


























- Using the Ladder Logic pdf, implement your own OpenPLC Ladder Logic program.
- To exactly correspond to the Modbus register mappings with the tags in the ignition project, use the following variable table as a reference (note the register locations for your own OpenPLC project):

| # | Name | Class | Type | Location | Initial Value | Documentation | Debug |
|----|-----------------------|-------|------|----------|---------------|---------------|---|
| 0 | Fill_State | Local | BOOL | %IX0.0 | 0 | |  |
| 1 | Mix_State | Local | BOOL | %IX0.1 | 0 | |  |
| 2 | Drain_Status | Local | BOOL | %IX0.2 | 0 | |  |
| 3 | Valve_A | Local | BOOL | %QX0.0 | | |  |
| 4 | Tank_Level | Local | DINT | %MD0 | 10 | |  |
| 5 | Level_Maxed | Local | BOOL | %QX0.1 | | |  |
| 6 | SR0 | Local | SR | | | |  |
| 7 | Level_Minimum | Local | BOOL | %QX0.2 | | |  |
| 8 | Mixert_Time_Done | Local | BOOL | %QX0.3 | | |  |
| 9 | Mixer | Local | BOOL | %QX0.4 | | |  |
| 10 | Valve_B | Local | BOOL | %QX0.5 | | |  |
| 11 | SR1 | Local | SR | | | |  |
| 12 | Fill_State_Set_Reset | Local | SR | | | |  |
| 13 | Fill_State_Trigger | Local | BOOL | %QX0.6 | | |  |
| 14 | SR2 | Local | SR | | | |  |
| 15 | TON0 | Local | TON | | | |  |
| 16 | SR3 | Local | SR | | | |  |
| 17 | TON1 | Local | TON | | | |  |
| 18 | Tank_Level_Timer_Done | Local | BOOL | %QX0.7 | | |  |
| 19 | Level_raiser | Local | BOOL | %QX1.0 | | |  |
| 20 | level_faller | Local | BOOL | %QX1.1 | | |  |
| 21 | E_Stop | Local | BOOL | %IX0.3 | 0 | |  |
| 22 | Start | Local | BOOL | %IX0.4 | 1 | |  |
| 23 | System_Run | Local | BOOL | %QX1.2 | | |  |
| 24 | Stop | Local | BOOL | %IX0.5 | 0 | |  |

- Start up the OpenPLC Runtime server
 - **Note: The webpage UI end point that allows users to create their own virtual Modbus TCP slave device and monitor tags is only available for linux systems. To use it for Windows, Ubuntu will need to be installed.**
- In OpenPLC Editor, go to Devices → configuration and establish a connection to the OpenPLC Runtime server (IP address 172.29.77.60). This IP address where the virtual PLC resides.
- Enter the webpage using the base credentials (user: openplc | password: openplc).

- Create the virtual Modbus TCP slave device. Ensure it is configured as such:

| | |
|--|--|
| <p>Device Name</p> <div>VirtualPLC</div> <p>Device Type</p> <div>Generic Modbus TCP Device ▼</div> <p>Slave ID</p> <div>1</div> <p>IP Address</p> <div>172.29.77.60</div> <p>IP Port</p> <div>502</div> | <p>Discrete Inputs (%IX100.0)</p> <p>Start Address: <div>0</div> Size: <div>16</div></p> <p>Coils (%QX100.0)</p> <p>Start Address: <div>0</div> Size: <div>16</div></p> <p>Input Registers (%IW100)</p> <p>Start Address: <div>0</div> Size: <div>16</div></p> <p>Holding Registers - Read (%IW100)</p> <p>Start Address: <div>0</div> Size: <div>16</div></p> <p>Holding Registers - Write (%QW100)</p> <p>Start Address: <div>0</div> Size: <div>16</div></p> |
|--|--|

- In the Ignition gateway, navigate to OPC UA → Device Connections. There, create a Modbus TCP device named **OpenPLC_Sim**. Ensure the IP address and Port match what you put in for the virtual device in the OpenPLC Runtime.
- Pull the repo to your local machine and import the Ignition project as a zip file into your local ignition system.
- Note: To view the perspective session live in a browser, launch the perspective session and browse to: http://localhost:8088/data/perspective/client/Afolabi_Tank_Mixer_Simulation/tank_monitoring