

Download the installer application.

Run the installer.

Open the MobaXterm application.

Click *Start local terminal*.

Record the IP address provided in the local terminal. This will be used when starting GUI applications in VM’s.

## Install the OpenPLC editor in the PLC VM

Link: <https://www.openplcproject.com/plcopen-editor/>

sudo apt-get install git

git clone https://github.com/thiagoralves/OpenPLC\_Editor

cd OpenPLC\_Editor

./install.sh  
  
To start the OpenPLC Editor:

Open a terminal in the VM.

Use the command: *DISPLAY=192.168.86.63:0.0 plcopeneditor&*

Note: replace the IP address above with the IP address from Mobaxterm

## Install the Arduino IDE in the PLC VM

Link: <https://www.arduino.cc/en/guide/linux>

In your PLC VM, download https://downloads.arduino.cc/arduino-1.8.13-linux64.tar.xz

Uncompress the IDE installer (tar -xvf arduino-1.8.13-linux64.tar.xz)

cd Downloads/Arduino-1.8.13

./install.sh

DISP

To run: DISPLAY=192.168.86.63:0.0 arduino& (a window will open on the host PC)

*Note: replace the IP address above with the IP address from Mobaxterm*

## Install SCADABr in the HMI VM

Link: https://www.openplcproject.com/reference/scadabr/

git clone https://github.com/thiagoralves/ScadaBR\_Installer.git

cd ScadaBR\_Installer

./install\_scadabr.sh  
  
To open SCADABr on a browser (option 1):

Open a terminal in the VM.

Use the command: *DISPLAY=192.168.86.63:0.0 firefox&*

*Note: replace the IP address above with the IP address from Mobaxterm*

Navigate to <http://localhost:9090/ScadaBR>

*Note: you may need to reboot the VM for the proxy to take effect.*

To start the OpenPLC Editor (option 2):

Create a port forwarding rule to map VM port 9090 to host port 9000.

In your host’s browser Navigate to <http://localhost:9000/ScadaBR>

Note: There may be problems with the apache server.

Navigate to the tomcat directory (*cd /opt/tomcat6/apache-tomcat-6.0.53/*).

*cd logs*

*cat catalina.out | less* (look for SEVERE)

restart apache (*sudo /opt/tomcat6/apache-tomcat-6.0.53/bin/startup.sh*)

## Linux Command Line tips:

1. Set your display for MOBAXTERM.

Use *export DISPLAY=192.168.1.1:0.0*

Change the IP address above to match your MOBAXTERM IP address. Start a local terminal in MOBAXTERM to find your MOBAXTERM IP address.

1. Shutdown a Linux VM with the *shutdown now* command.
2. Reboot a Linux VM with the *shutdown now -r* command
3. Capture or release mouse (move mouse control from VM to host or host to VM).
   1. Press the capture key. This is the right CTRL key by default.

# Part 2: Creating your first OpenPLC Project

Follow directions from the OpenPLC Project webpage to build your first project.

Review the material at the links below for background information needed to complete this exercise.

* [What is a PLC?](https://www.openplcproject.com/reference/basics/what-is-a-plc)
* [Introduction to Ladder Logic](https://www.openplcproject.com/reference/basics/introduction-to-ladder-logic)
* [Contact](https://www.openplcproject.com/reference/basics/contact)
* [Coil](https://www.openplcproject.com/reference/basics/coil)

Using the Arduino kit provided, follow the instructions at the link below to build the circuit shown and then create ladder logic to run on your OpenPLC.

Link: <https://www.openplcproject.com/reference/basics/first-project>

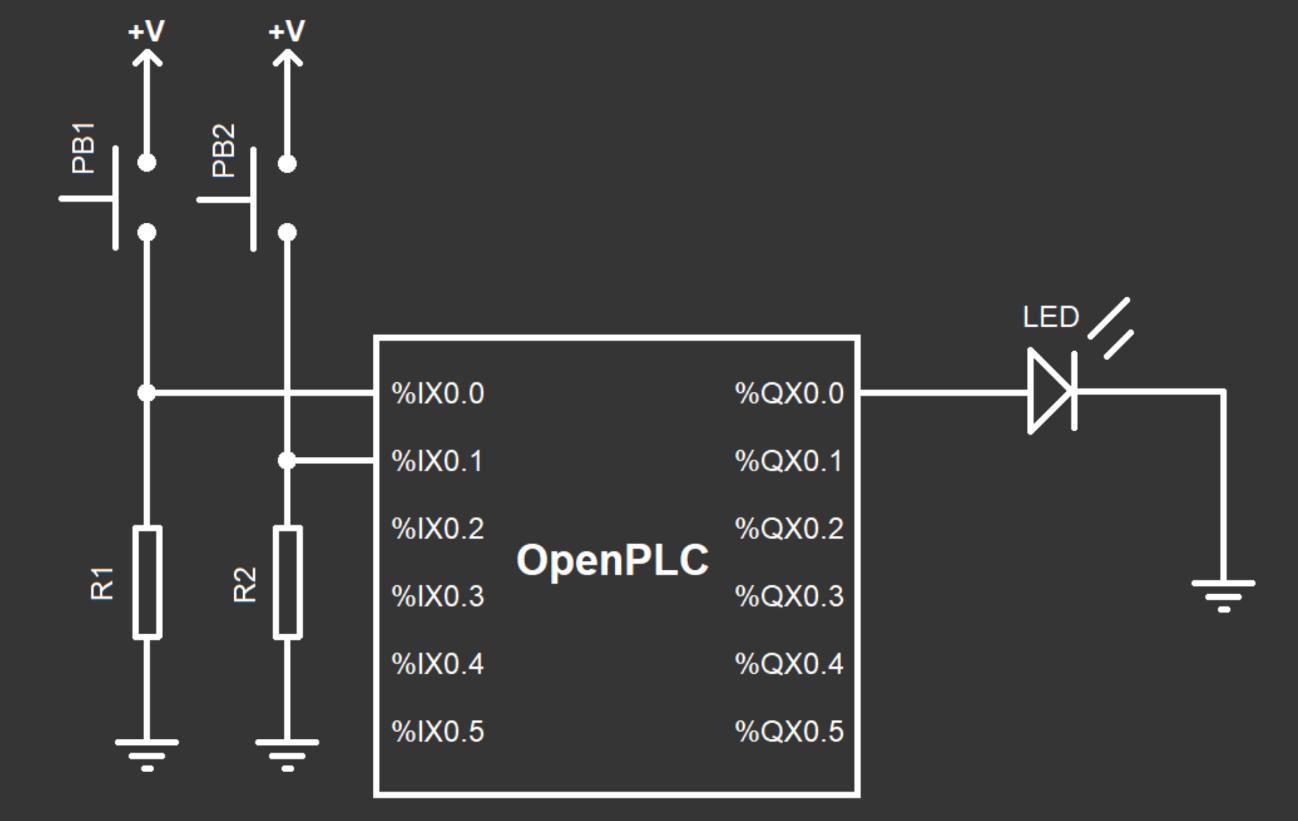


Figure 1: Circuit for First

Additional Information:

1. Refer to slide deck 03-04-Boarding\_C.pptx for more information.
   1. Arduino pinout: Slides 15-16
   2. Arduino to OpenPLC V3 pin name conversion: Slide 8
   3. Resistor sizes via color bands: Slide 20
   4. Switch orientation: Slides 21-22
   5. LED pin orientation: Slide 25
2. The figure shows a generic OpenPLC. You are connecting to pins on the Arduino Uno.
3. There is no resistor shown before between the LED and the PLC. Add one to avoid burning your LED. Use between a 1K – 10K ohm resistor. The lower the resistance the brighter the LED.
4. R1 and R2 are pull down resistors. All input pins that you use should have a pull down resistor between 1K and 10K ohms. Note: without the pull down the pin will “float”. This will cause unpredictable behavior which is unpredictable and therefore hard to debug.
5. The pin numbers for Arduino Uno are slightly different. Use %IX100.0, %IX100.1 and %QX100.0 (use these values in the OpenPLC editor).
6. There is additional pin mapping information for Arduino here: <https://www.openplcproject.com/runtime/arduino/>. This maps the %IX100.0, %IX100.1 and %QX100.0 values from Figure 1 to Arduino pin numbers.

# Post Exercise Report

1. Which machine (virtual or physical) serves as the PLC for the SCADA project completed in part 2? In other words where is the microprocessor that implements the PLC?
2. Which machine (virtual or physical) serves as an input/output board to connect the PLC to the physical system implemented in part 2?
3. Describe the steps to observe the value of a switch on the breadboard. What occurs at the switch, the Arduino, PLC, network, and HMI? What order does this occur?
4. Describe the steps to turn on a LED on the breadboard. What occurs at the switch, the Arduino, PLC, network, and HMI? What order does this occur?