

ScadaLTS

ScadaLTS will serve as the **HMI (Human-Machine Interface)** for this simulation. It provides capabilities to manipulate the simulation and monitor the status of the water network in real time.

Automatic Configuration

The ScadaLTS environment is automatically configured during deployment using a `.zip` configuration file located at:
`/scadalts/config.zip`

The configuration is uploaded and applied using a mechanism described in the following GitHub discussion:
[SCADA-LTS Discussion #2463](#).

This process is integrated into the `docker-compose.yaml` file, ensuring seamless setup without manual intervention.

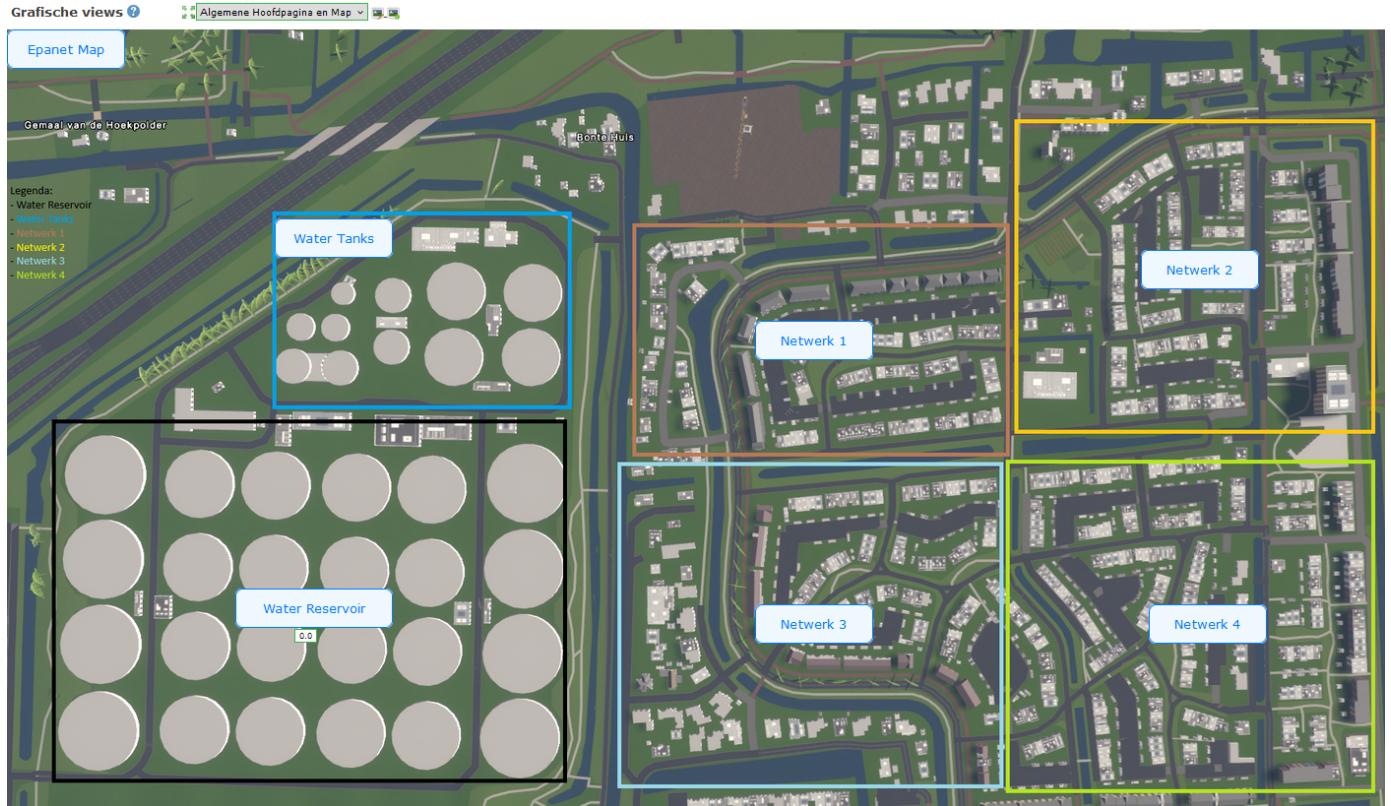
```
curl -d "username=admin&password=admin&submit=Login" -c cookies http://localhost:8080/ScadaBR/login.htm  
curl -b cookies -v -F importFile=/config.zip http://localhost:8080/ScadaBR/import_project.htm  
curl 'http://localhost:8080/ScadaBR/dwr/call/plaincall/EmportDwr.loadProject.dwr' -X POST -b cookies --  
data-raw  
$'callCount=1\ncallName=/ScadaBR/import_project.htm\nhttpSessionId=\nscriptSessionId=D15BC242A0E69D4251D5585A  
07806324697\nc0-scriptName=EmportDwr\nc0-methodName=loadProject\nc0-id=0\nbatchId=5\n'
```

HMI Water Network Overview

General Overview of the Water Network

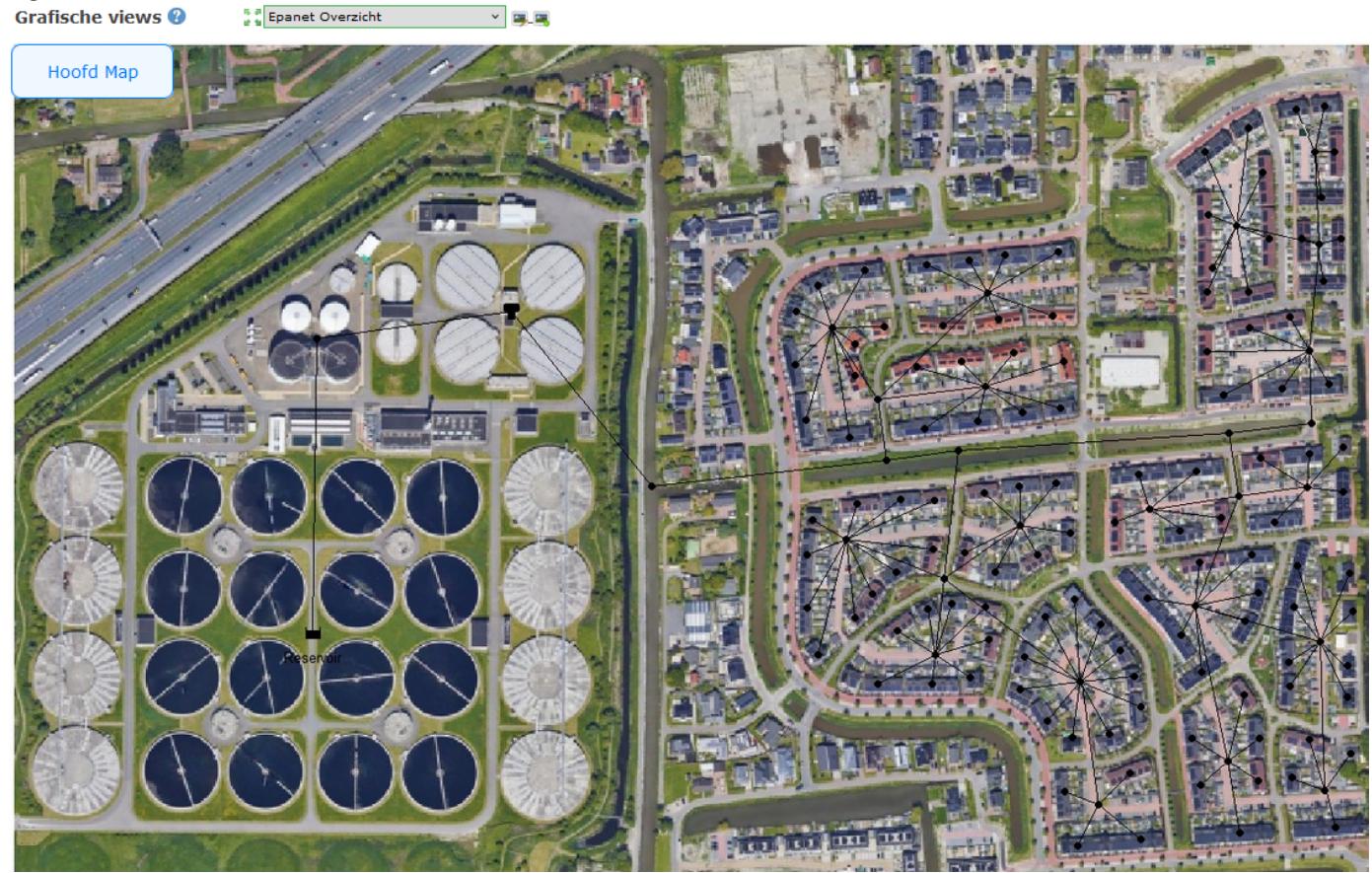
The main view of our HMI consists of the SCADA general overview of the water network. In this view, there are also nodes that link to the rest of the network and provide a general overview of how the network looks.

Figure 1: General Overview



Our water network was developed from the beginning in an **EPANET file**. In this file, the initial water network was designed, and all values in this document originate from it. It is important to note that both **Figure 1** and **Figure 2** are based on this EPANET file.

Figure 2: Overview of the EPANET Water Network



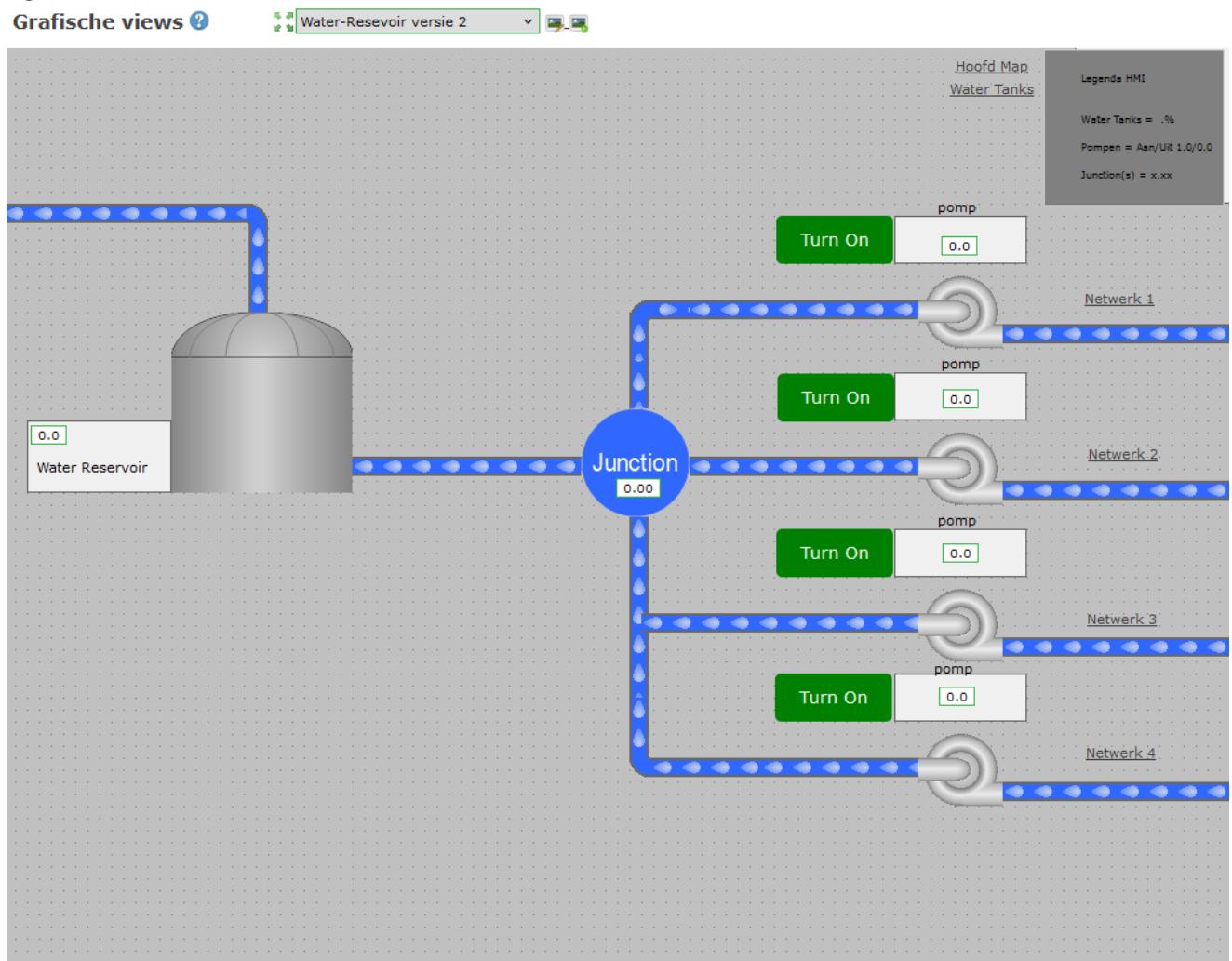
Network Structure

The network is divided into two sections:

1. Water Reservoir

Water is sent from the water reservoir—which we have decided, in this case, to be an infinite water supply—to the different networks.

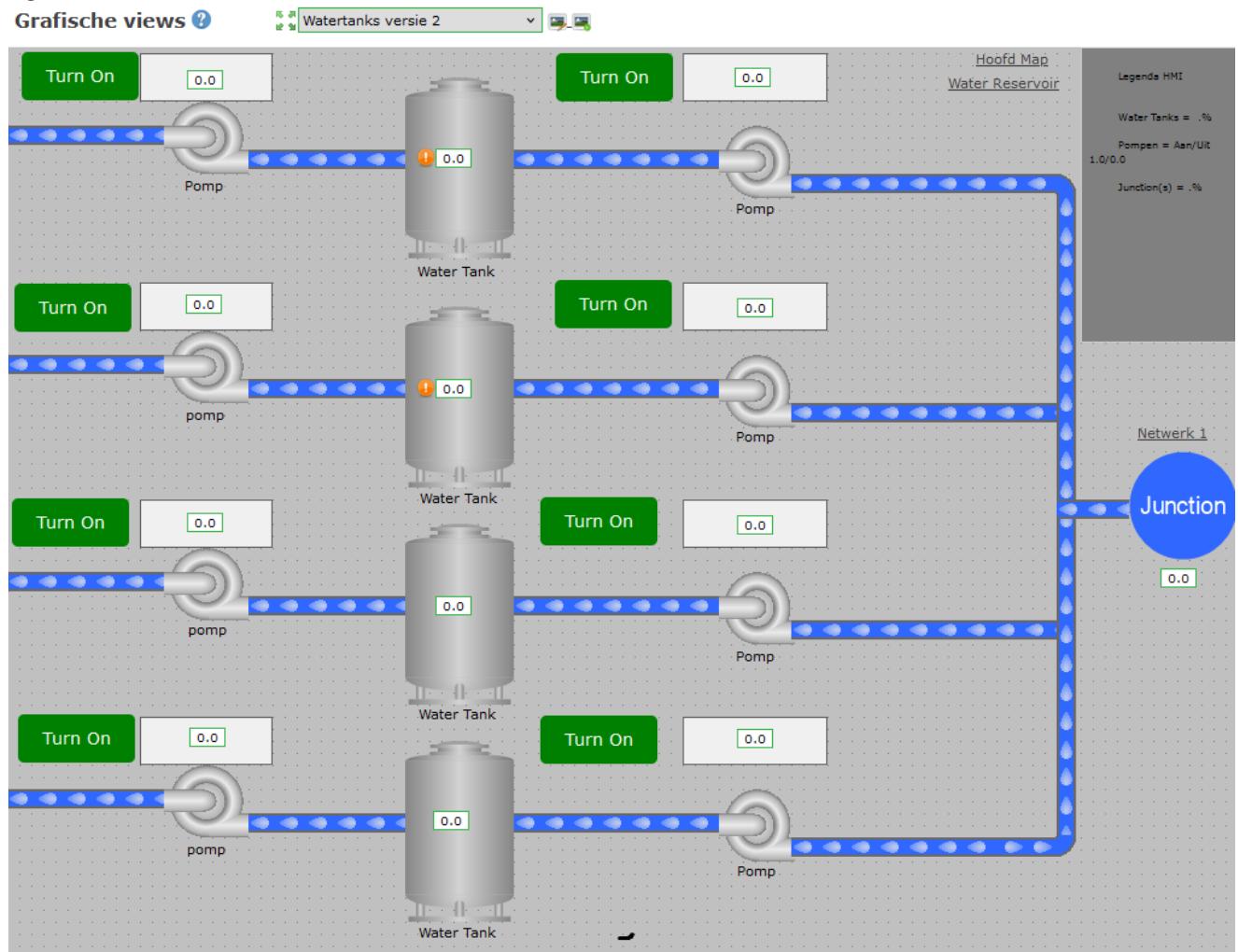
Figure 3: Overview of the Water Reservoir



2. Water Tanks

Each water network corresponds to different water tanks.

Figure 4: Overview of the Water Tanks



Networks Version 2

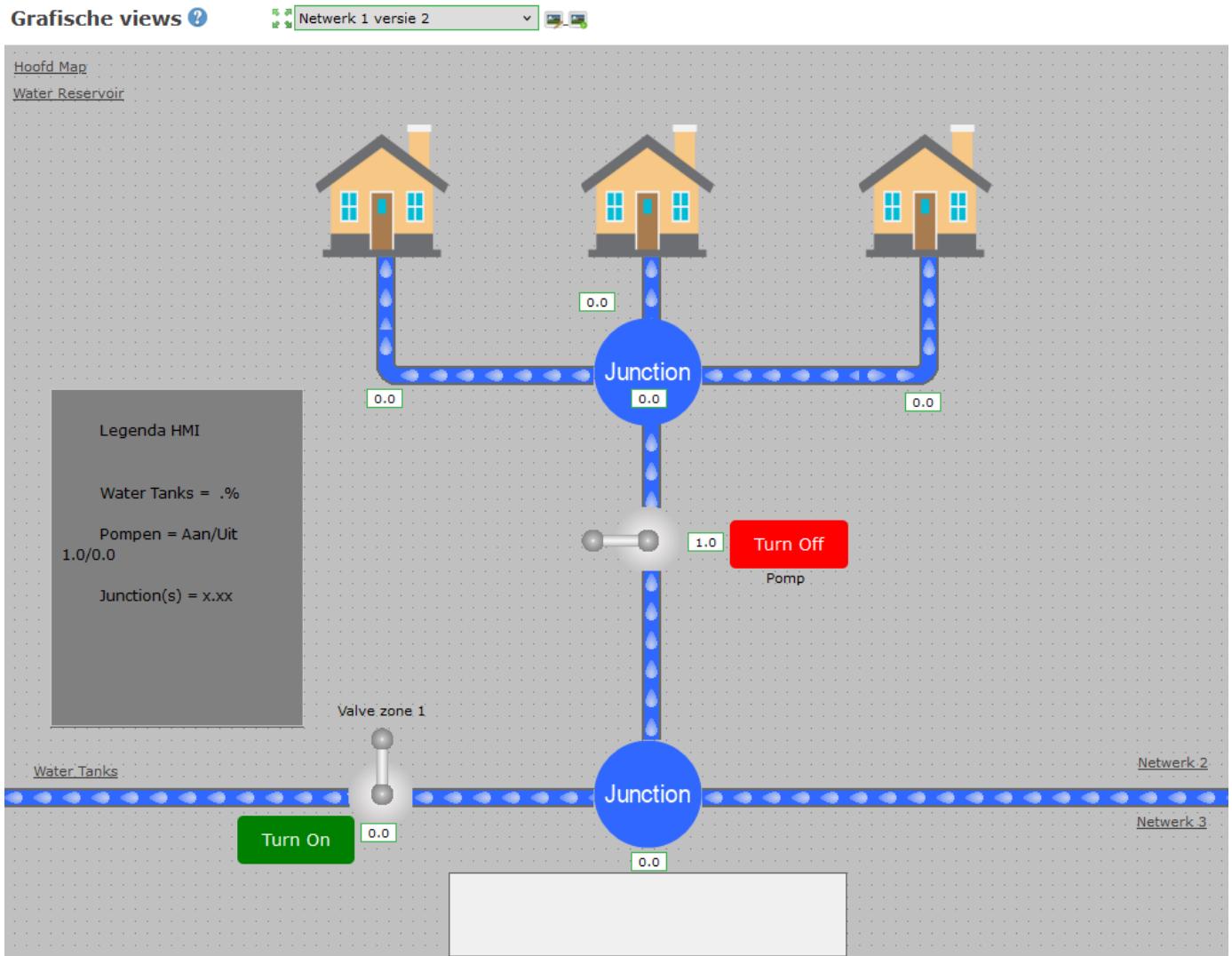
The network HMI designs were created based on the EPANET file we developed. This file contains a total of **four networks**. This version is referred to as **Version 2**, as it is a redesign compared to the HMIs from **Phase 3**.

In the new HMIs, **PMManger** was used as SCADA software, and more suitable designs were created based on it.

Note: All data points are set to 0 in all other network HMIs besides **Network 1** because, at the time of taking the screenshots, the EPANET file was not being read out anymore due to technical issues.

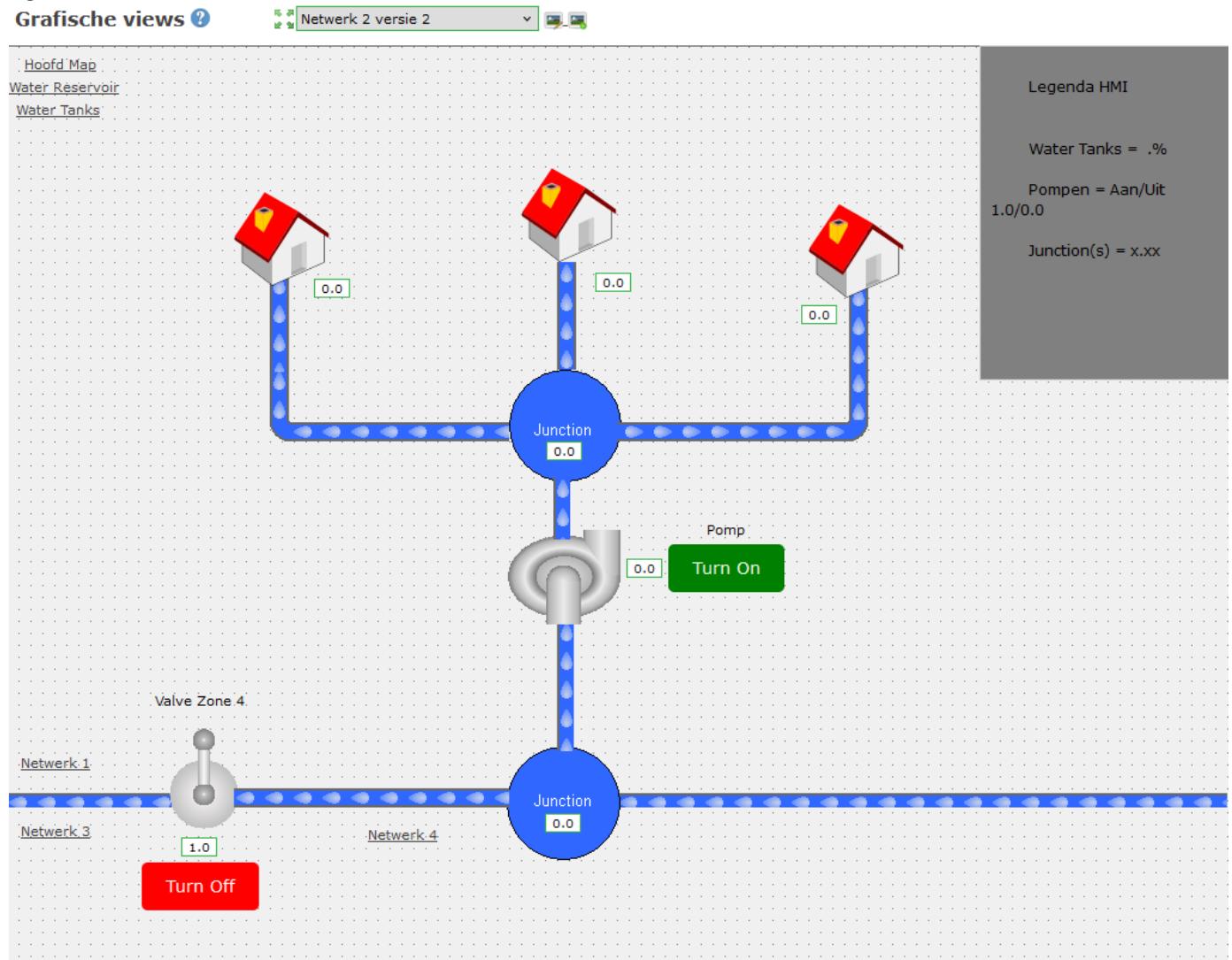
Network 1 Version 2

Figure 5: Network 1 Version 2



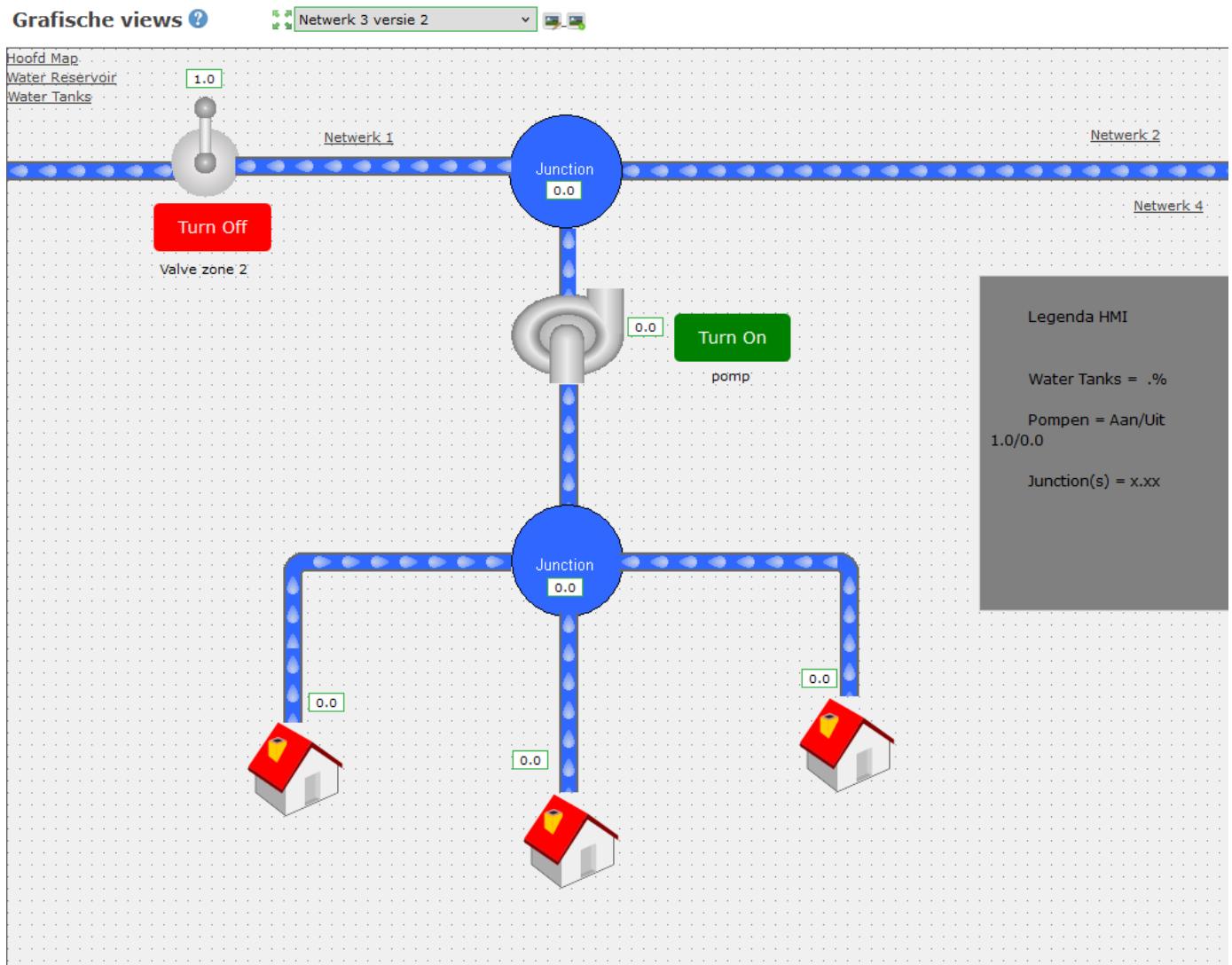
Network 2 Version 2

Figure 6: Network 2 Version 2



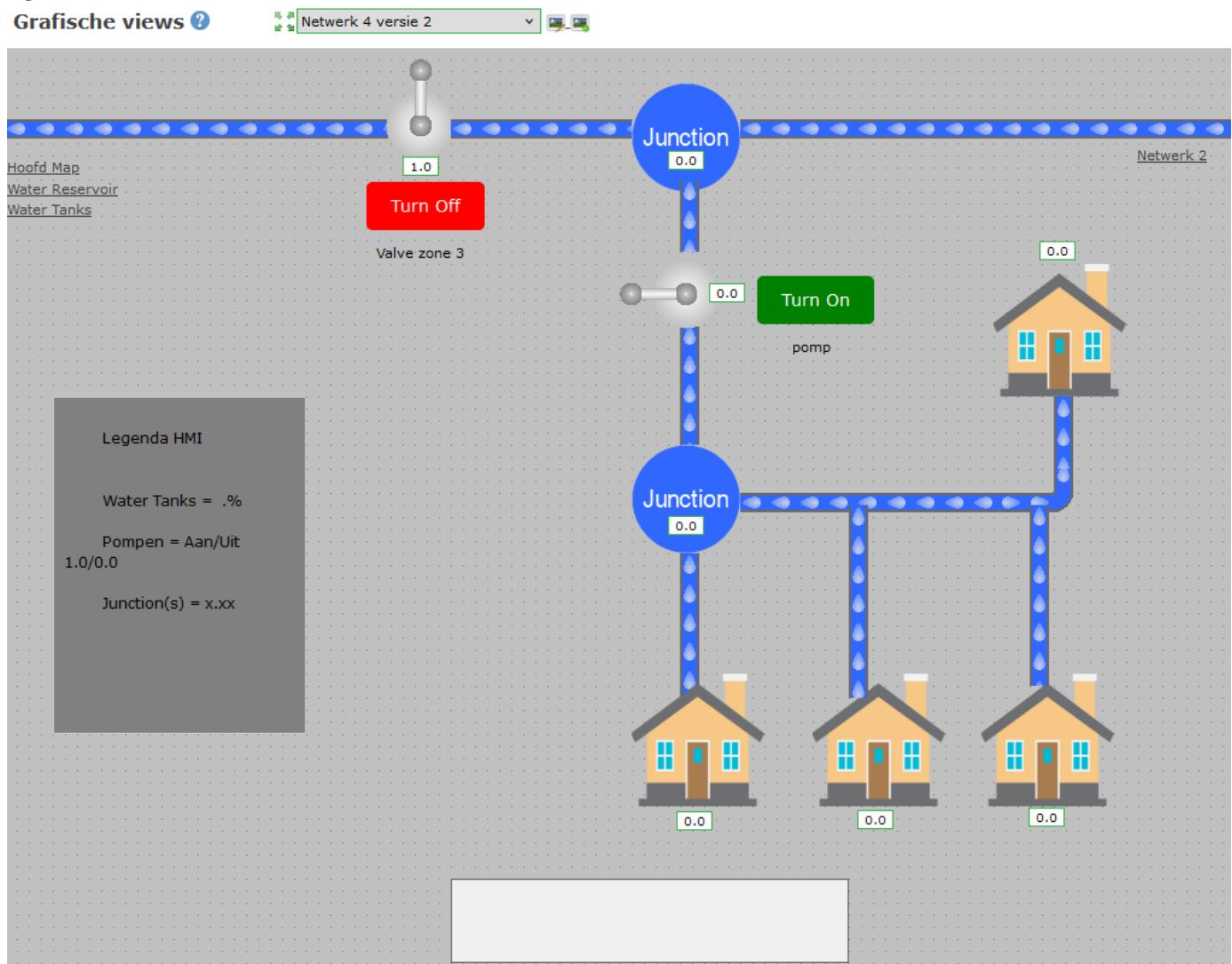
Network 3 Version 2

Figure 7: Network 3 Version 2



Network 4 Version 2

Figure 8: Network 4 Version 2



Creating the HMI Screens

The HMI screens that we have created during the 4th and final phase of our project were developed using the software called **PMMManager** from Promotic. This is specialized software designed for creating SCADA views for HMIs.

Link to PMMManager: [PMMManager Download](#)

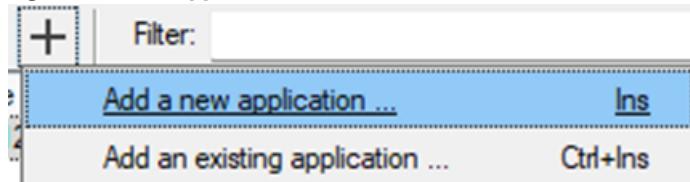
Figure 9: PMMManager Software



Creating a New Application

After downloading the software, start by creating a new application by clicking the "+" button.

Figure 10: New Application

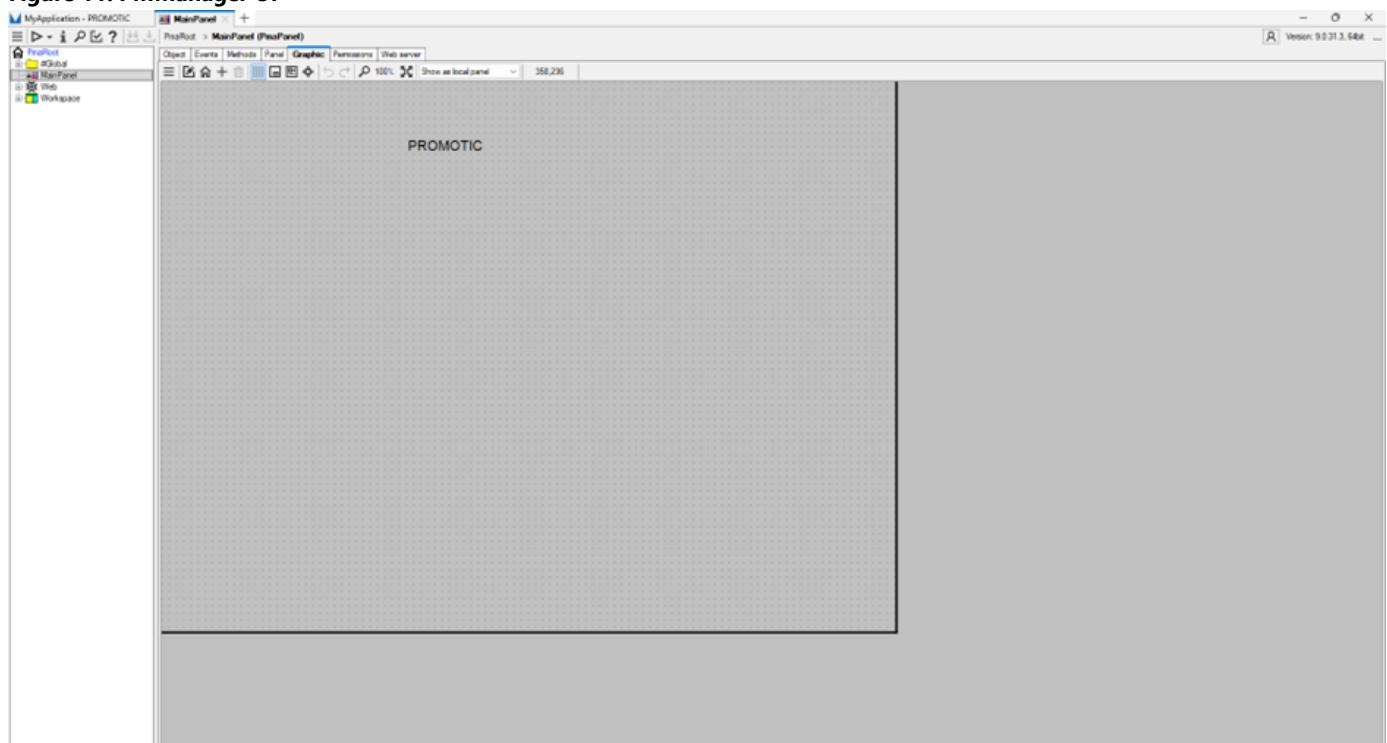


Once this is done, select "Next" on all options shown. After you've selected all options, you will be in your new application. Proceed by going to the **Main Panel Tab**.

Designing the HMI

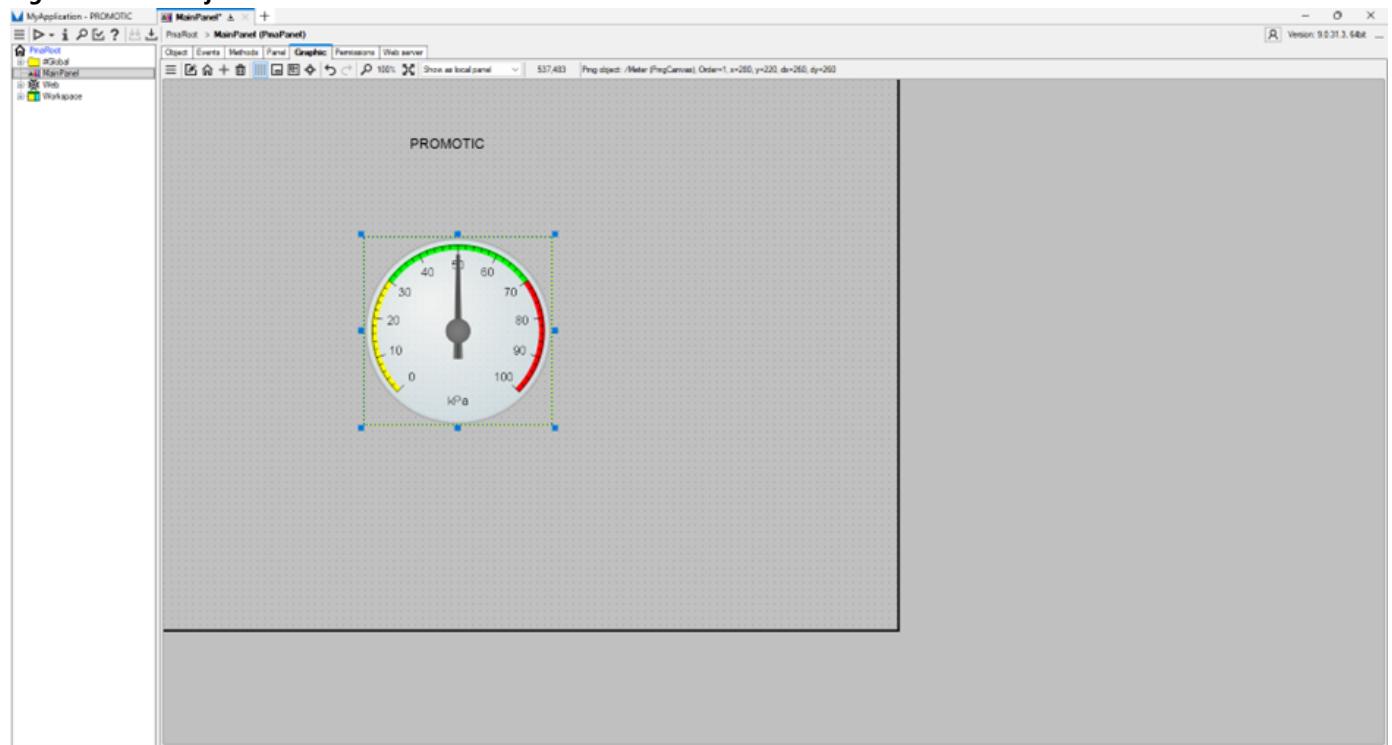
Here you will be able to add objects and design the HMI to your liking.

Figure 11: PMManager UI



Once you have the application open, the next step is to add "New object" to your HMI. This will allow you to start creating your HMI screens.

Figure 12: New Objects Added



Adding Data Resources to Your SCADA Project

Navigate to the **Data Resources** icon in ScadaLTS. Here, you will find a screen where you can add data resources.

Figure 13: Data Resources in ScadaLTS

A screenshot of the ScadaLTS application. The title bar says "SCADA-LTS Powered by Scada-LTS team v2.7.5.2 build 4551883606 (GitHub ref: 5cbf497); runs on Linux6.8.0-41-generic". The menu bar includes "File", "Edit", "View", "Panel", "Data", "Events", "Methods", "Panel", "Graphics", "Permissions", and "Help". The toolbar has icons for file operations, database, and other tools. A search bar at the top right contains the text "1-wire". The main area is titled "Gegevensbronnen" and displays a table of data resources. The table has columns: Naam, Type, Connectie, Status, and Status description. The data rows are: "plc-zone0" (Modbus IP, plc-zone0:502, green circle, "The user has changed the on/off status of datasource"), "plc-zone1" (Modbus IP, plc-zone1:502, green circle, "The user has changed the on/off status of datasource"), "plc-zone2" (Modbus IP, plc-zone2:502, green circle, "The user has changed the on/off status of datasource"), "plc-zone3" (Modbus IP, plc-zone3:502, green circle, "The user has changed the on/off status of datasource"), and "plc-zone4" (Modbus IP, plc-zone4:502, green circle, "The user has changed the on/off status of datasource"). Each row has a set of icons for edit, delete, and status monitoring.

These data resources are correlated with the **EPANET file** you have created or any other SCADA-readable file from which you want to take data.

Figure 14: Adding a New Data Resource

1-Wire eigenschappen

- Naam: [empty input]
- Export ID (XID): DS_451402
- Poort: [dropdown menu]
- Update periode: 5 minu(u)t(en)
- Geplande herscan: 1 Geen
- Gebeurtenis alarm niveaus**
 - Data bron exceptie: Urgent
 - Punt lees exceptie: Urgent
 - Punt schrijf exceptie: Urgent

1-wire Netwerk Scan

Scan uitvoeren
Specified adapter name "DS9097U" is not known

Fill in the values as shown in figure 15

Figure 15: Data Resource with Update Period of 1 Second

Modbus IP eigenschappen

- Naam: plc-zone0
- Export ID (XID): DS_054290
- Update periode: 1 second(en)
- Kwantiseer: [checkbox]
- Timeout (ms): 500
- Herhalingen: 2
- Alleen aaneengesloten batches: [checkbox]
- Creëer slave monitor punten: [checkbox]
- Max read bit count: 2000
- Max read register count: 125
- Max write register count: 120
- Transport type: TCP met keep-alive
- Host: plc-zone0
- Poort: 502
- Encapsuleerd: [checkbox]
- Create socket monitoring point: [checkbox]
- Gebeurtenis alarm niveaus**
 - Data bron exceptie: Urgent
 - Punt lees exceptie: Urgent
 - Punt schrijf exceptie: Urgent

Modbus node scan

Nodes scannen Afbreken
Nodes gevonden

Modbus read data

- Slave id: 1
- Register bereik: Coil status
- Offset: 0
- Number of registers: 100
- Read data

Point locator test

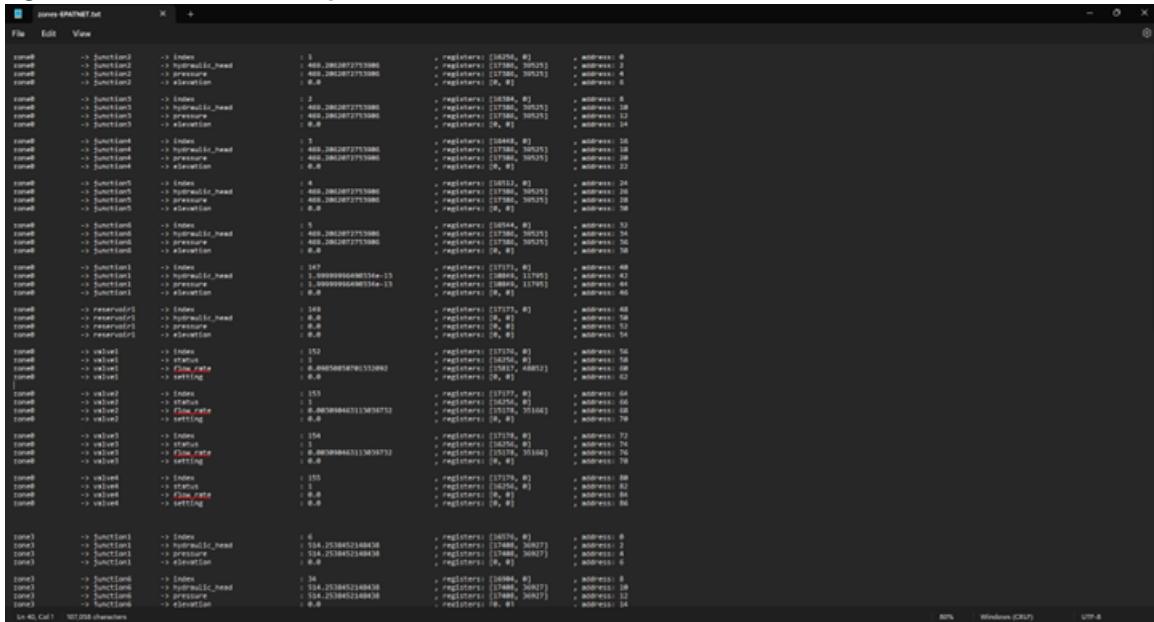
- Slave id: 1
- Register bereik: Coil status
- Modbus data type: Binary
- Offset: 0
- Bit: 0
- Number of registers: 0
- Character encoding: ASCII
- Read Add point

This resource continuously updates its information and is set to TCP with "Keep Alive." Once these two options are selected, the **Modbus Read Data** section becomes important. Once an offset is filled in, the data registers will be available to be read out.

Figure 16: Offset Data Being Read Out Starting from 1000

```
1000 ==> 0000
1001 ==> 0000
1002 ==> 3f80
1003 ==> 0000
1004 ==> 0000
```

Figure 17: EPANET Data File Exported



```
jones-EPANET.net x + File Edit View
node1
    > junction1    > Index : 1      . registers: [0400, #]   . address: 0
    > junction1    > hydraulic_head : 400_20010072753006 . registers: [07400, 00025] . address: 2
    > junction1    > pressure   : 400_20010072753006 . registers: [07380, 00025] . address: 4
    > junction1    > elevation  : 0.0      . registers: [0, #]    . address: 6
    > junction2    > Index : 2      . registers: [0400, #]   . address: 8
    > junction2    > hydraulic_head : 400_20010072753006 . registers: [07400, 00025] . address: 10
    > junction2    > pressure   : 400_20010072753006 . registers: [07380, 00025] . address: 12
    > junction2    > elevation  : 0.0      . registers: [0, #]    . address: 14
    > junction3    > Index : 3      . registers: [0400, #]   . address: 16
    > junction3    > hydraulic_head : 400_20010072753006 . registers: [07400, 00025] . address: 18
    > junction3    > pressure   : 400_20010072753006 . registers: [07380, 00025] . address: 20
    > junction3    > elevation  : 0.0      . registers: [0, #]    . address: 22
    > junction4    > Index : 4      . registers: [0400, #]   . address: 24
    > junction4    > hydraulic_head : 400_20010072753006 . registers: [07400, 00025] . address: 26
    > junction4    > pressure   : 400_20010072753006 . registers: [07380, 00025] . address: 28
    > junction4    > elevation  : 0.0      . registers: [0, #]    . address: 30
    > junction5    > Index : 5      . registers: [0400, #]   . address: 32
    > junction5    > hydraulic_head : 400_20010072753006 . registers: [07400, 00025] . address: 34
    > junction5    > pressure   : 400_20010072753006 . registers: [07380, 00025] . address: 36
    > junction5    > elevation  : 0.0      . registers: [0, #]    . address: 38
    > junction6    > Index : 6      . registers: [0400, #]   . address: 40
    > junction6    > hydraulic_head : 400_20010072753006 . registers: [07400, 00025] . address: 42
    > junction6    > pressure   : 400_20010072753006 . registers: [07380, 00025] . address: 44
    > junction6    > elevation  : 0.0      . registers: [0, #]    . address: 46
    > junction7    > Index : 147     . registers: [07370, #]   . address: 48
    > junction7    > hydraulic_head : 1.000000000000000e+13 . registers: [08840, 01000] . address: 50
    > junction7    > pressure   : 1.000000000000000e+13 . registers: [08860, 01000] . address: 52
    > junction7    > elevation  : 0.0      . registers: [0, #]    . address: 54
    > reserved1    > Index : 148     . registers: [07370, #]   . address: 56
    > reserved1    > hydraulic_head : 0.0      . registers: [0, #]    . address: 58
    > reserved1    > pressure   : 0.0      . registers: [0, #]    . address: 60
    > reserved1    > elevation  : 0.0      . registers: [0, #]    . address: 62
    > valued1      > Index : 151     . registers: [07370, #]   . address: 64
    > valued1      > status    : 1      . registers: [05000, 00000] . address: 66
    > valued1      > flow_rate : 4.000000000000000e+002 . registers: [05800, 00002] . address: 68
    > valued1      > setting   : 0.0      . registers: [0, #]    . address: 70
    > valued2      > Index : 152     . registers: [07370, #]   . address: 72
    > valued2      > status    : 1      . registers: [05000, 00000] . address: 74
    > valued2      > flow_rate : 4.000000000000000e+002 . registers: [05800, 00004] . address: 76
    > valued2      > setting   : 0.0      . registers: [0, #]    . address: 78
    > valued3      > Index : 154     . registers: [07370, #]   . address: 80
    > valued3      > status    : 1      . registers: [05000, 00000] . address: 82
    > valued3      > flow_rate : 4.000000000000000e+002 . registers: [05800, 00004] . address: 84
    > valued3      > setting   : 0.0      . registers: [0, #]    . address: 86
    zone1
        > junction1    > Index : 6      . registers: [0050, #]   . address: 0
        > junction1    > hydraulic_head : 154_2536451248038 . registers: [07400, 00027] . address: 2
        > junction1    > pressure   : 154_2536451248038 . registers: [07380, 00027] . address: 4
        > junction1    > elevation  : 0.0      . registers: [0, #]    . address: 6
    zone3
        > junction1    > Index : 34     . registers: [00000, #]  . address: 8
        > junction1    > hydraulic_head : 154_2536451248038 . registers: [07400, 00027] . address: 10
        > junction1    > pressure   : 154_2536451248038 . registers: [07380, 00027] . address: 12
        > junction1    > elevation  : 0.0      . registers: [0, #]    . address: 14
    
```

Here we can see the memory addresses associated with different sensors and values that we want to include in the HMI.

Customizing the HMI

Once your file is loaded and your data resources are configured, the only thing left to do is customize your HMI in **ScadaLTS**.

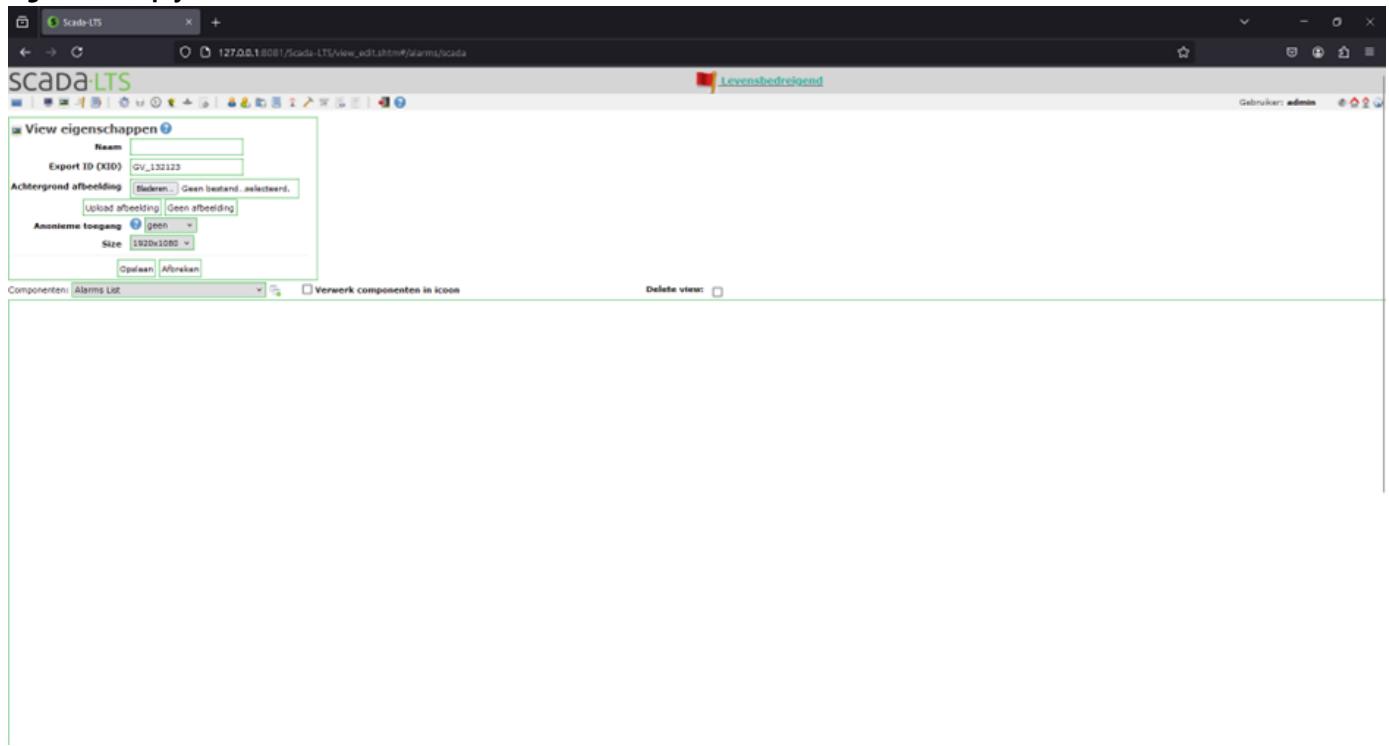
Customizing your HMI can be as simple or as complicated as you wish. A good starting point is to draw an HMI screen in software such as **PMMManager**. After you've completed your general HMI screen, you're ready to upload the image to your SCADA views. From here, you can add as many actuators and sensors as you like. Essentially, anything is possible.

Figure 18: Graphical View for the HMI Screens



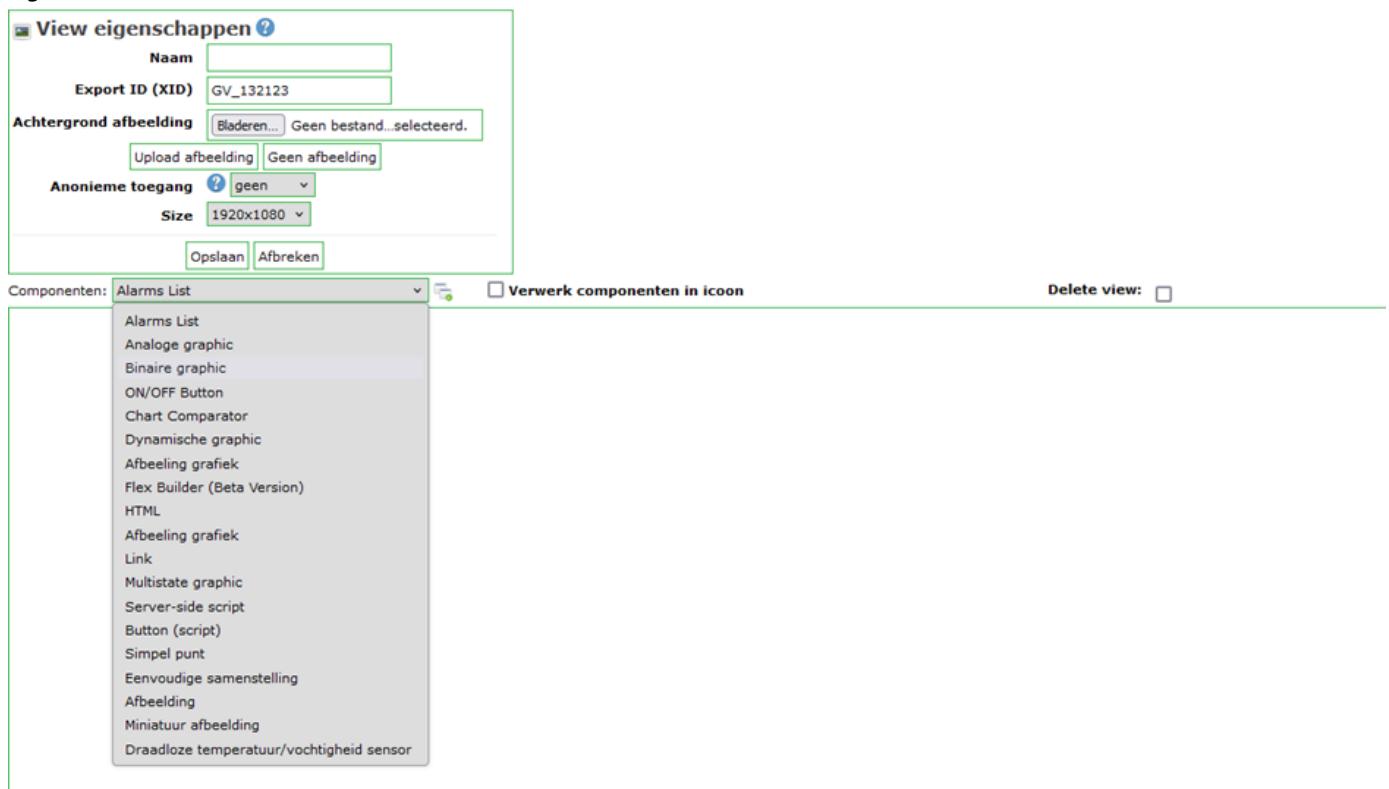
To create a new screen in the graphical view, click on the second icon to the right of the title of your HMI.

Figure 19: Empty HMI Screen



Now you can upload your HMI image and freely design and add whatever you need. In our project, we mostly used links to different network views and simple data points to read out values from our data resources. We also added buttons to control the valves (turn them on or off).

Figure 20: Final HMI Screen



Here you can upload your image and start adding data points.

End of the Guide

This concludes the guide for SCADA and how to create your own HMI.