# Exercise 1: Getting Started with SCADA

## 1 Objective

This exercise has two parts.

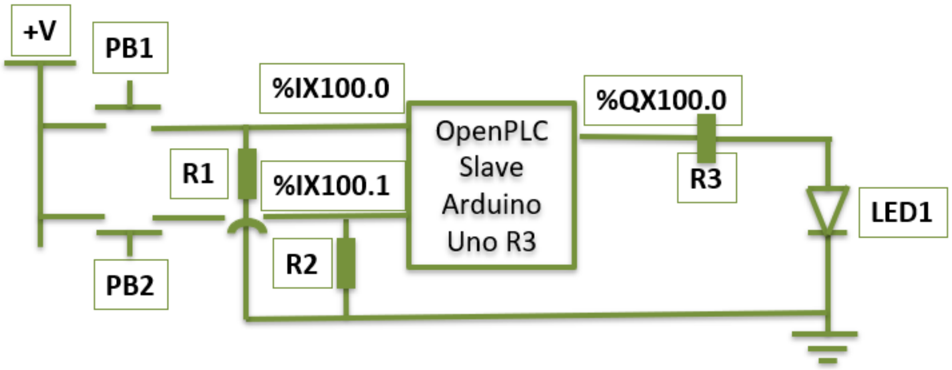
1. Setup our lab environment on your laptop/computer. You will install and configure:
    a. Install the OpenPLC runtime (on a VM).
    b. Install OpenPLC Editor on your host PC.
2. Design ladder logic and implement logic circuitry for your first SCADA system.

## 2 Installing Software

1. Download a Kali Linux Virtual Box VM here: https://www.kali.org/get-kali/#kali-virtual-machines
2. Download VirtualBox and VirtualBox Extension Pack from here: https://www.virtualbox.org/wiki/Downloads
3. Download OpenPLC Editor and install on your host PC from: https://autonomylogic.com/
4. Install VirtualBox - just double-click the installer file and follow along with the setup wizard.
5. Install VirtualBox Extension Pack - just double-click the file **after** VirtualBox has been installed. This is crucial as it enables USB support on your guest virtual machines.
6. Unzip the Kali Linux VM image with 7-zip.
7. Din the Virtual Box Manager – Use Machine -> Add to import the .vbox file from the unzipped Kali directory.
8. Start “PLC VM” and install the OpenPLC run time on it following the instructions on the OpenPLC Project website. (Hint: default user is ***kali*** with password ***kali***) https://autonomylogic.com/docs/installing-openplc-runtime-on-linux-systems/
9. Shutdown the VM and rename the VM – “PLC VM” or something descriptive. Note: you may need to change the permission in Windows Explorer to make the unzipped director writeable.
10. Make sure everything is working:
    a. Launch the PLC VM
    b. Login and check IP address of each VM with the $ip\ a$ command
    c. Open the browser on your host machine (Edge, Chrome, Firefox, etc) and launch the web portal for OpenPLC
    d. OpenPLC -> http://[PLC VM IP ADDRESS]:8080
    e. OpenPLC Run Time default user is ***openplc*** with password ***openplc.***

# 4 Building your circuit

Build the circuit in the figure below on the breadboard from your Arduino kit.



- +V is 5 Volts from the Arduino
- PB1 and PB2 are momentary pushbuttons
- R1 = R2 = R3 = 10 k Ohm resistor
- LED1 is a light emitting diode (LED)

# 6 Creating Ladder Logic

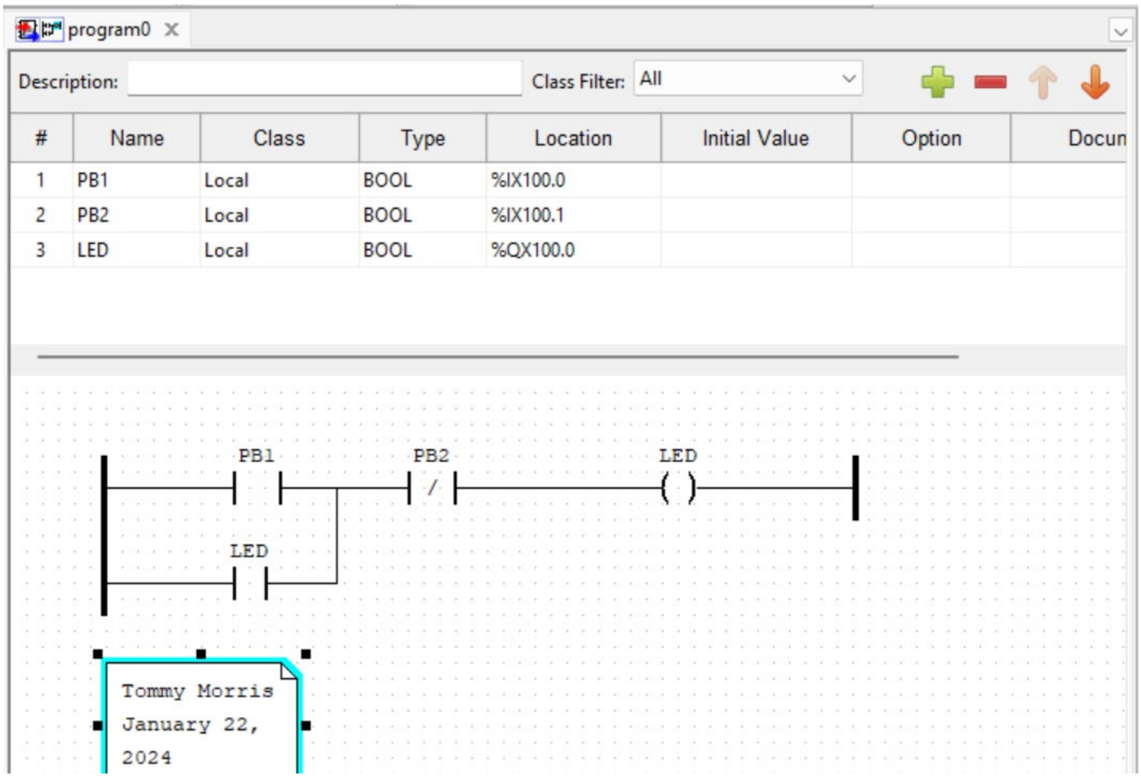
Follow directions from the OpenPLC webpage to build your first ladder logic project in the OpenPLC Editor.

https://autonomylogic.com/docs/3-2-creating-your-first-project-on-openplc-editor/

Add a comment with your name and the date. You will take a similar screen shot for submission.

- Click this button in the menu bar: CMT
- Click, drag, and release in the ladder editing window to place the comment.

The figure below shows the final ladder logic and variable declarations.



Description:

Class Filter: All

| # | Name | Class | Type | Location | Initial Value | Option | Docun |
|---|---|---|---|---|---|---|---|
| 1 | PB1 | Local | BOOL | %IX100.0 | | | |
| 2 | PB2 | Local | BOOL | %IX100.1 | | | |
| 3 | LED | Local | BOOL | %QX100.0 | | | |

Tommy Morris
January 22,
2024

# 8 Creating a Arduino OpenPLC Slave Device

### *Part 1:*

- Make sure to plug in the Arduino to your PLC (via USB port) before following the instructions below.
- Follow the instructions here to create an Arduino Slave Device using an empty project. Scroll to the bottom of the page for instructions. https://autonomylogic.com/docs/2-6-slave-devices/

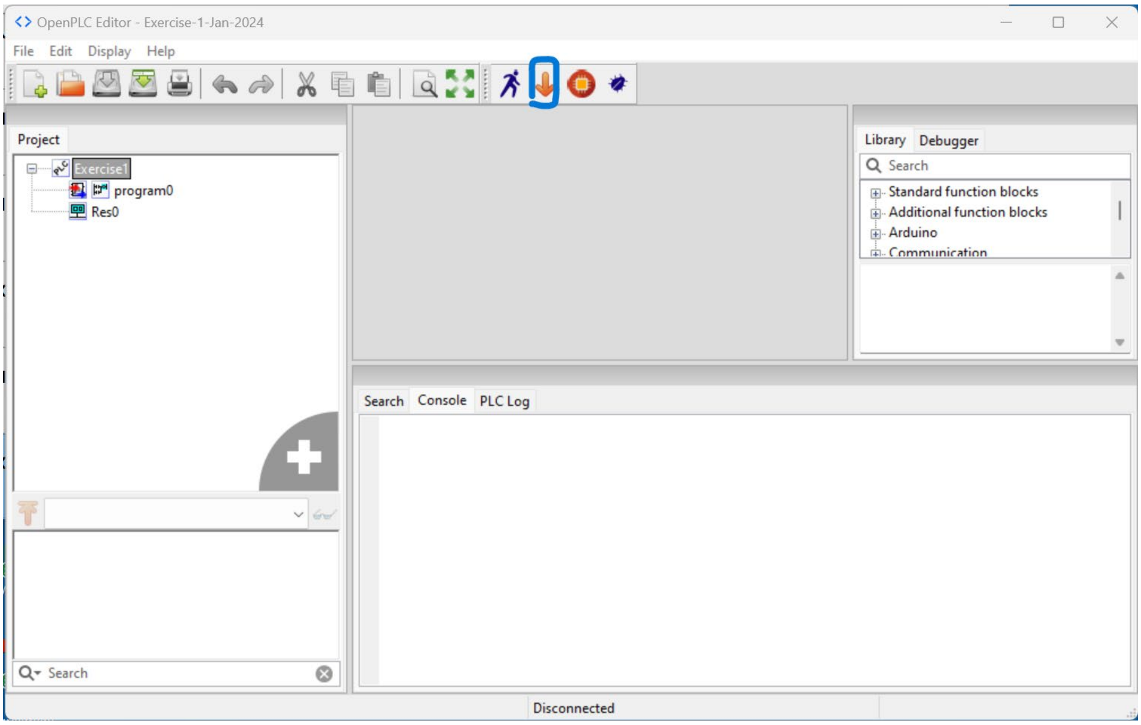
- In the Transfer tab
  - Select Board Type Arduino Uno
  - COM Port – look for Arduino Uno in the list
- In the Communications tab
  - Enable Modbus Serial
  - Set the Slave ID to 0
  - before clicking Transfer to PLC.
- Back to the Transfer tab
  - Click Transfer to PLC
  - The process takes a few minutes.

### *Part 2:*

- In your host browser navigate to the OpenPLC run time web interface.
  - OpenPLC run time URL-> http://[PLC VM IP ADDRESS]:8080
  - OpenPLC Run Time default user is ***openplc*** with password ***openplc.***
- On your OpenPLC VM, capture the USB port associated with the Arduino.
- Click “Slave Devices” and “Add New Device”
- Provide a device name and select the COM Port associated with your Arduino (there should only be one /dev/ttyACM0 or similar)
- Scroll down and click “Save Device”

# 9 Uploading to the PLC Run Time

- Save your program from the OpenPLC Editor. Click the down arrow on the menu bar and name your “.st” file. Remember where you put the file.



In the OpenPLC Run Time Web Interface

- Click Programs
- Choose File
  - Select your ".st" file
  - Click Upload Program
  - Type a name in the Program Name field
  - Scroll down and click Upload Program
  - Scroll down and click Go to Dashboard
  - On the left click Start PLC
- If the program is running correctly two LEDs (labeled TX and RX) just above the USB port should be blinking rapidly.
- Test your circuit by pressing button 1 and button 2.

Additional Information:

1. Refer to slide deck 03-04-Boarding_C.pptx for more information.
   a. Arduino pinout: Slides 15-16
   b. Arduino to OpenPLC V3 pin name conversion: Slide 8
   c. Resistor sizes via color bands: Slide 20
   d. Switch orientation: Slides 21-22
   e. LED pin orientation: Slide 25
2. The figure shows a generic OpenPLC. You are connecting to pins on the Arduino Uno.
3. 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 hard to debug.

# 10 Submitting your project

A. You will upload a screen capture of your ladder logic, variable declarations, and comment with your name and date.

B. You will also need to upload a cell phone video demonstrating the correct functionality of your circuit and ladder logic. Capture the following sequence for your video.

   a. Press button one – show the LED turns on
   b. Press button two – show the LED turns off
   c. Press button one – show the LED turns on
   d. Press button two – show the LED turns off