Lab worksheet one

Objectives of Lab

- Set up FactoryIO on your virtual machine.
- Set up the Raspbian operating system on a micro-SD card.
- Set up OpenPLC Runtime on your Raspberry Pi.
- Set up OpenPLC Editor on your virtual machine.
- Create a test scene to check the two applications are set up correctly.

Background

"Cyber-Physical Systems (CPSs) are engineered systems that are built from, and depend upon, the seamless integration of computation, and physical components". (The Cyber Security Body Of Knowledge, 2019). Cyber-Physical Systems are used to automate many physical infrastructures, including water systems, power grids and chemical reactors, all of which are critical to daily life. As Cyber-Physical Systems have been developed, new attacks have formed. Through a series of tutorials, we aim to educate you on this CyBOK knowledge area and the different attack vectors in Cyber-Physical Systems security.

Throughout the series of tutorials, we will be using FactoryIO as a simulator for the physical part of the system, and OpenPLC as our controller. In this worksheet, you will learn how to download and install each of these programs onto your Raspberry Pi and your Windows 10 virtual machine. You will also learn how to set up a test scene and configure FactoryIO and OpenPLC to work together to control the scene.

System requirements and Prerequisites

- No prerequisites to this lab
- Raspberry Pi 3B+
- 16GB Micro-SD card
- USB Card Reader
- Raspberry Pi Imager V1.5
- Raspbian Operating System (32-Bit Released 11.01.2021)
- OpenPLC Runtime V3
- OpenPLC Editor V1.0
- FactoryIO V2.4.6

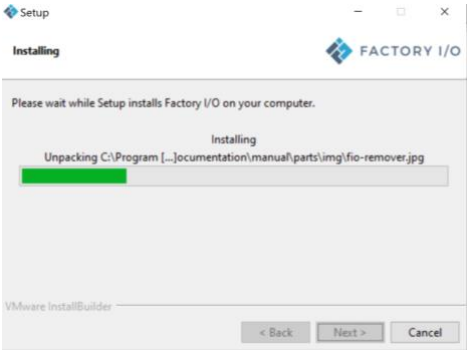
Task One – Install FactoryIO

Before you download and setup FactoryIO, you will need to acquire the rights to use it. This can be done via the 30-day free trial of FactoryIO, or by purchasing one of the editions to use.

Step 1

Once you have the rights to use FactoryIO, you will need to download it. To do so you should go to https://factoryio.com/download-archive and select the installer for version 2.4.6.

Once the installer has downloaded, open it to run the installer. Select the correct language and follow the installer wizard.



Once the wizard has finished installing the software, click finish. You should now be able to open the software.

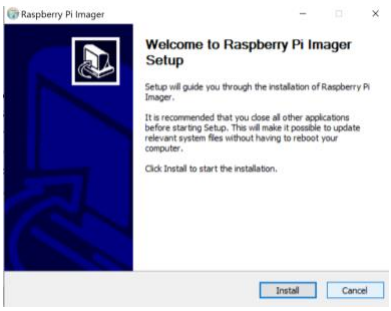
Task Two – Install Raspbian

Unless you have bought a Raspberry Pi kit that comes with Raspbian pre-installed, you will need to install the Raspbian operating system on your Pi. For this task, you will need a micro-SD card and a card reader.

Step 1

The easiest way to install Raspbian onto an SD card is to download and use the Raspberry Pi Imager. To download this software, go to https://www.raspberrypi.org/software/

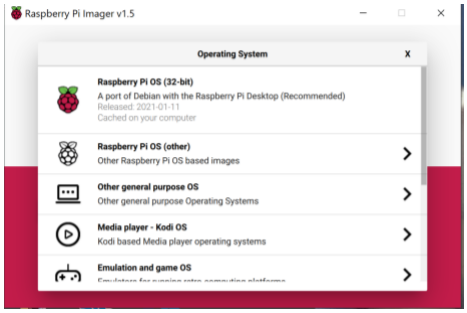
Once you are on the website, select the download for windows. Once the download is complete, open the .exe and run the install wizard which is shown in the image below.



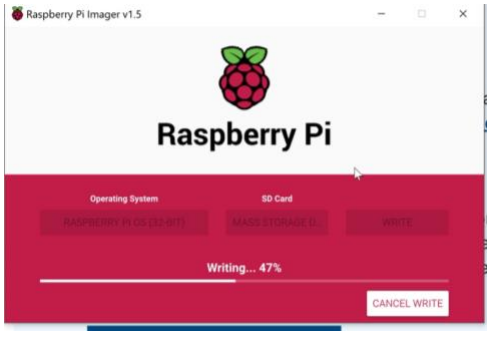
Step 2

Once the application has been installed, open the Raspberry Pi Imager application which should appear as it does in the image below.

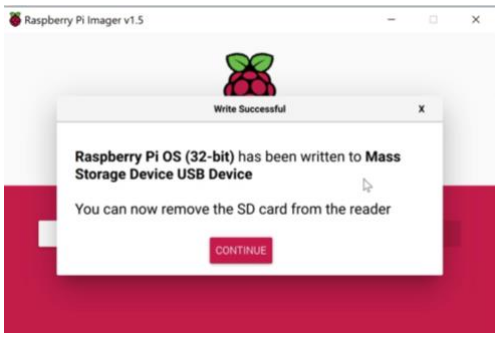
Once you have connected your micro-SD card, select the operating system on the application that you wish to use. In this case, we are using Raspberry Pi OS (32-Bit) from the list shown below.



You will also need to select your micro-SD card in the SD card drop-down list. Once the selections are correct, simply press the write button and the application will do the rest for you. Be aware that if anything is currently stored on your micro-SD card, it will be erased during this process.



As the installation is in progress, you will see a percentage complete bar for writing to the micro-SD card, followed by the verification. Do not remove the micro-SD until you see the pop-up message



To test that the operating system has installed properly, simply pop the micro-SD into your Pi and boot the machine.

Task Three – Install OpenPLC Runtime

For this task we will download and install OpenPLC on our Raspberry Pi, which we installed the Raspbian operating system on during the previous task.

Step 1

To install OpenPLC on the Pi, we will use three lines of code on your Pi's terminal and clone OpenPLC from git. Having installed the Raspbian operating system in our last task, git should already be installed on your system. If for any reason, you do not have git pre-installed, the following line of code can be run within the terminal to install it.

sudo apt-get install git

The three lines of code below are what you need to type to install OpenPLC, each line should be run individually.

git clone https://github.com/thiagoralves/OpenPLC_v3.git
cd OpenPLC_v3
./install.sh rpi

Be aware that the OpenPLC can take a while to install, it took thirty-four minutes on our Pi to finish, however it can take up to an hour depending on your device.

Step 2

Once the install has finished, simply restart the Pi and it will automatically start once the reboot is complete. You will not be able to visibly see that OpenPLC has started, but if you have completed the correct steps then it will be running in the background.

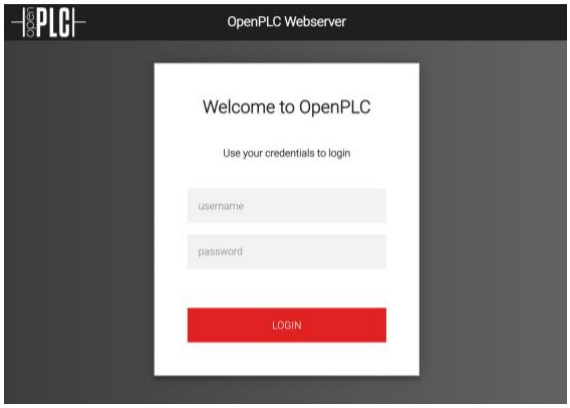
To access OpenPLC runtime, you will need to either already know your Pi's IP address, or be able to find the address. If you do not know your IP address, navigate to the terminal on your Raspberry Pi. Then enter the following command, which will return the IP address.

hostname -I

Once your IP address is known, you should navigate to port 8080 of that IP. To do this you will open a browser, and type in your IP followed by :8080. As an example, if the IP address was 192.168.0.64, then the line below is what you would enter into the address bar of your browser.

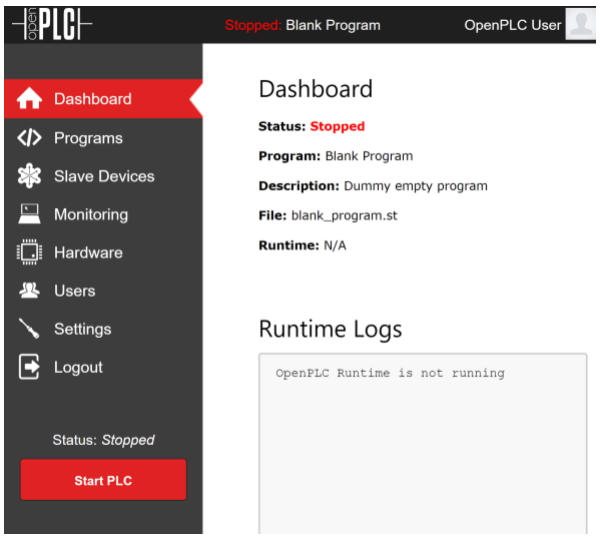
192.168.0.64:8080

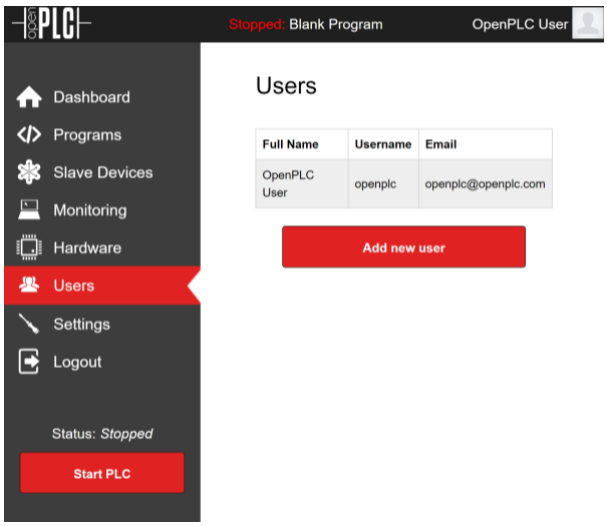
This will take you to the login page for OpenPLC, as shown in the image below.



To begin with, the default login credentials are as follows;
Username: openplc
Password: openplc

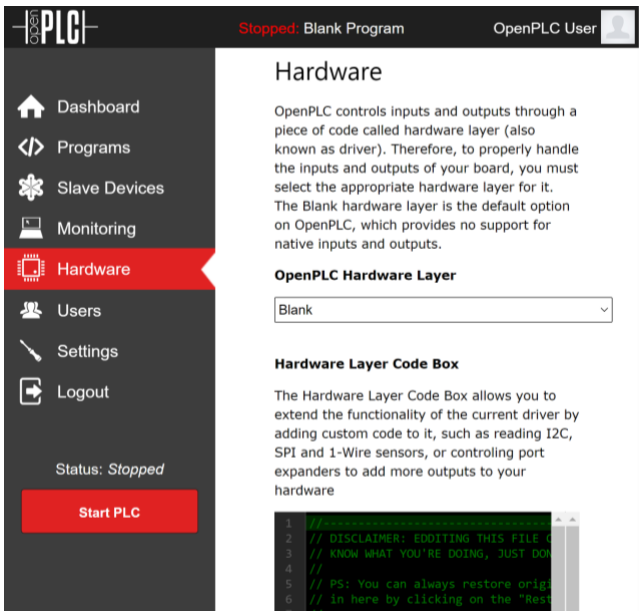
At this point, you can navigate from the dashboard to the users section, as shown in the image below, to change the username and password. To do so, click the user and it will take you to an edit user screen.





Step 3

As a default, on the Raspberry Pi the hardware setting will be set to Blank, as shown in the image below. This is due to the different variations of the Pi, meaning we need to manually select what Pi we are using.



In the drop-down, select Raspberry Pi, ensuring you do not select the 2011 old model option. Then scroll down and save the change. This will allow OpenPLC to interact with the correct hardware pins.

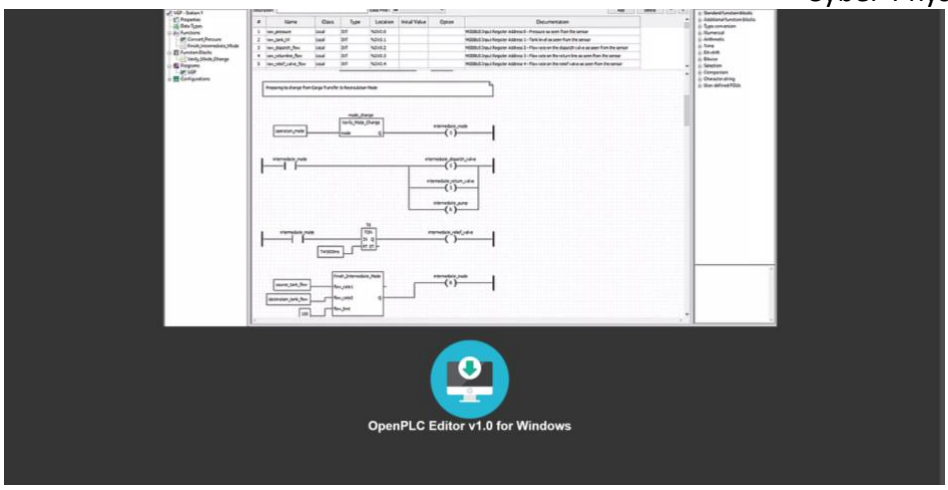
Task Four – Install OpenPLC Editor

For this task we will download and install OpenPLC Editor on our Windows 10 virtual machine.

Step 1

To download the Editor program for OpenPLC, you will need to navigate to https://www.openplcproject.com/plcopen-editor/

From this page you should see a button with a download link for the Windows version of this program.



Click the button, and wait for it to download. You will then need to unzip the downloaded folder. This will give you access to the OpenPLC Editor programme.

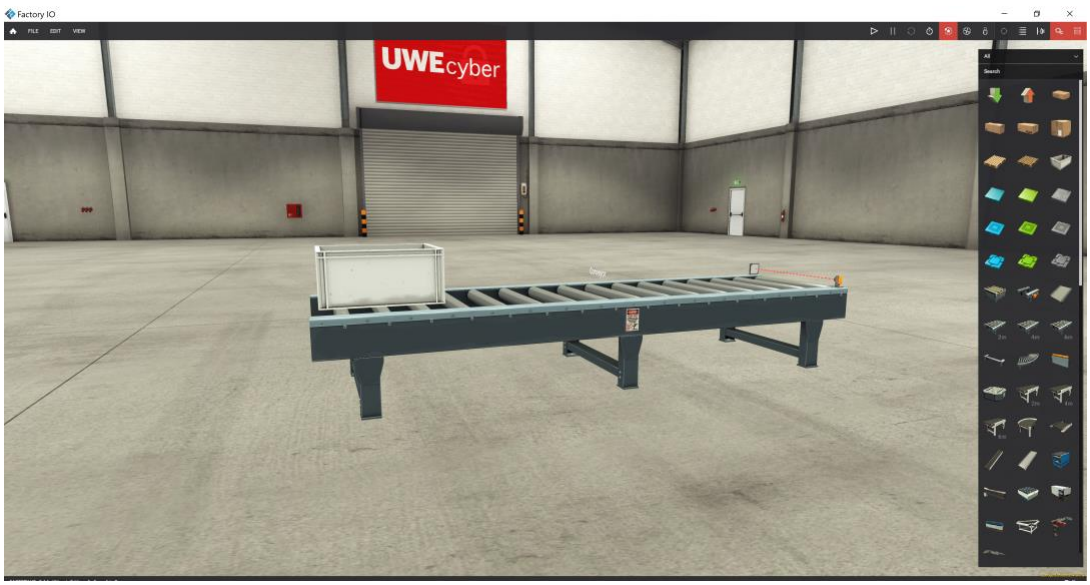
Task Five – Create a scene to set up and test FactoryIO and OpenPLC connection

To set up the connection between FactoryIO and OpenPLC runtime, we will need to create a test scene within FactoryIO and an OpenPLC program to control the scene.

Step 1

For our test scene in FactoryIO, we are going to modify one of FactoryIO's pre-built scenes. The scene we will use is scene 1 – from A to B. this can be found under the scenes option once FactoryIO has been opened.

Upon opening, it will look similar to the image below. The initial scene consists of a 4m roller conveyor with an attached retroreflective sensor on one end, and a stackable box on the other end.



To modify the scene and fully test the link we create between FactoryIO and OpenPLC, we will add a start and stop button. To do so, you will need to add a column, and electric switchboard, and the start and stop buttons themselves. These can be easily found if you change the dropdown list, from all to operators, where you will see each of these items.

The exact positioning of the items does not matter, if all four components are connected, they will work. The image below shows where we placed our electrical components.

To set the scene to work with OpenPLC as its controller you will need to edit the drivers. To do this, go to file, and then select drivers from the list. On the drop-down list, where it will initially say none, you will need to set it to Modbus TCP/IP server. This will allow FactoryIO to be set up as a slave device to OpenPLC Runtime.

Once you have set it to Modbus TCP/IP server, select the configuration option, where we can set the options that we will need to use to create the slave device. It is likely that your IP address will already be filled in for you, however if it is not, then you can find this through the command prompt by simply typing *ipconfig*. The correct IP address will usually be within the last section, under IPv4 address, as shown below.

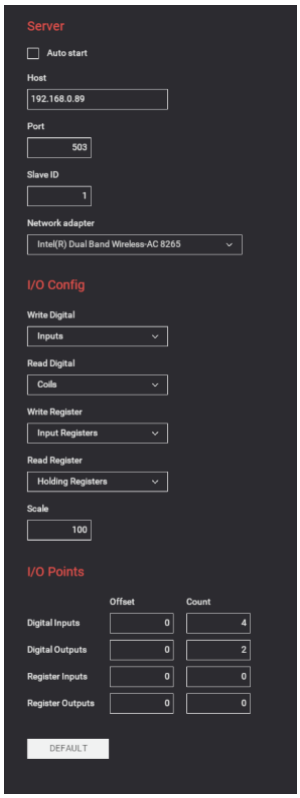
```
Wireless LAN adapter WiFi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::205b:ee1:ebf5:a2fe%13
   IPv4 Address. . . . . . . . . . . : 192.168.0.89
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

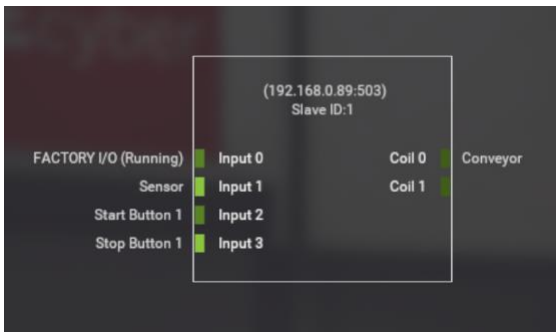
Once the IP address is filled in, change the port being used from 502 to 503. This is because OpenPLC will use port 502, as it is the default for Modbus. By using 503 for FactoryIO, this prevents any conflict.

You will also need to set the slave ID, which we have set to 1 as for this exercise we will only have the one slave device.

The final change to make, is to set the correct count for our digital inputs and outputs. In this scene we have four inputs, and one output, although we will leave the outputs set to its default of two. All of these changes are shown in the image below.



Having changed the configuration, you will need go back to the previous section, and add all of our inputs as shown below. This ensures that OpenPLC can see our running FactoryIO scene, and take input from the sensor and the two buttons. To add or move the inputs, drag them from the list on the left-hand side of the screen to the central configuration point.

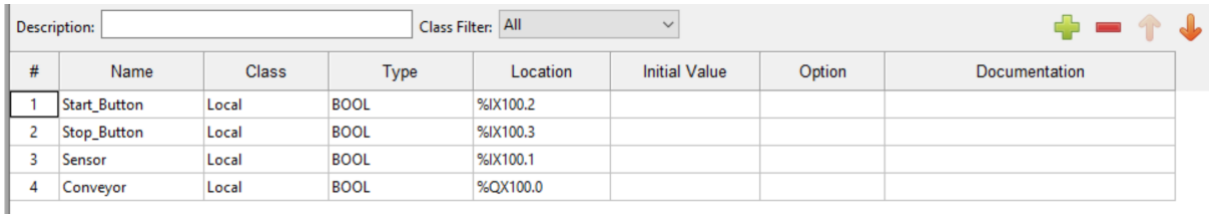


Once you have set the options above, you will need to click the start button in the top right corner of the application.

Step 2

To enable us to program and control the scene, we need to create a logic program within OpenPLC Editor.

To start, open OpenPLC Editor and create a new LD program. You will need to create a new directory for this to go in, and name it appropriately. Once you have created the file, click the green plus symbol to add variables to your program. You will need four variables, which I have shown in the image below. If you have set your test scene correctly, you will only need to fill in the variables. If your inputs from the test scene are in a different order then the location of the variables will need to be adjusted.



#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	Start_Button	Local	BOOL	%IX100.2			
2	Stop_Button	Local	BOOL	%IX100.3			
3	Sensor	Local	BOOL	%IX100.1			
4	Conveyor	Local	BOOL	%QX100.0			

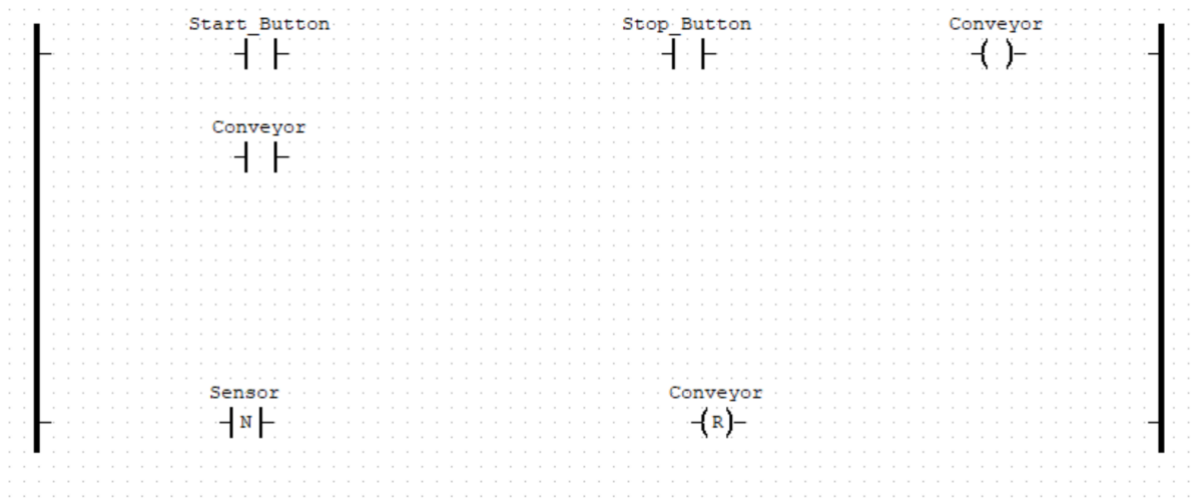
The variable location is specified based on the input number from FactoryIO. As such, location 100.1 in OpenPLC equates to input 1 in FactoryIO. This would also be the same for outputs, but we do not have those in the test scene.

You now need to create to ladder logic using the set variables. The first thing needed to create he ladder, is to add a beginning and end power rail. To do this, you will need to click on the button with the symbol shown below. You will need to select this twice, once to create the left rail, and once for the right rail.

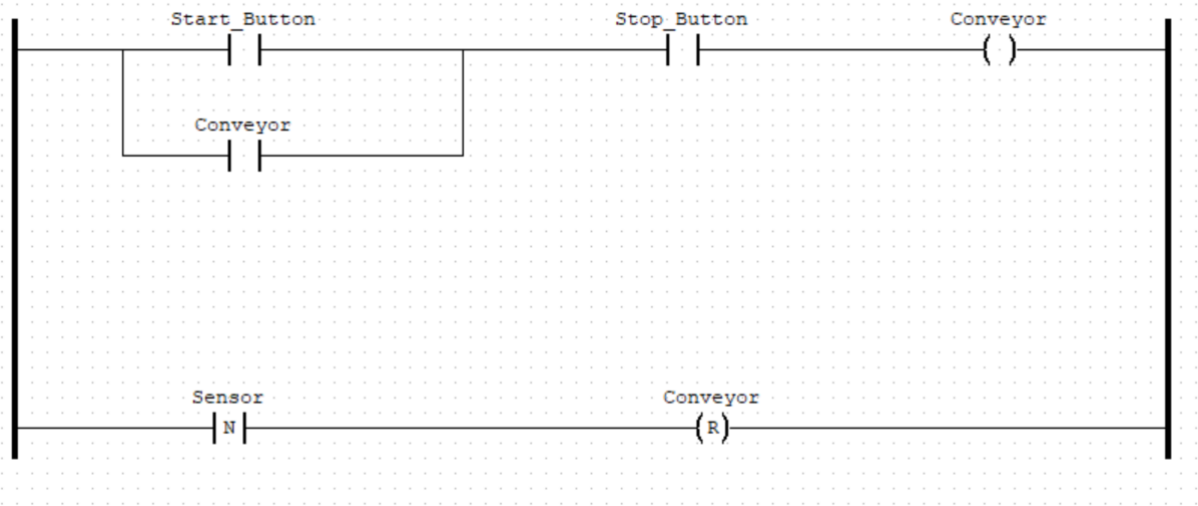
You can adjust these to become larger to suit your needs. You will now need to create new contacts, based on the variables we have. To create a contact, you will need to use the button shown below.

We need four contacts, one for each variable. The Start_Button, Stop_Button and the Conveyor variable need to be normal contacts, whilst the Sensor needs to be a falling edge variable.

You will also need to create two coils, from the Conveyor variable. One of the coils needs to be a normal coil, whilst the other should be a reset coil. Arrange the contacts, coils and rails as shown below.



The final stage to setting up your ladder logic is to join the variables together, as shown below. This is done by dragging the mouse from one part of the ladder to another.



Once that is done, save the program and compile it to create a .st file, which we can then use in OpenPLC Runtime.

Step 3

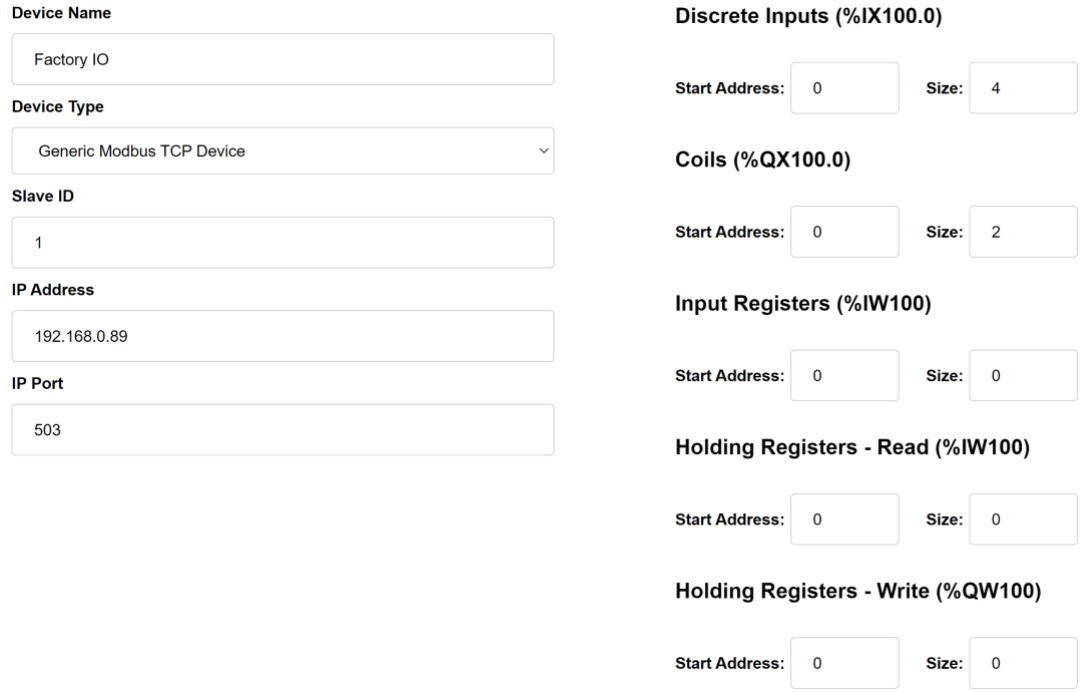
To upload the program, login to OpenPLC Runtime, as we previously did when setting up the hardware settings.

Navigate to the programs page, which should show just the blank program that OpenPLC installs with. At the bottom of the page, select choose file, and select the .st file we just created. Then select upload. Wait for it to upload before navigating to the dashboard to check that it has uploaded correctly.

Step 3

From within OpenPLC Runtime, navigate to the slave devices section, where we will now set up our factory as the slave device.

Select add new device, which will take you to a page that you will need to fill in based on our FactoryIO setup. The image below shows you how we set the slave device up.



As long as you followed the tutorial correctly, the only information you will need to set differently to what is shown above, is that of the IP address, as your machine with FactoryIO will have a differing IP address. Ensure that you fill out all the boxes before saving, as it may look like some were pre-filled however they are not, and will cause you an error upon saving.

Once you have set up the device, you can run the program by selecting start PLC. If you then navigate back to the test scene, your scene will be able to be controlled via the start and stop buttons set up, and the conveyor will automatically stop prior to the box falling off.

Takeaways

Having completed this worksheet, you should now have a baseline understanding of FactoryIO and OpenPLC. You will be able to set up a connection between the two programs, enabling you to control a factory scene with your ladder logic program.

Further reading

Cyber-Physical Systems Security Knowledge Area Issue 1.0 – available from https://www.cybok.org/knowledgebase/

FactoryIO – available from https://docs.factoryio.com/

OpenPLC – available from https://www.openplcproject.com/

Ladder Logic – https://www.plcacademy.com/ladder-logic-tutorial/

References

The Cyber Security Body Of Knowledge (2019) *Cyber-Physical Systems Security Knowledge Area Issue 1.0* [online]. Bristol: CyBOK © Crown Copyright, The National Cyber Security Centre 2018. Available from: https://www.cybok.org/knowledgebase/ [Accessed 12 March 2021].