OpenPLC in the Simulation Environment

This project integrates OpenPLC for simulating programmable logic controllers (PLCs) within the virtual environment. It leverages the repository OpenPLC-Docker-AutoStart to automate the import and configuration of PLCs using pre-defined `.ST` files during the build process.

Purpose and Functionality

The OpenPLC setup allows for the inclusion of multiple PLCs to control various zones within the simulation environment. Each PLC can be configured with a specific `.ST` (Structured Text) file, representing the logic for that zone. This provides flexibility in designing complex scenarios involving different operational zones.

Adding a PLC

To add a new PLC to the environment, you can modify the `docker-compose.yaml` file. Use the following syntax as a template:

```yaml
plc-zone*:
  container_name: plc-zone* # Replace * with the zone name (e.g., plc-zone1, plc-zone2)
  depends_on:
    - scadalts
  build:
    context: ./openplc/.
    dockerfile: Dockerfile
    args:
      script: watertanklogica.st # Replace with the .ST file the PLC should execute
      database: database.sh # Database script; do not modify
  ports:
    - "8082:8080" # Port mapping for external access and debugging; can be disabled if not required
  expose:
    - "502" # Modbus/TCP communication port
    - "8080" # WebUI access; can be removed if not needed
  networks:
    - plc_network
```

Ladder Logic for ScadaLTS Integration

This repository contains the ladder logic program designed to interact with ScadaLTS via holding registers. The program controls a pump's speed and interacts with EPANET for simulation purposes.

Download to the openplc editor to make the ladderlogic:

- Openplc-editor.

Overview

The ladder logic program uses holding registers to communicate with ScadaLTS. The `PumpSpeed` variable is mapped to `%QW1000`, which corresponds to the pump speed in EPANET.

Variables

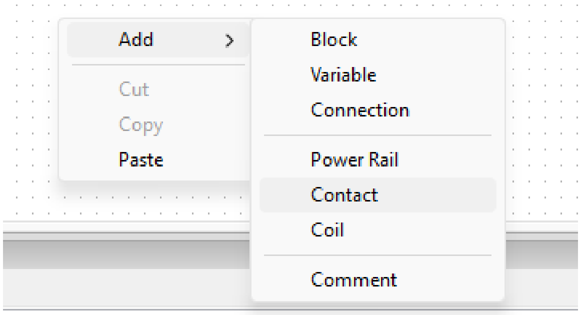
Below are the primary variables used in the ladder logic:

- **PumpSpeed**: Located at `%QW1000` in EPANET.
- **StartButton**: A functional button within ScadaLTS. Can be assigned between `%QX0.0 - %QX99.7`.
- **Reference**: For additional details on Modbus addressing, visit OpenPLC-Adressing.

Components

- Contacts: Represent inputs such as StartButton.

- Coils: Control outputs like pump activation.
- Blocks: Handle logic operations and data manipulation.
-



Logic Explanation

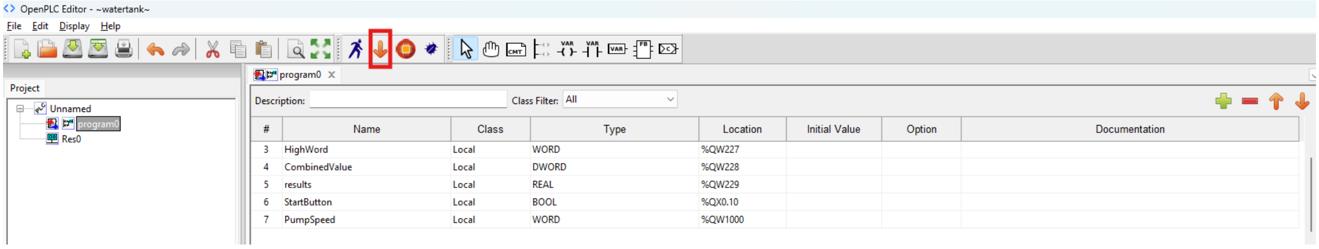
The ladder logic program performs the following tasks:

1. Reads the start button from ScadaLTS.
2. Writes the pump speed to `%QW1000` for EPANET simulation.
3. This turns the pump on and makes the water flow through.

Upload ST file to openplc program

1. Export the Ladder logic.

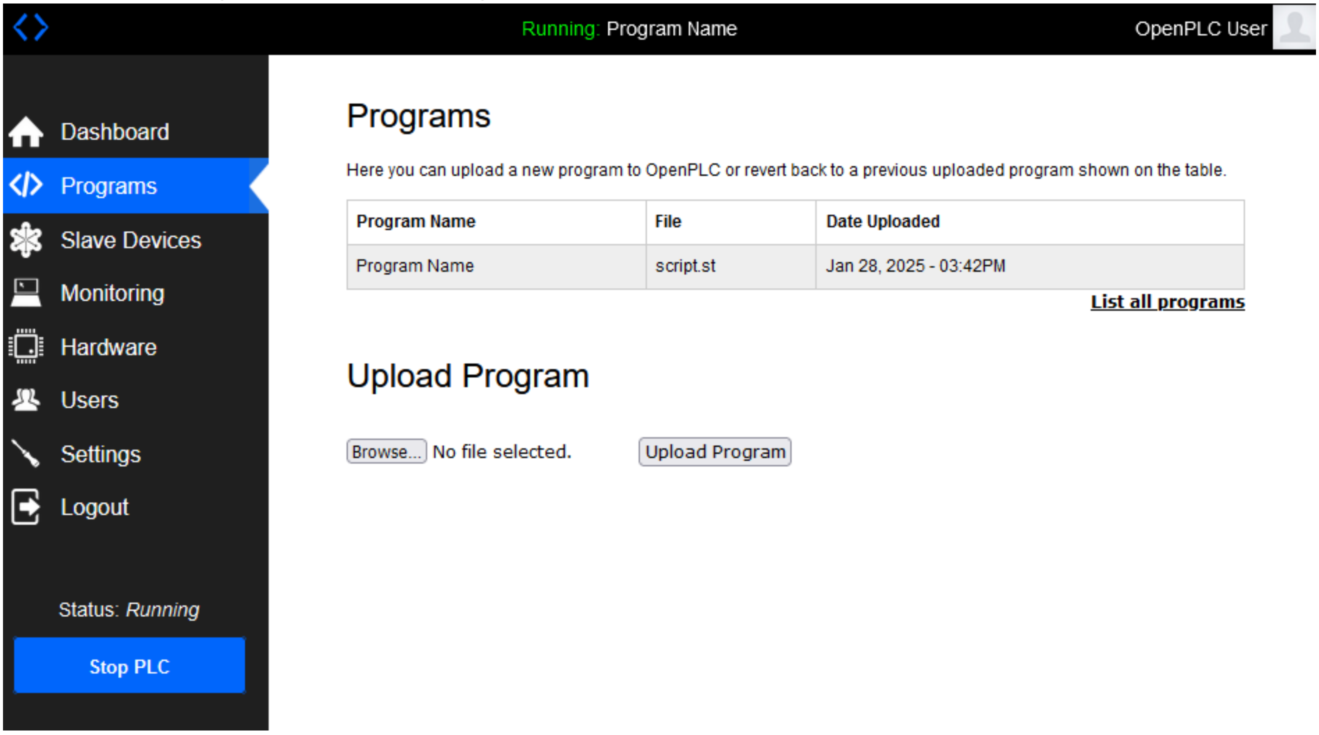
- Save it as an NameOfProgram.ST.



Description:
Class Filter: All

#	Name	Class	Type	Location	Initial Value	Option	Documentation
3	HighWord	Local	WORD	%QW227			
4	CombinedValue	Local	DWORD	%QW228			
5	results	Local	REAL	%QW229			
6	StartButton	Local	BOOL	%QX0.10			
7	PumpSpeed	Local	WORD	%QW1000			

2. Then open the running Openplc and upload Program.



Programs

Here you can upload a new program to OpenPLC or revert back to a previous uploaded program shown on the table.

Program Name	File	Date Uploaded
Program Name	script.st	Jan 28, 2025 - 03:42PM

List all programs

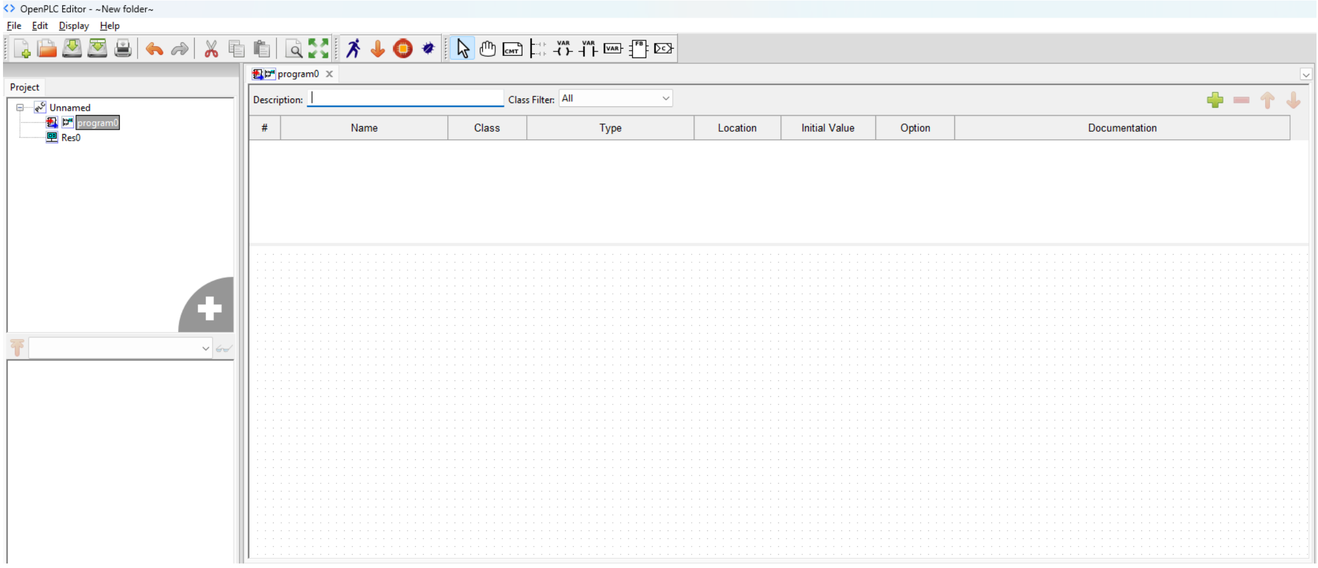
Upload Program

Browse... No file selected. Upload Program

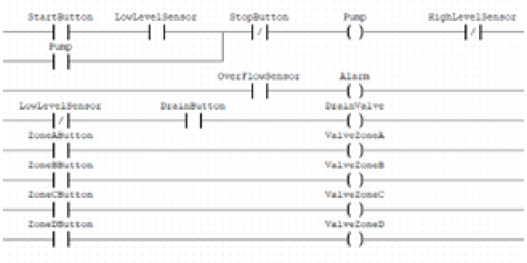
Screenshots

Below are screenshots of the ladder logic program and variable mappings:

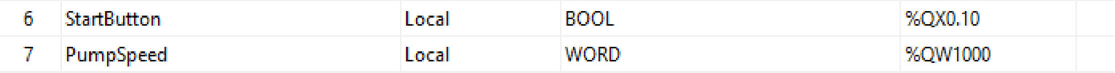
- **Ladder Logic Diagram**:
- empty:



#	Name	Class	Type	Location	Initial Value	Option	Documentation

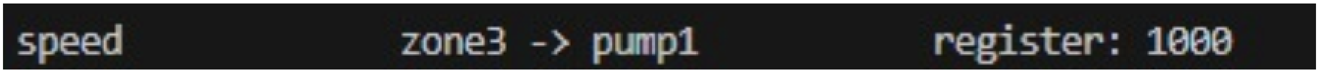


- With data:
- **Variable Mappings**:

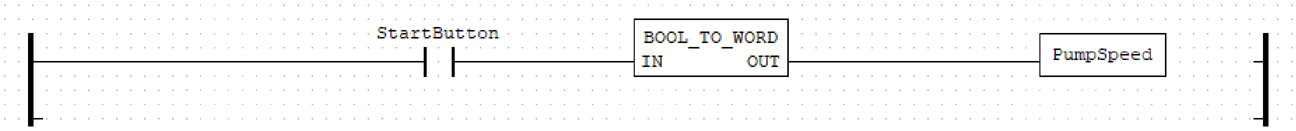


6	StartButton	Local	BOOL	%QX0.10	
7	PumpSpeed	Local	WORD	%QW1000	

- **EPANET Integration**:



```
speed                  zone3 -> pump1                  register: 1000
```


StartButton
BOOL_TO_WORD
IN
OUT
PumpSpeed