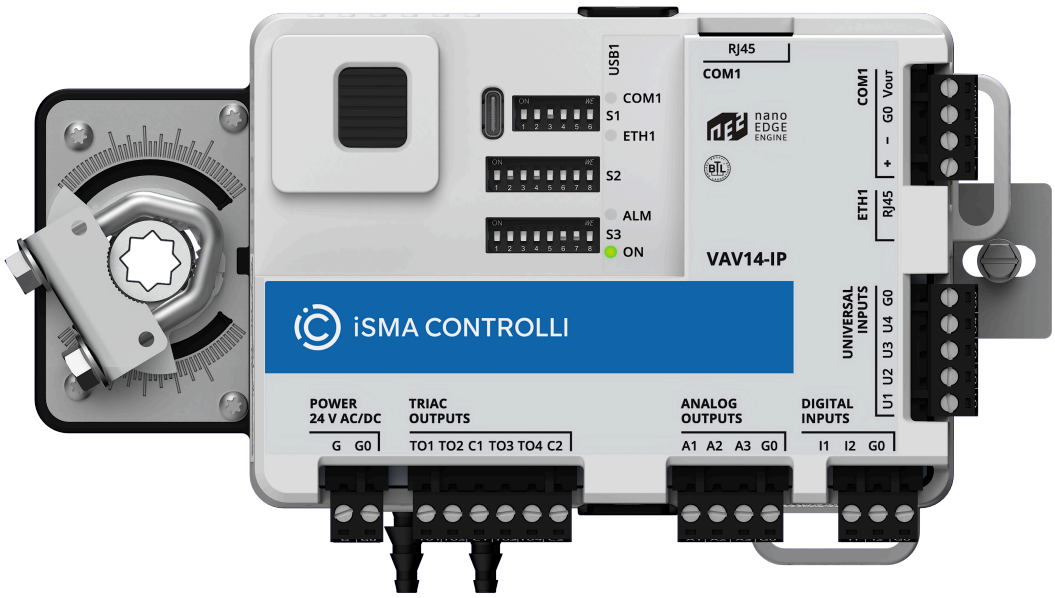
iSMA CONTROLLI

VAV14-IP

User Manual

Software



iSMA CONTROLLI S.p.A. - Via Carlo Levi 52, 16010 Sant'Olcese (GE) - Italy | support@ismacontrolli.com

Table of Contents

1 Introduction

The VAV14-IP is a configurable and freely programmable controller with BACnet IP, BACnet MS/TP, Modbus TCP/IP, and Modbus RTU protocols onboard. It is delivered with a built-in application, which supports the most popular types of VAV boxes. The possibility of creating a tailor-made application with the power and flexibility offered by the nano EDGE ENGINE makes the controller useful not only for typical VAV boxes but for all types, even the most advanced ones. The VAV14-IP controllers are developed on the nano EDGE ENGINE software platform, which offers cloud connectivity, real-time programming, and automatic exposure of Data Points. This enables a seamless integration with BMS. The platform supports remote control, real-time monitoring, and data analysis, enhancing energy consumption tracking, system performance, and maintenance needs.

1.1 Revision History

Date	Rev.	Description
18 Jun 2025	1.0.0	First edition
15 Jan 2025	Beta	Beta edition

2 VAV14-IP Controller

The VAV14-IP is a controller designed for energy-efficient airflow control in HVAC systems. It is equipped with a built-in configurable application and at the same time it can be freely programmable thanks to the versatility of its software platform, the nano EDGE ENGINE, which allows to adapt the controller's operation to the most advanced systems and equipment.

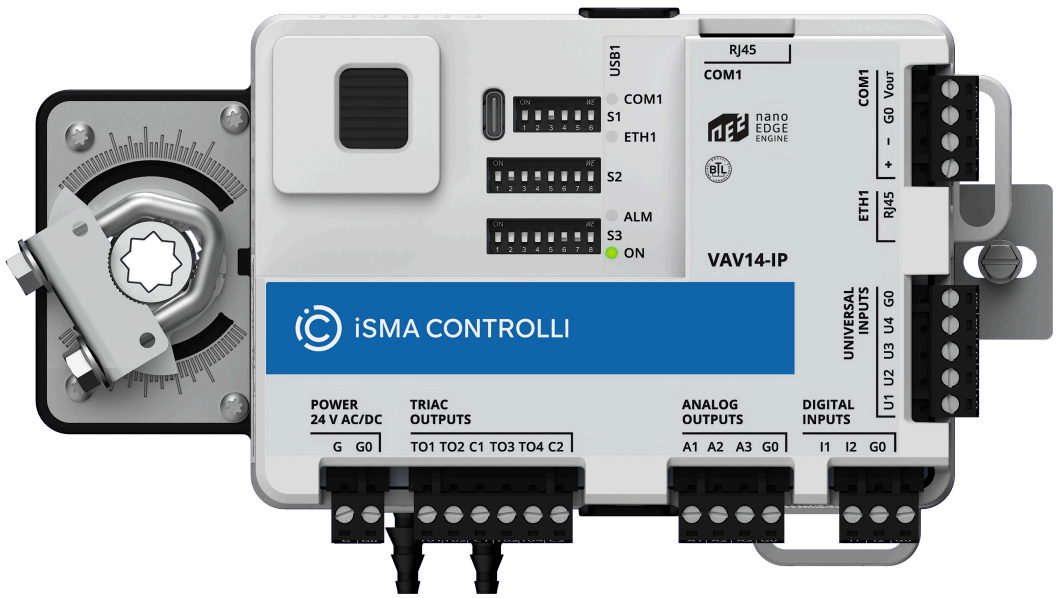


Figure 1. The VAV14-IP controller

The controller is equipped with:

- inputs and outputs (4 universal inputs, 2 digital inputs, 3 analog outputs, and 4 triac outputs);
- COM1 RS485 port (with RJ45 and screw connectors);
- ETH1 Ethernet port with 2 fail-safe protected switches (RJ45);
- USB1 port (USB type C for 5 V power supply and device connection in iC Tool);
- SD card for the nano EDGE ENGINE software platform;
- damper actuator powered from the controller;
- 3 DIP switches,
- LED diodes;
- 2 spigots.

2.1 Software Requirements

The VAV14-IP controller if developed for the nano EDGE ENGINE software platform. The controller requires at least the nano EDGE ENGINE OS V1.7.0 to operate; it has a built-in mechanism preventing from downgrading the OS version below V1.7.0.

2.2 Tools

The VAV14-IP controller can act either as a configurable device, which operation is based on the default pre-loaded application or as a freely-programmable device, which can be programmed to operate on most advanced VAV installations. For both these ends, there are relevant tools designed to configure or program the device.

Tools that enable configuration of the default application are:

- S3 DIP switch;
- iSMA Configurator;
- Control Point VAV panel;

Warning!

Before using the Control Point VAV panel with the VAV14-IP controller, make sure to upgrade its firmware to version V2.5.
Instructions how to upgrade firmware: iSMA Configurator.

- iC Device Manager (Niagara module);
- iC Tool (using components from the dedicated libraries)
- BACnet objects/Modbus registers (using dedicated objects/registers).

2.3 Configuration of the VAV Application

To configure the default VAV application, choose a most suited tool and follow the instructions contained in this manual (see the table below).

Tool	App configuration	Flow settings
S3 DIP switch	DIP Switches	-
iSMA Configurator	App configuration	Flow settings
Control Point VAV	App configuration	Flow settings
iC Device Manager	App configuration	Flow settings
iC Tool	nano EDGE ENGINE libraries for VAV application	
BACnet objects/Modbus registers	BACnet objects/Modbus registers	

Warning!

Please keep in mind that the setting of the first switch on the S3 DIP switch is vital for the configuration possibilities:

- switch 1 on the S3 DIP switch set to **ON**: configuration of the fan, reheater, and perimeter types (and referring parameters) possible **only** from the S3 DIP switch,
- switch 1 on the S3 DIP switch set to **OFF**: configuration of the fan, reheater, and perimeter types (and referring parameters) possible from the other tools: Control Point VAV/BACnet/Modbus/iC Configurator/iC Device Manager.

The S3 DIP switch setting holds priority over other tools. If it is set to ON, no change made to the mentioned parameters in any other tool (Control Point VAV or software) will be written to the controller!

2.4 Balancing

VAV (Variable Air Volume) balancing is the process of ensuring that a building's HVAC system delivers the right amount of air to different spaces. **The idea of balancing is to find the most optimal operating point of the ventilation system**. Proper balancing provides:

- energy efficiency: minimum consumption of electricity, heating and cooling;
- user comfort: temperature comfort, noise, air quality.

Read more about the process: Balancing.

2.4.1 Balancing Tools

Tools that enable carrying out balancing actions are:

- iSMA Configurator;
- Control Point VAV panel;
- iC Device Manager (Niagara module).

2.5 Freely-programmable Controller

Tools that enable free programming of a user application are:

- iC Tool (using components from the dedicated libraries);
- Workbench (using the Niagara nE2 Link module).

3 Quick Start-up

- First Connection
- Setting IP Address
- Commissioning

3.1 First Connection

3.1.1 Power Supply

A first step of the VAV14-IP commissioning is connecting the power supply.

The VAV14-IP controller requires 24 V AC/DC power supply.

It is also possible to power the controller using a USB cable (5 V power supply).

More

Find out more: VAV14-IP Hardware.

3.1.2 Communication and Device Configuration

To enable the device configuration and BACnet/Modbus communication management, connect the VAV14-IP controller to the PC using:

- RJ45 cable (ETH1 or COM1 ports),
- USB-C cable.

More

Find out more about device configuration options in the nano EDGE ENGINE: System.
Find out more about networks configuration in the nano EDGE ENGINE: quick start-up of LocalIO, quick start-up of BACnet, quick start-up of Modbus.
Find out more about the USB connection: USB connection.

3.2 Setting IP Address

When the VAV14-IP device is connected to the IP network, open the iC Tool and go to the IP Manager.

3.2.1 IP Manager

The IP Manager view is a convenient tool for discovering devices connected to the IP network and configuring their network settings. It facilitates a first connection to a device, providing a simple way to set IP addresses on discovered devices.

Important

For a device to get discovered by the IP Manager, the **IP Manager Discover Enabled** slot in the Ethernet1 component (Platform/System container) on the device must be set to true, which is its default value. To prevent the device from being discovered by unwanted agents, the slot should be set to false.

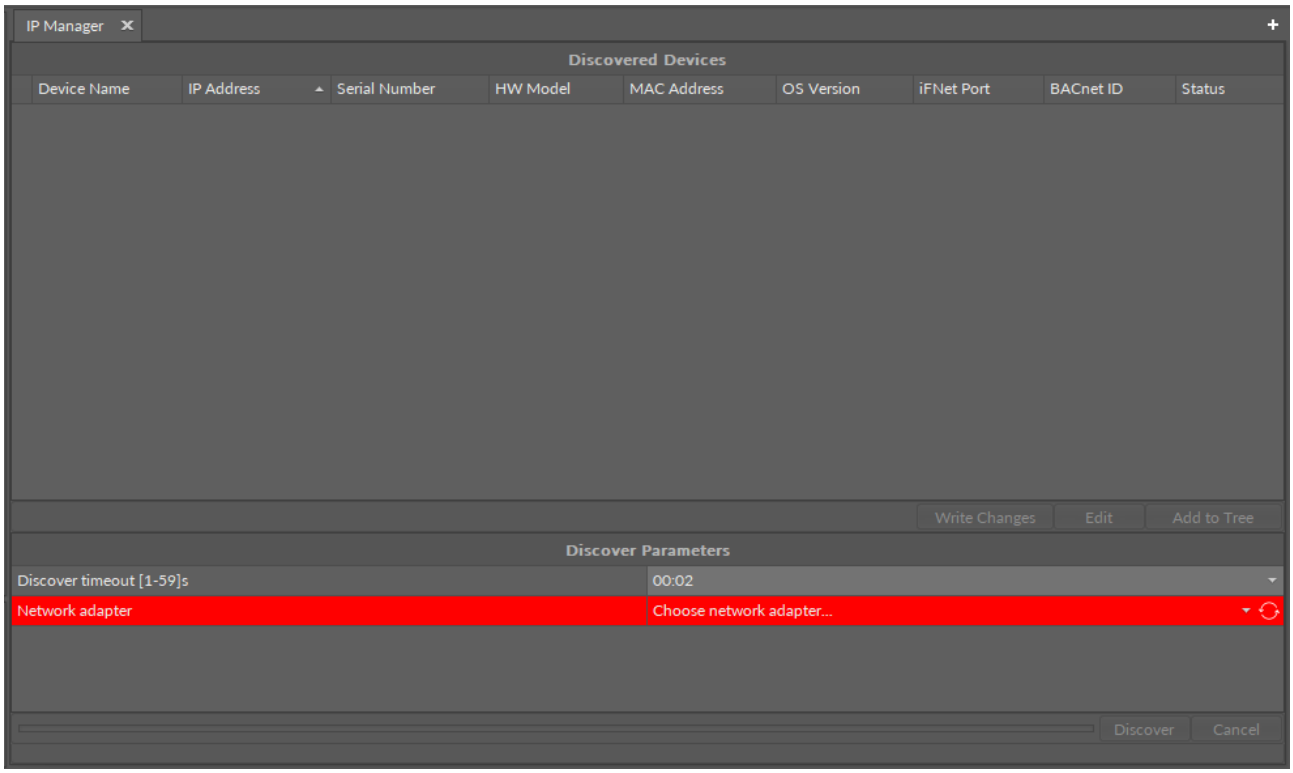


Figure 2. IP Manager

The IP Manager can be accessed from the main menu of iC Tool, by a dedicated button on the top bar, or in the context menu of the Site.

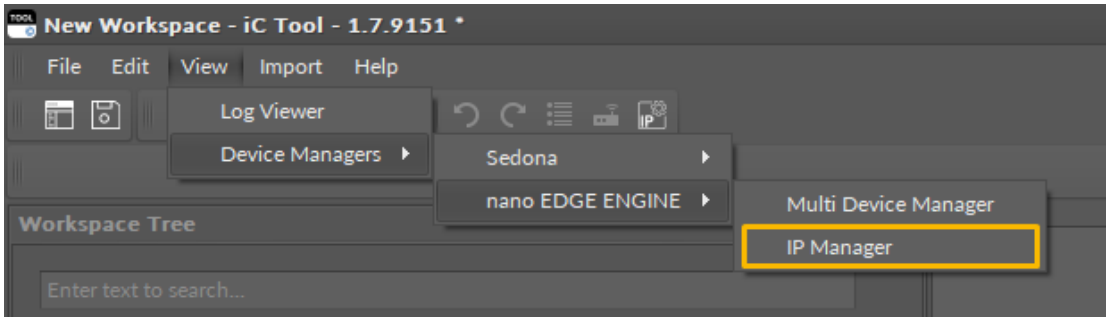


Figure 3. Accessing the IP Manager from the main menu

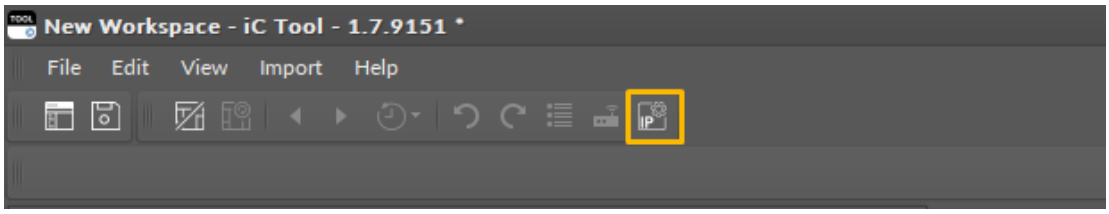


Figure 4. Accessing the IP Manager from the top bar

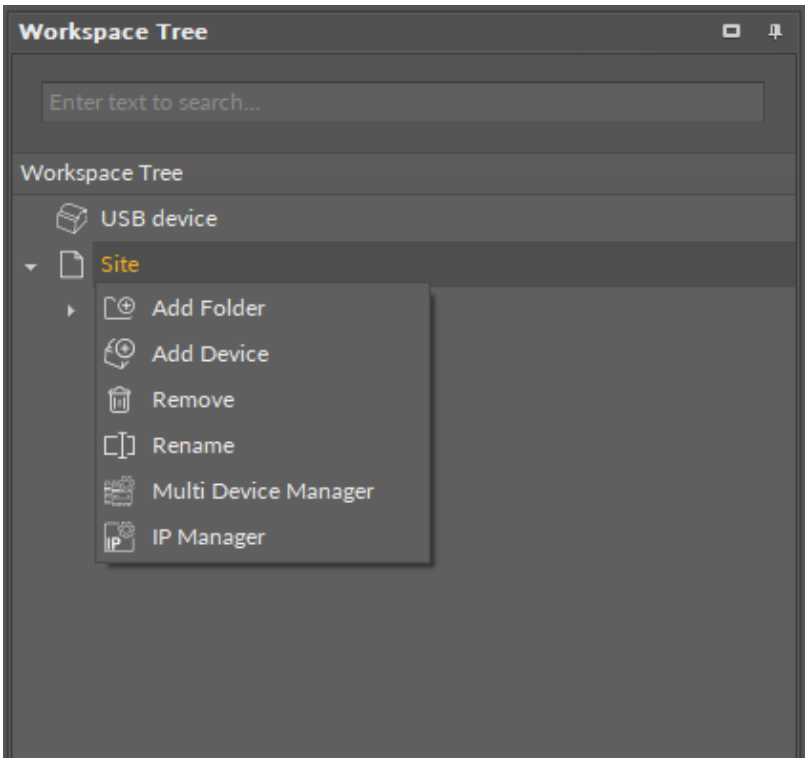


Figure 5. Accessing the IP Manager from the context menu of the Site

The IP Manager view is divided in two sections: Discovered Devices and Discover Parameters.

The Discovered Devices section shows devices, which are on the IP network. The section presents the following information about the device:

- device name,
- IP address of the device,
- serial number,
- HW model,
- MAC address,
- OS version;
- iFnet port,
- BACnet ID,
- status of the device:
    - available statuses:
        - Modified – applicable when any change has been introduced by the user in the edit view (the device is then highlighted in blue),
        - Written – set after the Write Changes command has been invoked and there is a positive writing confirmation from the device,
        - Error – set after the Write Changes command has been invoked and there is a negative response from the device or there is no response from the device (the device is then highlighted in red).

The second section, Discover Parameters, allows to select the IP network adapter and launch the discover process.

Note

If the device has already been added to the Workspace Tree and is still available on the network on the launch of the Discover process, it will be listed in the Discovered Devices section. Such device will be grayed out and it will not be possible to edit its parameters or add it again to the tree.

3.2.2 Using IP Manager

First, go to the Discover Parameters section and expand the list with available network adapters.

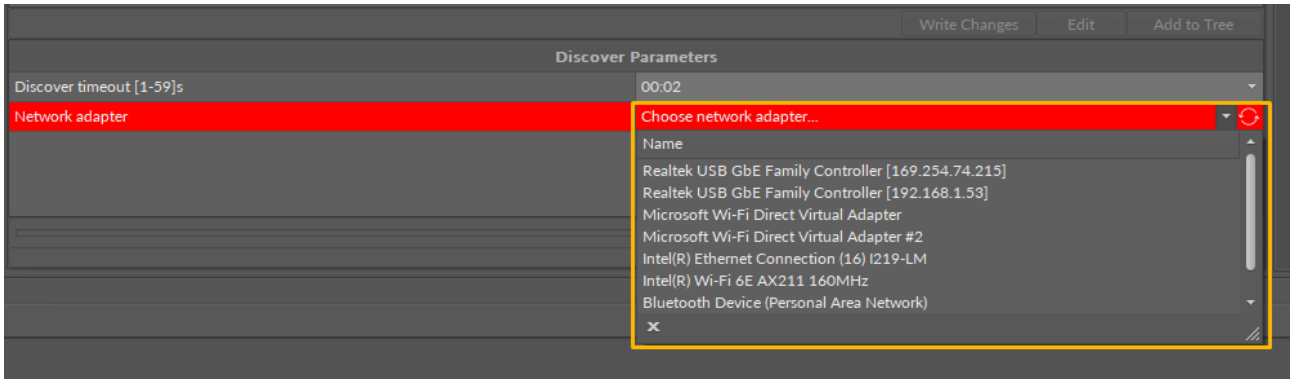


Figure 6. List of available network adapters

Select the required one and confirm with the Discover button.

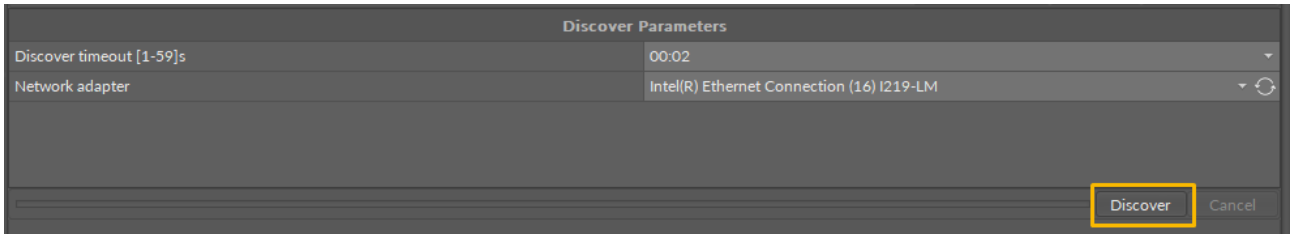


Figure 7. Discover button

The discovered devices are listed in the top section.

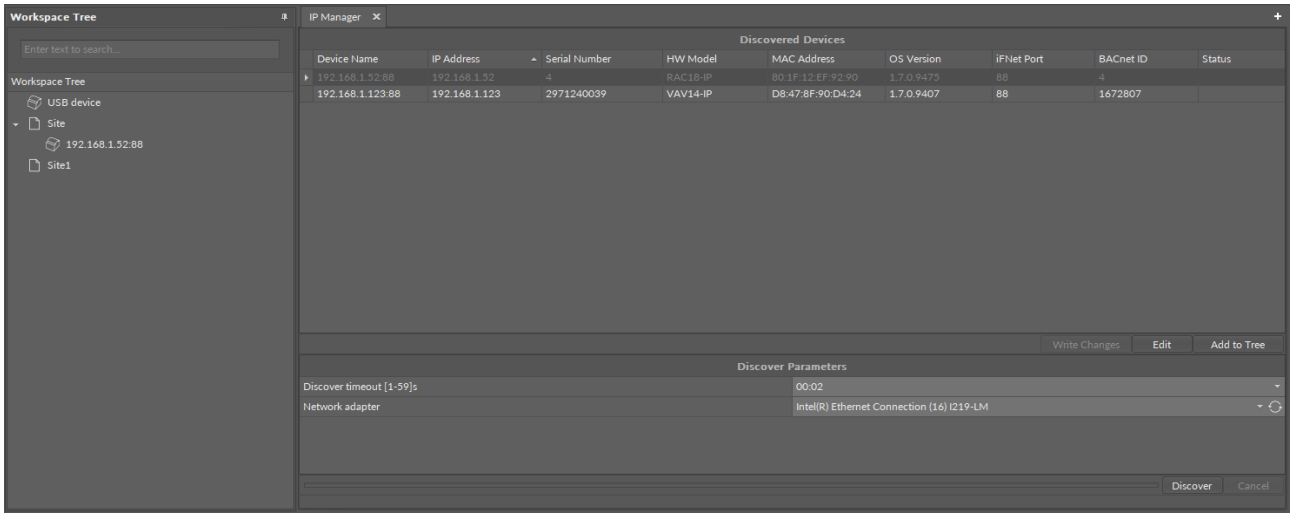


Figure 8. Discovered devices

Devices, which are on the network but have already been added to the Workspace Tree, are grayed out and their settings may only be changed in the System container when the device is connected.

For new devices, there are two main actions, Edit and Add to Tree.

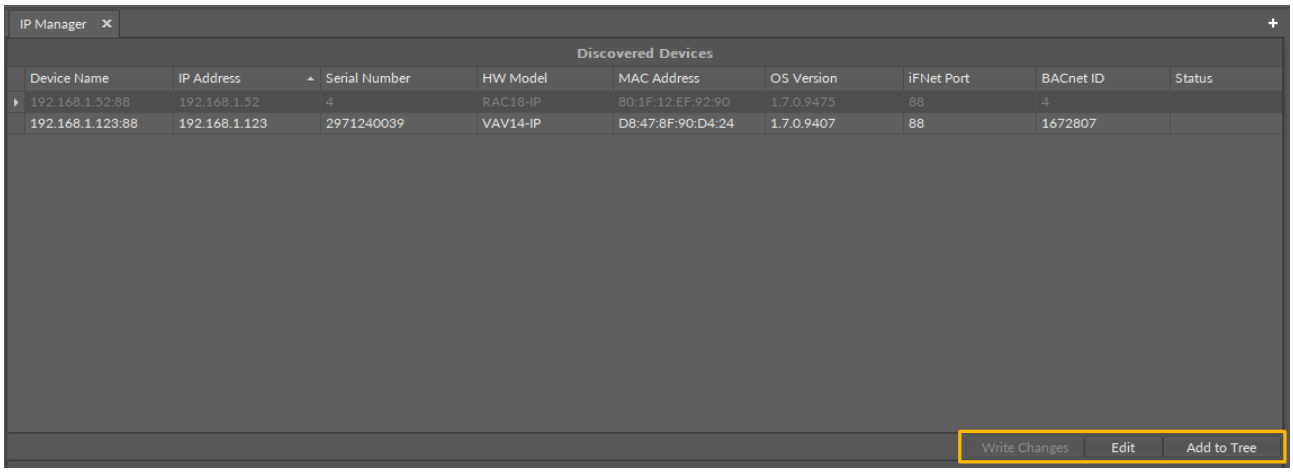


Figure 9. Option for new discovered devices

- **Edit (single device):** allows to configure network settings for the device:
  - DHCP mode,
  - IP address,
  - subnet mask,
  - gateway address,
  - DNS mode,
  - DNS server 1,
  - DNS server 2,
  - iFnet port;

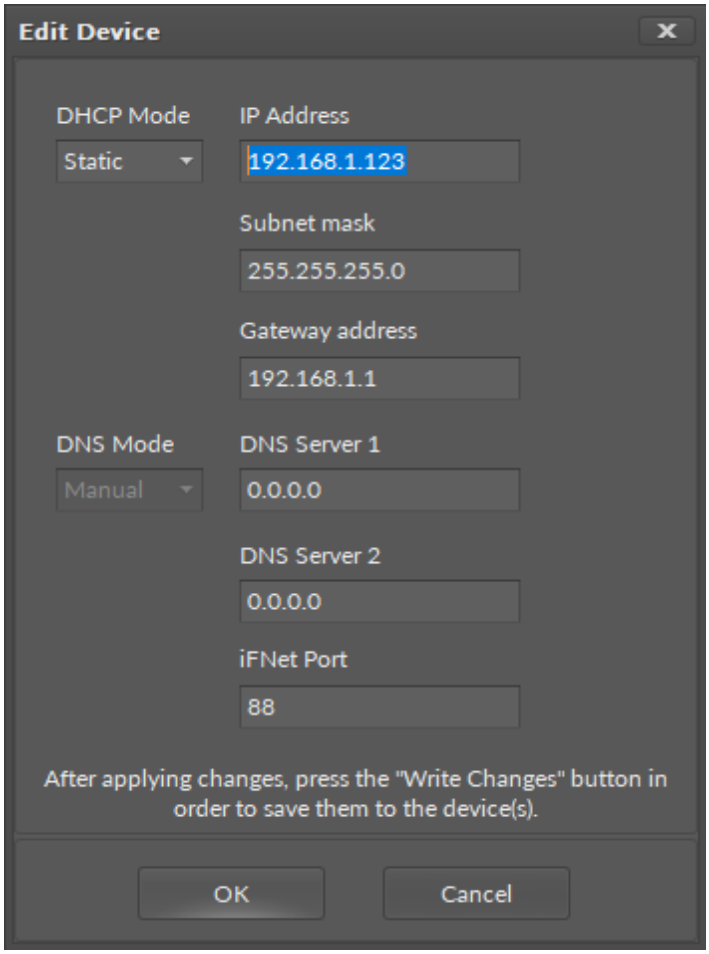


Figure 10. Editing options for a single device

- **Edit (more than one device):** allows to configure network settings for the device:
  - DHCP mode,
  - Base IP address: IP address of the first discovered device,

- Increment: a number by which IP addresses of the second and next devices will be assigned to based on the IP address of the first device,
- subnet mask,
- gateway address,
- DNS mode,
- DNS server 1,
- DNS server 2,
- iFnet port.

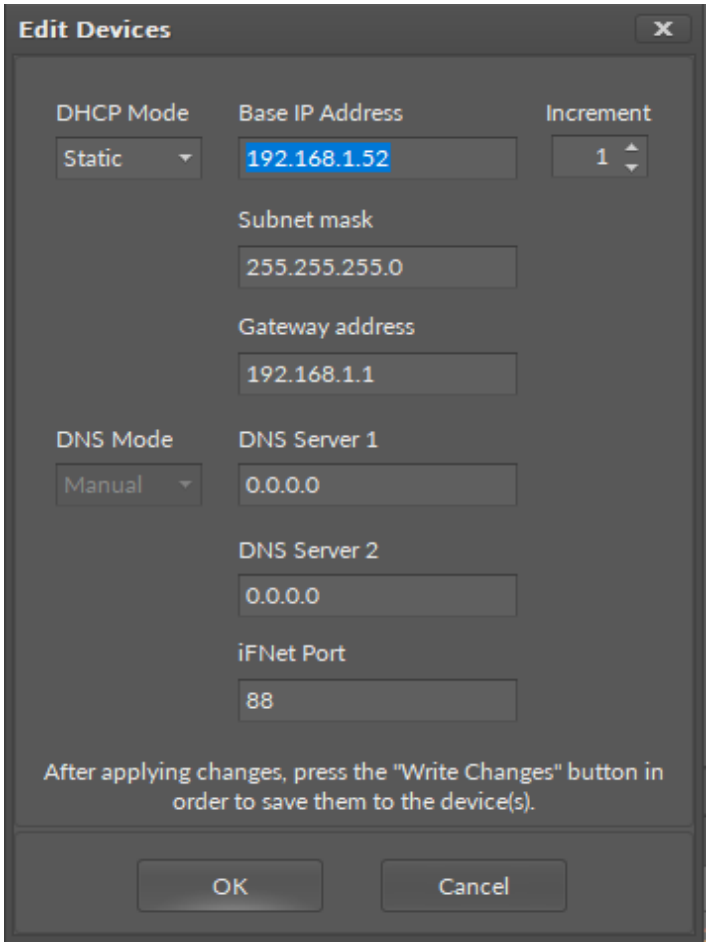


Figure 11. Editing option for more than one device

Warning!

Please make sure to confirm the edited values with the **Write Changes** button, which will send new values to the device. Please remember that any change made to the device requires a restart and the following alert pops up:

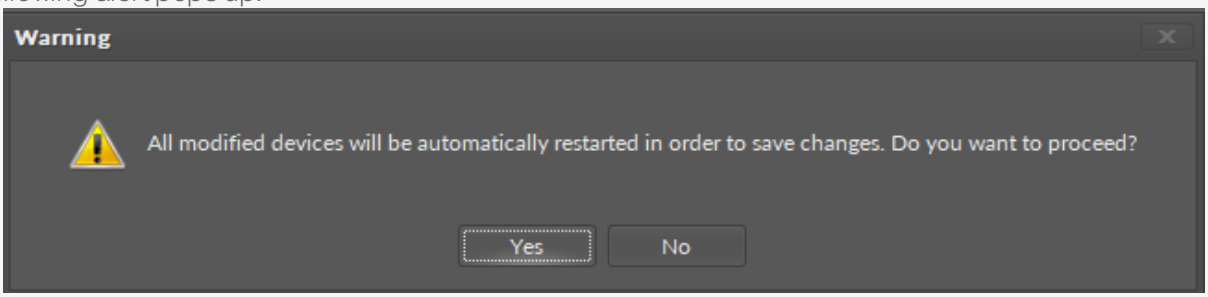


Figure 12. Restart alert

Please note that if any parameter has been changed, without the Write Changes action, it is impossible to add the devices to the tree.

- **Add to Tree:** allows to add the device to the Workspace Tree in a selected Site.

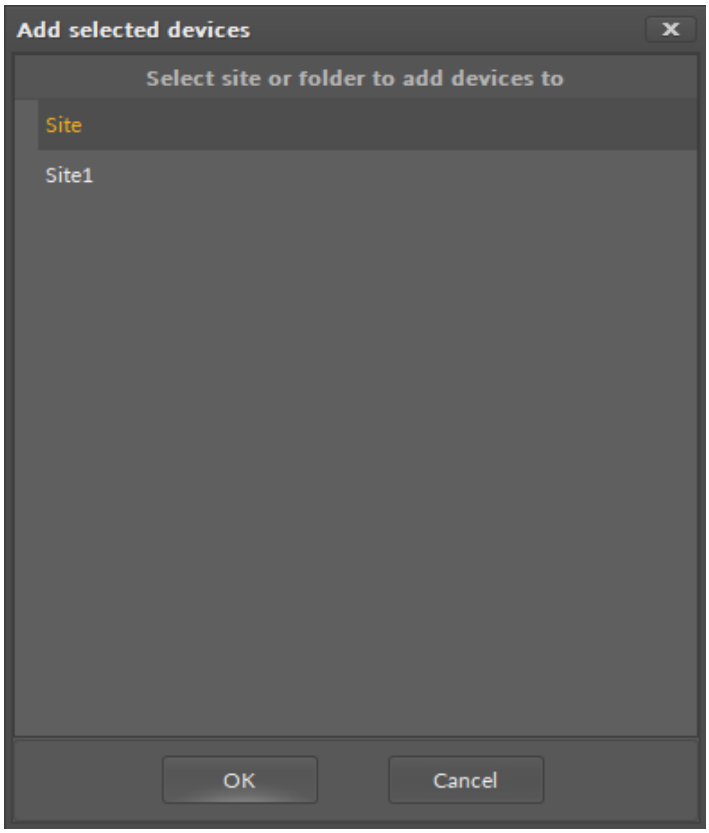


Figure 13. Add device dialog window

Note

It is also possible to add the device to the selected Site by drag-and-drop. This option work both for a single device and multi-selected devices. Make sure to keep the pointer directly on the selected Site.

After adding the device to the Workspace Tree, it is possible to connect to it and use its full potential.

3.3 Commissioning

Once the VAV14-IP controller is connected and configured in scope of device and networks parameters, proceed to working with application. The VAV14-IP controller is equipped with a pre-loaded application, which suits most of the VAV boxes available on the market (VAV14-IP as a Configurable Controller).

Credentials

With the default application on the controller, the VAV14-IP default credentials are:
- Username: admin
- Password: Admin123!

For more information, please see: Default Communication Settings and Credentials.

It also offers the possibility of further application enhancement or even building a user-custom application from scratch (VAV14-IP as a Freely-programmable Controller), which makes the controller suitable for most advanced VAV systems.

To proceed, select the tool, which is most suited for user requirements and continue to configuring of the application or building a new one.

Tool	App configuration	Flow settings
S3 DIP switch	DIP Switches	-
iSMA Configurator	App configuration	Flow settings
Control Point	App configuration	Flow settings
iC Device Manager	App configuration	Flow settings
iC Tool	nano EDGE ENGINE libraries for VAV application	
BACnet objects/Modbus registers	BACnet objects/Modbus registers	

Warning!

Please keep in mind that the setting of the first switch on the S3 DIP switch is vital for the configuration possibilities:
- switch 1 on the S3 DIP switch set to **ON**: configuration of the fan, reheater, and perimeter types (and referring parameters) possible **only** from the S3 DIP switch,
- switch 1 on the S3 DIP switch set to **OFF**: configuration of the fan, reheater, and perimeter types (and referring parameters) possible from the other tools: Control Point VAV/BACnet/Modbus/iC Configurator/iC Device Manager.

The S3 DIP switch setting holds priority over other tools. If it is set to ON, no change made to the mentioned parameters in any other tool (Control Point VAV or software) will be written to the controller!

4 VAV14-IP as a Configurable Controller

The VAV14-IP is delivered with a built-in application, which supports the most popular types of VAV boxes. The application is built in the nano EDGE ENGINE software platform. It offers easy configuration options in a number of tools fitting various user requirements.

- VAV application

The controller also allows for performing balancing with pre-designed actions available in the same tools which are used for configuration purposes.

- Balancing

4.1 VAV Application

Credentials

With the default application on the controller, the VAV14-IP default credentials are:
- Username: admin
- Password: Admin123!

For more information, please see: Default Communication Settings and Credentials.

A default application for the VAV14-IP controls the airflow adjusting it to a current demand for ventilation intensity depending on:

- occupancy status,
- space temperature,
- setpoint temperature,
- CO_2 level.

The application allows to optimize energy consumption of the air handling units.

The airflow control is based on the currently measured airflow in relation to the calculated setpoint for the airflow, which is calculated based on the demand for cooling, heating, or due to excessive CO_2 concentration, as well as due to forcing from the HVAC mode. The airflow demand is affected by the level of damper opening resulting from the airflow control loop.

The control loop for the CO_2 concentration in the room has a priority effect on the calculated airflow setpoint. This means that the airflow can be higher than the cooling or heating demand if the CO_2 concentration in the room is too high in order to ventilate the room. In the absence of a CO_2 sensor, the control loop does not affect the calculated flow setpoint.

The supply air temperature control loop is used to limit the supply air temperature specified by the current supply air setpoint based on the supply air temperature measurement. It can limit the reheater or perimeter control. The supply air temperature control loop operates only when there is a heating demand. If there is no supply air sensor, the control loop has no effect on limiting the supply air temperature.

The application is constructed to work in the following device configurations:

- VAV cooling only,
- VAV cooling/heating,
- VAV cooling with a water reheater and optional perimeter,
- VAV cooling with a 2-stage electric reheater and optional perimeter,
- series fan powered VAV cooling with a water reheater and optional perimeter,
- series fan powered VAV cooling with a 2-stage electric reheater and optional perimeter,
- parallel fan powered VAV cooling with a water reheater and optional perimeter,
- parallel fan powered VAV cooling with a 2-stage electric reheater and optional perimeter.

4.1.1 Configuration Scope

The VAV default application can be configured in the following areas:

- device type,
- damper direction,
- heating device type and priority,
- fan type,
- occupancy source, window and presence sensors,
- temperature source,
- temperature setpoint and offset,
- CO_2 source,

- airflow loop settings.

For more details, please see Methods of Configuration.

4.1.2 Application Modes

Taking into consideration all configuration options, types of devices used, and airflow settings, the VAV default application can be adapted to work in the following modes:

- VAV cooling only,
- VAV cooling/heating,
- VAV with water reheater and optional perimeter,
- VAV with up to 2-stage electric reheater and optional perimeter
- series fan powered VAV cooling with water reheater and optional perimeter,
- series fan powered VAV cooling with up to 2-stage electric reheater and optional perimeter,
- parallel fan powered VAV cooling with water reheater and optional perimeter,
- parallel fan powered VAV cooling with up to 2-stage electric reheater and optional perimeter.

For more details, please see Configuration Variants.

4.1.3 Application Algorithm

The VAV application algorithm is designed to maximize the user comfort and energy efficiency, achieving the set temperature by adjusting the airflow based on the occupancy status, measured temperature and CO_2 levels. It functioning is formulated on the operation the following control loops.

- space temperature control loop determines the demand for cooling or heating based on the room temperature measurement and the corresponding setpoint temperature. The control loops for cooling and heating are mutually exclusive, i.e., only one of them regulates the demand, while the other is disabled.
- discharge air temperature control loop is used to limit the discharge air temperature specified by the current discharge air setpoint based on the discharge air temperature measurement. It has the effect of limiting the reheater control. The discharge air temperature control loop operates only when there is a need for heating. If there is no discharge air sensor, the control loop has no effect on limiting the discharge air temperature (it has a value of 0).
- room CO_2 concentration control loop is used to adjust the airflow according to the measured CO_2 concentration. It has a priority effect on the calculated airflow setpoint. It means that the airflow can be higher than the cooling or heating demand if the CO_2 concentration in the room is too high in order to ventilate the room. In the absence of a CO_2 sensor, the control loop does not affect the calculated airflow setpoint (it has a value of 0).
- airflow control loop is based on the currently measured airflow in relation to the calculated airflow setpoint, which is based on the demand for cooling/heating, measured CO_2 concentration, and the HVAC mode settings. The demand for airflow is affected by the level of damper opening resulting from the airflow control loop.

Temperature

Temperature

Temperature

The set temperature is a central part of the VAV application algorithm. Reaching and maintaining the set temperature is the goal of the algorithm's operation by adjusting the airflow based on the current temperature measurements and occupancy status.

The VAV application’s algorithm includes a number of variables referring to the temperature:

- temperature setpoint,
- temperature setpoint’s offset,
- space temperature,
- discharge temperature.

Space Temperature

One of the essential configuration settings for a proper functioning of the VAV application is selecting the temperature measurements source. The available options are:

- Control Point VAV panel,
- dedicated universal input, U2 (connected sensor),
- network.

Automatic Detection

Additionally, the VAV application has a mechanism to provide an automatic temperature detection in case the selected temperature source does not provide any value. First, the application checks for the Control Point VAV panel availability. If the value cannot be read from the panel, the algorithm checks the dedicated U2 input and if no sensor is connected, it proceeds to check the value from the network. In case all these steps fail to provide the measured temperature, the algorithm uses the calculated effective setpoint value, which forces the application to work in minimal ranges.

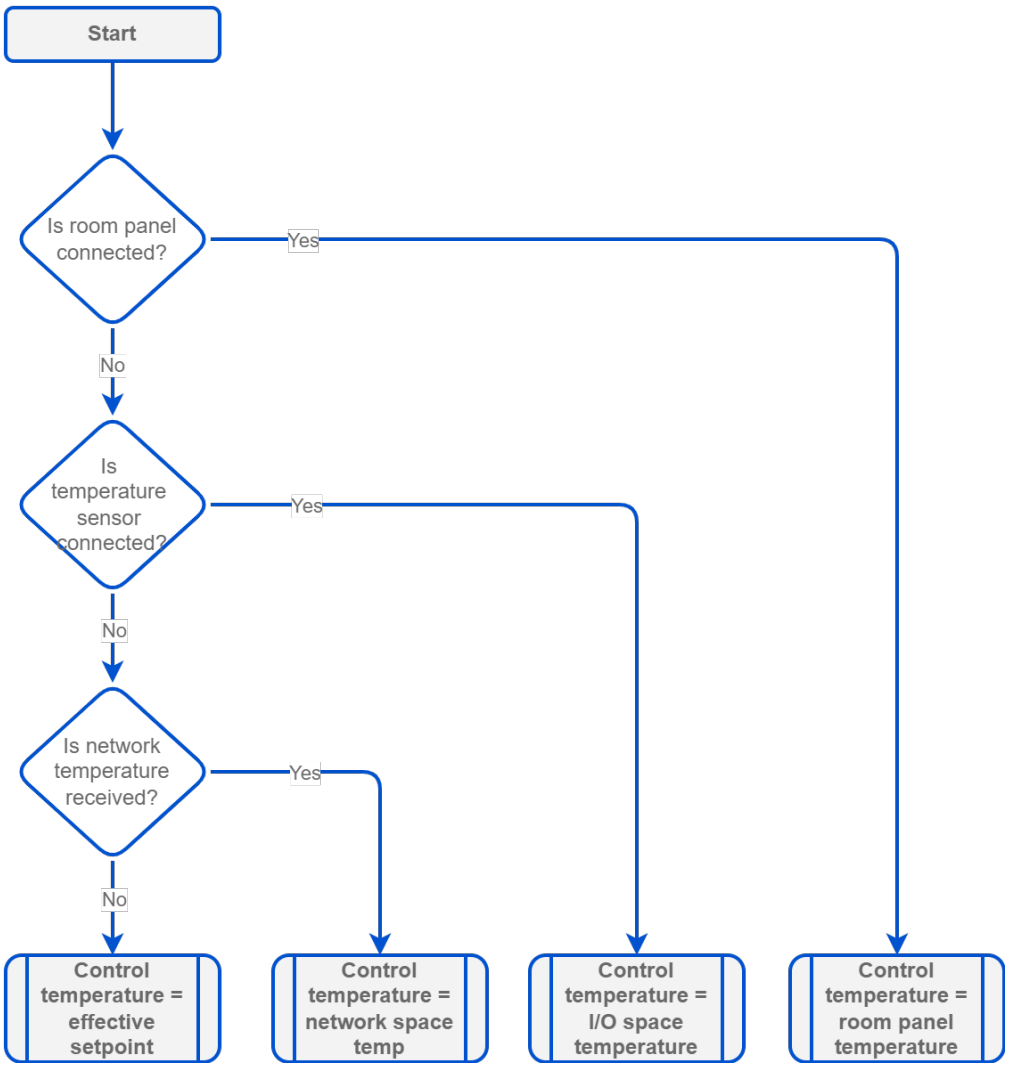


Figure 14. Temperature automatic detection mechanism

Note

In case more than one temperature source is connected and active (for example, the Control Point VAV panel and sensor connected to the U2 input), it is possible to indicate one of them as a leading one using the Input Selector slot (TemperatureSensorSelector component in the TemperatureSelector folder of the VAV application) or calculate an average using the Temperature Averaging slot (the same location).

Discharge Air Temperature (Only for Heating)

Discharge temperature is a discharge air temperature measured in two possible variants:

- before VAV in models without a reheater: detecting the temperature of the discharge air to determine operating mode (cooling/heating),
- after VAV in models with a reheater or reheater and fan: acting as high limit for discharge air - the reheater is controlled by the space temperature and the discharge temperature limits the reheater when the discharge air is too hot.

The universal input U1 is dedicated for the sensor of the discharge temperature.

Automatic Detection

Similarly, as in case of the space temperature, the application has a mechanism providing an automatic discharge temperature detection in case the sensor is not available or its value cannot be read from it. In

such event, the algorithm takes the MinDischTempSetpoint component's value to calculations (located in the TemperatureSetpointSelector folder of the VAV application).

Temperature Setpoint

Temperature setpoint is a fundamental setting in the VAV application's algorithm. It determines the airflow calculations depending on the area's occupancy status. There are two temperature setpoint variables depending on the application operating mode, cooling or heating. The algorithm will always work to achieve the temperature setpoint, either in the cooling or heating mode.

Cooling Setpoint

The application starts cooling if the space temperature rises above the cooling setpoint temperature for the corresponding occupied state.

Heating Setpoint

The heating setpoint is used for heating control with high limit in systems with the reheater or sets the reference point for a warm air temperature.

For heating control, information on the discharge air temperature from the discharge temperature sensor connected to the U1 input or from the network is required. In systems with the reheater, if there is no discharge air temperature measurement, heating is based on the space temperature control loop. In systems without the reheater and without the discharge air temperature measurement, heating is disabled.

The application starts heating if the space temperature falls below the heating setpoint temperature for the corresponding occupied state.

> **Note**
> The CentralSetpoint value (TemperatureSetpointSelector component in the TemperatureSetpointSelector of the VAV application) is an average of the heating and cooling setpoints according to the occupancy mode.

Effective Setpoint

The setpoint temperatures for the occupied mode are further adjusted by:

- offset: the value is added or subtracted from the setpoints;
- panel-derived setpoint temperature depending on the setting (offset or setpoint) of the PanelSetpointMode variable (TemperatureSetpointSelector folder of the VAV application):
  - offset: added or subtracted from the setpoints,
  - setpoint: the setpoint set on the Control Point VAV panel is corrected by half the difference between the OccCoolTempSetpoint and OccHeatTempSetpoint values, added or subtracted from the panel's setpoint providing setpoints for heating and cooling.

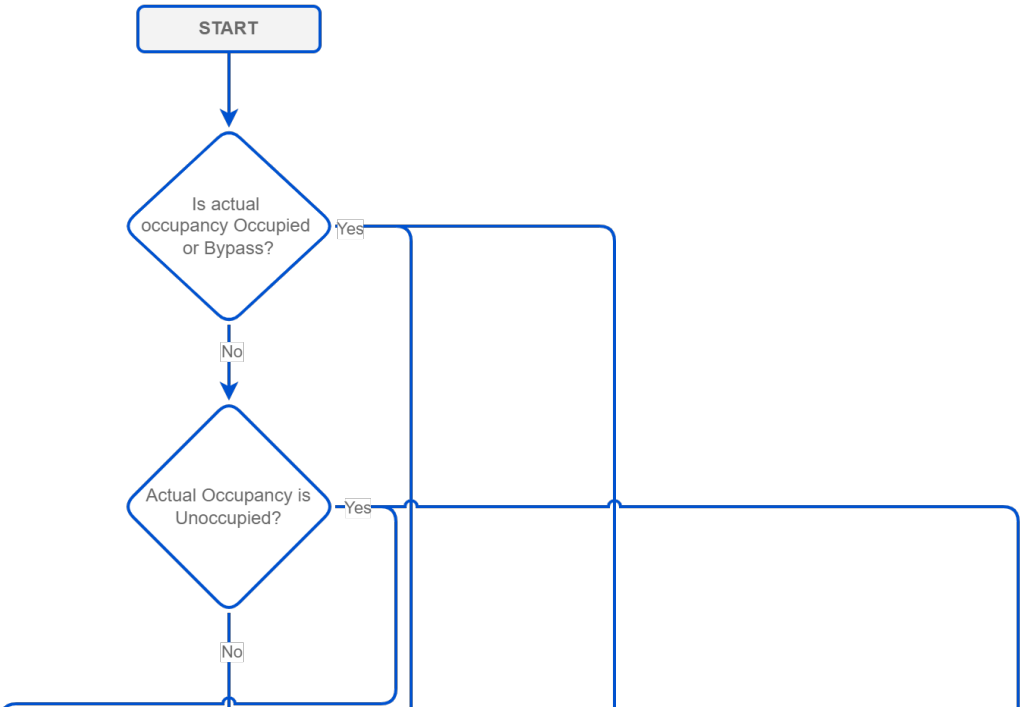
This selection between cooling or heating setpoint impacting the effective setpoint temperature is made when the HVAC Mode slot (set in the AirflowSetpointSelector component):

- Auto: selection between heating/cooling setpoint:
  - heating demand active: effective setpoint equals the heating setpoint,
  - cooling demand active: effective setpoint equals the cooling setpoint;
- Heat: effective setpoint equals the heating setpoint depending on the occupancy mode:
  - occupied: effective setpoint equals the OccHeatTempSetpoint value,
  - bypass: effective setpoint equals the OccHeatTempSetpoint value,
  - standby: effective setpoint equals the StandbyHeatTempSetpoint value,
  - unoccupied: effective setpoint equals the UnoccHeatTempSetpoint value;
- Cool: effective setpoint equals the cooling setpoint depending on the occupancy mode:
  - occupied: effective setpoint equals the OccCoolTempSetpoint value,

    - bypass: effective setpoint equals the OccCoolTempSetpoint value,
    - standby: effective setpoint equals the StandbyCoolTempSetpoint value,
    - unoccupied: effective setpoint equals the UnoccCoolTempSetpoint value;
- MorningWarmUp: effective setpoint equals the heating setpoint depending on the occupancy mode:
    - occupied: effective setpoint equals the OccHeatTempSetpoint value,
    - bypass: effective setpoint equals the OccHeatTempSetpoint value,
    - standby: effective setpoint equals the StandbyHeatTempSetpoint value,
    - unoccupied: effective setpoint equals the UnoccHeatTempSetpoint value;
- PreCool: effective setpoint equals the cooling setpoint depending on the occupancy mode:
    - occupied: effective setpoint equals the OccCoolTempSetpoint value,
    - bypass: effective setpoint equals the OccCoolTempSetpoint value,
    - standby: effective setpoint equals the StandbyCoolTempSetpoint value,
    - unoccupied: effective setpoint equals the UnoccCoolTempSetpoint value;
- NightPurge: effective setpoint equals the cooling setpoint depending on the occupancy mode:
    - occupied: effective setpoint equals the OccCoolTempSetpoint value,
    - bypass: effective setpoint equals the OccCoolTempSetpoint value,
    - standby: effective setpoint equals the StandbyCoolTempSetpoint value,
    - unoccupied: effective setpoint equals the UnoccCoolTempSetpoint value;
- Fire: effective setpoint equals the cooling setpoint depending on the occupancy mode:
    - occupied: effective setpoint equals the OccCoolTempSetpoint value,
    - bypass: effective setpoint equals the OccCoolTempSetpoint value,
    - standby: effective setpoint equals the StandbyCoolTempSetpoint value,
    - unoccupied: effective setpoint equals the UnoccCoolTempSetpoint value;
- Off: effective setpoint equals the cooling setpoint depending on the occupancy mode:
    - occupied: effective setpoint equals the OccCoolTempSetpoint value,
    - bypass: effective setpoint equals the OccCoolTempSetpoint value,
    - standby: effective setpoint equals the StandbyCoolTempSetpoint value,
    - unoccupied: effective setpoint equals the UnoccCoolTempSetpoint value.

Discharge Air Setpoint

The discharge air temperature setpoint is obtained by a linear function from the current value of the effective heating setpoint temperature to the effective setpoint increased by a value of 15 degrees with a low limit MinDischTempSetpoint and high limit MaxDischSetpoint. The value is calculated based on the output value from the heating control loop. The x value is based on the heating demand and depending on whether the DualHeat options is turned on (50-100% default for a secondary source or 0-50% for primary source) or turned off (0-100%).



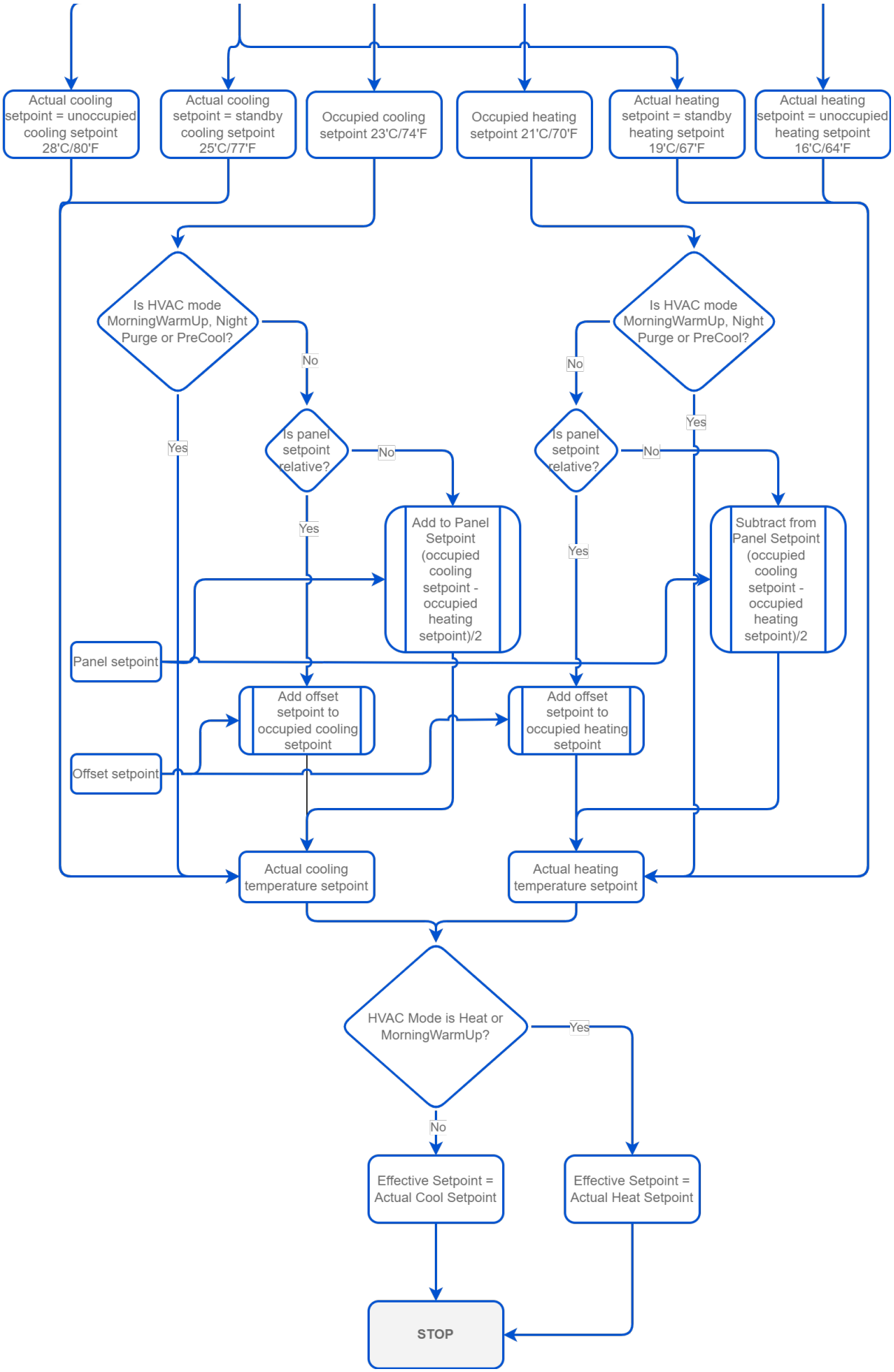


Figure 15. Temperature setpoint

Temperature Setpoint Offset

The offset value for the temperature setpoint can be set from the Control Point VAV panel or using a resistance adjuster like SP connected to the U3 input. It is possible to select a leading offset source in the Input Selector variable (SetpointOffsetSelector in the OffsetCalculator folder of the VAV application).

From the Control Point VAV panel and iSMA Configurator, the user has the option to change the range in the SetpointOffsetRange variable for a resistance adjuster connected to the U3 input, which by default is set to 3°C (from -3 to +3) or 5°F (from -5 to +5). Changing the variable also triggers a change in the offset setpoint range in the Control Point VAV panel.

Temperature Control Loops

The temperature control loop in the VAV application determines the demand for cooling or heating based on the space temperature measurement and the corresponding setpoint temperature.

The control loops for cooling and heating are mutually exclusive, i.e., only one of them regulates the demand, while the other is disabled.

Space Temperature Control Loop

The space temperature control loop for heating and cooling is based on the space temperature sensor measurements (Control Point VAV panel or connected to the U2 input). It operates according to the HVAC Mode settings (AirflowSetpointCalculator):

- Auto: the cooling control loop is always active,
- Heat: the heating control loop is always active,
- Cool: the cooling control loop is always active,
- MorningWarmUp: the heating control loop is always active,
- PreCool: the cooling control loop is always active,
- NightPurge: the cooling control loop is always active,
- Fire: both control loops are disabled,
- Off: both control loops are disabled.

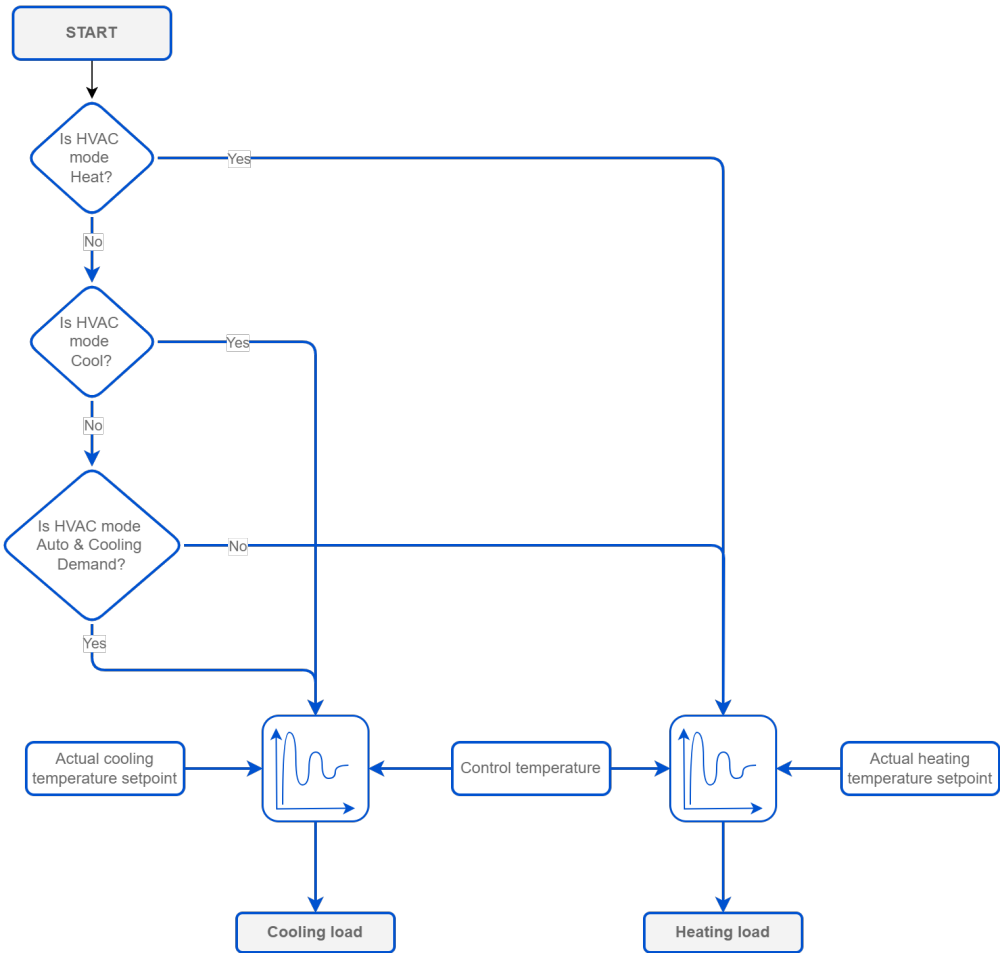


Figure 16. Temperature control

Discharge Air Temperature Control Loop

The discharge air temperature control loop for heating and cooling is active when there is the discharge air sensor connected to the U1 input or there is the discharge air signal available from the network. It operates according to the HVAC mode set in the HVAC Mode settings (AirflowSetpointCalculator component):

- **Auto:** the heating or cooling control loop will be active, depending on the measured temperature:
  - the heating control loop is activated when:
    - the measured temperature exceeds the value of the HotAirTempSetpoint variable (by default, 26°C/78°F), taking into account the hysteresis (+/-3 degrees),
    - the above temperature is maintained above the time specified in the ChangeoverDelay (by default, 2 min);
  - the cooling control loop is activated when:
    - the measured temperature falls below the value of the HotAirTempSetpoint variable (by default, 26°C/78°F), taking into account the hysteresis (+/-3 degrees),
    - the above temperature is maintained above the time specified in the ChangeoverDelay (by default, 2 min);
- **Heat:** the heating control loop is always active,
- **Cool:** the cooling control loop is always active,
- **MorningWarmUp:** the heating control loop is always active,
- **PreCool:** the cooling control loop is always active,
- **NightPurge:** the cooling control loop is always active,
- **Fire:** both control loops are disabled,
- **Off:** both control loops are disabled.

Occupancy

In the VAV application, occupancy is one of the factors determining the calculated airflow. In the application algorithm, the occupancy status has an impact on the control of a dead zone range to achieve maximum energy efficiency taking into account the user's comfort temperature. The dead zone means that no cooling or heating is executed and the airflow is minimized. The algorithm is designed to adjust the dead zone to the occupancy status in order to maximize energy efficiency and minimize the effect on the user's comfort – even the smallest range of the dead zone in the occupied mode generates savings for the system, while the effect on the user's comfort is non-distinctive.

Primarily, the occupancy status is retrieved from the network. If there is no update from the network and the OccupancyMode Data Point falls into status different than OK, the local schedule is checked next and becomes a source for the occupancy status.

The occupancy status is also affected by the motion detector and window open/close state.

The motion detector, if available, is connected to the I1 digital input. Its state is read to the PresureSensor variable. The PresenceSensorInvert variable has two modes, normal and invert, which are used to control the sensor state interpretation. In the normal mode, if the value from I1 is true, it means the sensor has detected presence and the occupancy status is switched to occupied. If I1 is false, it means no motion has been detected. If the sensor does not detect motion after the StandbyTimeOverride time expires, it switches the status to standby (temporarily unoccupied). In the invert mode, if I1 is true, it means no presence has been detected, and if I1 is false, it means that motion has been detected.

Note

If there is no presence sensor connected, it is recommended to use the inverted mode, because in the inverted mode the constant state is motion detected.

The window contact switch is connected to the I2 digital input. Its open/close state is read to the WindowContact variable. The WindowContantInvert variable has two modes, normal and invert, which are used to control the sensor state interpretation. In the normal mode, if the value from the I2 is true, the window is open, if the value is false, the window is closed. In the invert mode, if the I2 value is true, it means the window is closed, and if the I2 is false, the window is open. Most contact switches work in the invert mode. If there is no contact switch installed on the window, the WindowContantInvert works in its default, normal, mode.

Occupied Mode

In the occupied mode, the application algorithm works to reach a desired comfort temperature taking into account the narrow dead zone (21-23°C/70-74°F) and provides the calculated airflow according to the following rules:

- if the temperature rises above the dead zone range (>23°C/74°F), the algorithm increases the airflow;
- if the temperature falls below the dead zone range (<21°C/70°F), the algorithm decreases the airflow.

Standby Mode (Temporarily Unoccupied)

The standby mode means that the area is temporarily unoccupied (e.g., an employee has left the room for a meeting). It is normally used in combination with the presence sensor. In the standby mode, the dead zone is expanded to the range of 19-25°C/67-77°F. The occupied status changes to standby after the time set in the StandbyTimeOverride variable (by default, 15 minutes). If the presence sensor detect motion again, the status changes back to occupied. Occupancy in the standby mode does not change automatically to unoccupied, only the signal form the BMS can trigger the unoccupied status (in the OccupancyMode variable).

Bypass Mode (Temporarily Occupied)

The bypass mode means that the area is temporarily occupied (e.g., in a spare conference room, which is normally unoccupied, the bypass mode would be used for an occasional meeting), the dead zone is the

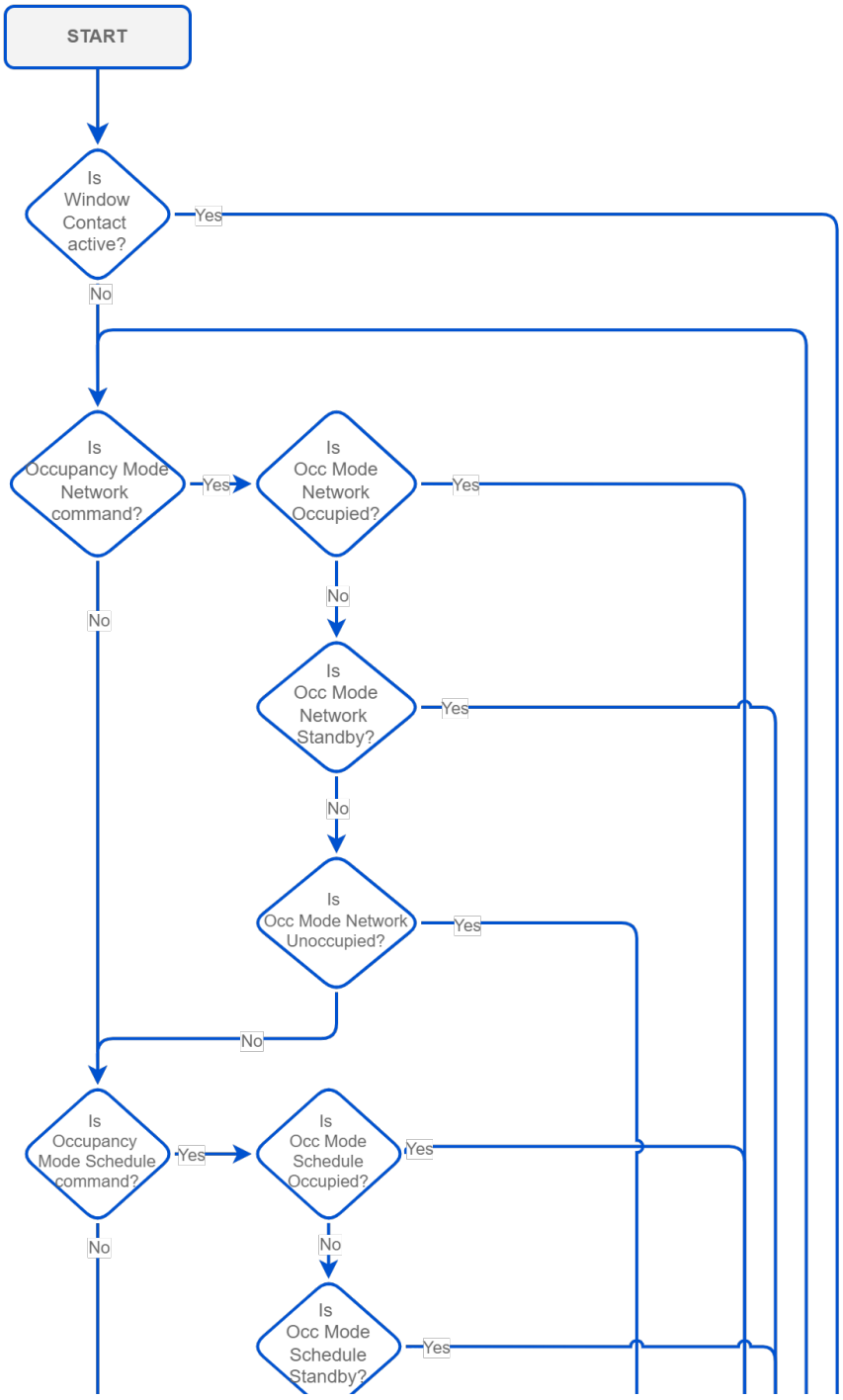
same as in the occupied mode (21-23°C/70-74°F) but it is active for a specific time (by default, 2 hours) and switches back to a previous state.

Unoccupied Mode

The unoccupied mode triggers the biggest range of the dead zone (16-28°C/64-80°F). The unoccupied mode is usually activated from the BMS or the local schedule.

Auto Occupancy

An additional option is the Auto Occupancy function which, when enabled in the AutoOccMode variable, examines the change in airflow in relation to the setpoint – when the measured value rises above 30% (the default value set in the OccupancyCalculator component, Increased Air Flow Level slot) the setpoint, the occupancy mode will switch to the Occupied state; when the measured value falls below 25% (by default, -5%) and maintains this difference for the Flow Occupied Time (by default, 5 min), it will return to the Unoccupied mode.



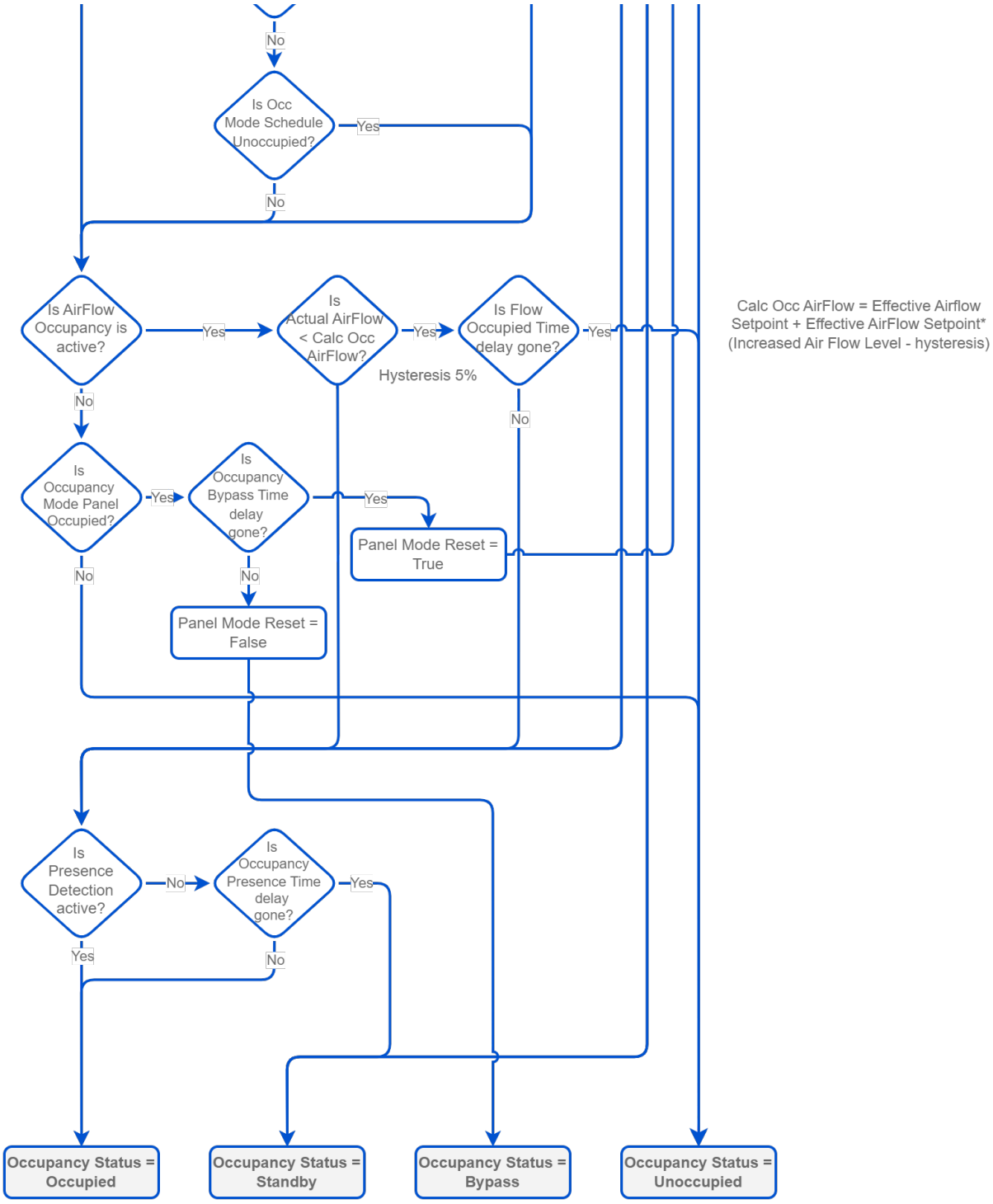


Figure 17. Occupancy diagram

CO2

CO2 Measurement

The CO_2 measurement provides an additional factor in the VAV application's algorithm, which takes it into calculations along with the temperature control loop. The CO_2 measurements can be derived from three different sources:

- Control Point VAV panel,
- dedicated universal input, U4 (connected sensor),
- network.

Automatic Detection

The VAV application has a mechanism to provide an automatic CO_2 detection in case the selected source does not provide any value. First, the application checks for the Control Point VAV panel availability. If the value cannot be read from the panel, the algorithm checks the dedicated U4 input and if no sensor is connected, it proceeds to check the value from the network. In case all these steps fail to provide the measured CO_2 level, the algorithm uses the SpaceCO2Setpoint (by default, 1000 ppm), which has no effect on the VAV operation.

Note

In case more than one CO_2 source is connected and active (for example, the Control Point VAV panel and sensor connected to the U4 input), it is possible to indicate one of them as a leading one using the Input Selector slot (RoomCO2Selector component in the CO2Selector folder of the VAV application) or calculate an average using the CO2 Averaging slot (the same location).

In the Control Point VAV and iSMA Configurator, it is possible to adjust the range of the U4CO2SensorRange variable for the CO_2 sensor connected to the U4 input, which by default is 2000 ppm.

CO2 Control Loop

The CO_2 measurement in the VAV application algorithm activates the CO_2 control loop. If the set CO_2 level is exceeded and the CO_2 control loop output is:

- 0-50%: the VAV application increases the airflow scaling the 0-50% CO_2 control loop to 0-100% of airflow;

Example

The airflow is controlled in the range of 100-200 l/s. The temperature control loop output is 10%, which means the airflow should be set to 110 l/s. However, the CO_2 control loop output is 50%, which forces the 200 l/s airflow.
When the CO_2 measurement is available, the algorithm compares it with the temperature control loop output and always the higher value of the two impacts the airflow control.

- 50-100%: if the CO_2 control loop output still rises in spite of the VAV increasing the airflow, the output in the range of 50-100% is forwarded to the AHUOutDoorDamper variable (scaled to 0-100%) for the air handling unit to increase the opening level of the fresh air damper.

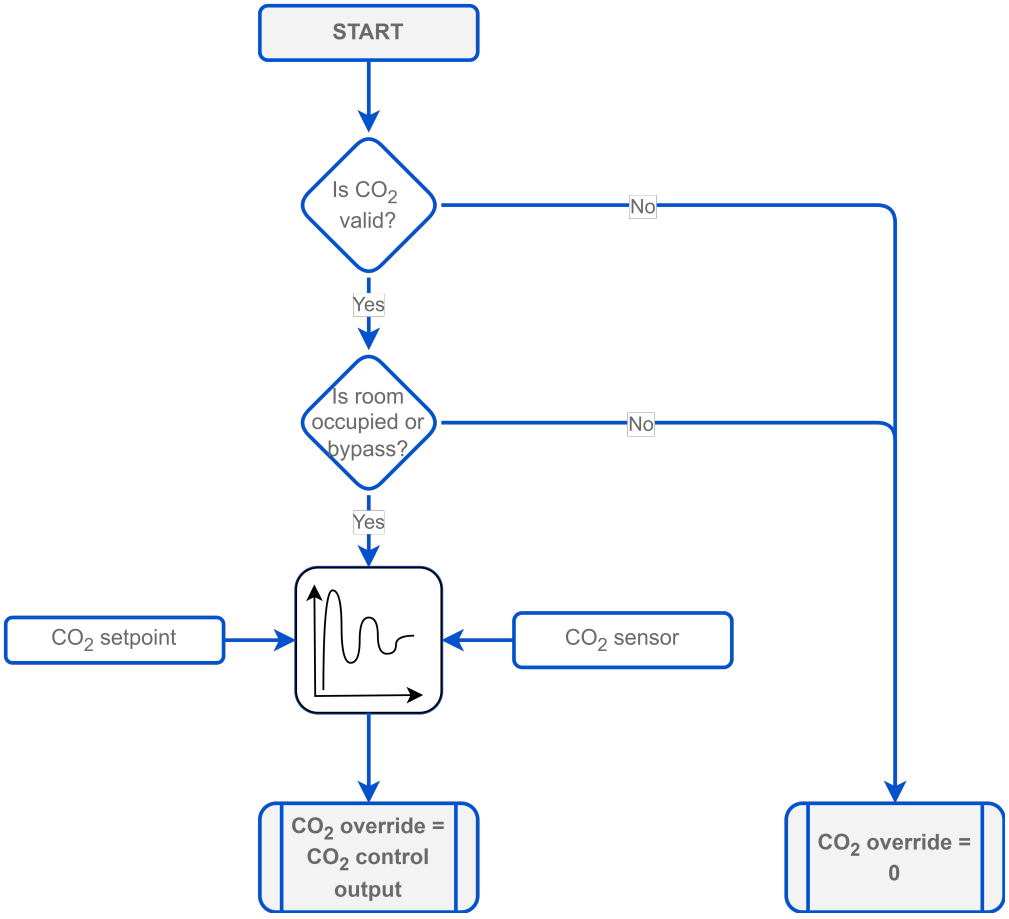


Figure 18. CO2 control

Airflow

Airflow Control

Airflow in the VAV application is the main factor used to achieve a given temperature setpoint. The current airflow rate in l/s or cfm [V=K*sqrt(dP)] is calculated based on the differential pressure measurement using a built-in pressure transducer and the K-Factor variable.

K-Factor

The K-Factor variable, set in the calibration process from the Control Point VAV panel, iSMA Configurator, or BACnet/Modbus, is a parameter provided by the VAV box manufacturer. It determines the airflow through the unit at 1 Pa/1 inH2O.

The basic element enabling the regulation of the airflow stream is the measurement of its flow, which is obtained by measuring the pressure difference carried out on the measuring element (e.g., on the measuring cross) and performing appropriate mathematical calculations using the K-factor coefficient related to the VAV box.

Because the K-factor is determined at the factory laboratory-like conditions, it often requires modification by airflow calibration.

Additionally, the pressure transducer itself, although factory-calibrated, may require calibration during commissioning (zeroing).

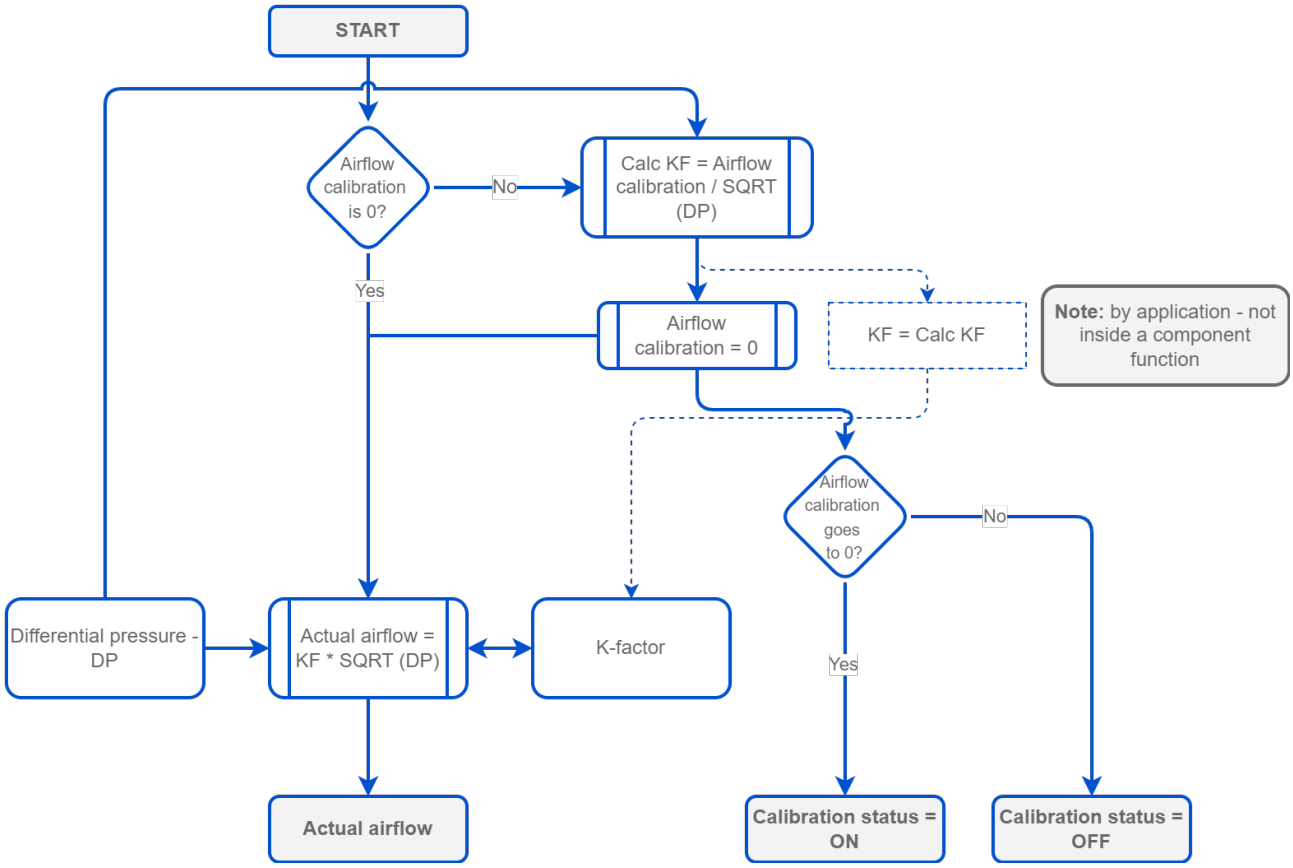


Figure 19. Actual airflow diagram

During the calibration process, the balancing technician takes the actual measurement of the airflow and enters the measured value into the AirIlowCalibration component (in the AirFlowCalculator folder of the VAV application), triggering a recalculation of the pre-set K-Factor parameter so that the airflow indicated by the device corresponds to the actual airflow.

The built-in pressure transducer is also equipped with a zeroing function, available from the Control Point VAV panel, iSMA Configurator, or BACnet/Modbus (to obtain a zero measurement in the absence of airflow), which is performed in the absence of airflow (additionally, it is advisable to close the throttle) when the measured pressure difference is non-zero (minimal reading fluctuations are allowed due to measurement error).

Concerning the output of the cooling or heating control loop and the specified airflow limits (in the occupied state), the current setpoint value for the airflow can be determined as a linear function.

The heating loop is used when hot air is detected (using a discharge air temperature sensor), and in the case of a system with a duct reheater, the airflow setpoint can be shaped as a linear function of the heating loop within the airflow limits (generally used as the 2nd heating stage) or as a constant minimum airflow setpoint for heating when the reheater is to be the main heat source (no Dual Heating).

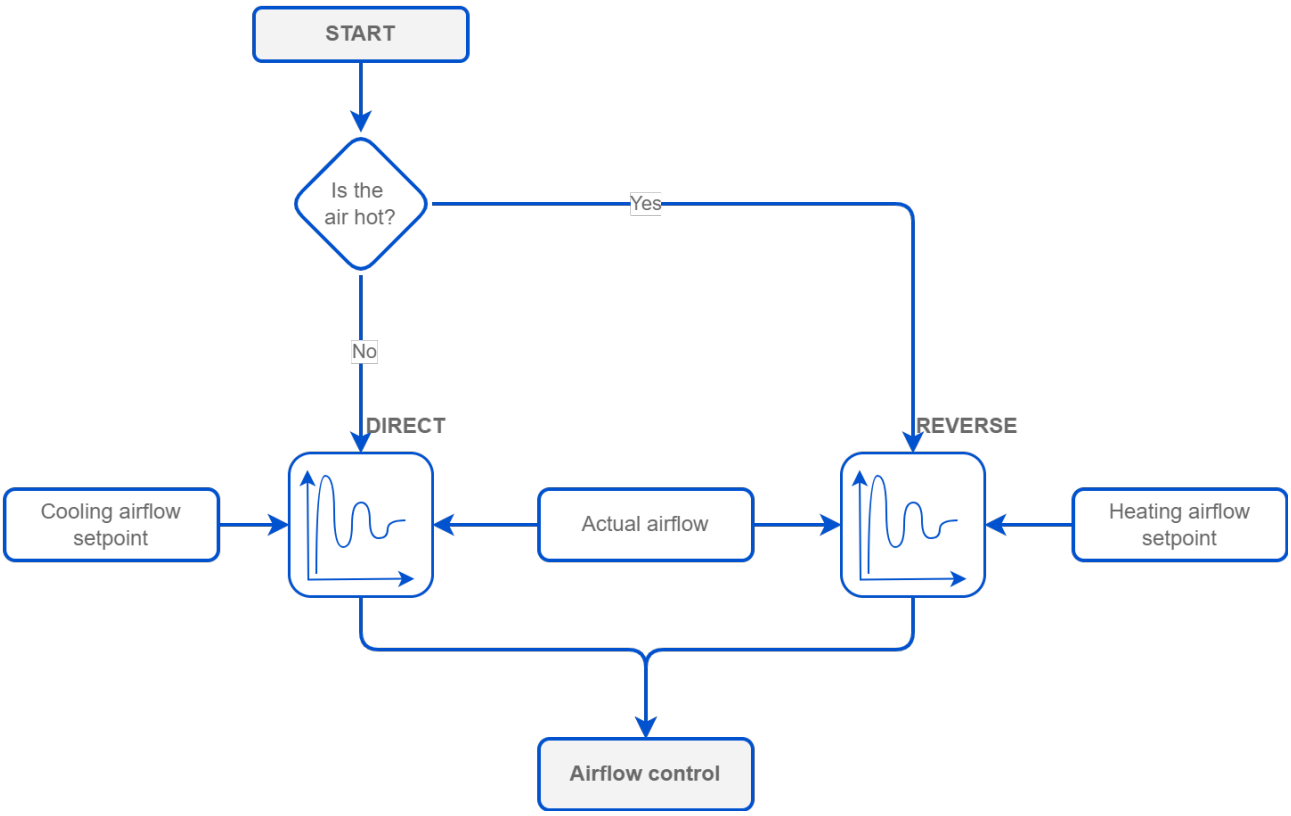


Figure 20. Airflow control

Airflow Setpoint

The airflow setpoint depends on the HVAC Mode settings (AirflowSetpointCalculator component) and may depend on the occupancy status.

- Auto: in the Auto HVAC mode and the operating mode also set to auto, the airflow setpoint is determined by the occupancy status:
  - occupied and bypass:
    - heating: MaxOccHeatAirFlowSetpoint and MinOccHeatAirFlowSetpoint,
    - cooling: MaxOccCoolAirFlowSetpoint and MinOccCoolAirFlowSetpoint;
  - standby (StandbyAirFlowSetpoint) and unoccupied (UnoccAirFlowSetpoint): the setpoint is the same for heating and cooling.

By default, the maximum airflow setpoint for heating is set as half of the value of the airflow setpoint for cooling.

The setpoints are also determined by the demand resulting from the control loops calculated as a linear function between the minimum and maximum values for heating or cooling, respectively. The values

resulting from the heating and cooling control loops are mutually exclusive, which means that only one of them sets the setpoint for the airflow (the other is off - it has a zero value).

- **Heat:** the airflow setpoint is set depending on the occupancy status:
  - occupied and bypass: between the MaxOccHeatAirFlowSetpoint and MinOccHeatAirFlowSetpoint values - it is a function of the linear heating demand resulting from the heating control loop;
  - unoccupied (UnoccAirFlowSetpoint) and standby (StandbyAirFlowSetpoint), the airflow setpoint is identical to the Auto mode.
- **Cool:** the airflow setpoint is set depending on the occupancy status:
  - occupied and bypass: between the MaxOccCoolAirFlowSetpoint and MinOccCoolAirFlowSetpoint values - it is a function of the linear cooling demand resulting from the cooling control loop;
  - unoccupied (UnoccAirFlowSetpoint) and standby (StandbyAirFlowSetpoint), the airflow setpoint is identical to the Auto mode;
- **MorningWarmUp:** the airflow setpoint is set to the MaxOccHeatAirFlowSetpoint, regardless the occupancy status;
- **PreCool:** the airflow setpoint is set to the MaxOccCoolAirFlowSetpoint, regardless the occupancy status;
- **NightPurge:** the airflow setpoint is set to the MaxOccCoolAirFlowSetpoint, regardless the occupancy status;
- **Fire:** the airflow setpoint is set to the MaxOccCoolAirFlowSetpoint, regardless the occupancy status;
- **Off:** the airflow setpoint is set to 0, regardless of the occupancy status.

Notes

- In addition, the airflow setpoint that would result from heating or cooling is affected by the value resulting from the increased CO_2 concentration in the room resulting from the control loop, and it takes priority in setting the airflow setpoint as a linear function between the MaxOccCoolAirFlowSetpoint and MinOccCoolAirFlowSetpoint values (for cooling) over the heating or cooling demands.
- In the case of a VAV box version with the parallel fan, when it is started and run for the period set in the StartupFanDelay (by default, 1 min), the airflow setpoint is set to the StartupAirFlowSetpoint value.

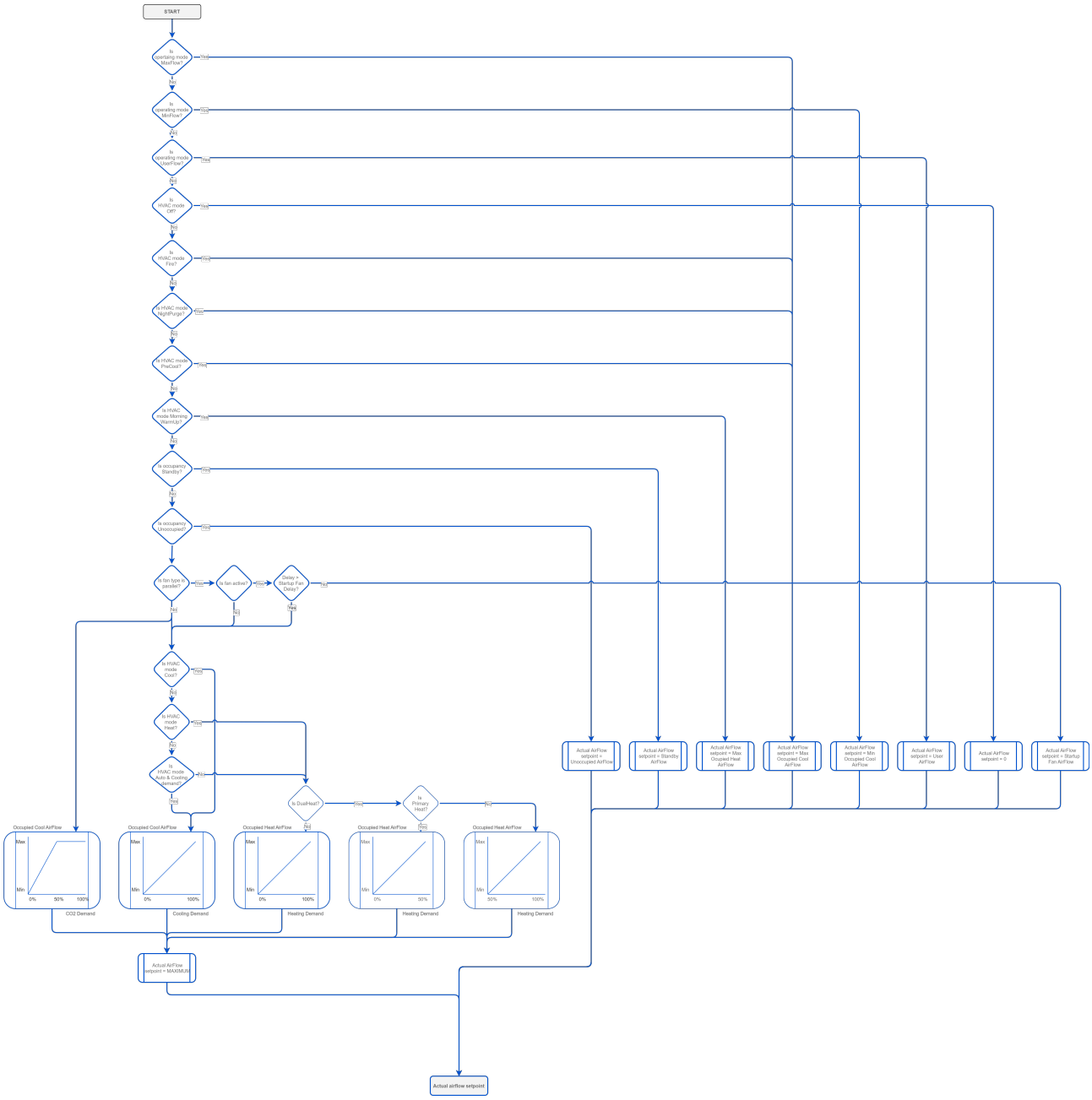


Figure 21. Airflow setpoint diagram

Damper Control

The DamperControl variable is used to directly control the motor which is responsible for opening or closing the damper. The motor is controlled in the form of a floating type of control, where a specific drive time is required for correct (full opening) control, allowing for full opening of the damper under load.

Due to the possibility of the transmitted signal diverging from the actual opening (e.g., in the event of jamming, blocking, etc.), a calibrated mechanism is used, which involves a calibration process for the duration of the motor's drive time. This function is always performed after the power supply voltage fails and returns, and cyclically, e.g. every day at midnight.

The priority signal is the damper operating mode resulting from, among others, the balancing or testing procedure. In case of the version with a configured series fan, starting the fan is preceded by closing the damper waiting a specific time after the fan starts, and then opening it again.

In systems without the reheater, the damper is used as a heating source (requires the discharge air temperature sensor). In systems with the reheater, it is used as one of the heating stages (by default, as the second stage).

Damper control in the VAV application depends on the type of device selected in the DeviceType variable (DamperControl folder of the VAV application):

- VAV (variable air volume system): damper control based on airflow demand (pressure independent),
- VVT (variable volume and temperature): damper control based directly on cooling or heating demand (pressure dependent).

Damper control is carried out in the range defined by the MinDamperPosition and MaxDamperPosition variables, scaling the corresponding demands.

For the correct operation of the damper, the correct transition time for a full cycle – from fully closed to fully open – must be set in the DamperDriveTime variable (by default, 90 s) and the correct direction of damper operation must be set in the DirectionDirection variable as CW (clockwise) or CCW (counterclockwise).

Warning!

Changing the DamperDriveTime without a valid reason is strongly not recommended.

The DamperPosition variable returns the actual value of the damper opening, and the DamperControl variable returns its control.

The Damper Command variable, accessible from the Control Point VAV panel and iSMA Configurator, is mainly used to perform balancing and force testing of the set airflow. It allows to force the setting of the set airflow as maximum flow (for cooling), minimum flow (for cooling), as well as any airflow set in the UserSetAirFlow variable.

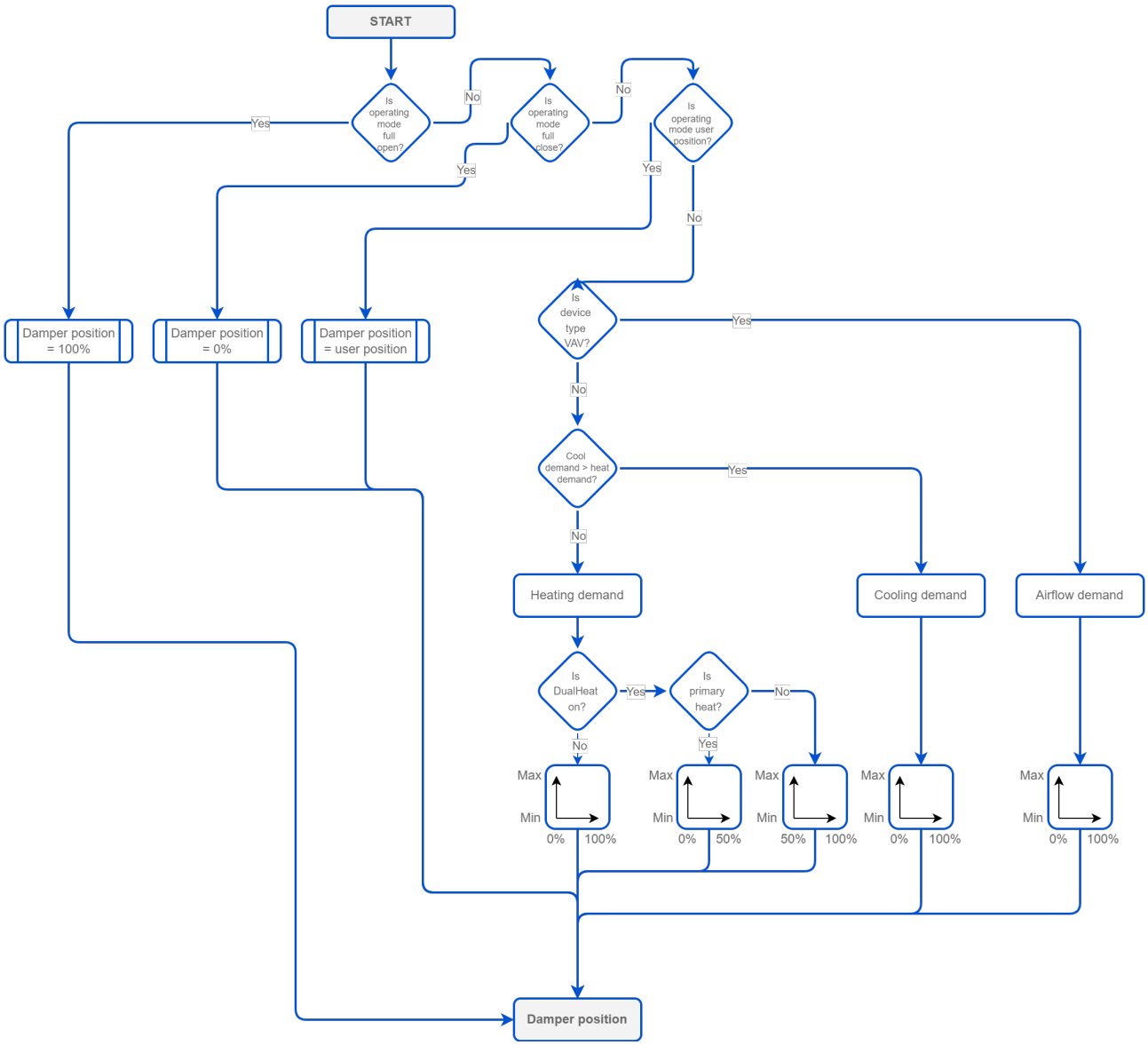


Figure 22. Damper control

Damper Operating Modes

The damper can be configured to work in the following operating modes:

- **Auto:** the damper is controlled according to the airflow control loop output (for VAV device type) or temperature control loops (for VVT device type),
- **Full Open:** the damper control is forced to 100% open position,
- **Full Close:** the damper control is forced to 0% open position (closed), even when the MinDamperPosition value is higher than 0%,
- **User Position:** the damper control is forced to the user set position (the UserSetPosition variable),
- **Min Flow:** the damper control is forced to fit the set minimum airflow (set in the the MinOccCoolAirflowSetpoint),
- **Max Flow:** the damper control is forced to fit the set maximum airflow (set in the the MaxOccCoolAirflowSetpoint),
- **User Flow:** the damper control is forced to fit the set user airflow (set in the UserSetAirflow),
- **Calibrate:** forces the damper calibration (the damper goes to 100%, then to 0%, and goes back to the control loop output).

Initial Position

The Initial Position function is an auto-synchronization of the actuator, which allows for maintaining proper airflow parameters. In the process, the damper goes to 0% and goes back to the control loop output. It is performed each first day of the month, on the time calculated by the formula:

Calibration time = 00:00 + (serial number MOD 100)

This way helps to avoid the situation when all controllers perform the process at one moment, compromising the stability of the system.

DamperResponse

In case of significant fluctuations of airflow demand (resulting from the respective control loop), it is possible to mitigate the changes of value using the DamperResponse variable. It allows to enforce the change of value on the DamperControl component if the change of output demand is higher than set in the DamperResponse In16 slot (by default, 2%). It means that the DamperControl output will change only if the heating/cooling or airflow demand changes more than 2%.

DualHeat

The DualHeat function for damper allows to meet energy efficiency requirements by allowing to select between two modes of heating, primary or secondary. The damper is then controlled according to the setting of the HeatDamperPriority variable. The primary and secondary mode of heating work depending on the type of device:

- VVT:
  - primary mode: the damper is controlled in a full range of 0-100% at the corresponding heating demand from 0-50%,
  - secondary mode: the damper is controlled in a full range of 0-100% at the corresponding heating demand from 50-100%;
- VAV:
  - primary mode: the damper is controlled in a full range of 0-100% at the corresponding airflow for heating demand from 0-50%,
  - secondary mode: the damper is controlled in a full range of 0-100% at the corresponding airflow for heating demand from 50-100%.

Fan Control

Fan control in the VAV application is carried out based on the type of fan installed in the system and occupancy status:

- **series fan:** works only as ventilator regardless of the heating/cooling demand, in occupied or standby statuses (standby only if the StandbySeriesFan variable is activated),
- **parallel fan:** works on heating demand in the HVAC mode set to Auto or Heat, regardless of the occupancy status.

For fan configuration methods, see the Application Configuration section.

Each type of fan can controlled in digital and analog mode set in the FanControlMode variable:

- **digital (TO4):** on/off,
- **analog (AO3):** min./max. speed (binary mode available in occupied, bypass and, optionally, standby statuses) or linearly according to the heating/cooling demand and within limit speeds set in the MinFanSpeed (by default, 20%) and MaxFanSpeed (by default, 100%) components.

Depending on the HVAC Mode settings (AirflowSetpointCalculator component):

- Auto:
  - parallel fan: working on heating demand,
  - series fan: working if occupied/bypass or standby (if allowed);
- Heat:
  - parallel fan: working on heating demand,
  - series fan: working only on heating demand if occupied/bypass or standby (if allowed);
- Cool:
  - parallel fan: inactive,
  - series fan: working only on cooling demand if occupied/bypass or standby (if allowed);
- MorningWarmUp:
  - parallel fan: if the WarmUpParallelFan variable is set to true, the parallel fan is active in this mode,
  - series fan: ventilator set to the maximum speed;
- PreCool:
  - parallel fan: inactive,
  - series fan: ventilator set to the maximum speed;
- NightPurge:
  - parallel fan: inactive,
  - series fan: ventilator set to the maximum speed;
- Off: both types of fan disabled;
- Fire:
  - parallel fan: inactive,
  - series fan: set to the maximum speed.

For safety reasons, the fan is started with a delay set in the FanDelayOff variable (by default, 30 s), which purpose is to cool down the reheater (for systems with a water reheater, the delay may be cancelled and set to 0 s).

The fan can work in four modes defined in the FanCommand variable:

- **Auto:** fan works according to the demand and type of the fan,
- **MaxSpeed:** fan works with the maximum speed set in the Max Fan Speed slot,
- **MinSpeed:** fan works with the minimum speed set in the Mix Fan Speed slot,
- **Stop:** fan stops working.

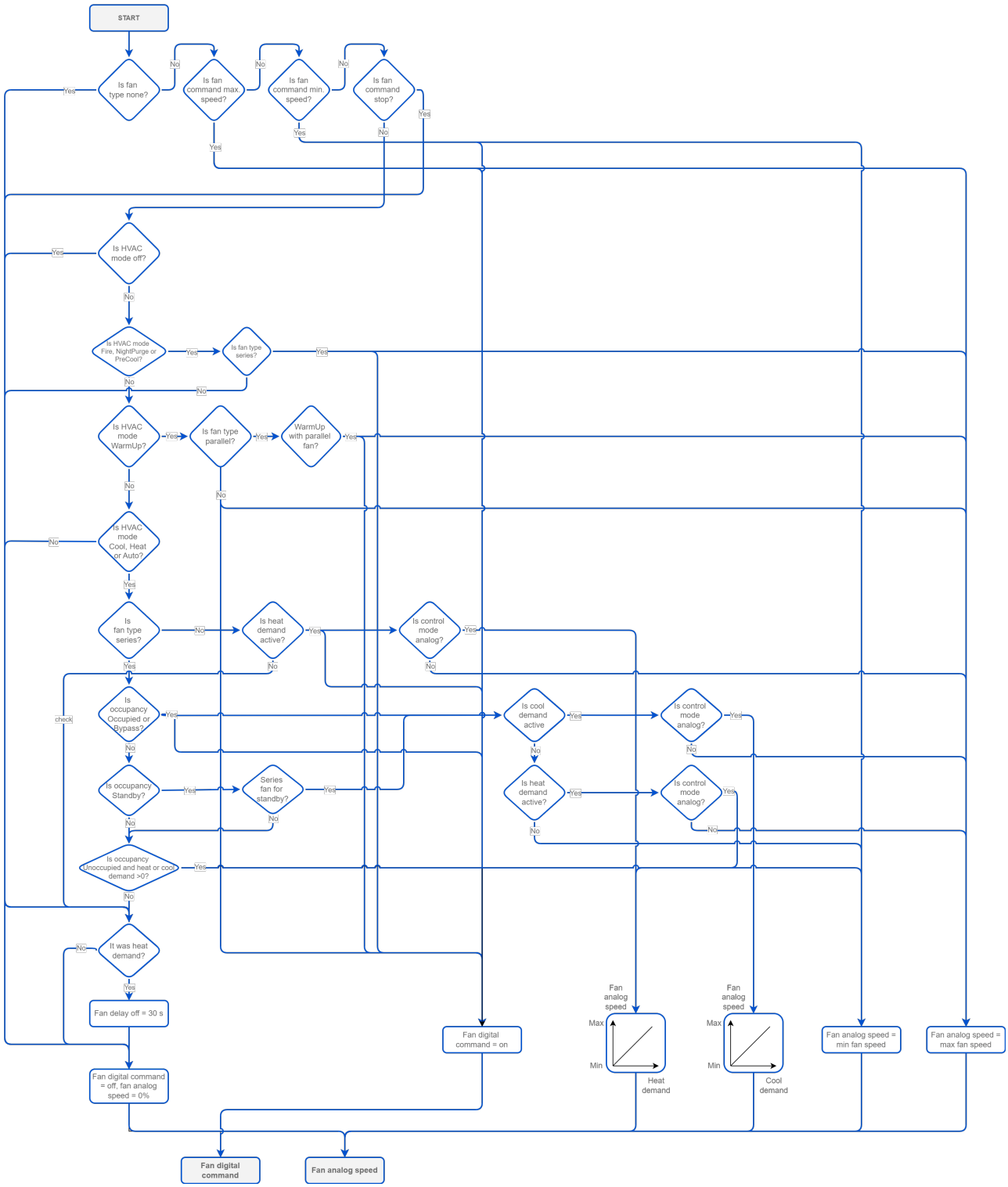


Figure 23. Fan control

Reheater Control

The VAV application is designed to work with either water or electric reheaters. For safety reasons, reheater control is possible only for the minimum airflow set in the MinOccupiedHeatAirFlow variable. Reheater auto control is based on the heating demand from the discharge air temperature sensor or the space temperature sensor (if no discharge air temperature sensor is in place).

If the discharge air temperature sensor is available (required in systems with an electric reheater), the reheater is controlled based on the discharge air temperature control loop (in full range of the loop), taking into account the sensor measurements and setpoint calculated based on the space temperature control loop, up to the maximum set limit (valid for versions with electric reheater), or as the first (or second) heating stage (in half range of the loop).

If there is no discharge air temperature sensor, the reheater is controlled based on the main heating temperature control loop directly – in full range of the loop – or as the first (or second) heating stage (in half range of the loop).

Additionally, if used simultaneously with a perimeter, it can be specified whether both heaters should work together simultaneously or which should be the first and second stage of the primary/secondary heating level. In version with a parallel fan, the reheater is always the second heating stage (the fan is the first).

Note

An additional factor in the reheater auto control is the level of energy load cut-off, which decreases the value of the reheater control output:
RH control output = (100% - LoadShedding)*HeatDemand

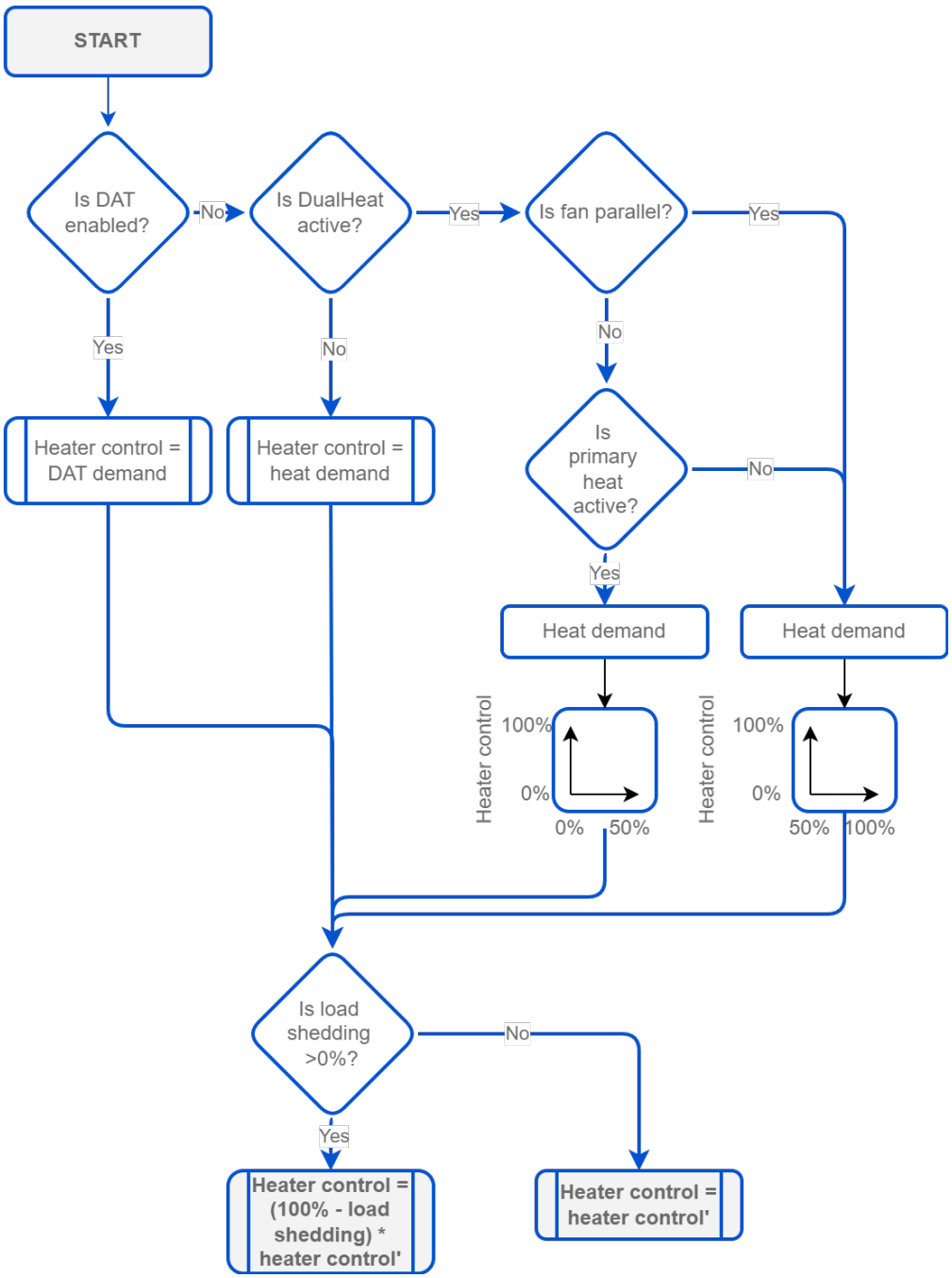
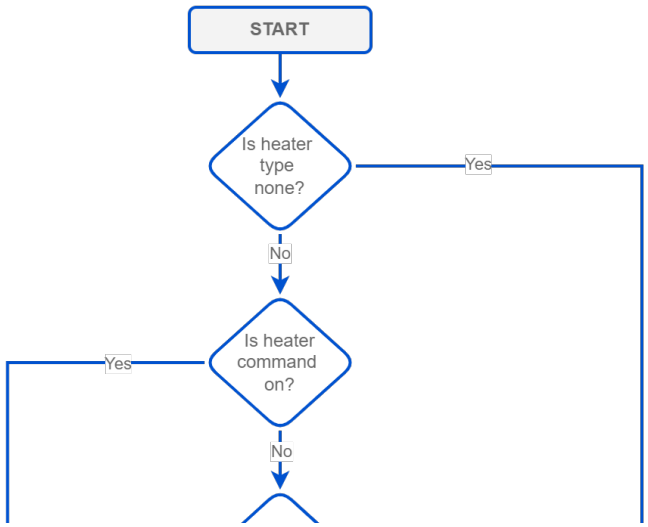


Figure 24. Heater control with or without the DualHeat function



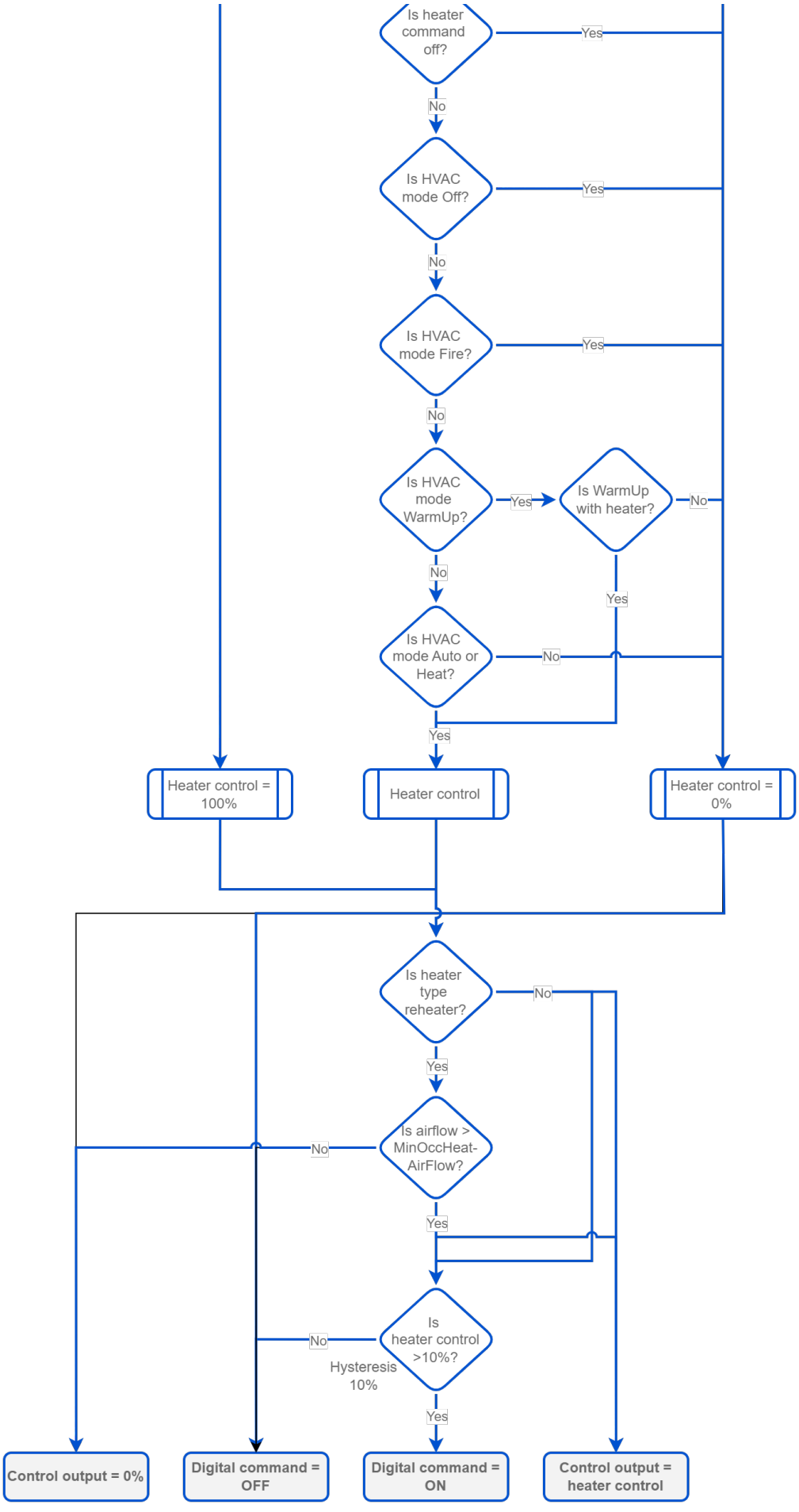

Is heater command off?
Yes
No
Is HVAC mode Off?
Yes
No
Is HVAC mode Fire?
Yes
No
Is HVAC mode WarmUp?
Yes
Is WarmUp with heater?
No
Yes
No
Is HVAC mode Auto or Heat?
No
Yes
Heater control = 100%
Heater control
Heater control = 0%
Is heater type reheater?
No
Yes
Is airflow > MinOccHeat-AirFlow?
No
Yes
Is heater control >10%?
No
Hysteresis 10%
Yes
Control output = 0%
Digital command = OFF
Digital command = ON
Control output = heater control

Figure 25. Reheater control

Reheater Control According to HVAC Modes

Depending on the HVAC Mode settings (AirflowSetpointCalculator component):

- **Auto:** working according to the heating demand,
- **Heat:** working according to the heating demand,
- **Cool:** switched off,
- **MorningWarmUp:** if the WarmUpHeater variable is set to true, the reheater works according to the heating demand,
- **PreCool:** switched off,
- **NightPurge:** switched off,
- **Off:** switched off,
- **Fire:** switched off.

The reheater can work in three modes defined in the ReheaterCommand variable:

- **Auto:** reheater works based on the heating demand,
- **Full Open:** reheater is set to the maximum control output, 100%,
- **Full Close:** reheater is set to the minimum control output, 0%.

DualHeat

The DualHeat function for reheater allows to meet energy efficiency requirements by allowing to select between two modes of heating. The DualHeat function is activated according to the setting of the HeatDamperPriority variable:

- primary stage (true): heater(s) works in the range of 0-50% (indirectly from the discharge air temperature heating loop and directly from the space temperature heating loop),
- secondary stage (false): damper in the range of 50-100% (directly from the space temperature heating loop).

The heaters priority is set in the HeaterPriority variable (ReheaterControl folder of the VAV application):

- HeaterPriority set to reheater:
  - primary mode: reheater in the range of 0-50% (indirectly from the discharge air temperature heating loop or directly from the space temperature heating loop),
  - secondary mode: perimeter in the range of 50-100% (directly from the space temperature heating loop);
- HeaterPriority set to perimeter:
  - primary mode: perimeter in the range of 0-50% (directly from the space temperature heating loop),
  - secondary mode: reheater in the range of 50-100% (indirectly from the discharge air temperature heating loop or directly from the space temperature heating loop);
- HeaterPriority set to simultaneous:
  - reheater and perimeter work in full range of the heating loop.

> **Note**
> If the DualHeat is active and the fan is of parallel type, the fan is always the primary heating and the reheater is the second.

Reheater Types

Depending on the configuration (see: Methods of Configuration), the reheater can be controlled by triac outputs (TO1, TO2) as follows:

- time-proportional (PWM) water reheater with a valve and actuator or time-proportional (PWM) electric reheater with a period set in the TimePropReheaterPeriod variable, controlled by the TO1 output,
- water reheater with a valve and actuator controlled as floating type with a full-position switch time set in the FloatReheaterValveDriveTime variable, controlled by the TO1 output for closing and TO2 output for opening,

- electric reheater controlled in one or two heating stages, on/off type (the second stage switched on above the 50% heating demand), controlled by TO1 output for 1-stage heating and TO2 output for 2-stage heating,
- water reheater with a valve and actuator, on/off type, controlled by the TO1 output.

Regardless of the configuration of the control type, the reheater is simultaneously controlled in analog mode by the AO1 output.

If the outside temperature signal is connected (from the BACnet/Modbus network), the reheater is blocked over the setpoint in the MaxOatReheater variable (by default, 32°C/90°F).

Perimeter Control

The perimeter is an additional heating source, independent from the ventilation system, for example, an underfloor heating. If configured in the VAV application, perimeter control does not require checking of the minimum airflow as it is not a part of the ventilation duct. The perimeter control in the auto mode is based on the heating demand from the space temperature control loop.

> **Note**
>
> An additional factor in the perimeter auto control is the level of energy load cut-off, which decreases the value of the perimeter control output:
> PM control output = (100% - LoadShedding)*HeatDemand

Perimeter Control According to HVAC Modes

Depending on the HVAC Mode settings (AirflowSetpointCalculator folder):

- **Auto:** working according to the heating demand,
- **Heat:** working according to the heating demand,
- **Cool:** switched off,
- **MorningWarmUp:** if the WarmUpHeater variable is set to true, the perimeter works according to the heating demand,
- **PreCool:** switched off,
- **NightPurge:** switched off,
- **Off:** switched off,
- **Fire:** switched off,

The perimeter can work in three modes defined in the ReheaterCommand variable:

- **Auto:** perimeter works based on the heating demand,
- **Full Open:** perimeter is set to the maximum control output, 100%,
- **Full Close:** perimeter is set to the minimum control output, 0%.

DualHeat

The DualHeat function for perimeter allows to meet energy efficiency requirements by allowing to select between two modes of heating. The DualHeat function is activated according to the setting of the HeatDamperPriority variable:

- primary stage (true): heater(s) works in the range of 0-50% (if DAT sensor is available: from the discharge air temperature heating loop with DAT setpoint based on the space temperature loop, or if DAT sensor is nor available: directly from the space temperature heating loop),
- secondary stage (false): damper in the range of 50-100% (directly from the space temperature heating loop).

The heaters priority is set in the HeaterPriority variable (PerimeterControl folder of the VAV application):

- HeaterPriority set to reheater:
  - primary mode: reheater in the range of 0-50% (indirectly from the discharge air temperature heating loop or directly from the space temperature heating loop),
  - secondary mode: perimeter in the range of 50-100% (directly from the space temperature heating loop);
- HeaterPriority set to perimeter:
  - primary mode: perimeter in the range of 0-50% (directly from the space temperature heating loop),
  - secondary mode: reheater in the range of 50-100% (indirectly from the discharge air temperature heating loop or directly from the space temperature heating loop);
- HeaterPriority set to simultaneous:
  - reheater and perimeter work in full range of the heating loop.

Perimeter Types

Depending on the configuration (see: Methods of Configuration), the perimeter can be controlled by the triac output (TO3) as follows:

- time-proportional (PWM) water perimeter with a valve and actuator, controlled by the TO3 output,
- water perimeter with a valve and actuator, on/off type, controlled by the TO3 output.

Regardless of the configuration of the control type, the perimeter is simultaneously controlled in analog mode by the AO2 output.

If the outside temperature signal is connected (from the BACnet/Modbus network), the perimeter is blocked over the setpoint in the MaxOatPerimeter variable (by default, 18°C/65°F).

Highlights

HVAC Mode Dependencies

The HVAC Mode settings in the AirflowSetpointControl folder affects a lot of factors in the VAV application. The below tables offers a basic summary.

HVAC Mode	Effective setpoint	Space temp. control loop	DAT control loop	Airflow setpoint	Fan control	Reheater control	Perimeter control
Auto	Heating demand active: effective setpoint equals the heating setpoint Cooling demand active: effective setpoint equals the cooling setpoint	Cooling CL active	Heating CL active: • DAT > HotAirTempSetpoint • the above temp. is maintained > ChangeoverDelay Cooling CL active: • DAT < HotAirTempSetpoint • the above temp. is maintained > ChangeoverDelay	Occupied and bypass: • heating: linear function between the MaxOccHeatAirFlowSetpoint and MinOccHeatAirFlowSetpoint • cooling: linear function between theMaxOccCoolAirFlowSetpoint and MinOccCoolAirFlowSetpoint Standby (StandbyAirFlowSetpoint) and unoccupied (UnoccAirFlowSetpoint): the same setpoint for heating and cooling	Parallel fan: working on heating demand Series fan: working if occupied or standby (if allowed)	According to the heating demand	According to the heating demand

Heat	Occupied: OccHeatTempSetpoint Bypass: OccHeatTempSetpoint Standby: StandbyHeatTempSetpoint Unoccupied: UnoccHeatTempSetpoint	Heating CL active	Heating CL active	Occupied and bypass: linear function between the MaxOccHeatAirFlowSetpoint and MinOccHeatAirFlowSetpoint values Standby (StandbyAirFlowSetpoint): setpoint the same as the auto mode Unoccupied (UnoccAirFlowSetpoint) and	Parallel fan: working on heating demand Series fan: working only on heating demand if occupied or standby (if allowed)	According to the heating demand	According to the heating demand
Cool	Occupied: OccCoolTempSetpoint Bypass: OccCoolTempSetpoint Standby: StandbyCoolTempSetpoint Unoccupied: UnoccCoolTempSetpoint	Cooling CL active	Cooling CL active	Occupied and bypass: linear function between the MaxOccCoolAirFlowSetpoint and MinOccCoolAirFlowSetpoint values Standby (StandbyAirFlowSetpoint): setpoint the same as the auto mode Unoccupied (UnoccAirFlowSetpoint): setpoint the same as the auto mode	Parallel fan: inactive Series fan: working only on cooling demand if occupied or standby (if allowed)	Off	Off
Morning WarmUp	Occupied: OccHeatTempSetpoint Bypass: OccHeatTempSetpoint Standby: StandbyHeatTempSetpoint Unoccupied: UnoccHeatTempSetpoint	Heating CL active	Heating CL active	MaxOccHeatAirFlowSetpoint	Parallel fan: if the WarmUpParallelFan variable is set to true, the parallel fan is active in this mode Series fan: ventilator set to the maximum speed	According to the heating demand (if WarmUpHeater is true)	According to the heating demand (if WarmUpHeater is true)

PreCool	Occupied: OccCoolTempSetpoint Bypass: OccCoolTempSetpoint Standby: StandbyCoolTempSetpoint Unoccupied: UnoccCoolTempSetpoint	Cooling CL active	Cooling CL active	MaxOccCoolAirFlowSetpoint	Parallel fan: inactive Series fan: ventilator set to the maximum speed	Off	Off
Night Purge	Occupied: OccCoolTempSetpoint Bypass: OccCoolTempSetpoint Standby: StandbyCoolTempSetpoint Unoccupied: UnoccCoolTempSetpoint	Cooling CL active	Cooling CL active	MaxOccCoolAirFlowSetpoint	Parallel fan: inactive Series fan: ventilator set to the maximum speed	Off	Off
Fire	Occupied: OccCoolTempSetpoint Bypass: OccCoolTempSetpoint Standby: StandbyCoolTempSetpoint Unoccupied: UnoccCoolTempSetpoint	Disabled	Disabled	MaxOccCoolAirFlowSetpoint	Disabled	Off	Off
Off	Occupied: OccCoolTempSetpoint Bypass: OccCoolTempSetpoint Standby: StandbyCoolTempSetpoint Unoccupied: UnoccCoolTempSetpoint	Disabled	Disabled	0	Off	Off	Off

4.1.4 Configuration Variants

- VAV Cooling Only
- VAV Cooling/Heating
- VAV with Water Reheater and Optional Perimeter
- VAV with up to 2-stage Electric Reheater and Optional Perimeter
- Series Fan Powered VAV Cooling with Water Reheater and Optional Perimeter
- Series Fan Powered VAV Cooling with up to 2-stage Electric Reheater and Optional Perimeter
- Parallel Fan Powered VAV Cooling with Water Reheater and Optional Perimeter
- Parallel Fan Powered VAV Cooling with up to 2-stage Electric Reheater and Optional Perimeter

VAV Cooling Only

The VAV cooling application is a configuration factory default; it does not require any further configuration of the reheater, perimeter, or fan. Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

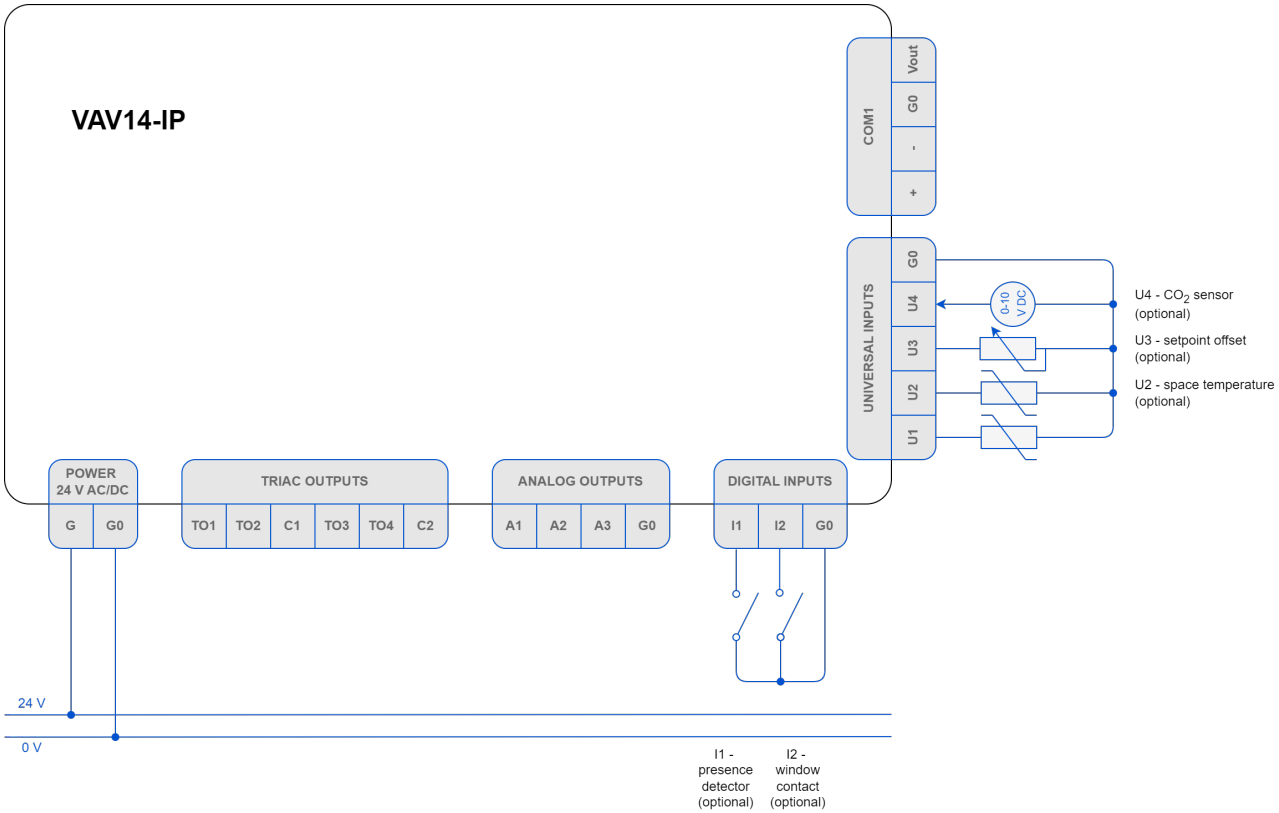


Figure 26. VAV cooling - application connections diagram

The Control Point VAV panel or sensors on inputs (space temperature, setpoint offset, CO_2 sensor) are required for the application to operate properly. Optionally, it is possible to connect the presence detector and window contact sensors to minimize energy efficiency loss.

VAV Cooling/Heating

The VAV cooling/heating application does not require any further configuration of the reheater, perimeter, or fan. **To enable the variant, use factory default configuration of the VAV cooling and connect the discharge air temperature (DAT) sensor.** Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

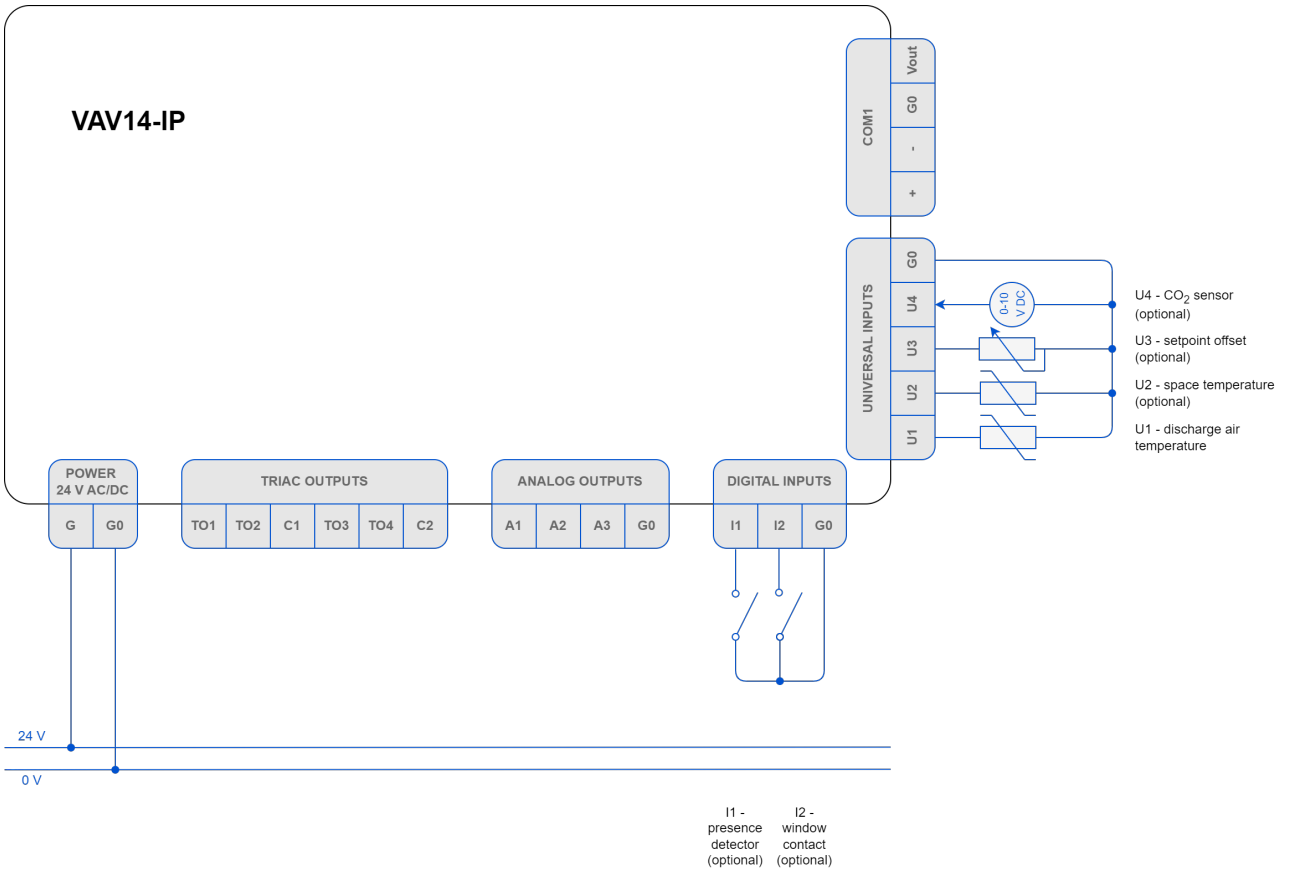


Figure 27. VAV cooling/heating - application connections diagram

For the application to operate properly, it is required to connect the following elements:

- Control Point VAV panel or sensors on inputs (space temperature, setpoint offset, CO_2 sensor), and
- DAT sensor.

Optionally, it is possible to connect the presence detector and window contact sensors to minimize energy efficiency loss.

VAV with Water Reheater and Optional Perimeter

The VAV cooling with water reheater and optional perimeter application requires to configure the following parameters:

- reheater's control type (time proportional - PWM, digital, or floating),
- (if applicable) perimeter's control type (time proportional - PWM, on/off).

Configuring of any type of the reheater or perimeter automatically activates an analog control mode, so it does not require any further configuration.

Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

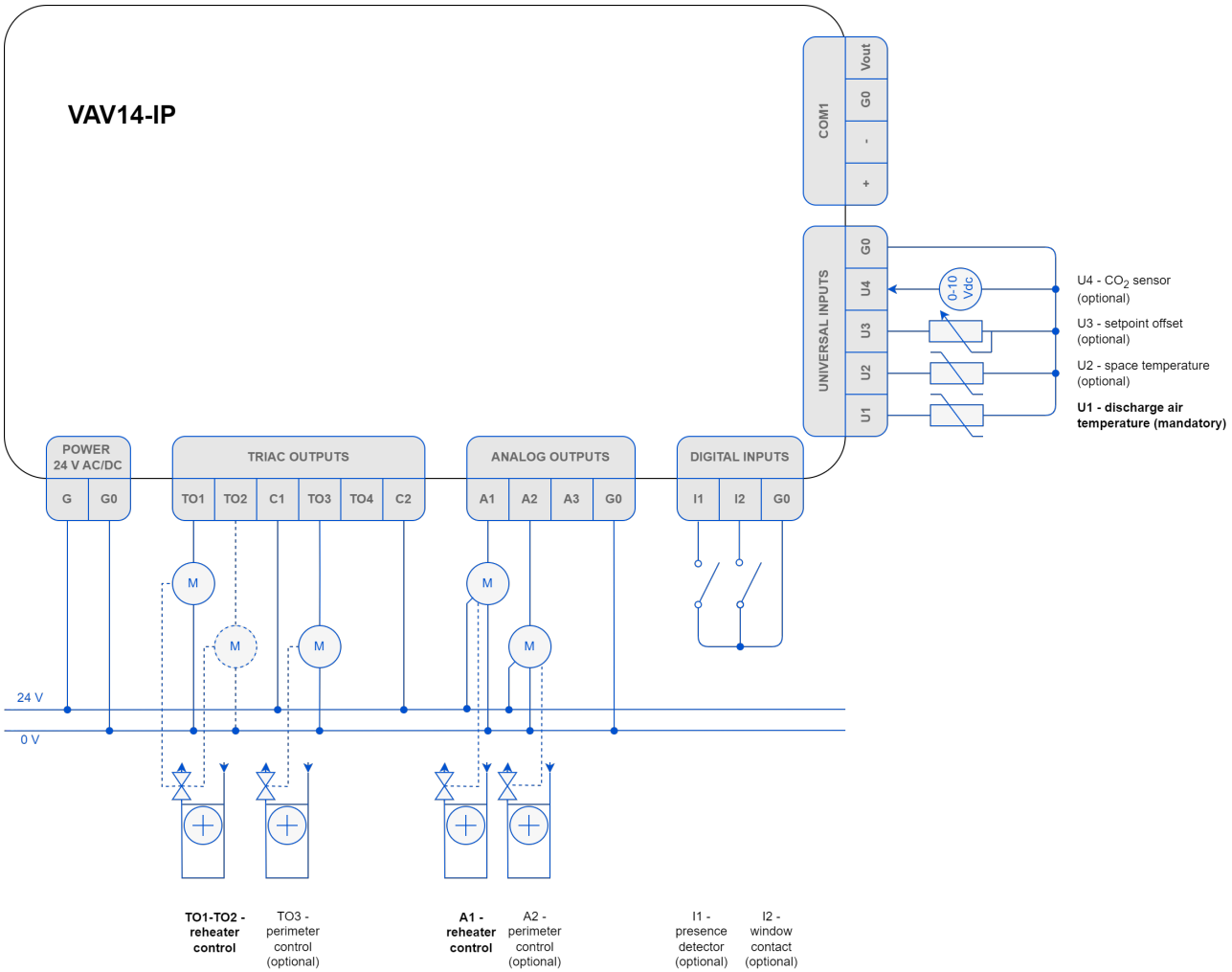


Figure 28. VAV cooling with water reheater and optional perimeter - application connections diagram

For the application to operate properly, it is required to connect the following elements:

- Control Point VAV panel or sensors on inputs (space temperature, setpoint offset, CO_2 sensor), and
- DAT sensor located in the supply air duct before VAV.

Optionally, it is possible to connect the presence detector and window contact sensors to minimize energy efficiency loss. To control the reheater and, optionally, perimeter, dedicated triac and analog output are available, as depicted on the diagram above.

VAV with up to 2-stage Electric Reheater and Optional Perimeter

The VAV cooling with up to 2-stage electric reheater and optional perimeter application requires to configure the following parameters:

- reheater's control type:
  - 1-stage reheater: time proportional (PWM), digital (on/off),
  - 2-stage reheater: digital (on/off);
- (if applicable) perimeter's control type (time proportional - PWM, on/off).

Configuring of any type of the reheater or perimeter automatically activates an analog control mode, so it does not require any further configuration.

Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

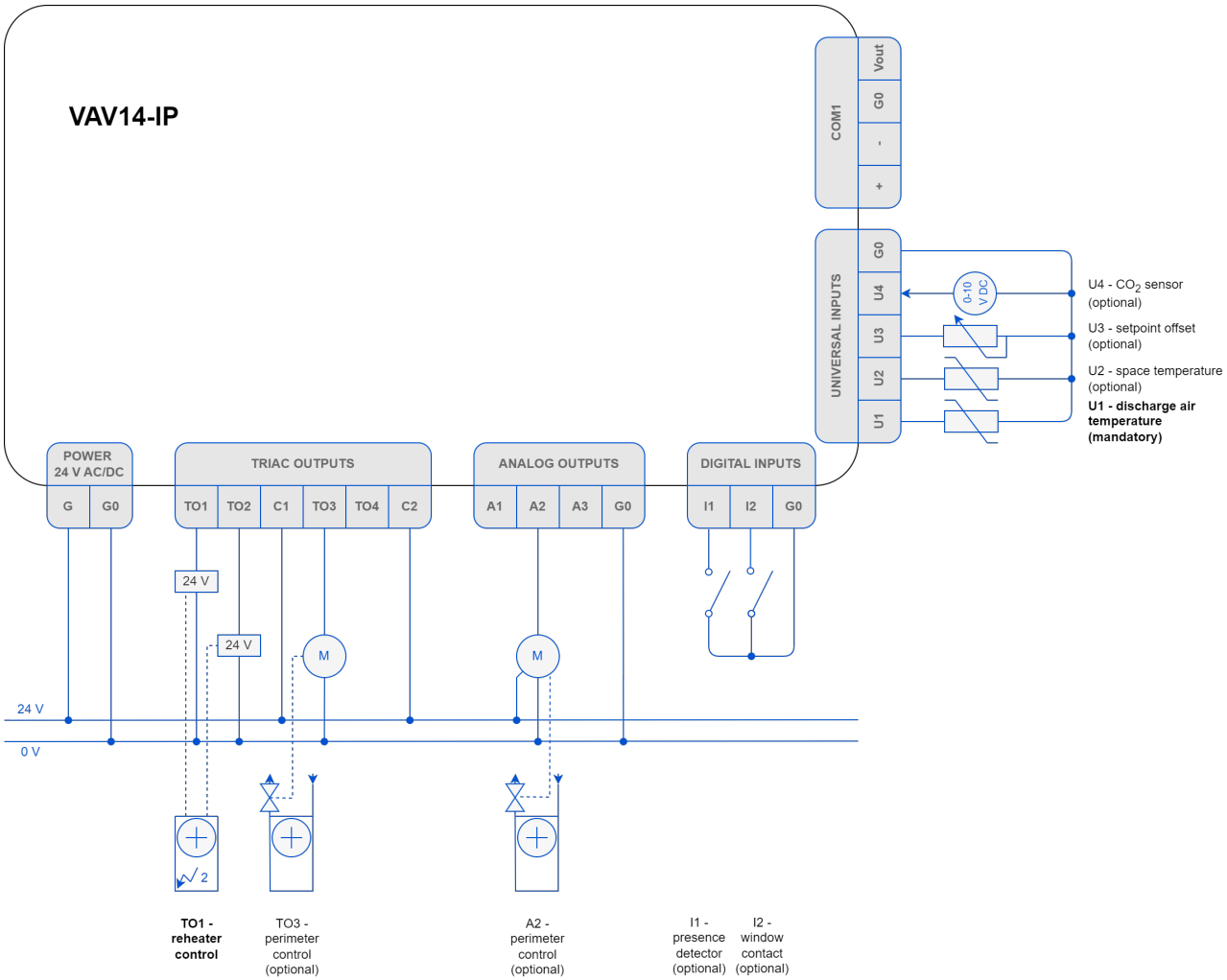


Figure 29. VAV cooling with up to 2-stage electric reheater and optional perimeter - application connections diagram

For the application to operate properly, it is required to connect the following elements:

- Control Point VAV panel or sensors on inputs (space temperature, setpoint offset, CO_2 sensor), and
- DAT sensor located in the supply air duct before VAV.

Optionally, it is possible to connect the presence detector and window contact sensors to minimize energy efficiency loss. To control the reheater and, optionally, perimeter, dedicated triac and analog output are available, as depicted on the diagram above.

Series Fan Powered VAV Cooling with Water Reheater and Optional Perimeter

The series fan powered VAV cooling with water reheater and optional perimeter application requires to configure the following parameters:

- fan's control mode (series),
- reheater's control type (time proportional – PWM, digital, or floating),
- (if applicable) perimeter's control type (time proportional - PWM, on/off).

Configuring of any type of the fan, reheater, or perimeter automatically activates an analog control mode, so it does not require any further configuration.

Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

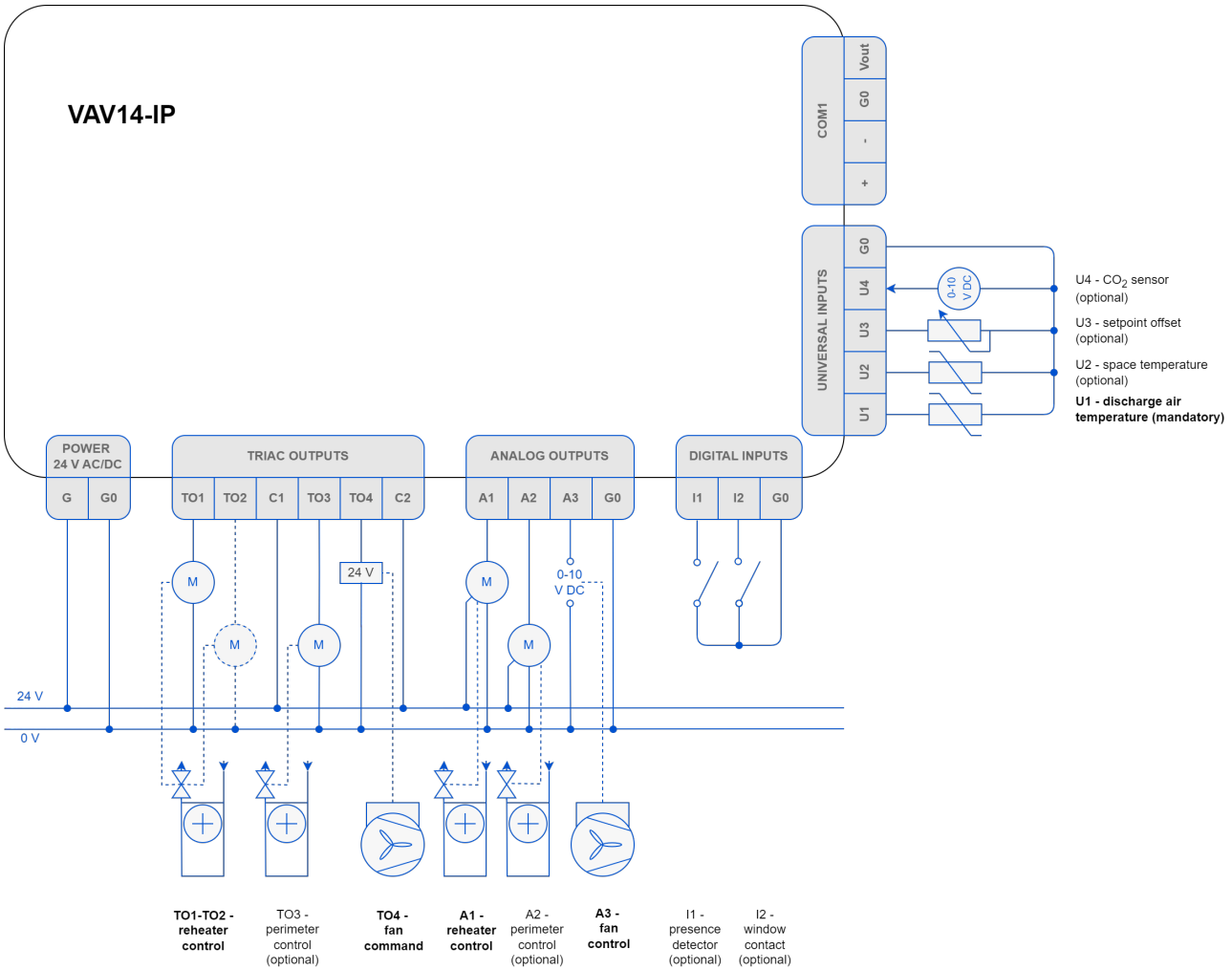


Figure 30. Series fan powered VAV cooling with water reheater and optional perimeter - application connections diagram

Series Fan Powered VAV Cooling with up to 2-stage Electric Reheater and Optional Perimeter

The series fan powered VAV cooling with up to 2-stage electric reheater and optional perimeter application requires to configure the following parameters:

- fan's control mode (series),
- reheater's control type:
  - 1-stage reheater: time proportional (PWM), digital (on/off),
  - 2-stage reheater: digital (on/off);
- (if applicable) perimeter's control type (time proportional - PWM, on/off).

Configuring of any type of the fan, reheater, or perimeter automatically activates an analog control mode, so it does not require any further configuration.

Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

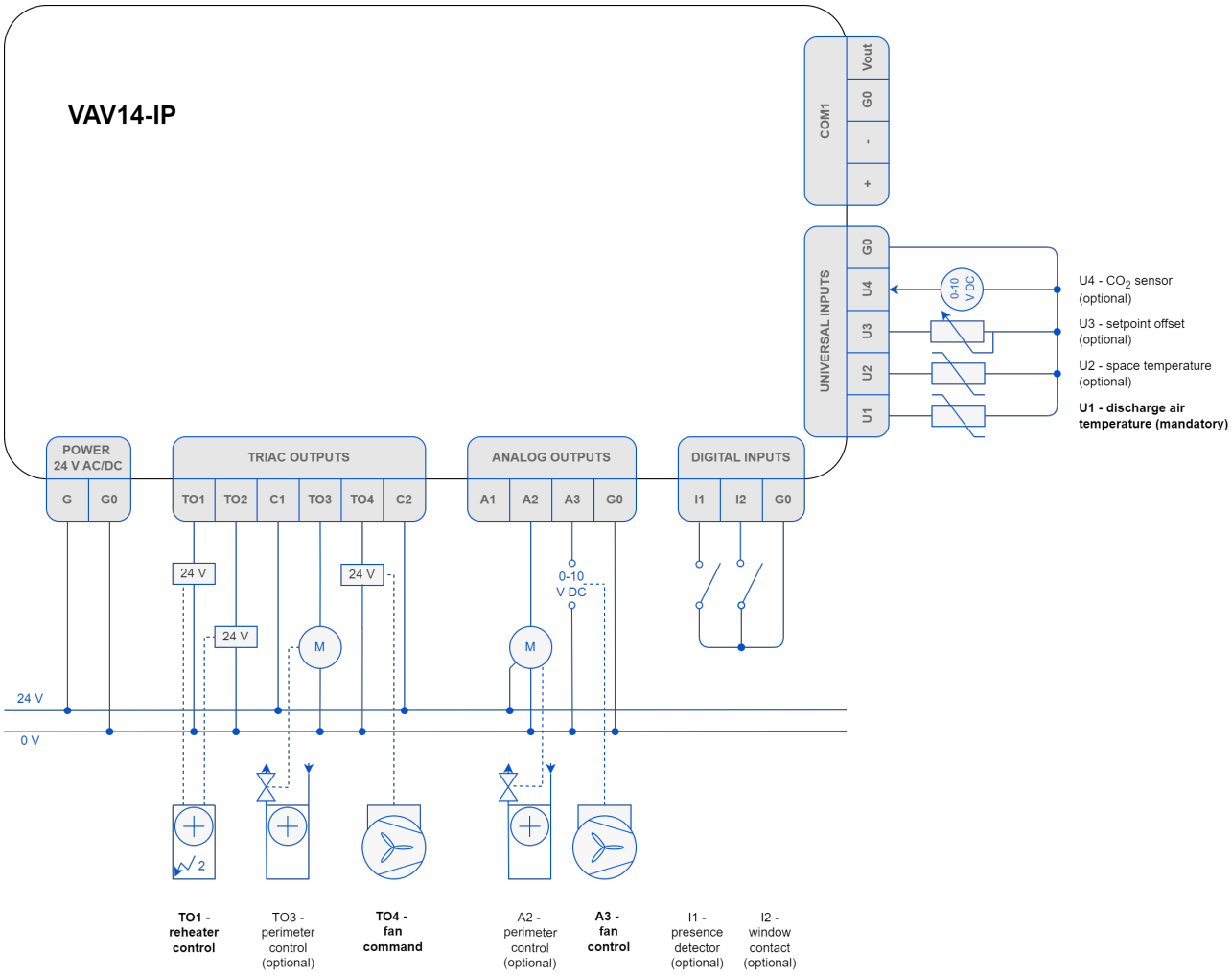


Figure 31. Series fan powered VAV cooling with up to 2-stage electric reheater and optional perimeter - application connections diagram

For the application to operate properly, it is required to connect the following elements:

- Control Point VAV panel or sensors on inputs (space temperature, setpoint offset, CO_2 sensor), and
- DAT sensor located in the supply air duct before VAV.

Optionally, it is possible to connect the presence detector and window contact sensors to minimize energy efficiency loss. To control the reheater and, optionally, perimeter, dedicated triac and analog output are available, as depicted on the diagram above.

Parallel Fan Powered VAV Cooling with Water Reheater and Optional Perimeter

The parallel fan powered VAV cooling with water reheater and optional perimeter application requires to configure the following parameters:

- fan's control mode (parallel),
- reheater's control type (time proportional – PWM, digital, or floating),
- (if applicable) perimeter's control type (time proportional - PWM, on/off).

Configuring of any type of the fan, reheater, or perimeter automatically activates an analog control mode, so it does not require any further configuration.

Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in

the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

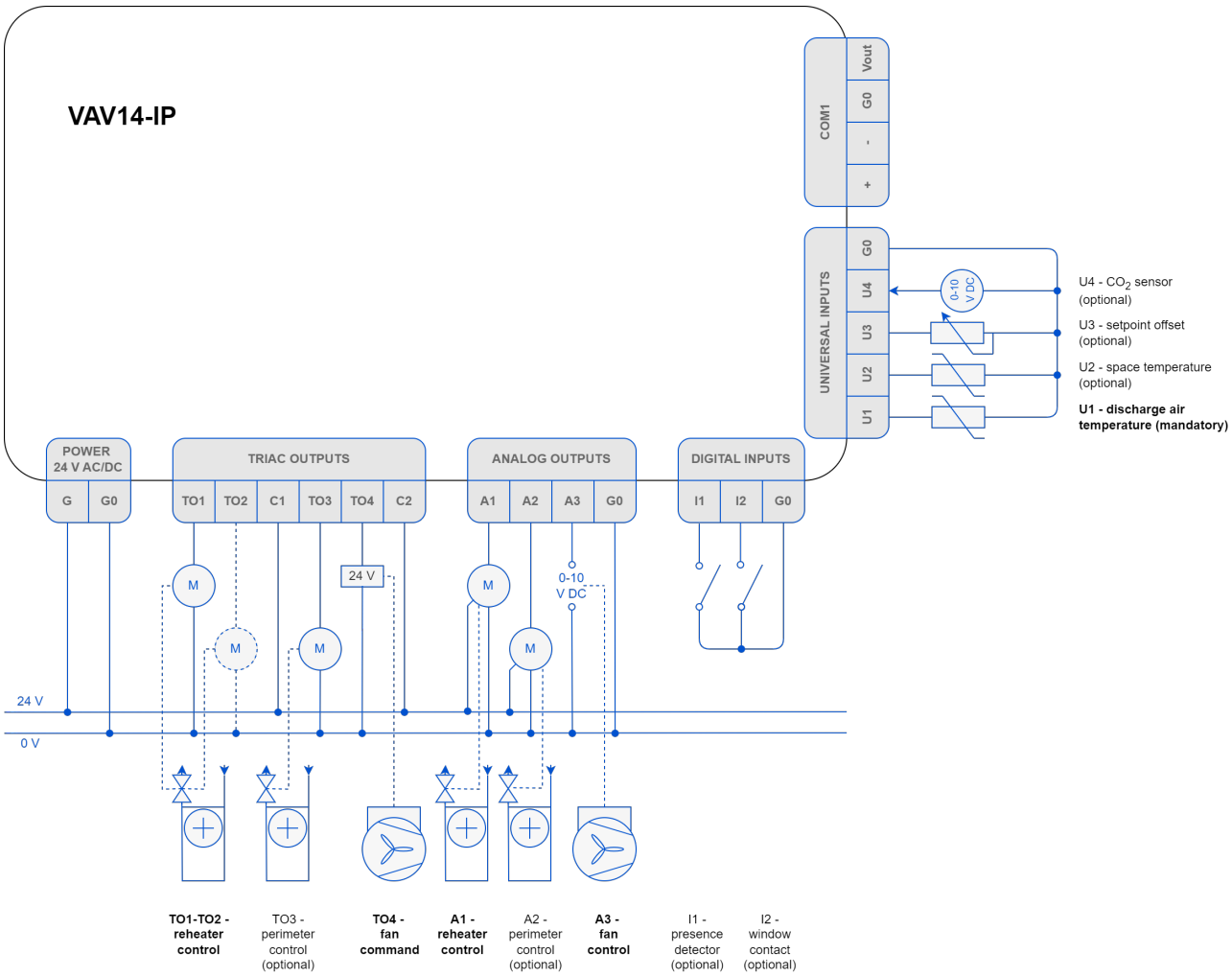


Figure 32. Parallel fan powered VAV cooling with water reheater and optional perimeter - application connections diagram

For the application to operate properly, it is required to connect the following elements:

- Control Point VAV panel or sensors on inputs (space temperature, setpoint offset, CO_2 sensor), and
- DAT sensor located in the supply air duct before VAV.

Optionally, it is possible to connect the presence detector and window contact sensors to minimize energy efficiency loss. To control the reheater and, optionally, perimeter, dedicated triac and analog output are available, as depicted on the diagram above.

Parallel Fan Powered VAV Cooling with up to 2-stage Electric Reheater and Optional Perimeter

The parallel fan powered VAV cooling with up to 2-stage electric reheater and optional perimeter application requires to configure the following parameters:

- fan's control mode (parallel),
- reheater's control type:
  - 1-stage reheater: time proportional (PWM), digital (on/off),
  - 2-stage reheater: digital (on/off);
- (if applicable) perimeter's control type (time proportional - PWM, on/off).

Configuring of any type of the fan, reheater, or perimeter automatically activates an analog control mode, so it does not require any further configuration.

Other configuration parameters (such as device type – VAV or VVT, damper direction – CW or CCW, default sensor source, type and range or polarity, dual heating function, or heater priority) are available in the Control Point VAV panel or software programming tools (iSMA Configurator/iC Device Manager module/iC Tool).

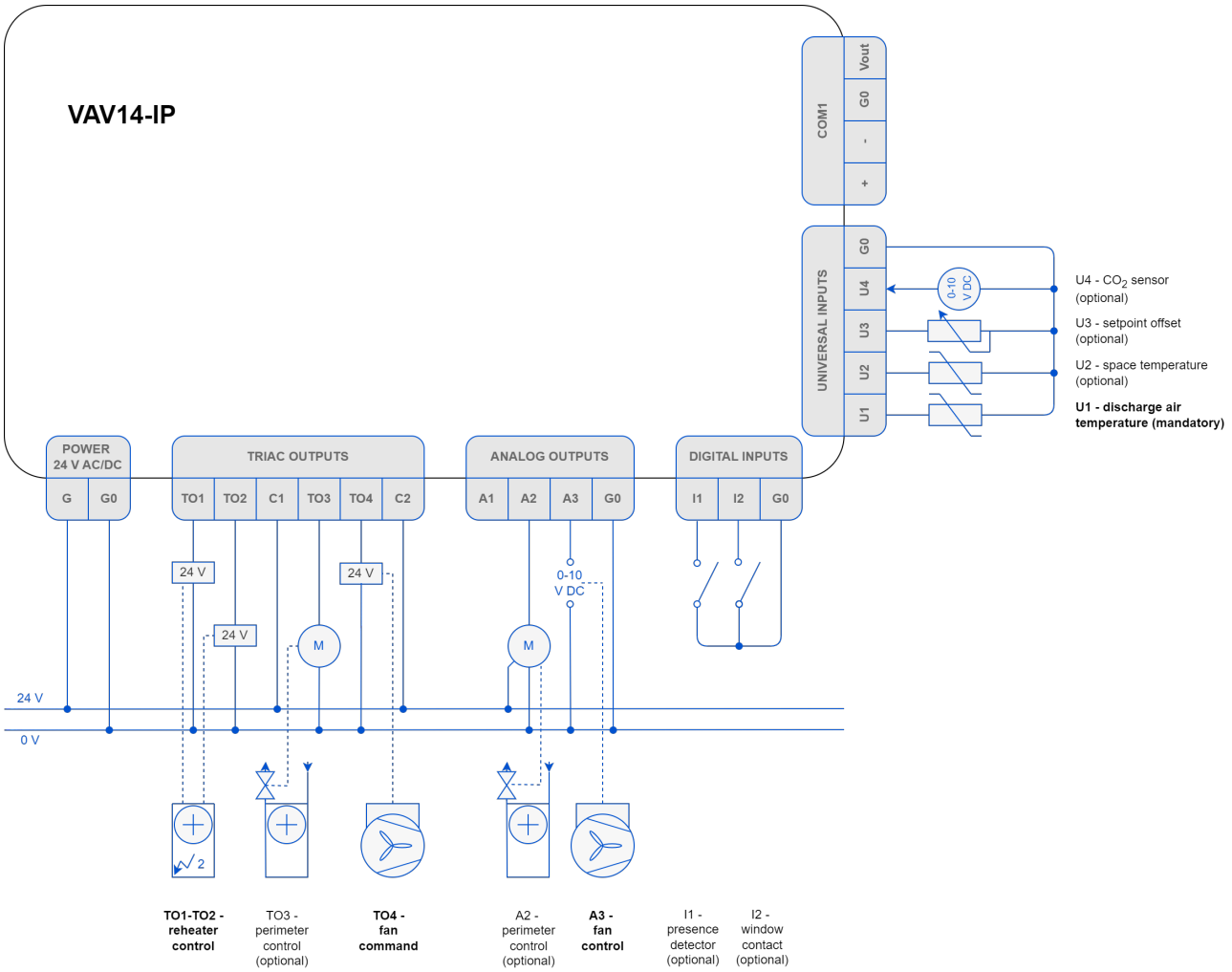


Figure 33. Parallel fan powered VAV cooling with up to 2-stage electric reheater and optional perimeter - application connections diagram

For the application to operate properly, it is required to connect the following elements:

- Control Point VAV panel or sensors on inputs (space temperature, setpoint offset, CO_2 sensor), **and**
- DAT sensor located in the supply air duct before VAV.

Optionally, it is possible to connect the presence detector and window contact sensors to minimize energy efficiency loss. To control the reheater and, optionally, perimeter, dedicated triac and analog output are available, as depicted on the diagram above.

4.1.5 Methods of Configuration

Configuration Sources Priority

Configuration	S3 DIP Switch	OFF (0)	ON (1)
Configuration priority	Switch 1	CP/Software (BACnet/Modbus/iC Configurator/iC Device Manager)	DIP switch

Please note that if the switch 1 on the S3 DIP switch is on, than configuration from any other source is blocked, for example, any changes made in the iSMA Configurator will not be written to the device.

Configuration by Type of Device

Configuration parameters	S3 DIP switch	iSMA Configurator	Control Point VAV	BACnet objects/Modbus registers
Reheater type	Switch 2 and 3	☑	☑	MSV12/40073
Fan type	Switch 4 and 5	☑	☑	MSV11/40072
Perimeter type	Switch 6 and 7	☑	☑	MSV13/40074

DIP Switches

> **Note**
>
> Please note that the default state of all DIP switches, S1, S2, and S3, is off (0).

S1 DIP Switch

6-position DIP switch (only the last switch is used for factory reset):

Number	Description	OFF (0)	ON (1)
1	Units	Metric	Imperial
6	Factory Reset	Down - perform an action	Up - preparation for action

S2 DIP Switch

The S2 8-position DIP switch is reserved for future IP configuration.

S3 DIP Switch

8-position DIP switch configures the application settings:

> **Note**
>
> Please note that if the switch 1 on the S3 DIP switch is on, than configuration of the fan, reheater, and perimeter type from any other source is blocked, as well as any changes made to the referring parameters (for example, Heater Priority) will not be written to the device.

Number	Description	OFF (0)	ON (1)
1	Configuration priority	Control Point VAV/Software (BACnet/Modbus/iC Configurator/iC Device Manager)	DIP switch
2 3	Reheater type	2 - off, 3 - off: no reheater 2 - on, 3 - off: reheater PWM (time proportional) 2 - off, 3 - on: reheater digital/staged (up to 2 stages) 2 - on, 3 - on: reheater floating	
4 5	Fan type	4 - off, 5 - off: no fan 4 - on, 5 - off: series fan 4 - off, 5 - on: parallel fan 4 - on, 5 - on: parallel fan	
6 7	Perimeter type	6 - off, 7 - off: no perimeter 6 - on, 7 - off: perimeter PWM (time proportional) 6 - off, 7 - on: perimeter digital (on/off) 6 - on, 7 - on: no perimeter	

BACnet Objects/Modbus Registers

Note

Please note that if the switch 1 on the S3 DIP switch is on, than configuration of the fan, reheater, and perimeter type from any other source is blocked (BACnet/Modbus), as well as any changes made to the referring parameters (for example, Heater Priority) will not be written to the device.

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
Configuration						
AV37	40238 (237)	SetpointOffsetRange	N/a	Read/ write	6	Sets the setpoint offset range to be changed by the user (e.g., a value of 5 means a range from -5 to +5), (min = 0, max = 10) Connected to CP AV59
AV38	40239 (238)	CO2SensorRange	N/a	Read/ write	2000	Sets the range of the CO_2 sensor used (min = 0, max = 10000) Connected to CP AO37
BV4	00105 (104)	PresenceSensorInvert	True/ false	Read/ write	False	Allows to invert the I1 input signal if required: false (normal), true (invert) Connected to CP BO66
BV5	00106 (105)	WindowContactInvert	True/ false	Read/ write	False	Allows to invert the I2 input signal if required: false (normal), true (invert) Connected to CP BO67
BV6	00107 (106)	DeviceType	True/ false	Read/ write	False	Allows to choose VAV device control type from airflow demand (VAV also known as pressure independent) or from cooling/heating demand (VVT also known as pressure dependent): false (VAV), true (VVT) Connected to CP BO68

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
BV8	00109 (108)	DualHeat	True/ false	Read/ write	False	Allows to enable dual heating mode with repeater/perimeter and damper (no effect without damper): false (no), true (yes) Connected to CP BO69
BO2	00123 (122)	Units	True/ false	Read/ write	False	Allows to change the unit in the application provided that the BACnet configuration is allowed in the Device Configuration Source variable: false (metric), true (imperial) Connected to CP BO2
MSV11	40272 (271)	FanType	N/a	Read/ write	None	Allows to set the fan type: 1: none 2: series 3: parallel Connected to CP MSV9
MSV12	40273 (272)	ReheaterType	N/a	Read/ write	None	Allows to set the reheater type: 1: none 2: TimeProp 3: staged 4: float Connected to CP MSV10
MSV13	40274 (273)	PerimeterType	N/a	Read/ write	None	Allows to set the perimeter type: 1: none 2: TimeProp 3: digital Connected to CP MSV11
MSV14	40275 (274)	HeaterPriority	N/a	Read/ write	Reheater	Possibility to select heating priority for heating sources: 1: reheater 2: perimeter 3: simultaneous Connected to CP MSV14

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
MSV21	40282 (281)	U1DischargeTempType	N/a	Read/ write		Allows to select the discharge temperature type: 1: Voltage measurement 2: Current 3: Resistance input 4: 10K3A1 NTC 5: 10K4A1 NTC 6: 10K NTC Carel 7: 20K6A1 NTC 8: 2,2K3A1 NTC 9: 3K3A1 NTC 10: 30K6A1 NTC 11: SIE1 12: TAC1 13: SAT1 14: PT1000 15: NI1000 16: NI1000 21C 17: NI1000 LG F 18: 10K Type2 NTC F 19: 10K Type3 NTC F 20: 20K NTC F 21: 3K NTC F 22: PT1000 F 23: NI1000 32F 24: NI1000 70F Connected to CP MSV12

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
MSV22	40283 (282)	U2SpaceTempType	N/a	Read/ write		Allows to select the space temperature type: 1: Voltage measurement 2: Current 3: Resistance input 4: 10K3A1 NTC 5: 10K4A1 NTC 6: 10K NTC Carel 7: 20K6A1 NTC 8: 2,2K3A1 NTC 9: 3K3A1 NTC 10: 30K6A1 NTC 11: SIE1 12: TAC1 13: SAT1 14: PT1000 15: NI1000 16: NI1000 21C 17: NI1000 LG F 18: 10K Type2 NTC F 19: 10K Type3 NTC F 20: 20K NTC F 21: 3K NTC F 22: PT1000 F 23: NI1000 32F 24: NI1000 70F Connected to CP MSV13
Flow Settings						
AV11	40212 (211)	MaxOccCoolAirflowSetpoint	L/s / cfm	Read/ write	400/800	Maximum airflow setpoint for cooling in the occupied state (min = 0) Connected to CP AO29
AV12	40213 (212)	MinOccCoolAirflowSetpoint	L/s / cfm	Read/ write	100/200	Minimum airflow setpoint for cooling in the occupied state (min = 0) Connected to CP AO30
AV13	40214 (213)	MaxOccHeatAirflowSetpoint	L/s / cfm	Read/ write	200/400	Maximum airflow setpoint for heating in the occupied state (min = 0) Connected to CP AO31

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
AV14	40215 (214)	MinOccHeatAirflowSetpoint	L/s / cfm	Read/ write	100/200	Minimum airflow setpoint for heating in the occupied state (min = 0) Connected to CP AO32
AV15	40216 (215)	StandbyAirFlowSetpoint	L/s / cfm	Read/ write	50/100	Airflow setpoint in the standby state (min = 0) Connected to CP AO33
AV16	40217 (216)	UnoccAirFlowSetpoint	L/s / cfm	Read/ write	25/50	Airflow setpoint in the unoccupied state (min = 0) Connected to CP AO34
AV17	40218 (217)	StartupFanAirflowSetpoint	L/s / cfm	Read/ write	35/75	Airflow setpoint for a fan startup only (min = 0) Connected to CP AO35
BV9	00110 (109)	DamperDirection	True/ false	Read/ write	False	Sets the damper opening direction: false (CW), true (CCW) Connected to CP BO65
AV0	40201 (200)	OccCoolTempSetpoint	°C/°F	Read/ write	23/74	Temperature setpoint for cooling in the occupied state
AV1	40202 (201)	OccHeatTempSetpoint	°C/°F	Read/ write	21/70	Temperature setpoint for heating in the occupied state
AV2	40203 (202)	StandbyCoolTempSetpoint	°C/°F	Read/ write	25/77	Temperature setpoint for cooling in the standby state
AV3	40204 (203)	StandbyHeatTempSetpoint	°C/°F	Read/ write	19/67	Temperature setpoint for heating in the standby state
AV4	40205 (204)	UnoccCoolTempSetpoint	°C/°F	Read/ write	28/80	Temperature setpoint for cooling in the unoccupied state
AV5	40206 (205)	UnoccHeatTempSetpoint	°C/°F	Read/ write	16/64	Temperature setpoint for heating in the unoccupied state
AV6	40207 (206)	SpaceCO2Setpoint	ppm	Read/ write	1000	CO_2 setpoint for the room (regardless of the occupancy status)
AV9	40210 (209)	HotAirTempSetpoint	°C/°F	Read/ write	26/78	Discharge temperature setpoint for detecting warm air to change the heating/cooling mode

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
Balancing						
BV13	00114 (113)	PressureZeroing	True/ false	Read/ write	False	Allows to perform zeroing action on the built-in PI pressure transducer if there is a non-zero value in the absence of airflow. **Important:** perform in the absence of airflow through the pressure transducer ports False (no), true (zeroing) Connected to CP BO64
AI11	30012 (11)	ActualAirFlow	L/s / cfm	Read-only	N/a	Actual calculated airflow based on the K-Factor value and differential pressure measure Connected to CP AO25
AV61	40289 (288)	K-Factor	N/a	Read/ write	1000	Sets the K-Factor (the flow rate at a differential pressure of 1 inWC/1 Pa on the measuring cross) for the given VAV model (min = 0) Connected to CP AO23
AV64	40292 (291)	UserSetPosition	%	Read/ write	0	User-set damper opening level (after selecting the User Position mode in the Damper Command variable) (min = 0, max = 100) Connected to CP AO26
AV65	40293 (292)	UserSetAirflow	L/s / cfm	Read/ write	0	User-set airflow setpoint (after selecting the User Position mode in the Damper Command variable) (min = 0) Connected to CP AO27
AV66	40294 (293)	AirflowCalibration	L/s / cfm	Read/ write	0	Allows to enter the airflow value measured by the hood for calibration Connected to CP AO28
AO0	30251 (250)	DamperPosition	%	Read-only	N/a	Actual damper opening level coming from the Rotary Actuator component (Position Feedback signal) Connected to CP AO24

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
MSV7	40268 (267)	DamperCommand	N/a	Read/ write	None	Allows to set different damper command modes for calibration, balancing, or testing: 1: Auto/none (null) 2: Max flow 3: Min flow 4: User flow 5: User position 6: Full open 7: Full close 8: Calibrate Connected to CP MSV5
Other						
AI0	30001 (0)	UpTime	s	Read-only	N/a	Device's up time
AI1	30005 (4)	SpaceTemperature	°C/°F	Read-only	N/a	Calculated space temperature
AI3	30004 (3)	U2SpaceTemperature	°C/°F	Read-only	N/a	Actual temperature sensor value from the local U2 input
AI4	30301 (300)	PanelTemperature	°C/°F	Read-only	N/a	Actual temperature sensor value from the room panel
AI5	30006 (5)	SpaceCO2	ppm	Read-only	N/a	Calculated space CO_2
AI6	30303 (302)	PanelCO2	ppm	Read-only	N/a	Actual CO_2 sensor value from the room panel
AI8	30009 (8)	U4CO2Sensor	ppm	Read-only	N/a	Actual CO_2 sensor value from the local U4 input
AI9	30010 (9)	EffectiveTempSetpoint	°C/°F	Read-only	N/a	Calculated effective temperature setpoint based on occupancy status and heating or cooling temperature setpoints (depending on demand)
AI10	30011 (10)	CentralTempSetpoint	°C/°F	Read-only	N/a	Calculated Central Temperature Setpoint based on setpoints and occupancy status

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
AI12	30013 (12)	EffectiveAirFlowSetpoint	L/s / cfm	Read-only	N/a	Calculated effective airflow setpoint based on the occupancy status and heating or cooling airflow setpoints
AI13	30014 (13)	DischargeTemperature	°C/°F	Read-only	N/a	Calculated discharge temperature
AI14	30015 (14)	ActDischTempSetpoint	°C/°F	Read-only	N/a	Calculated discharge temperature setpoint based on the effective temperature setpoint and heating demand for high-limiting discharge temperature
AI15	30016 (15)	DifferentialPressure	Pa / “WC	Read-only	N/a	Actual differential pressure sensor-measured value from the built-in pressure input
AI16	30017 (16)	U1DischTemperature	°C/°F	Read-only	N/a	Actual discharge temperature sensor value comes from the U1 input
AI17	30018 (17)	U3SetpointOffset	°C/°F	Read-only	N/a	Actual setpoint offset value (calculated from resistance) from the U3 input
AI100	30103 (102)	AppVersion	N/a	Read-only	1.0	Current application version
AV7	40208 (207)	MaxDischTempSetpoint	°C/°F	Read/write	32/90	High limit of discharge temperature supplied to the room
AV8	40209 (208)	MinDischTempSetpoint	°C/°F	Read/write	13/55	Low limit of discharge temperature supplied to the room
AV10	40211 (210)	NetTemperature	°C/°F	Read/write	-327	Temperature sensor value from the network
AV18	40219 (218)	NetDuctInTemp	°C/°F	Read/write	-327	Discharge temperature sensor value from the network
AV19	40220 (219)	NetOutdoorTemp	°C/°F	Read/write	-327	Outside air temperature sensor value from the network
AV20	40221 (220)	LoadShedding	%	Read/write	0	Load-shedding level required from the network (min = 0, max = 100)

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
AV21	40222 (221)	MaxDamperPosition	%	Read/ write	100	Maximum damper open position limit (min = 0, max = 100)
AV22	40223 (222)	MinDamperPosition	%	Read/ write	0	Minimum damper open position limit (min = 0, max = 100)
AV23	40224 (223)	MaxFanSpeed	%	Read/ write	100	Maximum fan speed level limit (min = 0, max = 100)
AV24	40225 (224)	MinFanSpeed	%	Read/ write	20	Minimum fan speed level limit (min = 0, max = 100)
AV25	40226 (225)	DamperDriveTime	s	Read/ write	90	Time required to reach full damper open position (from 0% to 100%) (min = 0)
AV26	40227 (226)	DamperResponse	%	Read/ write	15	Sets the value of how much the control signal must change to perform the next damper movement
AV28	40229 (228)	MaxOatReheater	°C/°F	Read/ write	32/90	Maximum outside air temperature limit that allows the use of a reheater
AV29	40230 (229)	MaxOatPerimeter	°C/°F	Read/ write	18/65	Maximum outside air temperature limit that allows the use of a perimeter
AV30	40231 (230)	FloatReheaterValveDriveTime	s	Read/ write	120	Time required to reach full reheater valve open (from 0% to 100%)
AV32	40233 (232)	TimePropReheaterPeriod	s	Read/ write	60	Time required for a PWM valve for a full cycle time of a reheater
AV33	40234 (233)	TimePropPerimeterPeriod	s	Read/ write	60	Time required for a PWM valve for a full cycle time of a perimeter
AV34	40235 (234)	BypassTimeOverride	min	Read/ write	120	Time required to maintain the bypass occupancy status (after expiration it returns to the previous status)
AV35	40236 (235)	StandbyTimeOverride	min	Read/ write	15	Time required to change the occupancy status to standby (after motion detection disappears)

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
AV36	40237 (236)	ChangeoverDelay	min	Read/ write	2	Delay time for changing the heating/cooling operating mode
AV39	40240 (239)	FanDelayOff	s	Read/ write	30	Delay time for switching off the fan
AV40	40241 (240)	NetCO2	ppm	Read/ write	N/a	CO_2 sensor value comes from the network
AV41	40242 (241)	NetOffset	°C/°F	Read/ write	0	Offset temperature setpoint value from the network
AV56	40502 (501)	PanelTempSetpoint	°C/°F	Read/ write	N/a	Temperature setpoint from the room panel
AV58	40504 (503)	OffsetTempSetpoint	°C/°F	Read/ write	N/a	Temperature offset for setpoint from the room panel
AV62	40290 (289)	CalculatedK-Factor	N/a	Read/ write	N/a	Calculated K-Factor value (meaning the flow rate at a differential pressure of 1 inWC/1 Pa on the measuring cross) based on the entered measured airflow (using the hood) and the measured differential pressure from the built-in PI input
AV67	40295 (294)	DamperControl	%	Read/ write	0	Damper opening level control resulting from the application
AO1	40252 (251)	ReheaterControl	%	Read-only	N/a	Reheater valve opening level control resulting from the application
AO2	40253 (252)	FanSpeed	%	Read-only	N/a	Fan speed level control resulting from the application
AO3	40254 (253)	PerimeterControl	%	Read-only	N/a	Perimeter valve opening level control resulting from the application
AO4	40255 (254)	AHUOutdoorDamper	%	Read-only	N/a	Required opening level of the fresh air damper in the AHU resulting from the current CO_2 level in the room (network signal to the AHU control)
BI0	10302 (301)	PresenceSensor	True/ false	Read-only	N/a	Actual presence sensor state from the I1 input

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
BI1	10304 (303)	WindowContact	True/ false	Read-only	N/a	Actual window contact state from the I2 input
BV0	00101 (100)	AutoOccMode	True/ false	Read/ write	False	Automatically changes occupancy to occupied when increased airflow is detected: false (disabled), true (enabled)
BV1	00102 (101)	WarmUpHeater	True/ false	Read/ write	False	Allows to use the reheater also in the Morning WarmUp mode (in the HVAC Mode variable): false (disabled), true (enabled)
BV2	00103 (102)	WarmUpParallelFan	True/ false	Read/ write	False	Allows to use the parallel fan also in the Morning WarmUp mode (in the HVAC Mode variable): false (disabled), true (enabled)
BV3	00104 (103)	StandbySeriesFan	True/ false	Read/ write	False	Allows to use the series fan also in standby occupancy: false (disabled), true (enabled)
BV10	00111 (110)	FanControlMode	True/ false	Read/ write	False	Allows to select the fan control mode: false (digital), true (analog)
BV11	00112 (111)	HeatDamperPriority	True/ false	Read/ write	False	Allows to set the priority of the damper heating operation when working in the active DualHeat mode: false (secondary), true (primary)
BV12	00113 (112)	DeviceConfigurationSource	True/ false	Read/ write	False	Allows to choose to set a source of the device application configuration: false (BACcnet), true (DIP switch)
BO0	00121 (120)	PerimeterStatus	True/ false	Read/ write	False	Perimeter digital control status (if configured as digital) from the application: false (off), true (on)
BO1	00122 (121)	FanStatus	True/ false	Read/ write	False	Fan digital control status (if configured as digital) from the application: false (off), true (on)

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
BO16	00017 (116)	PanelHeating	True/ false	Read/ write	N/a	Heating mode status to be displayed as an icon on the Control Point room panel false (no heating), true (heating) Connected to CP BO16
BO17	00118 (117)	PanelCooling	True/ false	Read/ write	N/a	Cooling mode status to be displayed as an icon on the Control Point room panel false (no cooling), true (cooling) Connected to CP BO17
BO55	00156 (155)	PanelSetpointMode	True/ false	Read/ write	N/a	Sets the temperature setpoint mode in the Control Point room panel - directly or as an offset false (offset), true (setpoint) Connected to CP BO55
BO56	00157 (156)	PanelSetpointDisplay	True/ false	Read/ write	N/a	Allows to select which value to be displayed during the offset editing according to the set Panel Setpoint Mode false (offset), true (setpoint) Connected to CP BO56
MSI1	30022 (21)	HVACStatus	N/a	Read-only	Unoccupied Cooling	Current HVAC status: 1: Occupied cooling 2: Occupied heating 3: Standby cooling 4: Standby heating 5: Unoccupied cooling 6: Unoccupied heating 7: Off 8: Fire
MSI2	30023 (22)	OccupancyStatus	N/a	Read-only	Unoccupied	Current occupancy status: 1: occupied 2: unoccupied 3: bypass 4: standby

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
MSV0	40601 (600)	PanelFanStatus	N/a	Read/ write	Off	Fan status to be displayed as an icon on the Control Point room panel: 1: off 5: speed 1 (auto) 6: speed 2 (auto) 7: speed 3 (auto)
MSV1	40262 (261)	HVACMode	N/a	Read/ write	Auto	Allows to set the operating mode for the VAV device: 1: Auto 2: Heat 3: Morning WarmUp 4: Cool 5: Night Purge 6: PreCool 7: Off (depressurize) 8: Fire (pressurize)
MSV2	40263 (262)	OccupancyMode	N/a	Read/ write	Unoccupied	Allows to set room the occupancy mode from the BMS system: 1: occupied 2: unoccupied 3: bypass 4: standby
MSV3	40701 (700)	PanelOccupancy Status	N/a	Read/ write	Unoccupied	Current room occupancy status dedicated to Control Point room panel 1: unoccupied 2: occupied 3: standby 4: bypass Connected to CP MSV3
MSV4	40702 (701)	PanelOccupancyMode	N/a	Read/ write	Unoccupied	Status of set room occupancy mode from Control Point room panel 1: unoccupied 2: occupied Connected to CP MSV4

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
MSV5	40703 (702)	PanelOccupancyReset	N/a	Read/ write	Unoccupied	Resetting the room occupancy mode from occupied to unoccupied in the Control Point room panel 1: unoccupied 2: occupied
MSV8	40269 (268)	FanCommand	N/a	Read/ write	None	Allows to set different fan command modes for testing 1: Auto/none (null) 2: Max speed 3: Min speed 4: Stop Connected to CP MSV6
MSV9	40270 (269)	ReheaterCommand	N/a	Read/ write	None	Allows to set different reheater command modes for testing 1: Auto/none (null) 2: Full open 3: Full close Connected to CP MSV7
MSV10	40271 (270)	PerimeterCommand	N/a	Read/ write	None	Allows to set different perimeter command modes for testing 1: Auto/none (null) 2: Full open 3: Full close Connected to CP MSV8
MSV23	40284 (283)	TemperatureInputSelector	N/a	Read/ write	Panel	Allows to choose the temperature source for control logic: 1: panel 2: input 3: network Connected to CP MSV15
MSV24	40285 (284)	SetpointOffsetSelector	N/a	Read/ write	Panel	Allows to choose the setpoint offset source for control logic: 1: panel 2: input 3: network Connected to CP MSV16

BACnet ID	Modbus address (dec)	Object/register name	Units	Access	Default value	Description
MSV25	40286 (285)	CO2InputSelector	N/a	Read/ write	Panel	Allows to select the CO2 source for control logic: 1: panel 2: input 3: network Connected to CP MSV17
MSO0	40291 (290)	ReheaterStatus	N/a	Read/ write	Off	Reheater staged control status (if configured as digital or staged) from application: 1: off 2: stage 1 on 3: stage 2 on

iSMA Configurator

> **Note**
>
> Please note that if the switch 1 on the S3 DIP switch is on, than configuration of the fan, reheater, and perimeter type from any other source is blocked (iSMA Configurator), as well as any changes made to the referring parameters (for example, Heater Priority) will not be written to the device.

In the iSMA Configurator, it is possible to configure parameters of the VAV14-IP controller in three areas:

- VAV application;
- flow settings;
- balancing.

VAV Configurator

The configuration is carried out in a dedicated tool, VAV Configurator, available as a built-in tool in the iSMA Configurator:

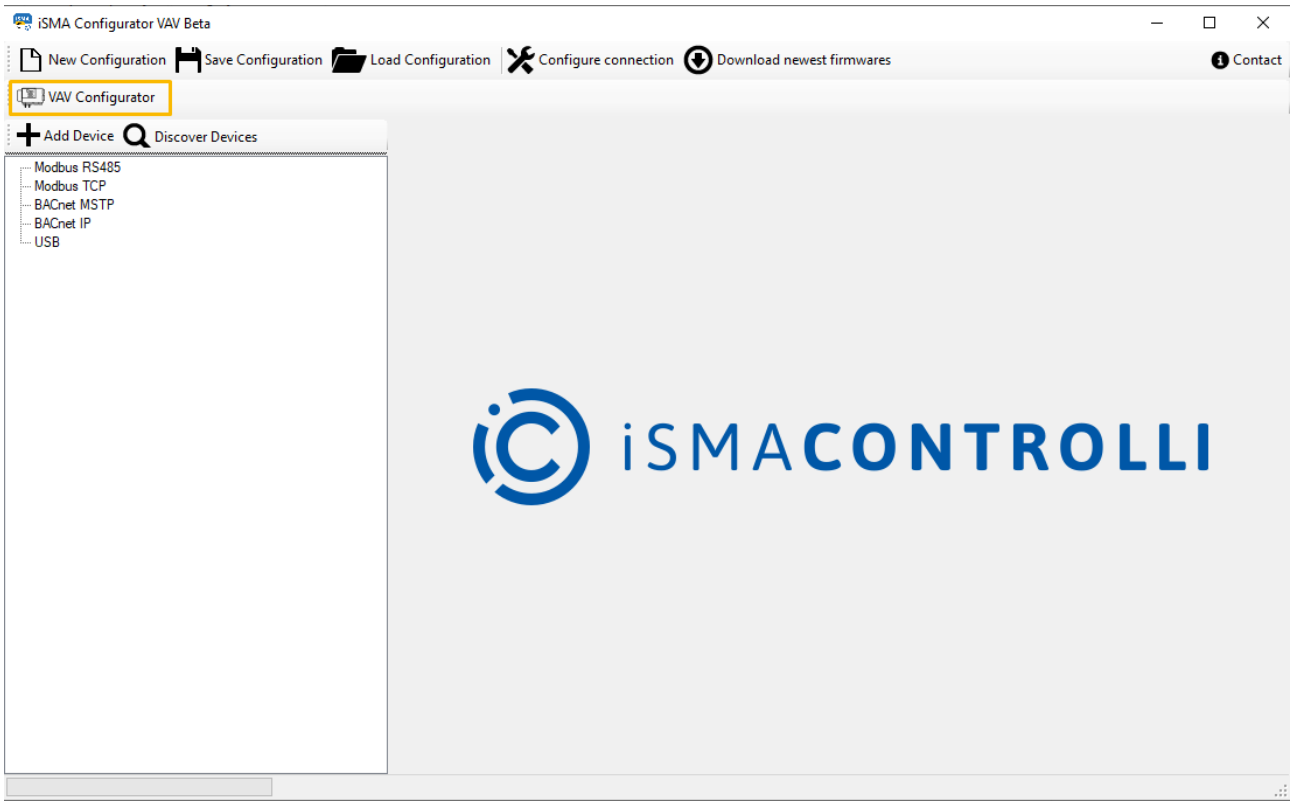


Figure 34. VAV Configurator button

Opening the VAV Configurator

The VAV Configurator is accessible under a dedicated button.

Before opening the VAV Configurator, it is required to start a BACnet IP transmission. Normally, a start-transmission prompt will be displayed automatically after pressing the VAV Configurator button:

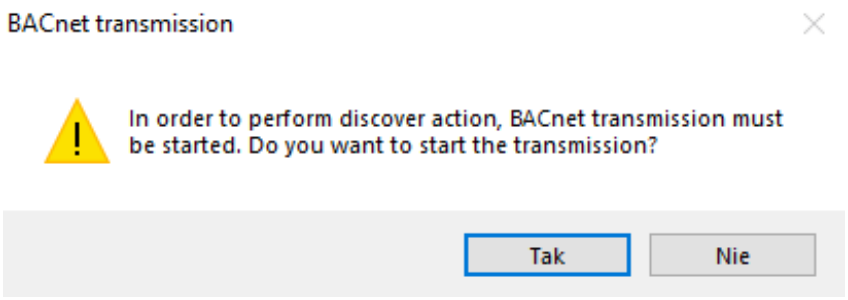


Figure 35. BACnet transmission prompt

Confirm the pop-up window and the BACnet IP transmission will be started automatically.

Note
In case there are any problems with an automatic start of the BACnet IP transmission, start it manually. Find out more here.
If BACnet IP communication is not started, the VAV Configurator button is not active.

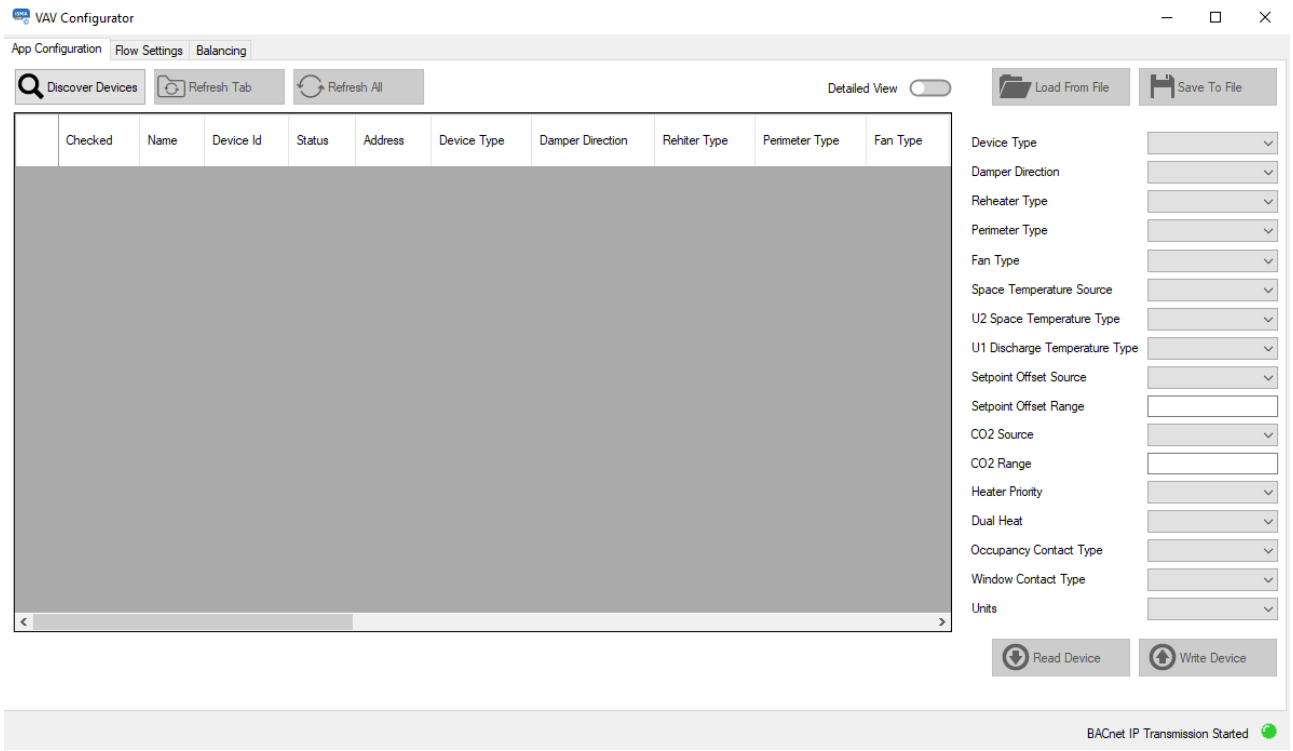


Figure 36. VAV Configurator started

Discovering Devices

The next required step is discovering devices. Use a dedicated button in the left top corner of the VAV Configurator window:

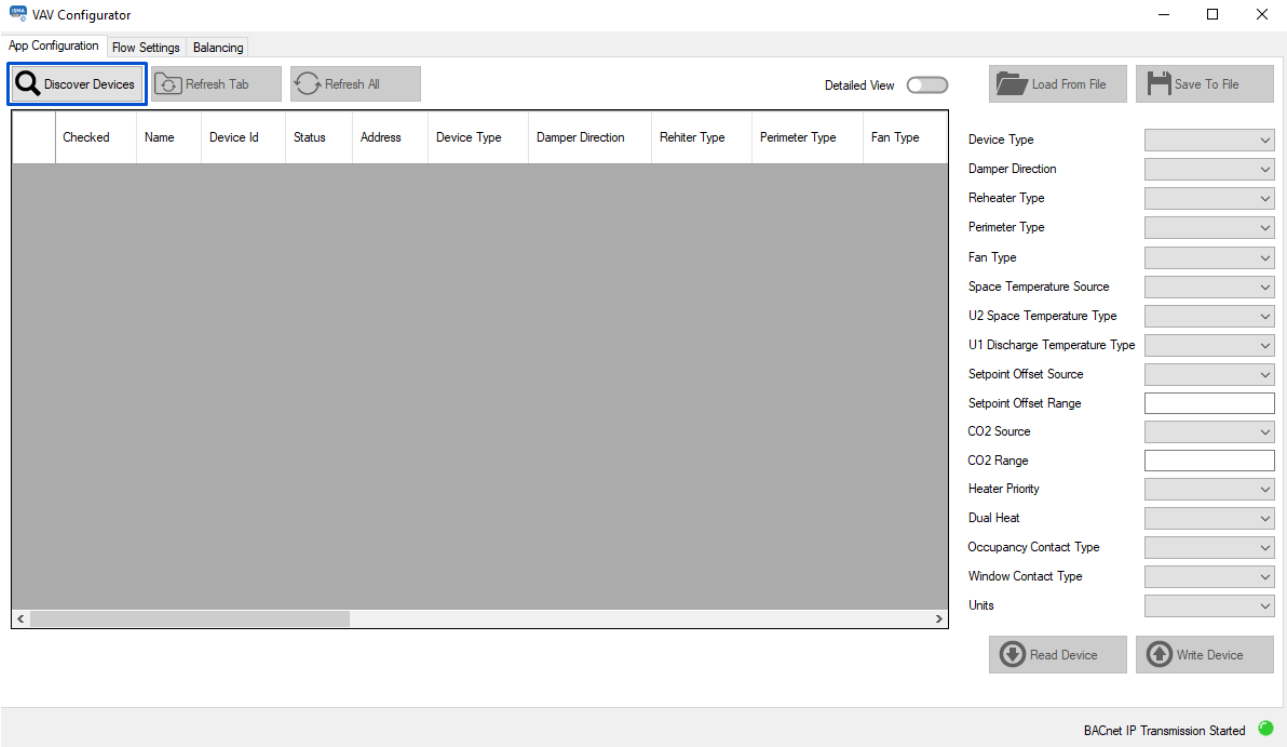


Figure 37. Discover Devices button

Warning!
For the discovering process to be successful, make sure that the subnet mask of the PC is compliant with the subnet mask of the device to be discovered. Network parameters of the controller can be checked and/or changed in the Ethernet component (System container).

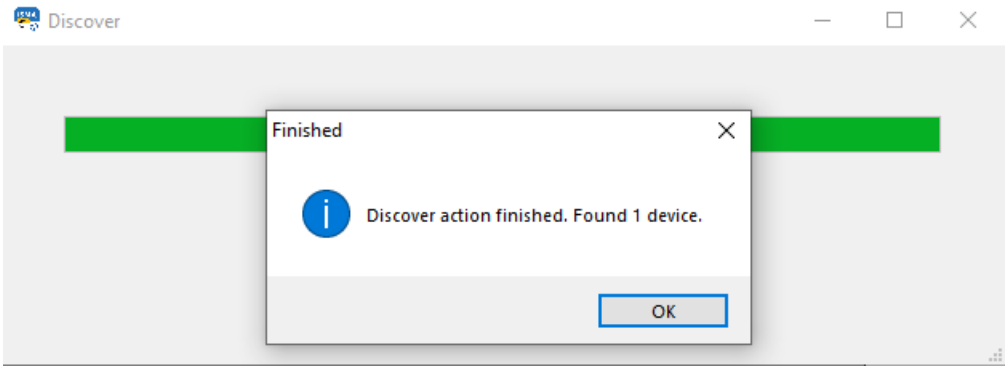


Figure 38. Finished discovering process

The VAV Configurator has two tabs where different of the application and airflow parameters can be adjusted:

- App Configuration
- Flow Settings

App Configuration

The first tab of the VAV Configurator is the App Configuration tab. Here, it is possible to execute four basic actions and configure VAV application parameters.

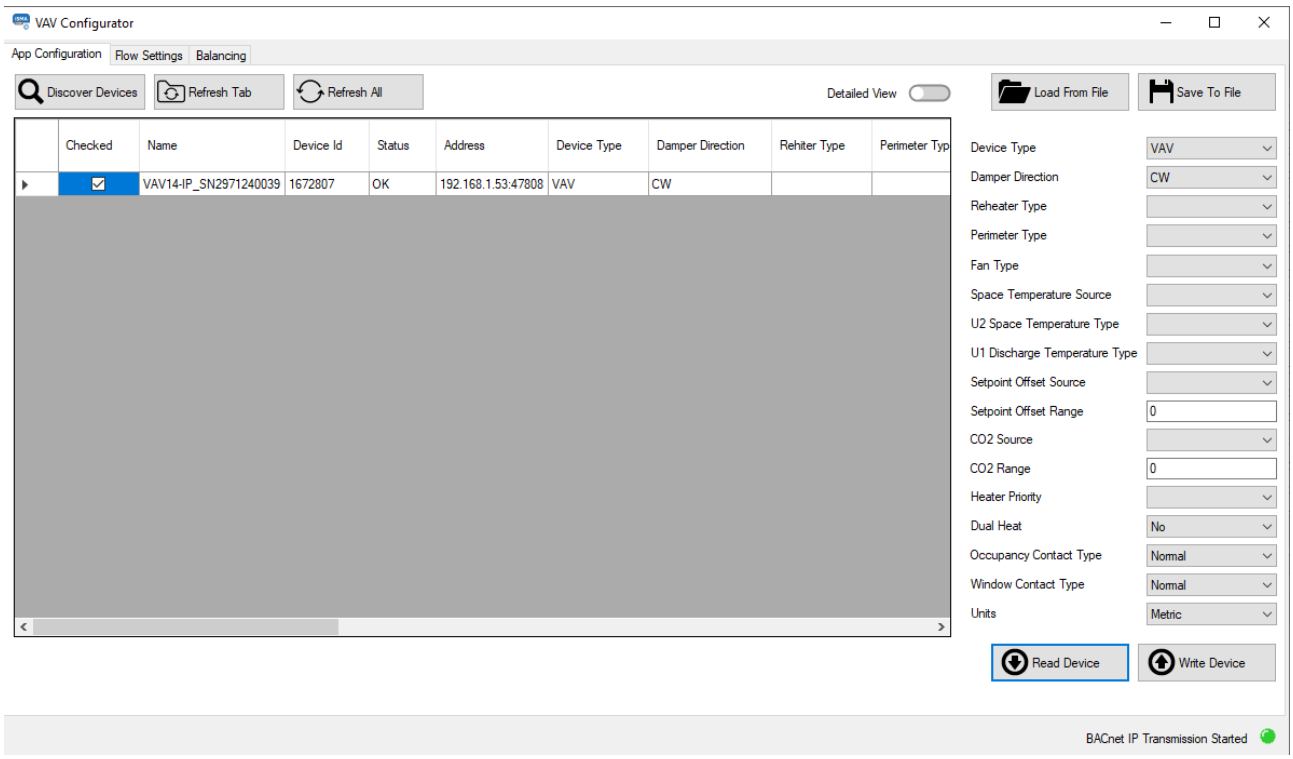


Figure 39. App Configuration view

The four available actions are:

- **Load From File:** allows to upload application parameters from a previously saved file (*.json);
- **Save To File:** allows to save current application settings to a .json file;
- **Read Device:** reads current application settings directly from the device;
- **Write Device:** sends new settings to the device.

Warning!

Please note that if the switch 1 on the S3 DIP switch is on, the Write Device action will take no effect.

The VAV application parameters available to configure are:

- **Device Type:** allows to choose a device type (DeviceType):
  - Available settings: VAV (volume air variable - control based on airflow demand) and VVT (variable volume and temperature - control based directly on heating and cooling demands);
- **Damper Direction:** allows to set a damper moving direction to clockwise or counterclockwise (DamperDirection);
- **Reheater Type:** allows to set a reheater type;
  - Available settings: none, TimeProp (PWM), staged, float;
- **Perimeter Type:** allows to set a perimeter type;
  - Available settings: none, TimeProp (PWM), digital;
- **Fan Type:** allows to set a fan type;
  - Available settings: none, series (works only as ventilator regardless of the heating/cooling demand, in occupied or standby statuses), parallel (works on heating demand in the HVAC mode set to Auto or Heat, regardless of the occupancy status);
- **Space Temperature Source:** allows to set a leading source for temperature measurement for the space temperature control loop calculations;

  - Available settings: panel (Control Point VAV), input (U2), network;
- **U2 Space Temperature Type:** allows to set a space temperature sensor type on the universal input 2;
  - Available settings: voltage measurement, current, resistance input, specific temperature sensor;
- **U1 Discharge Temperature Type:** if connected, allows to set a discharge temperature sensor type on the universal input 1 (used in the supply air temperature control loop calculations);
  - Available settings: voltage measurement, current, resistance input, specific temperature sensor;
- **Setpoint Offset Source:** allows to set a source for temperature setpoint offset settings;
  - Available settings: panel (Control Point VAV), input (U3), network;
- **Setpoint Offset Range:** allows to set a range for setpoint offset settings;
- **CO2 Source:** allows to set a leading source for CO_2 measurement for the CO_2 control loop calculations;
  - Available settings: panel (Control Point VAV), input (U4), network;
- **CO2 Range:** allows to set a range for CO_2 alarm settings – outside this range the CO_2 alarm state will be invoked;
- **Heater Priority:** allows to select a first priority heat source in the DualHeat function for the reheater and perimeter (if used in the system);
  - Available settings: reheater, perimeter, simultaneous;
- **Dual Heat:** allows to enable a possibility to use one or two heat sources (reheater and perimeter, if used in the system);
  - Available settings: no, yes;

DualHeat

The DualHeat function allows to select one or two stages of heating (two stages of heating meaning heating with a reheater and, optionally, perimeter and a damper). The DualHeat function is activated according to the setting of the HeatDamperPriority Data Point (linked to the Heat Priority slot in the DamperControlComponent):

- primary (false): damper in the range of 0-50% (directly from the space temperature heating loop) and heater(s) in the range of 50-100% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop),
- secondary (true): heater(s) in the range of 0-50% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop) and damper in the range of 50-100% (directly from the space temperature heating loop).

Heater(s) in the primary stage can be reheater and/or perimeter. If available in the system and properly configured, they can be arranged in order according to the HeaterPriority variable (ReheaterControl folder):

- reheater/perimeter
- perimeter/reheater
- simultaneous.

If there is no reheater and/or perimeter in the system, the DualHeat function cannot be activated and the damper is the only heating source. If the reheater is available in the system, it can be configured as the primary stage heater and the DualHeat function can be activated. However, if the system is equipped with the reheater and parallel fan, the reheater will always be the secondary:

- parallel fan/reheater.

- **Occupancy Contact Type:** allows to select an occupancy contact type;
  - Available settings: normal, invert;

Normal/invert Modes

If available, the motion detector is connected to the I1 digital input. It is possible to choose between two types of a presence sensor operation:

- normal (default): I1 true - presence detected, I2 false - no presence detected,
- invert: I2 true - no presence detected, I2 false - presence detected.

- **Window Contact Type:** allows to select a mode window reed switch operation;
  - Available settings: normal, invert;

Normal/invert Modes

If available, the window reed switch is connected to the I2 digital input. It is possible to choose between two types of a reed switch operation:

- normal (default): I2 true - window open, I2 false - window closed,
- invert: I2 true - window closed, I2 false - window open.

- **Units:** allows to select a units system;
  - Available settings: imperial, metric.

Flow Settings

The second tab of the VAV Configurator is the Flow Settings tab. Here, it is possible to execute four basic actions and set flow parameters for the use of VAV application.

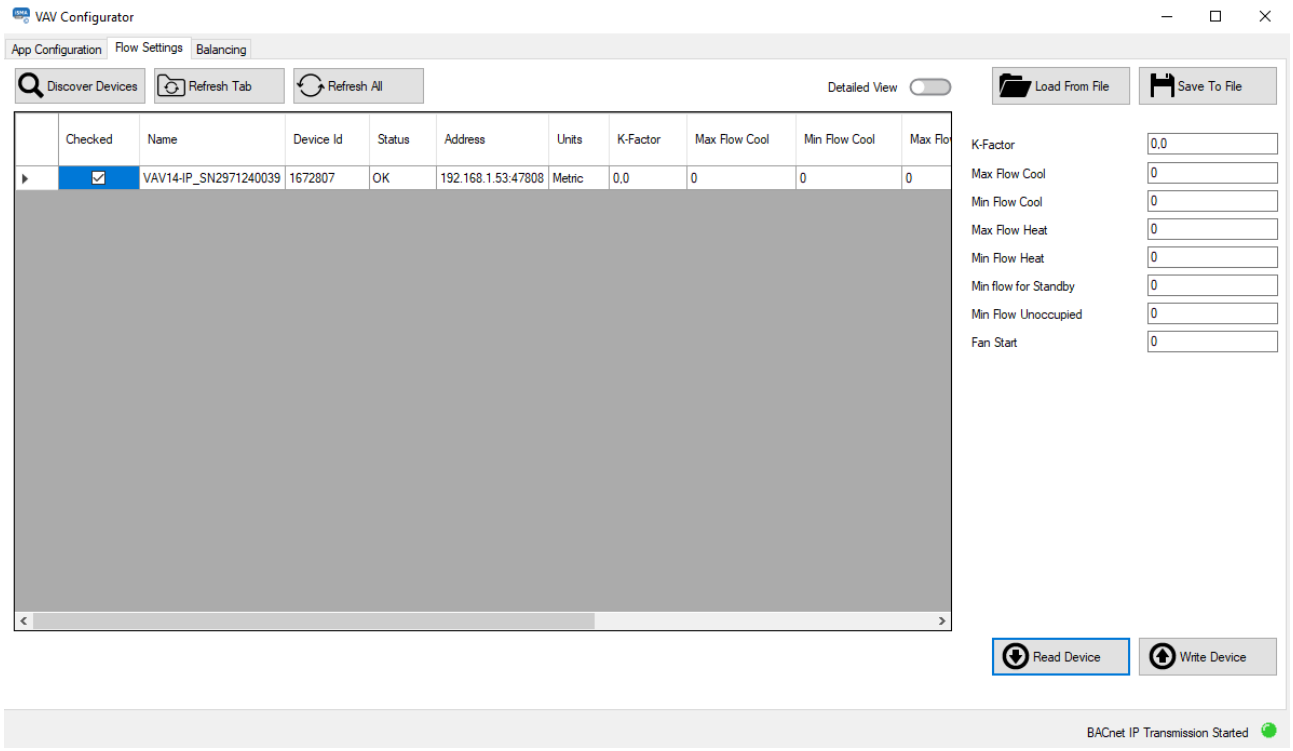


Figure 40. Flow Settings view

The four available actions are:

- **Load From File:** allows to upload airflow parameters from a previously saved file (*.json);
- **Save To File:** allows to save current airflow settings to a .json file;
- **Read Device:** reads current airflow settings directly from the device;
- **Write Device:** sends new settings to the device.

Warning!

Please note that if the switch 1 on the S3 DIP switch is on, the Write Device action will take no effect.

The VAV application airflow parameters available to configure are:

- **K-Factor:** allows to set a K-factor value according to the type of device for use in the airflow calculations;

K-Factor

The K-Factor variable, set in the calibration process from the Control Point VAV panel, iSMA Configurator, or BACnet/Modbus, is a parameter provided by the VAV box manufacturer. It determines the airflow through the unit at 1 Pa/1 inH2O.

- **Max Flow Cool:** allows to set a maximum airflow value for cooling in an occupied status (MaxOccCoolAirflowSetpoint);
- **Min Flow Cool:** allows to set a minimum airflow value for cooling in an occupied status (MinOccCoolAirflowSetpoint);
- **Max Flow Heat:** allows to set a maximum airflow value for heating in an occupied status (MaxOccHeatAirflowSetpoint);
- **Min Flow Heat:** allows to set a minimum airflow value for heating in an occupied status (MinOccHeatAirflowSetpoint);

- **Min Flow for Standby:** allows to set a minimum airflow value in a standby status (StandbyAirflowSetpoint);
- **Min Flow Unoccupied:** allows to set a minimum airflow value in an unoccupied status (UnoccAirflowSetpoint);
- **Fan Start:** allows to set a minimum airflow value for a fan to start operating (StartupFanAirflowSetpoint).

Control Point VAV

Note

Please note that if the switch 1 on the S3 DIP switch is on, than configuration of the fan, reheater, and perimeter type from any other source is blocked (Control Point VAV), as well as any changes made to the referring parameters (for example, Heater Priority) will not be written to the device.

Warning!

Before using the Control Point VAV panel with the VAV14-IP controller, make sure to upgrade its firmware to version V2.5.
Instructions how to upgrade firmware: iSMA Configurator.

The configuration of the VAV14-IP device is possible directly from the Control Point VAV panel. To do that, it is required to connect the panel to the controller with an RJ45-RJ45 cable.

Parameters available to configure are contained in the panel's submenu. In order to enter the submenu:

- long-press the OK button;
- edit PIN (default: 0000);
- short-press the OK button.

Figure 41. Panel access to the VAV submenu

VAV Submenu

In the Control Point panel's VAV submenu, it is possible to configure parameters of the VAV controller in two areas:

- App Configuration

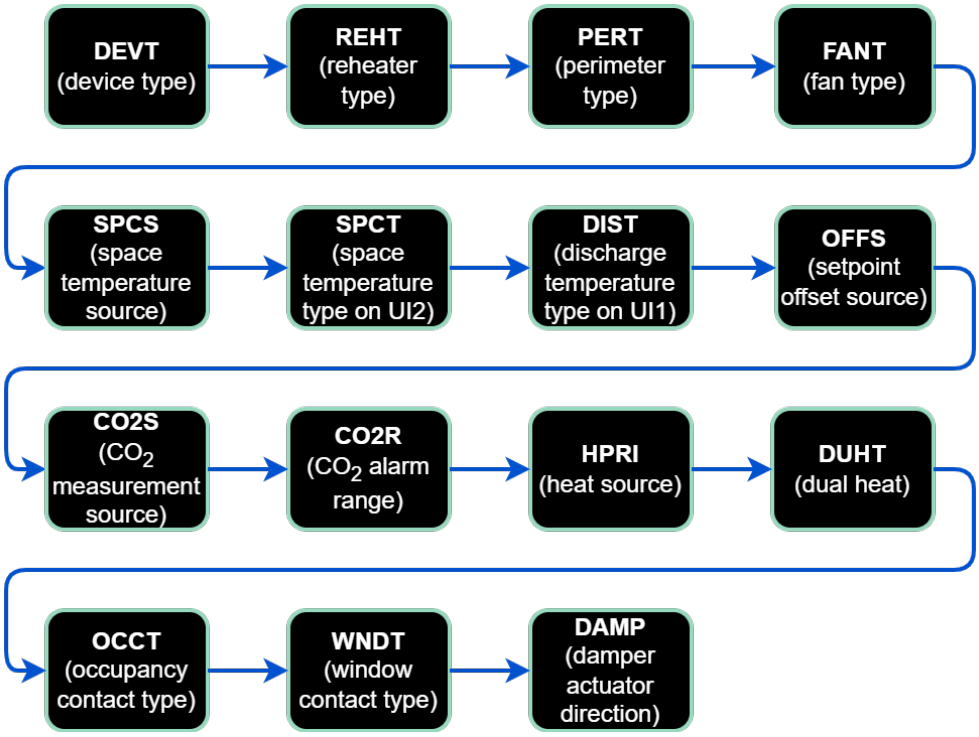


Figure 42. Sequence of app configuration options in the Control Point VAV submenu

- Flow Settings

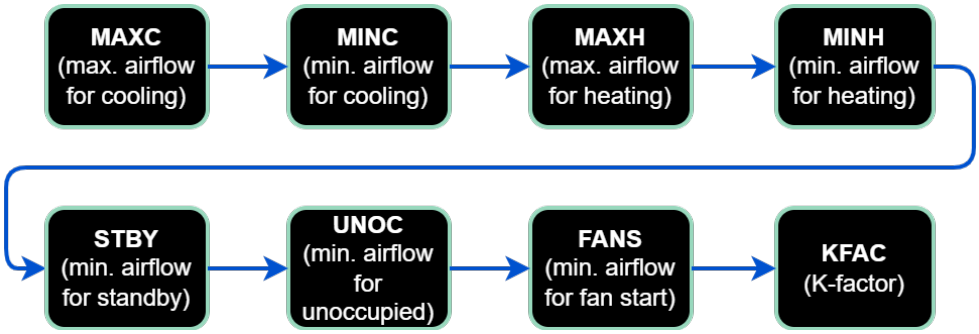


Figure 43. Sequence of flow settings options in the Control Point VAV submenu

App Configuration

In the Control Point panel's VAV submenu dedicated to the VAV application, it is possible to configure the following parameters:

DEVT: allows to choose a device type between VAV and VVT;

- **DeviceType:**
  - Modbus register: 00107;
  - BACnet object: BV6, property: Present Value;

Register/object value	Value
0/false	VAV
1/true	VVT

REHT: allows to set a reheater type;

- **ReheaterType:**
  - Modbus register: 40273;
  - BACnet object: MSV12, property: Present Value;

Register/object value	Value
1	None
2	Time proportional (PWM)
3	Staged
4	Float

PERT: allows to set a perimeter type;

- **PerimeterType:**
  - Modbus register: 40274;
  - BACnet object: MSV13, property: Present Value;

Register/object value	Value
1	None
2	Time proportional (PWM)
3	Digital

FANT: allows to set a fan type;

- **FanType:**
  - Modbus register: 40272;
  - BACnet object: MSV11, property: Present Value;

Register/object value	Value
1	None
2	Series
3	Parallel

SPCS: allows to set a source for temperature readings;

- **TemperatureInputSelector:**
  - Modbus register: 40284;
  - BACnet object: MSV23, property: Present Value;

Register/object value	Value
1	Panel
2	Input
3	Network

SPCT: allows to set a space temperature type on the universal input 2;

- **U2SpaceTempType:**
  - Modbus register: 40283;
  - BACnet object: MSV22, property: Present Value;

DIST: allows to set a discharge temperature type on the universal input 1;

- **U1DischargeTempType:**
  - Modbus register: 40282;
  - BACnet object: MSV21, property: Present Value;

Register/object value	Value
1	Voltage measurement
2	Current
3	Resistance input
4	10K3A1 NTC
5	10K4A1 NTC
6	10K NTC Carel
7	20K6A1 NTC
8	2,2K3A1 NTC

Register/object value	Value
9	3K3A1 NTC
10	30K6A1 NTC
11	SIE1
12	TAC1
13	SAT1
14	PT1000
15	NI1000
16	NI1000 21C
17	NI1000 LG F
18	10K Type2 NTC F
19	10K Type3 NTC F
20	20K NTC F
21	3K NTC F
22	PT1000 F
23	NI1000 32F
24	NI1000 70F

OFFS: allows to set a source for setpoint offset settings;

- **SetpointOffsetSelector:**
  - Modbus register: 40285;
  - BACnet object: MSV24, property: Present Value;

CO2S: allows to set a source for CO2 readings;

- **CO2InputSelector:**
  - Modbus register: 40286;
  - BACnet object: MSV25, property: Present Value;

Register/object value	Value
1	Panel
2	Input
3	Network

CO2R: allows to set a range for CO2 alarm settings;

- **CO2SensorRange:**
  - Modbus register: 40239;
  - BACnet object: AV38, property: Present Value;

Register/object value	Value
Min.	0
Max.	10000

HPRI: allows to select a priority order for the heat source;

- **HeaterPriority:**
  - Modbus register: 40275;
  - BACnet object: MSV14, property: Present Value;

Register/object value	Value
1	Reheater
2	Perimeter
3	Simultaneous

DUHT: allows to enable a possibility to use one or two heat sources;

- **DualHeat:**
  - Modbus register: 00109;
  - BACnet object: BV8, property: Present Value;

Register/object value	Value
0/false	No
1/true	Yes

OCCT: allows to select an occupancy contact type;

- **PresenceSensorInvert:**
  - Modbus register: 00105;
  - BACnet object: BV4, property: Present Value;

WNDT: allows to select a window contact type;

- **WindowContactInvert:**
  - Modbus register: 00106;
  - BACnet object: BV5, property: Present Value;

DAMP: allows to set a damper actuator's direction to open to clockwise or counterclockwise;

- **DamperDirection:**
  - Modbus register: 00110;

- BACnet object: BV9, property: Present Value.

Register/object value	Value
0/false	Clockwise
1/true	Counterclockwise

Flow Settings

In the Control Point panel's VAV submenu dedicated to airflow settings, it is possible to configure the following parameters:

MAXC: allows to set a maximum airflow value for cooling;

- MaxOccCoolAirflowSetpoint:
  - Modbus register: 40212;
  - BACnet object: AV11, property: Present Value;

MINC: allows to set a minimum airflow value for cooling;

- MinOccCoolAirflowSetpoint:
  - Modbus register: 40213;
  - BACnet object: AV12, property: Present Value;

MAXH: allows to set a maximum airflow for heating;

- MaxOccHeatAirflowSetpoint:
  - Modbus register: 40214;
  - BACnet object: AV13, property: Present Value;

MINH: allows to set a minimum airflow for heating;

- MinOccHeatAirflowSetpoint:
  - Modbus register: 40215;
  - BACnet object: AV14, property: Present Value;

STBY: allows to set a minimum airflow in a standby state;

- StandbyAirflowSetpoint:
  - Modbus register: 40216;
  - BACnet object: AV15, property: Present Value;

UNOC: allows to set a minimum airflow in an unoccupied state;

- UnoccAirflowSetpoint:
  - Modbus register: 40217;
  - BACnet object: AV16, property: Present Value;

FANS: allows to set a minimum airflow value for a fan to start operating;

- StartupFanAirflowSetpoint:
  - Modbus register: 40218;
  - BACnet object: AV17, property: Present Value;

KFAC: allows to set a K-factor value according to the type of device for use in the airflow calculations;

- K-Factor:
  - Modbus register: 40289;
  - BACnet object: AV61, property: Present Value.

iC Device Manager

> **Note**
>
> Please note that if the switch 1 on the S3 DIP switch is on, than configuration of the fan, reheater, and perimeter type from any other source is blocked (iC Device Manager), as well as any changes made to the referring parameters (for example, Heater Priority) will not be written to the device.

The iC Device Manager service for Niagara Framework is dedicated for iSMA CONTROLLI VAV14-IP controllers. The service enables configuration of the parameters in the VAV controller in three areas:

- VAV application;
- flow settings;
- balancing.

Using iC Device Manager

Adding the Module

The iC Device Manager service is a part of the iC Workbench and iC Niagara Expansion Pack (from version 4.14).

Note: For a correct operation of the iC Device Manager service, it is required also to have the latest iClib version.

To start using the iC Device Manager service, go to the Palette window (in iC Workbench or other Niagara tool) and select the Open Palette option.

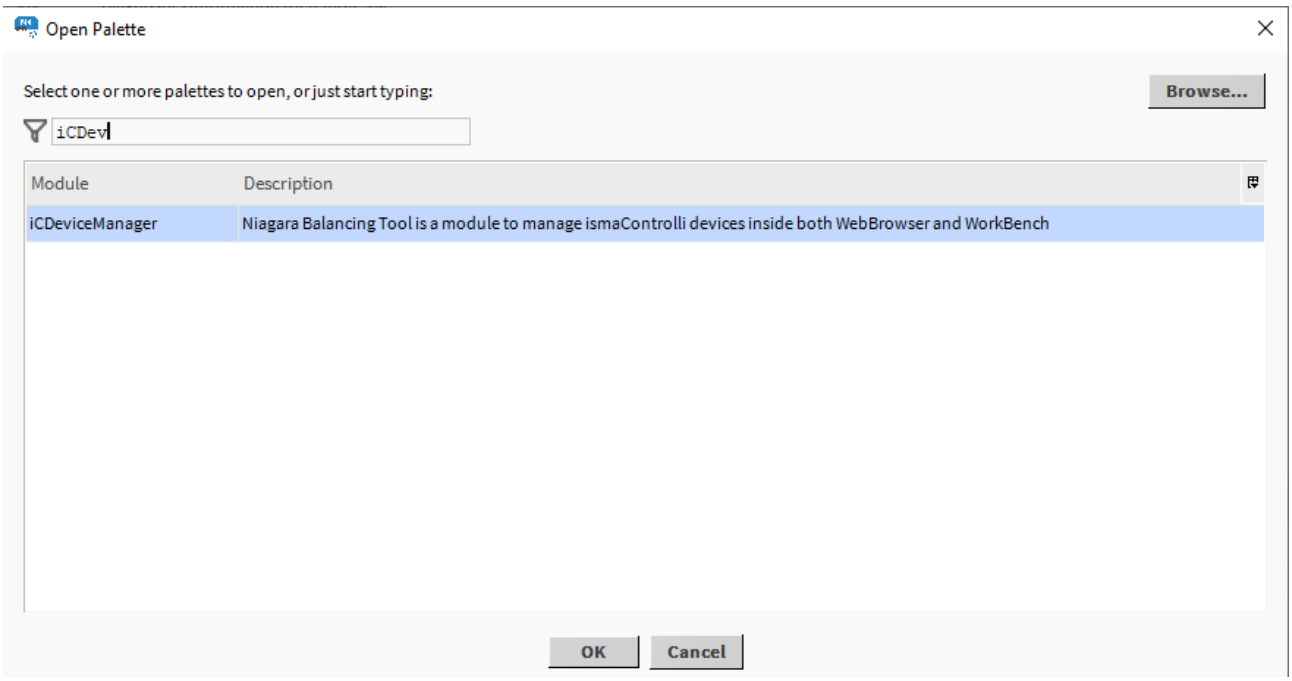


Figure 44. Opening palette

Confirm with OK, the palette is ready to use in the Palette window.

The only location where the service will operate properly is Config → Services.

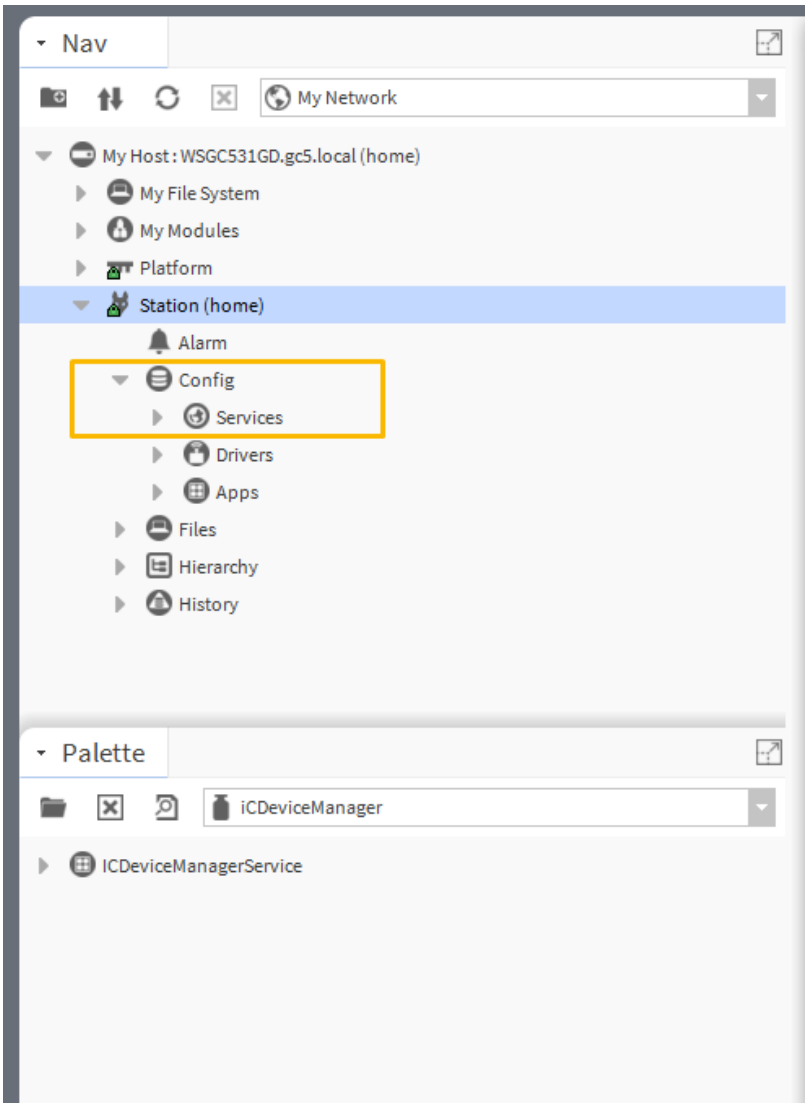
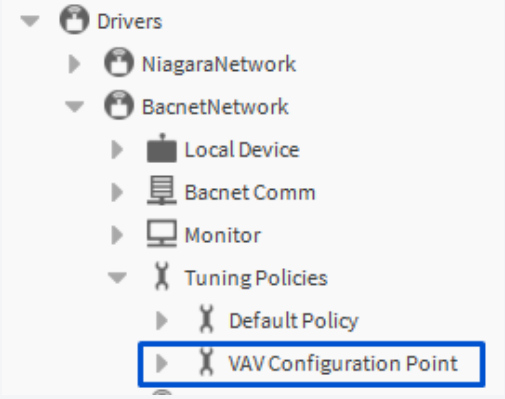


Figure 45. Services location

Drag and drop the iC Device Manager service to Services.

Warning!
For the iC Device Manager service to fully operate, make sure that the VAV14-IP controller is added to the BACnet network in Drivers.
If BACnet has been configured using the default BACnetNetwork module, it is required to add the VAV Configuration Point (from the iSMA_CONTROLLI-Library) to Tuning Policies:



Adding the VAV Device

Offline

1. Add the VAV14-IP device from the iSMA_CONTROLLI_Library (BACnetNetwork → ComfortManagement), according to user requirements: VAV14_CONFIG_POINT or VAV14_PROXY_POINT.
2. Go to BACnetNetwork in the station and invoke the Discover action.
3. Mark the offline device and the device to be matched with it, and confirm with the Match button.

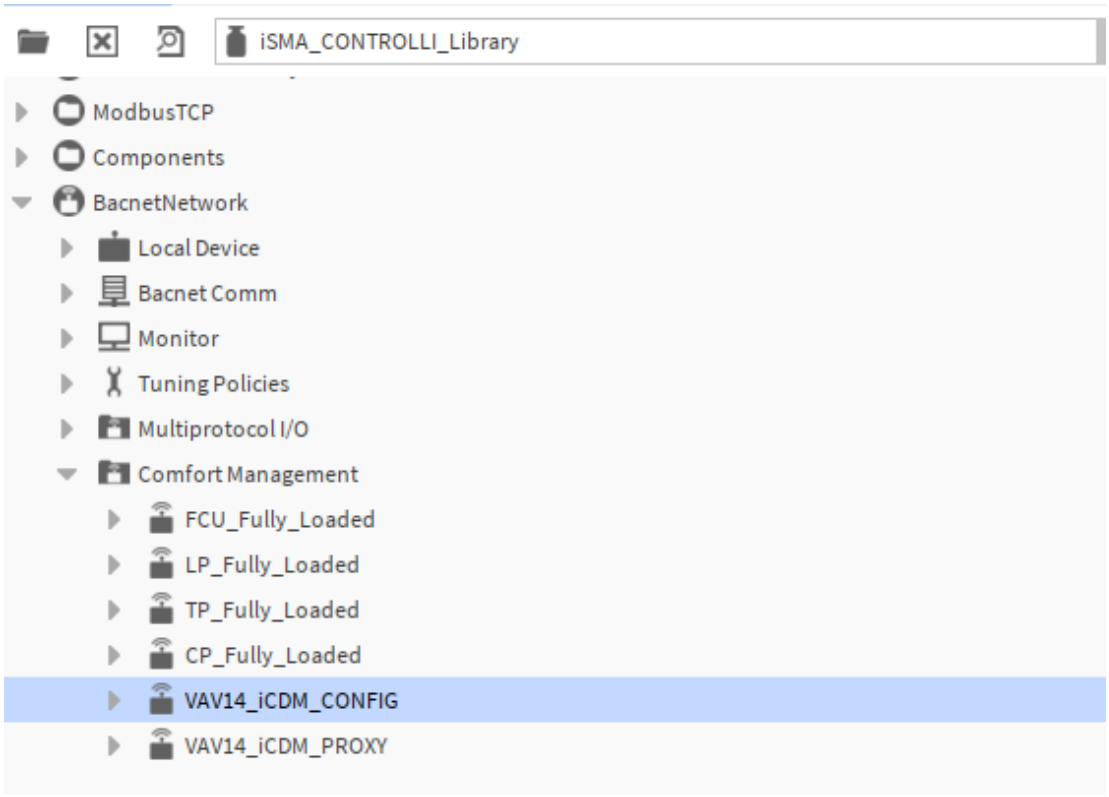


Figure 46. Config and proxy points

Online

1. Go to BACnetNetwork in the station and invoke the Discover action.
2. Add the discovered VAV14-IP device(s).
3. From the iSMA_CONTROLLI_Library select points for devices to be configured as proxy or config points.

Warning!
For a proper recognition of the device, the iC Device Manager verifies its hardware type, firmware version, and application version. From these values, only the application version is read from the AI 100 point. Make sure it is added to the device, otherwise, it may not be visible in the service.

Proxy and config points
Proxy points are BACnet points configured as proxy and placed under the Points folder. These consume Niagara license points.
Config points are BACnet points configured as config and placed under the Config folder. These points do not consume Niagara license points.

App Configuration

The first tab of the iC Device Manager service is the App Configuration tab. Here, it is possible to execute three basic actions and configure VAV application parameters.

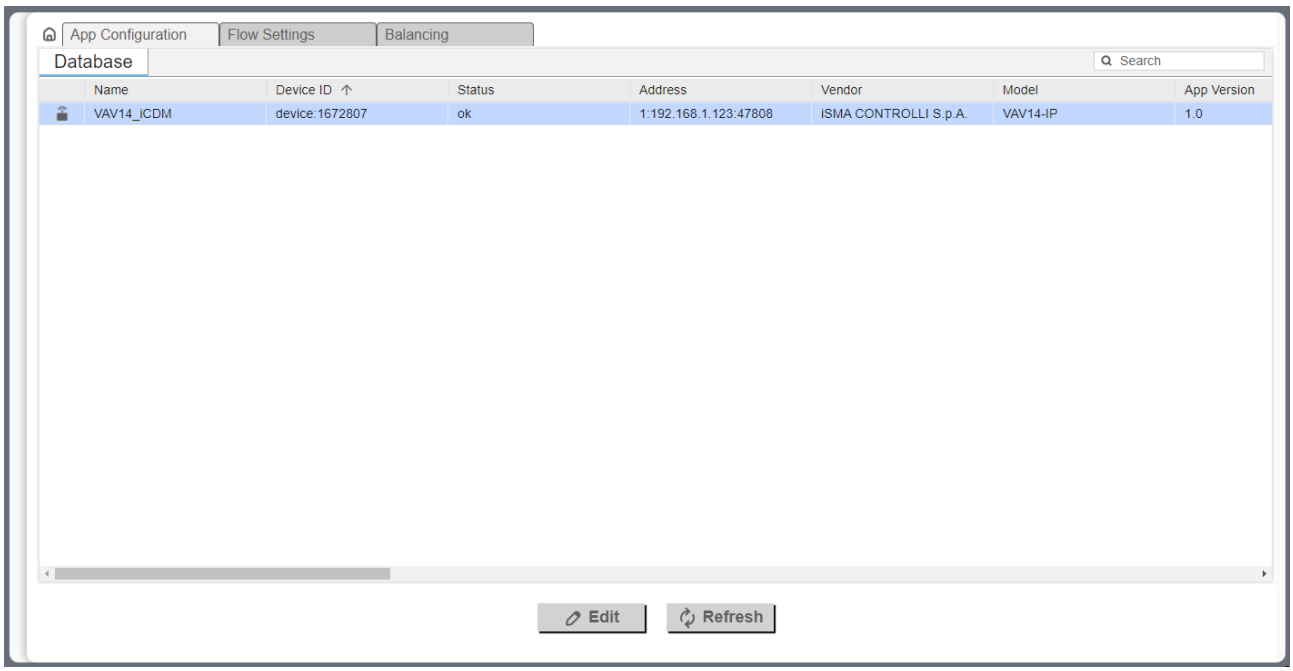


Figure 47. Application configuration main view

The main view of the tab shows data read from the device.

To make sure that the data are up to date, click the Refresh button.

To enter the application settings, click the Edit button. The configuration window pops up, where it is possible to set new values to the VAV application parameters, send them to the device (or download current settings).

Worth to notice

The configuration windows contains all parameters available to set and available actions. If any parameter from the list is meant to be left **as is** and **not** to be set to a new value, check the null option. If the null option is checked, then this parameter will be omitted when sending new values to the controller.

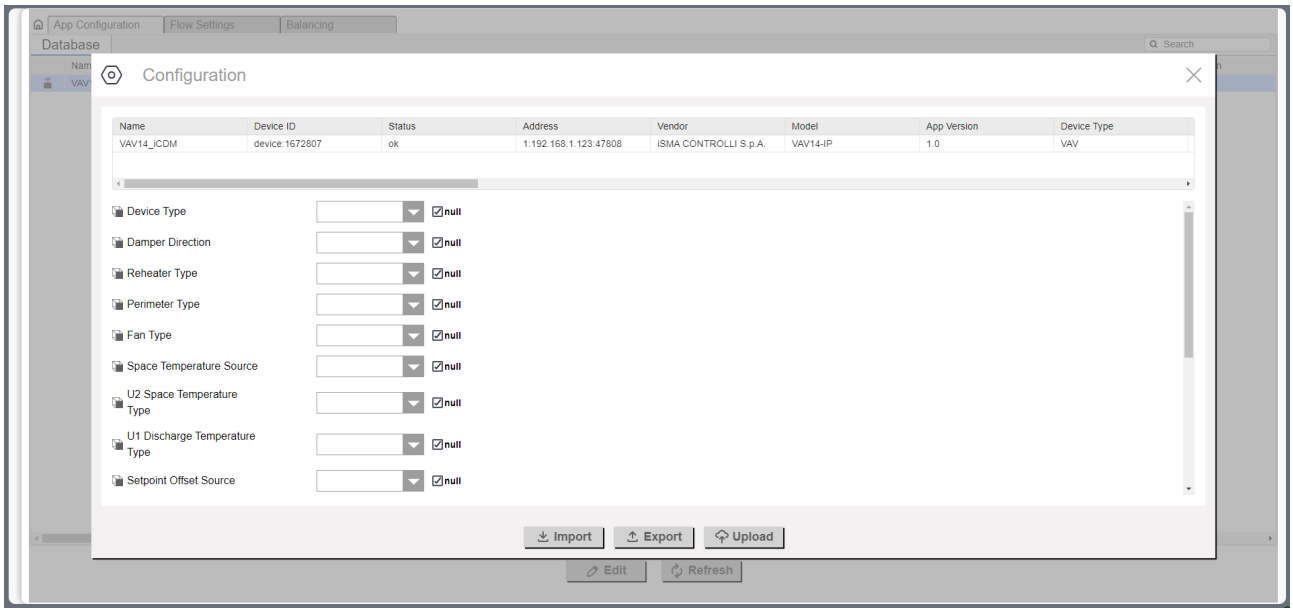


Figure 48. Application configuration details

The three actions available in the configuration tab are:

- **Import:** allows to upload application parameters from a previously saved file (*.json);
- **Export:** allows to save current application settings to a .json file;
- **Upload:** sends new settings the the device.

Warning!

Please note that if the switch 1 on the S3 DIP switch is on, the Upload action will take no effect.

The VAV application parameters available to configure are:

- **Device Type:** allows to choose a device type between VAV and VVT;
- **Damper Direction:** allows to set a damper direction to clockwise or counterclockwise;
- **Reheater Type:** allows to set a reheater type;
  - Available settings: none, TimeProp (PWM), staged, float;
- **Perimeter Type:** allows to set a perimeter type;
  - Available settings: none, TimeProp (PWM), digital;
- **Fan Type:** allows to set a fan type;
  - Available settings: none, series, parallel;
- **Space Temperature Source:** allows to set a source for temperature readings;
  - Available settings: panel, input, network;
- **U1 Discharge Temperature Type:** allows to set a discharge temperature type on the universal input 1;
  - Available settings: voltage measurement, current, resistance input, specific temperature sensor;
- **U2 Space Temperature Type:** allows to set a space temperature type on the universal input 2;
  - Available settings: voltage measurement, current, resistance input, specific temperature sensor;
- **Setpoint Offset Source:** allows to set a source for setpoint offset settings;
  - Available settings: panel, input, network;
- **Setpoint Offset Range:** allows to set a range for setpoint offset settings;
- **CO2 Source:** allows to set a source for CO2 readings;
  - Available settings: panel, input, network;
- **CO2 Range:** allows to set a range for CO2 alarm settings;
- **Heater Priority:** allows to select a priority order for the heat source;
  - Available settings: reheater, perimeter, simultaneous;
- **Dual Heat:** allows to enable a possibility to use one or two heat sources;
  - Available settings: no, yes;
- **Occupancy Contact Type:** allows to select an occupancy contact type;
  - Available settings: normal, invert;
- **Window Contact Type:** allows to select a window contact type;

    - Available settings: normal, invert;
- **Units:** allows to select a units system;
    - Available settings: imperial, metric.

Flow Settings

The second tab of the iC Device Manager service is the Flow Settings tab. Here, it is possible to execute three basic actions and set flow parameters for the use of VAV application.

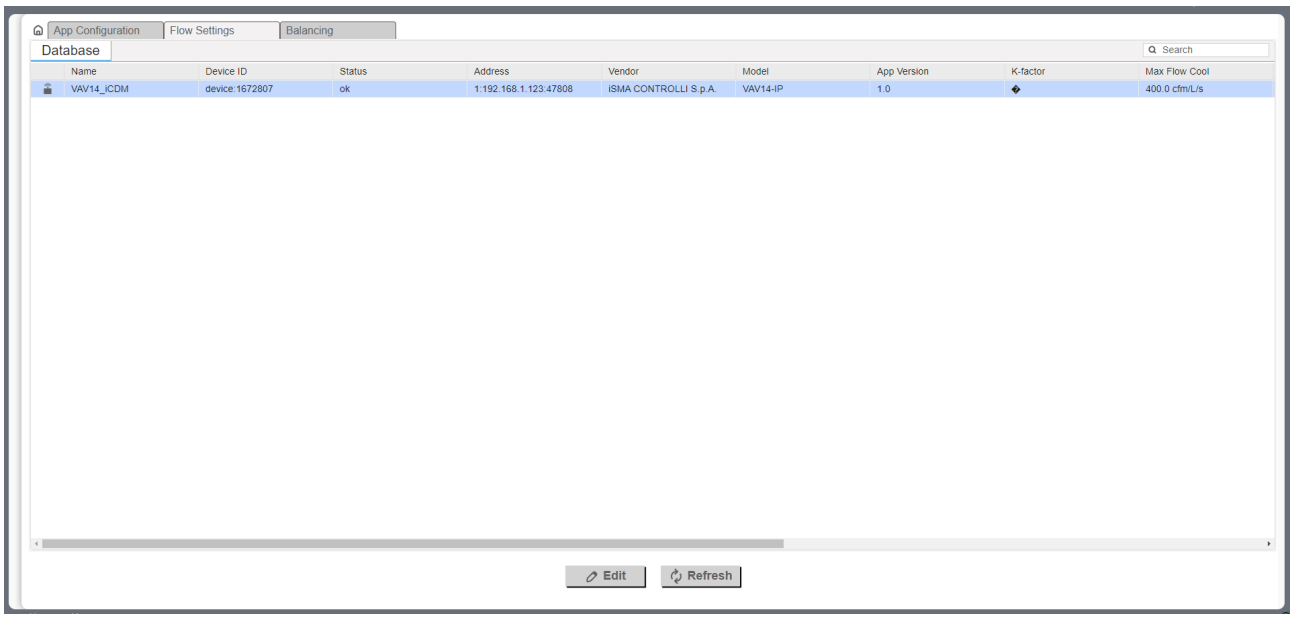


Figure 49. Airflow settings main view

The main view of the tab shows data read from the device.

To make sure that the data are up to date, click the Refresh button.

To enter the application settings, click the Edit button. The configuration window pops up, where it is possible to set new values to the VAV application parameters, send them to the device (or download current settings).

Worth to notice

The configuration windows contains all parameters available to set and available actions. If any parameter from the list is meant to be left **as is** and **not** to be set to a new value, check the null option. If the null option is checked, then this parameter will be omitted when sending new values to the controller.

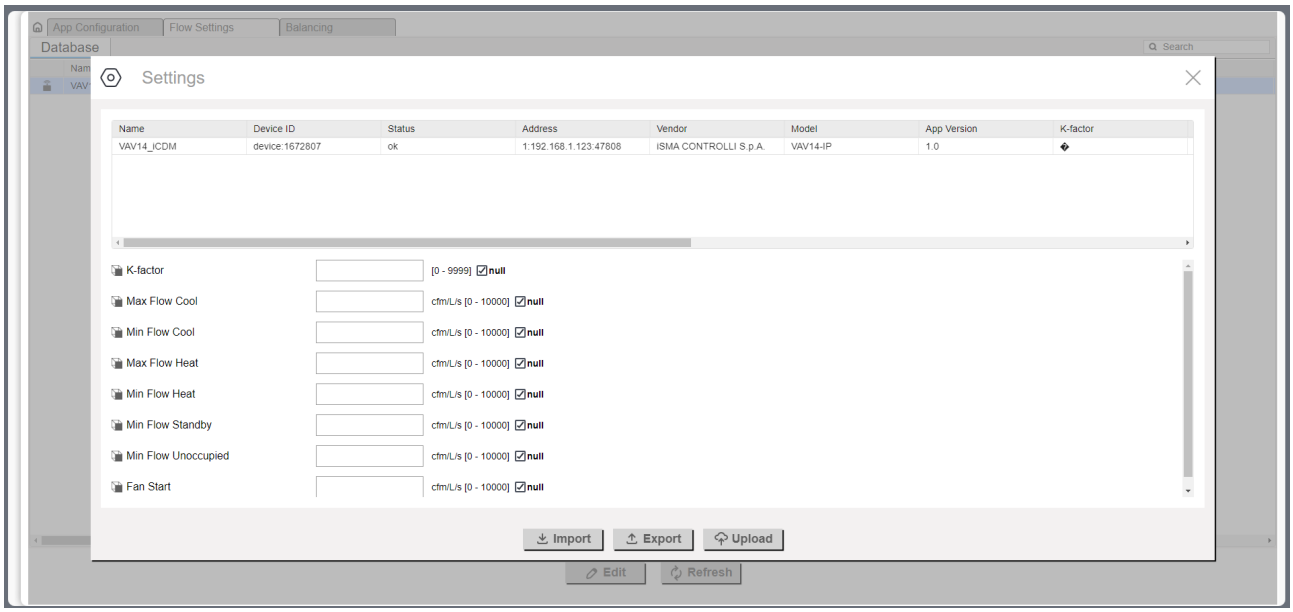


Figure 50. Airflow settings window

The three available actions are:

- **Import:** allows to upload airflow parameters from a previously saved file (*.json);
- **Export:** allows to save current airflow settings to a .json file;
- **Upload:** sends new settings the the device.

Warning!

Please note that if the switch 1 on the S3 DIP switch is on, the Upload action will take no effect.

The VAV application airflow parameters available to configure are:

- **K-Factor:** allows to set a K-factor value according to the type of device for use in the airflow calculations;
- **Max Flow Cool:** allows to set a maximum airflow value for cooling;
- **Min Flow Cool:** allows to set a minimum airflow value for cooling;
- **Max Flow Heat:** allows to set a maximum airflow value for heating;
- **Min Flow Heat:** allows to set a minimum airflow value for heating;
- **Min Flow for Standby:** allows to set a minimum airflow value in a standby state;
- **Min Flow Unoccupied:** allows to set a minimum airflow value in an unoccupied state;
- **Fan Start:** allows to set a minimum airflow value for a fan to start operating.

4.1.6 List of Variables in Application

The below table lists all Data Points used in the VAV application as variables.

The application employs 116 licensed Data Points.

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
OccupancyCalculator								
OccupancyStatus	MDP	Unocc.	N/a	True	MSI2	22	No	Current occupancy status: 1: occupied 2: unoccupied 3: bypass 4: standby
OccupancyMode	MDP	Unocc.	N/a	True	MSV2	262	No	Allows to set room the occupancy mode from the BMS system: 1: occupied 2: unoccupied 3: bypass 4: standby
PanelOccupancyStatus	MDP	Unocc.	N/a	True	MSV3	700	No	Current room occupancy status dedicated to Control Point room panel 1: unoccupied 2: occupied 3: standby 4: bypass
PanelOccupancyMode	MDP	Unocc.	N/a	True	MSV4	701	No	Status of set room occupancy mode from Control Point room panel 1: unoccupied 2: occupied
PresenceSensor	BDP	N/a (derived from the app)	True/ false	True	BI0	301	No	Actual presence sensor state from the I1 input

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
WindowContact	BDP	N/a (derived from the app)	True/ false	True	BI1	303	No	Actual window contact state from the I2 input
BypassTimeOverride	ADP	120	s	True	AV34	234	Yes	Time required to maintain the bypass occupancy status (after expiration it returns to the previous status)
StandbyTimeOverride	ADP	15	s	True	AV35	235	Yes	Time required to change the occupancy status to standby (after motion detection disappears)
AutoOccMode	BDP	Disabled	True/ false	True	BV0	100	Yes	Automatically changes occupancy to occupied when increased airflow is detected: false (disabled), true (enabled)
PanelOccupancyReset	MDP	Unocc.	N/a	True	MSV5	702	No	Resetting the room occupancy mode from occupied to unoccupied in the Control Point room panel 1: unoccupied 2: occupied
PresenceSensorInvert	BDP	Invert	True/ false	True	BV4	104	Yes	Allows to invert the I1 input signal if required: false (normal), true (invert)
WindowContactInvert	BDP	Normal	True/ false	True	BV5	105	Yes	Allows to invert the I2 input signal if required: false (normal), true (invert)
TemperatureSelector								

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
SpaceTemperature	ADP	N/a (derived from the app)	°C/°F	True	AI1	4	No	Calculated space temperature
NetTemperature	ADP	-327	°C/°F	True	AV10	210	No	Temperature sensor value from the network
U2SpaceTemperature	ADP	N/a (derived from the app)	°C/°F	True	AI3	3	No	Actual temperature sensor value from the local U2 input
PanelTemperature	ADP	N/a (derived from the app)	°C/°F	True	AI4	300	No	Actual temperature sensor value from the room panel
DischargeTemperature	ADP	N/a (derived from the app)	°C/°F	True	AI13	13	No	Calculated discharge temperature
NetDuctInTemp	ADP	-327	°C/°F	True	AV18	218	No	Discharge temperature sensor value from the network
U1DischTemperature	ADP	N/a (derived from the app)	°C/°F	True	AI16	16	No	Actual discharge temperature sensor value comes from the U1 input

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
U2SpaceTempType	MDP	10K3A1 NTC	°C/°F	True	MSV22	282	Yes	Allows to select the space temperature type: 1: Voltage measurement 2: Current 3: Resistance input 4: 10K3A1 NTC 5: 10K4A1 NTC 6: 10K NTC Carel 7: 20K6A1 NTC 8: 2,2K3A1 NTC 9: 3K3A1 NTC 10: 30K6A1 NTC 11: SIE1 12: TAC1 13: SAT1 14: PT1000 15: NI1000 16: NI1000 21C 17: NI1000 LG F 18: 10K Type2 NTC F 19: 10K Type3 NTC F 20: 20K NTC F 21: 3K NTC F 22: PT1000 F 23: NI1000 32F 24: NI1000 70F

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
U1DischargeTemp Type	MDP	10K3A 1 NTC	°C/°F	True	MSV21	281	Yes	Allows to select the discharge temperature type: 1: Voltage measurement 2: Current 3: Resistance input 4: 10K3A1 NTC 5: 10K4A1 NTC 6: 10K NTC Carel 7: 20K6A1 NTC 8: 2,2K3A1 NTC 9: 3K3A1 NTC 10: 30K6A1 NTC 11: SIE1 12: TAC1 13: SAT1 14: PT1000 15: NI1000 16: NI1000 21C 17: NI1000 LG F 18: 10K Type2 NTC F 19: 10K Type3 NTC F 20: 20K NTC F 21: 3K NTC F 22: PT1000 F 23: NI1000 32F 24: NI1000 70F Connected to CP MSV12
TemperatureInput Selector	MDP	Panel	N/a	True	MSV23	283	Yes	Allows to choose the temperature source for control logic: 1: panel 2: input 3: network
OffsetCalculator								
OffsetTempSetpoi nt	ADP	N/a (derive d from the app)	°C/°F	True	AV58	503	No	Temperature offset for setpoint from the room panel

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
SetpointOffsetRange	ADP	6	N/a	True	AV37	237	Yes	Sets the setpoint offset range to be changed by the user (e.g., a value of 5 means a range from -5 to +5), (min = 0, max = 10)
U3SetpointOffset	ADP	N/a (derived from the app)	°C/°F	True	AI17	17	No	Actual setpoint offset value (calculated from resistance) from the U3 input
PanelSetpointDisplay	BDP	N/a (derived from the app)	True/ false	True	BO56	156	Yes	Allows to select which value to be displayed during the offset editing according to the set Panel Setpoint Mode false (offset), true (setpoint)
NetOffsetSetpoint	ADP	0	°C/°F		AV41	241	No	Offset temperature setpoint value from the network
SetpointOffsetSelector	MDP	Panel	N/a		MSV24	284	Yes	Allows to choose the setpoint offset source for control logic: 1: panel 2: input 3: network
TemperatureSetpointSelector								
PanelSetpointMode	BDP	N/a (derived from the app)	True/ false	True	BO55	155	Yes	Sets the temperature setpoint mode in the Control Point room panel - directly or as an offset false (offset), true (setpoint)
OccCoolTempSetpoint	ADP	23°C/ 74°F	°C/°F	True	AV0	200	Yes	Temperature setpoint for cooling in the occupied state
OccHeatTempSetpoint	ADP	21°C/ 70°F	°C/°F	True	AV1	201	Yes	Temperature setpoint for heating in the occupied state

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
StandbyCoolTemp Setpoint	ADP	25°C/ 77°F	°C/°F	True	AV2	202	Yes	Temperature setpoint for cooling in the standby state
StandbyHeatTemp Setpoint	ADP	19°C/ 67°F	°C/°F	True	AV3	203	Yes	Temperature setpoint for heating in the standby state
UnoccCoolTempS etpoint	ADP	28°C/ 80°F	°C/°F	True	AV4	204	Yes	Temperature setpoint for cooling in the unoccupied state
UnoccHeatTempS etpoint	ADP	16°C/ 64°F	°C/°F	True	AV5	205	Yes	Temperature setpoint for heating in the unoccupied state
EffectiveTempSetp oint	ADP	N/a (derive d from the app)	°C/°F	True	AI9	9	No	Calculated effective temperature setpoint based on occupancy status and heating or cooling temperature setpoints (depending on demand)
CentralTempSetpo int	ADP	N/a (derive d from the app)	°C/°F	True	AI10	10	No	Calculated Central Temperature Setpoint based on setpoints and occupancy status
PanelTempSetpoin t	ADP	N/a (derive d from the app)	°C/°F	True	AV56	501	No	Temperature setpoint from the room panel
MaxDischTempSet point	ADP	32°C/ 90°F	°C/°F	True	AV7	207	Yes	High limit of discharge temperature supplied to the room
ActDischTempSet point	ADP	N/a (derive d from the app)	°C/°F	True	AI14	14	No	Calculated discharge temperature setpoint based on the effective temperature setpoint and heating demand for high-limiting discharge temperature
MinDischTempSet point	ADP	13°C/ 55°F	°C/°F	True	AV8	208	Yes	Low limit of discharge temperature supplied to the room

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
CO2Selector								
PanelCO2	ADP	N/a (derived from the app)	ppm	True	AI6	302	No	Actual CO_2 sensor value from the room panel
NetCO2	ADP	N/a (derived from the app)	ppm	True	AV40	240	No	CO_2 sensor value comes from the network
U4CO2Sensor	ADP	N/a (derived from the app)	ppm	True	AI8	8	No	Actual CO_2 sensor value from the local U4 input
SpaceCO2	ADP	N/a (derived from the app)	ppm	True	AI5	5	Yes	Calculated space CO_2
U4CO2SensorRange	ADP	2000	N/a	True	AV38	238	Yes	Sets the range of the CO_2 sensor used (min = 0, max = 10000)
CO2InputSelector	MDP	Panel	N/a		MSV25	285	Yes	Allows to select the CO2 source for control logic: 1: panel 2: input 3: network
AirflowCalculator								
K-Factor	ADP	1000	N/a	True	AV61	288	Yes	Sets the K-Factor (the flow rate at a differential pressure of 1inWC/1Pa on the measuring cross) for the given VAV model (min = 0)

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
DifferentialPressure	ADP	N/a (derived from the app)	Pa/WC	True	AI15	15	No	Actual differential pressure sensor-measured value from the built-in pressure input
ActualAirFlow	ADP	N/a (derived from the app)	L/s/cfm	True	AI11	11	No	Actual calculated airflow based on the K-Factor value and differential pressure measure
AirflowCalibration	ADP	0	L/s/cfm	True	AV66	293	No	Allows to enter the airflow value measured by the hood for calibration
CalculatedKFactor	ADP	N/a (derived from the app)	N/a	False	AV62	289	No	Calculated K-Factor value (meaning the flow rate at a differential pressure of 1 inWC/1 Pa on the measuring cross) based on the entered measured airflow (using the hood) and the measured differential pressure from the built-in PI input
PressureZeroing	BDP	No	True/ false	True	BV13	113	No	Allows to perform zeroing action on the built-in PI pressure transducer if there is a non-zero value in the absence of airflow. **Important:** perform in the absence of airflow through the pressure transducer ports False (no), true (zeroing)
AirflowSetpointCalculator								
EffectiveAirFlowSetpoint	ADP	N/a (derived from the app)	L/s/cfm	True	AI12	12	No	Calculated effective airflow setpoint based on the occupancy status and heating or cooling airflow setpoints

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
HVACMode	MDP	Auto	N/a	True	MSV1	261	No	Allows to set the operating mode for the VAV device: 1: Auto 2: Heat 3: Morning WarmUp 4: Cool 5: Night Purge 6: PreCool 7: Off (depressurize) 8: Fire (pressurize)
MaxOccCoolAirflowSetpoint	ADP	400	L/s/cfm	True	AV11	211	Yes	Maximum airflow setpoint for cooling in the occupied state (min = 0)
MaxOccHeatAirflowSetpoint	ADP	200	L/s/cfm	True	AV13	213	Yes	Minimum airflow setpoint for cooling in the occupied state (min = 0)
MinOccCoolAirflowSetpoint	ADP	100	L/s/cfm	True	AV12	212	Yes	Maximum airflow setpoint for heating in the occupied state (min = 0)
MinOccHeatAirflowSetpoint	ADP	100	L/s/cfm	True	AV14	214	Yes	Minimum airflow setpoint for heating in the occupied state (min = 0)
StandbyAirflowSetpoint	ADP	50	L/s/cfm	True	AV15	215	Yes	Airflow setpoint in the standby state (min = 0)
UnoccAirflowSetpoint	ADP	25	L/s/cfm	True	AV16	216	Yes	Airflow setpoint in the unoccupied state (min = 0)
StartupFanAirflowSetpoint	ADP	35	L/s/cfm	True	AV17	217	Yes	Airflow setpoint for a fan startup only (min = 0)
UserSetAirflow	ADP	0	L/s/cfm	True	AV65	292	No	User-set airflow setpoint (after selecting the User Position mode in the Damper Command variable) (min = 0)

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
Heat&CoolLoops								
HotAirTempSetpoint	ADP	26°C/ 78°F	°C/°F	True	AV9	209	Yes	Discharge temperature setpoint for detecting warm air to change the heating/ cooling mode
PanelHeating	BDP	N/a (derived from the app)	True/ false	True	BO16	116	No	Heating mode status to be displayed as an icon on the Control Point room panel false (no heating), true (heating)
PanelCooling	BDP	N/a (derived from the app)	True/ false	True	BO17	117	No	Cooling mode status to be displayed as an icon on the Control Point room panel false (no cooling), true (cooling)
ChangeoverDelay	ADP	2	min	True	AV36	236	No	Delay time for changing the heating/ cooling operating mode
Airflow/DAT/CO2Loops								
SpaceCO2Setpoint	ADP	1000	ppm	True	AV6	206	Yes	CO_2 setpoint for the room (regardless of the occupancy status)
AHUOutdoorDamper	ADP	N/a (derived from the app)	%	True	AO4	254	No	Required opening level of the fresh air damper in the AHU resulting from the current CO_2 level in the room (network signal to the AHU control)
DamperControl								

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
DeviceType	BDP	VAV	True/ false	True	BV6	106	Yes	Allows to choose VAV device control type from airflow demand (VAV also known as pressure independent) or from cooling/ heating demand (VVT also known as pressure dependent): false (VAV), true (VVT)
MaxDamperPosition	ADP	100	%	True	AV21	221	Yes	Maximum damper open position limit (min = 0, max = 100)
MinDamperPosition	ADP	0	%	True	AV22	222	Yes	Minimum damper open position limit (min = 0, max = 100)
UserSetPosition	ADP	0	%	True	AV64	291	No	User-set damper opening level (after selecting the User Position mode in the Damper Command variable) (min = 0, max = 100)
DualHeat	BDP	No	True/ false	True	BV8	108	Yes	Allows to enable dual heating mode with repeater/perimeter and damper (no effect without damper): false (no), true (yes)
HeatDamperPriority	BDP	Secondary	True/ false	True	BV11	111	Yes	Allows to set the priority of the damper heating operation when working in the active DualHeat mode: false (secondary), true (primary)
DamperPosition	ADP	N/a (derived from the app)	%	True	AO0	250	No	Actual damper opening level coming from the Rotary Actuator component (Position Feedback signal)

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
DamperResponse	ADP	2	%	True	AV26	226	Yes	Sets the value of how much the control signal must change to perform the next damper movement
DamperDriveTime	ADP	90	s	True	AV25	225	Yes	Time required to reach full damper open position (from 0% to 100%) (min = 0)
DirectionActuator	BDP	Clockwise	True/ false	True	BV9	109	Yes	Sets the damper opening direction: false (CW), true (CCW)
DamperControl	ADP	0	%	True	AV67	294	No	Damper opening level control resulting from the application
DamperCommand	MDP	None	N/a	True	MSV7	267	No	Allows to set different damper command modes for calibration, balancing, or testing: 1: Auto/none (null) 2: Max flow 3: Min flow 4: User flow 5: User position 6: Full open 7: Full close 8: Calibrate
FanControl								
FanCommand	MDP	Auto	N/a	True	MSV8	268	No	Allows to set different fan command modes for testing 1: Auto/none (null) 2: Max speed 3: Min speed 4: Stop
FanControlMode	BDP	Digital	True/ false	True	BV10	110	Yes	Allows to select the fan control mode: false (digital), true (analog)

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
MaxFanSpeed	ADP	100	%	True	AV23	223	Yes	Maximum fan speed level limit (min = 0, max = 100)
MinFanSpeed	ADP	20	%	True	AV24	224	Yes	Minimum fan speed level limit (min = 0, max = 100)
WarmUpParallelFan	BDP	Disabled	True/ false	True	BV2	102	Yes	Allows to use the parallel fan also in the Morning WarmUp mode (in the HVAC Mode variable): false (disabled), true (enabled)
StandbySeriesFan	BDP	Disabled	True/ false	True	BV3	103	Yes	Allows to use the series fan also in standby occupancy: false (disabled), true (enabled)
FanDelayOff	ADP	30	s	True	AV39	239	Yes	Delay time for switching off the fan
PanelFanStatus	MDP	Off	N/a	True	MSV0	600	No	Fan status to be displayed as an icon on the Control Point room panel: 1: off 5: speed 1 (auto) 6: speed 2 (auto) 7: speed 3 (auto)
FanStatus	BDP	Off	True/ false	True	BO1	121	No	Fan digital control status (if configured as digital) from the application: false (off), true (on)
FanSpeed	ADP	N/a (derived from the app)	%	True	AO2	252	No	Fan speed level control resulting from the application
ReheaterControl								

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
LoadShedding	ADP	0	%	True	AV20	220	No	Load-shedding level required from the network (min = 0, max = 100)
ReheaterCommand	MDP	Auto	N/a	True	MSV9	269	No	Allows to set different reheater command modes for testing 1: Auto/none (null) 2: Full open 3: Full close
HeaterPriority	MDP	Reheater	N/a	True	MSV14	274	Yes	Possibility to select heating priority for heating sources: 1: reheater 2: perimeter 3: simultaneous
WarmUpHeater	BDP	Disabled	True/ false	True	BV1	101	Yes	Allows to use the reheater also in the Morning WarmUp mode (in the HVAC Mode variable): false (disabled), true (enabled)
ReheaterControl	ADP	N/a (derived from the app)	%	True	AO1	251	No	Reheater valve opening level control resulting from the application
TimePropReheaterPeriod	ADP	60	s	True	AV32	232	Yes	Time required for a PWM valve for a full cycle time of a reheater
FloatReheaterValveDriveTime	ADP	120	s	True	AV30	230	Yes	Time required for a PWM valve for a full cycle time of a perimeter

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
ReheaterStatus	MDP	Off	N/a	True	MSO0	290	No	Reheater staged control status (if configured as digital or staged) from application: 1: off 2: stage 1 on 3: stage 2 on
PerimeterControl								
PerimeterCommand	MDP	Auto	N/a	True	MSV10	270	No	Allows to set different perimeter command modes for testing 1: Auto/none (null) 2: Full open 3: Full close
PerimeterControl	ADP	N/a (derived from the app)	%	True	AO3	253	No	Perimeter valve opening level control resulting from the application
PerimeterStatus	BDP	Off	True/ false	True	BO0	120	No	Perimeter digital control status (if configured as digital) from the application: false (off), true (on)
TimePropPerimeterPeriod	ADP	60	s	True	AV33	233	Yes	Time required for a PWM valve for a full cycle time of a perimeter
OutsideTemperatureControl								
MaxOatReheater	ADP	32°C/ 90°F	°C/°F	True	AV28	228	Yes	Maximum outside air temperature limit that allows the use of a reheater
MaxOatPerimeter	ADP	18°C/ 65°F	°C/°F	True	AV29	229	Yes	Maximum outside air temperature limit that allows the use of a perimeter
NetOutdoorTemp	ADP	-327	°C/°F	True	AV19	219	No	Outside air temperature sensor value from the network

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
DeviceConfiguration								
DeviceConfigurationSource	BDP	BACnet	True/ false	True	BV12	112	Yes	Allows to choose to set a source of the device application configuration: false (BACcnet), true (DIP switch)
PerimeterType	MDP	None	N/a	True	MSV13	273	Yes	Allows to set the perimeter type: 1: none 2: TimeProp 3: digital
ReheaterType	MDP	None	N/a	True	MSV12	272	Yes	Allows to set the reheater type: 1: none 2: TimeProp 3: staged 4: float
FanType	MDP	None	N/a	True	MSV11	271	Yes	Allows to set the fan type: 1: none 2: series 3: parallel
Units	BDP	Metric	True/ false	True	BO2	122	Yes	Allows to change the unit in the application provided that the BACnet configuration is allowed in the Device Configuration Source variable: false (metric), true (imperial)
UpTime	ADP	N/a (derived from the app)	s	True	AI0	0	No	Device's up time
AppVersion	ADP	1.0	N/a	True	AI100	102	No	Current application version
TriacOutputsConfig								

Name	DP type	Def. Out value	Units	Exposed on BACnet/ Modbus	BACnet ID	Modbus dec. addr.	Conf. data ext.	Description
TO4	ADP	0		False	996	996	No	
TO3	ADP	0		False	997	997	No	
TO2	ADP	0		False	998	998	No	
TO1	ADP	0		False	999	999	No	

4.1.7 List of I/Os

I/O number	Signal
U1	Discharge air temperature
U2	Space temperature
U3	Setpoint offset
U4	CO_2 sensor
I1	Presence detection
I2	Window contact
A1	Reheater analog
A2	Perimeter analog
A3	Fan analog
TO1	Reheater PWM/reheater digital/reheater 1-stage/reheater close
TO2	Reheater 2-stage/reheater open
TO3	Perimeter PWM/perimeter digital
TO4	Fan digital

4.2 Balancing

4.2.1 Balancing Goals

VAV (Variable Air Volume) balancing is the process of ensuring that a building's HVAC system delivers the right amount of air to different spaces. **The idea of balancing is to find the most efficient operating point of the ventilation system**. Proper balancing provides:

- energy efficiency: minimum consumption of electricity, heating and cooling;
- user comfort: temperature comfort, noise elimination, air quality.

It establishes a benchmark for the collaboration between AHU and VAV systems. The optimal operating point ensures the minimum amount of air required for the proper functioning of the entire VAV installation.

From a more technically oriented standpoint, there are two main goals that balancing aims to achieve:

- **Obtaining a correct flow from the pressure transducer:**
  - if the K-factor VAV box is known, set it in the application and verify the correctness of the flow;
  - if the K-factor of the VAV box is not known, force the damper to open to the maximum flow and set the measured flow in the application – the K-factor will be automatically calculated.
- **Damper direction verification:**
  - open the damper to the maximum flow – if it is not achieved, change the direction.

General aspects of balancing of ventilation VAV systems

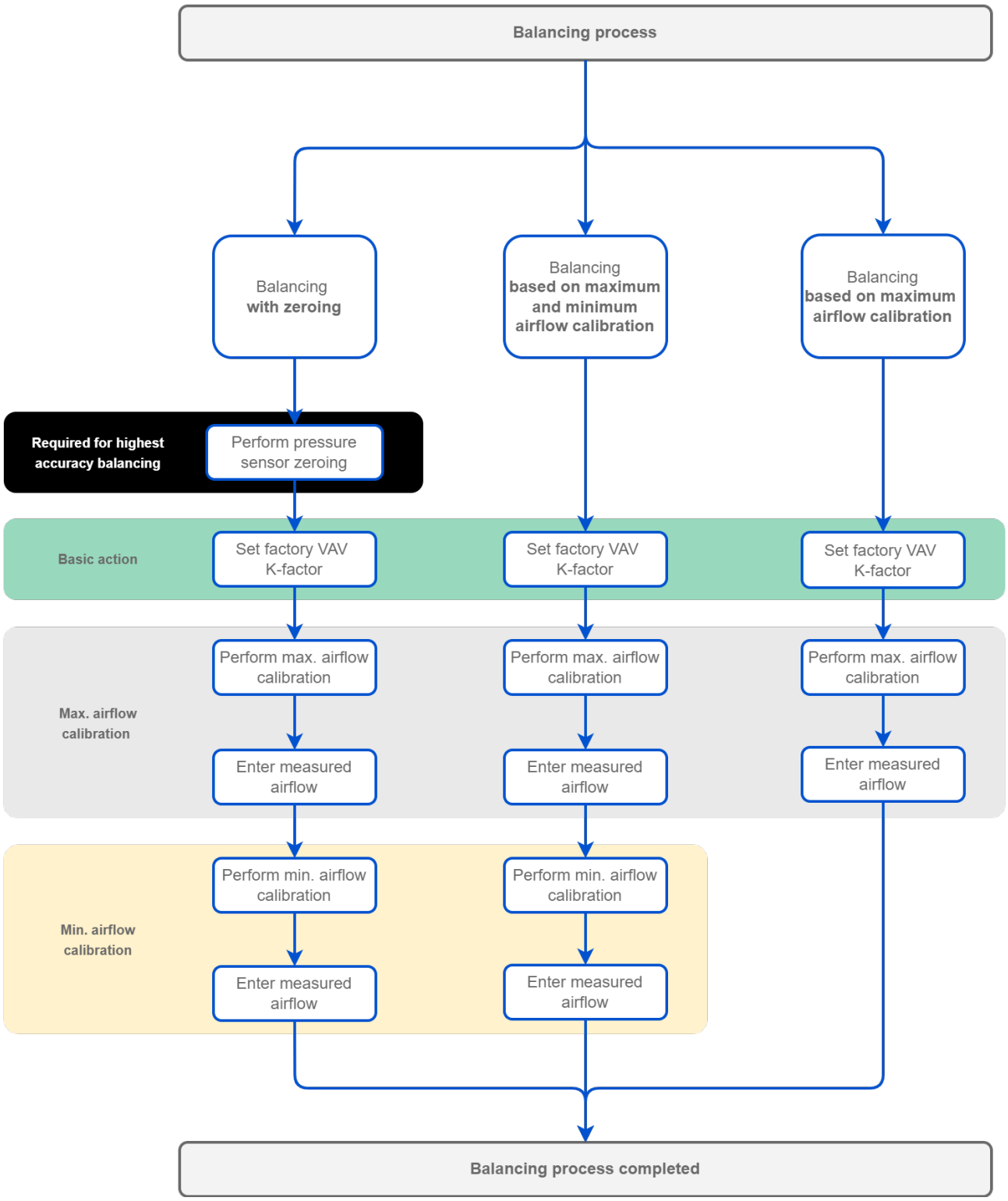
- The first step is to force all VAVs (supply and exhaust separately) to operate at their maximum defined airflow.
- Set the AHU air efficiency 5-10% higher than the ventilation design (should be consistent with the sum of the VAV air efficiency).
- The next step is to gradually reduce the AHU efficiency (supply and exhaust separately) and observe the degree of opening of the VAV dampers.
- Starting from the maximum efficiency of +10% in the AHU, a scenario is created where the VAVs are partially closed. By reducing the air efficiency of the AHU, the VAVs begin to open.
- The air efficiency of the AHU is lowered until the VAV is almost fully open (85-90%) – usually, this is the VAV at the end of the installation (from the AHU). Calibrate this VAV and recheck.
- In this way, the optimal operating point for the AHU's air efficiency is achieved, allowing all other VAVs to self-adjust.
- Calibrate the rest of the VAV.

4.2.2 VAV Balancing Process

The balancing process involves activities that aim eliminating any discrepancies in airflow adjustment and application calculations:

- pressure sensor zeroing,
- setting K-factor,
- performing maximum airflow calibration,
- performing minimum airflow calibration.

Based on the user requirements in the area of the balancing process accuracy, all of the above activities can be performed for a full and most advanced balancing outcome or some of them can be omitted. The below diagram shows activities required for the advanced, medium, and basic levels of the balancing accuracy:



1. If required, the first balancing action is pressure sensor zeroing (this action can be omitted in other variants).

Pressure Sensor Zeroing

Pressure sensor zeroing is a part of the VAV balancing process, which aims at eliminating a constant measurement error of differential pressure on a built-in pressure sensor. The zeroing process involves the following steps:

- Make sure the differential pressure sensor is disconnected from the measuring cross or other measuring method.
- Use a flexible hose of an appropriate diameter to connect the two spigots (+ and -).

- Make sure the hose is well secured and tight to equalize the pressures on both ports.
- Invoke the zeroing action using one of the methods:
    - using the action in the PressureInput component (iC Tool)
    - in the Balancing tab available in one of the tools (iSMA Configurator, iC Device Manager),
    - writing a value to the PressureZeroing variable (BACnet object: BV13, Modbus address: 13),
    - from the Control Point VAV panel.
- Detach the hose.
- Restore the normal connection of the sensor to the measuring cross or other target circuit, pay attention to the polarity.

2. Then, the common and mandatory action for all variants, setting the K-factor.

Note

The K-factor is provided by the manufacturer of the VAV box. If for any reason it is unavailable, it is recommended to use a default K-factor from the VAV application (100 l/s /1000 cfm).

3. Next action is the maximum airflow calibration. Performing it is the necessary minimum of the balancing process.
4. If required, the minimum airflow calibration is the last step of the balancing process.

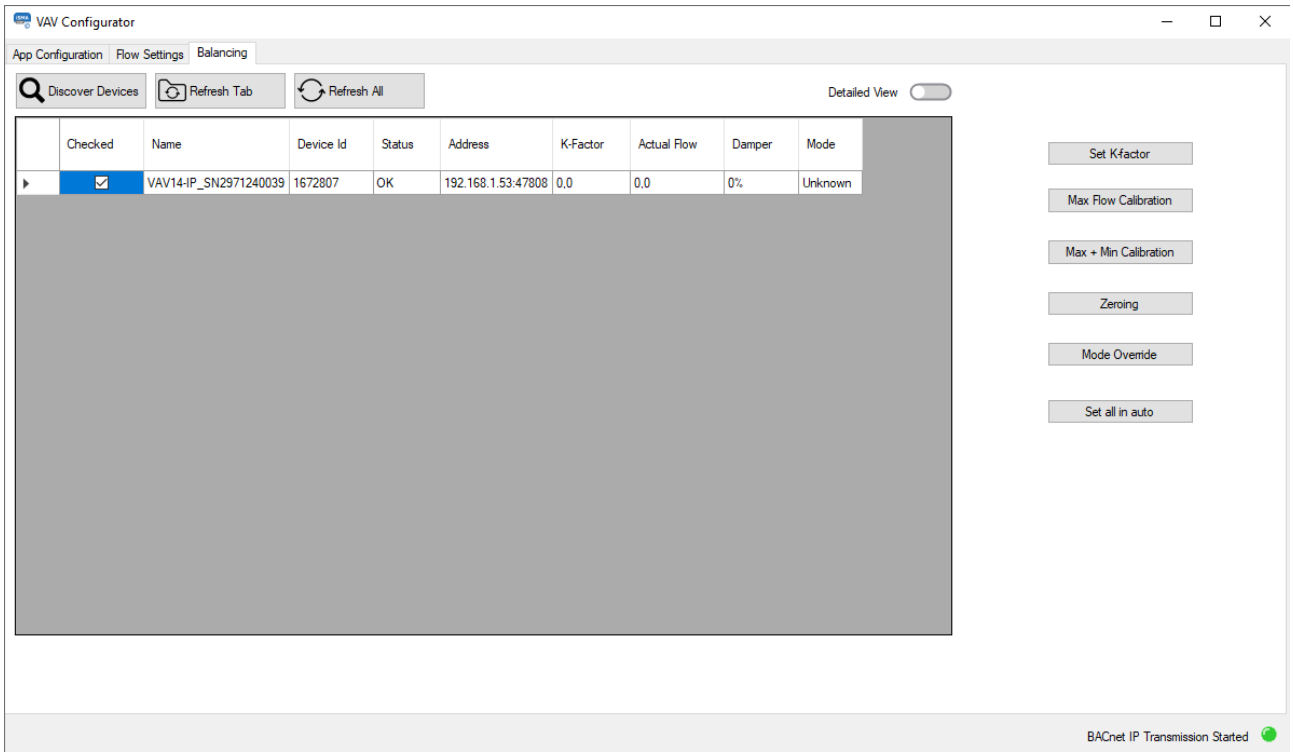
Tools

Balancing can be performed using one of the iC Tools provided for this purpose:

- iSMA Configurator,
- iC Device Manager,
- Control Point wall panel.

4.2.3 iSMA Configurator

Balancing actions in the iSMA Configurator are available in the last tab of the VAV Configurator, the Balancing tab. Here, it is possible to perform actions of the balancing process.



The available actions are:

- **Set K-Factor:** sets the K-factor value to the airflow calculations (AV61, 40289);
- **Max Flow Calibration:** executes a calibration action according to a new maximum airflow value (AV66, 40294);
- **Max + Min Calibration:** executes a calibration action according to a new maximum and minimum airflow values (AV66, 40294);
- **Zeroing:** performs the pressure sensor zeroing action (BV13, 00014);
- **Mode Override:** allows to force the device to operate in one of the available modes (MSV7, 40068):
  - Available settings: Auto, Max. Flow, Min. Flow, User Flow, User Position, Full Open, Full Close, Calibrate;
- **Set all In auto:** forces all dampers to the Auto mode (MSV7, 40068).

Balancing Steps

To perform the balancing process and regulate the efficiency point of the ventilation system, follow the below steps. These activities are to performed in the presented order. If required, perform the balancing process with zeroing, however, according to the specific user and accuracy requirements, it is possible to omit the pressure sensor zeroing and/or minimum airflow calibration.

Balancing with zeroing

1. Pressure sensor zeroing.
2. Set the factory K-factor.
3. Measure the airflow and enter the value in the designated field.
4. Perform the maximum airflow calibration.
5. Measure the airflow and enter the value in the designated field.
6. Perform the minimum airflow calibration.

Balancing based on the maximum and minimum airflow calibration

1. Set the factory K-factor.
2. Measure the airflow and enter the value in the designated field.
3. Perform the maximum airflow calibration.
4. Measure the airflow and enter the value in the designated field.
5. Perform the minimum airflow calibration.

Balancing based on the maximum airflow calibration

1. Set the factory K-factor.
2. Measure the airflow and enter the value in the designated field.
3. Perform the maximum airflow calibration.

Pressure Sensor Zeroing

If required, pressure sensor zeroing is an initial action of the balancing process. According to the specific user and accuracy requirements, this action can be omitted in the balancing process.

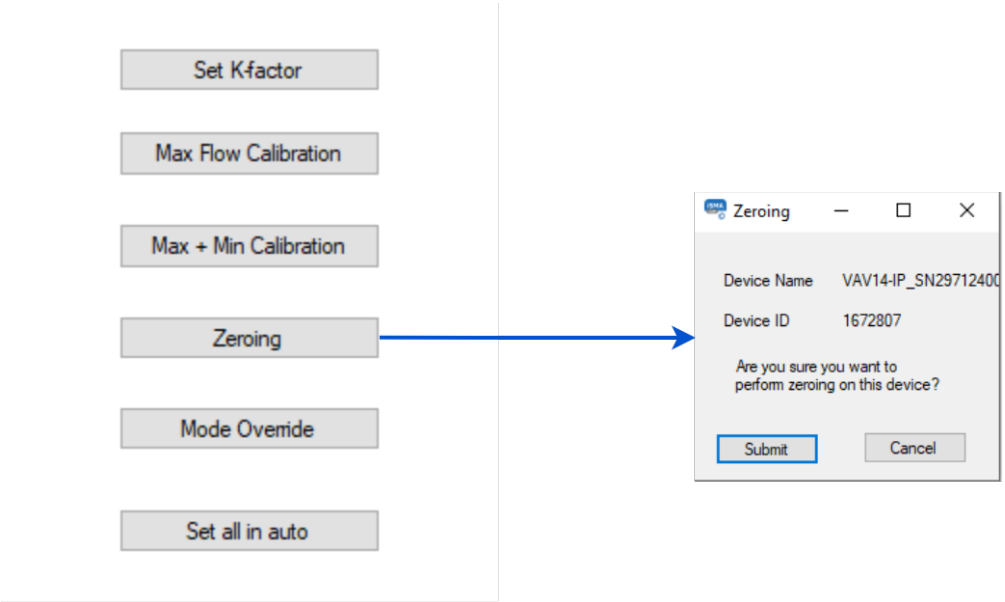


Figure 51. Zeroing action pop-up

Pressure sensor zeroing is a part of the VAV balancing process, which aims at eliminating a constant measurement error of differential pressure on a built-in pressure sensor. The zeroing process involves the following steps:

- Make sure the differential pressure sensor is disconnected from the measuring cross or other measuring method.
- Use a flexible hose of an appropriate diameter to connect the two spigots (+ and -).
- Make sure the hose is well secured and tight to equalize the pressures on both ports.
- Invoke the zeroing action.
- Detach the hose.
- Restore the normal connection of the sensor to the measuring cross or other target circuit, pay attention to the polarity.

Setting K-factor

Setting the K-factor is a mandatory part of the balancing process. It is normally provided by the manufacturer of the VAV box. If for any reason it is unavailable, it is recommended to use a default K-factor from the VAV application (100 l/s /1000 cfm).

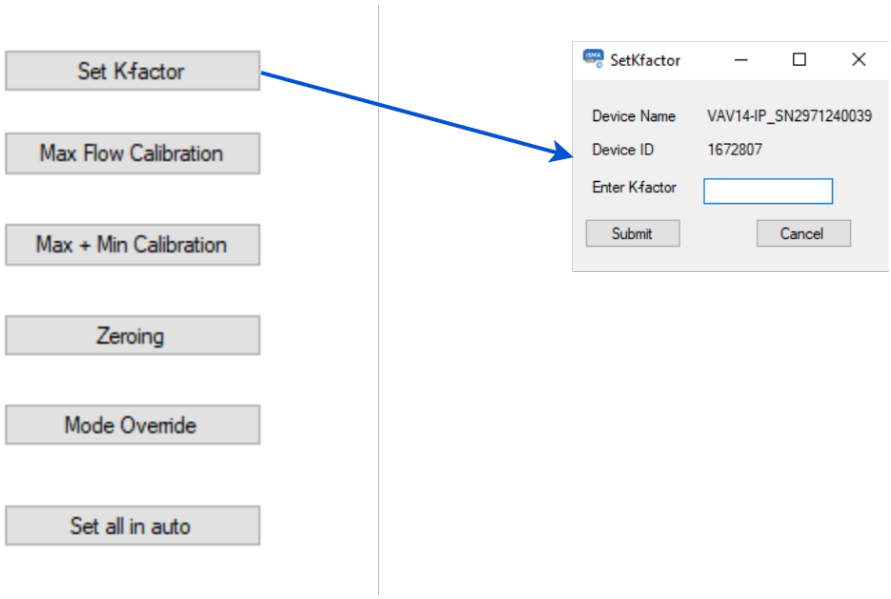


Figure 52. Setting K-factor pop-up

Maximum and Minimum Airflow Calibration

To perform the full balancing process, proceed to the maximum and minimum airflow calibration action, under the Max. + Min. Calibration button. The action invokes two pop-up windows, first for the maximum calibration in step 1, and second for the minimum airflow calibration in step 2.

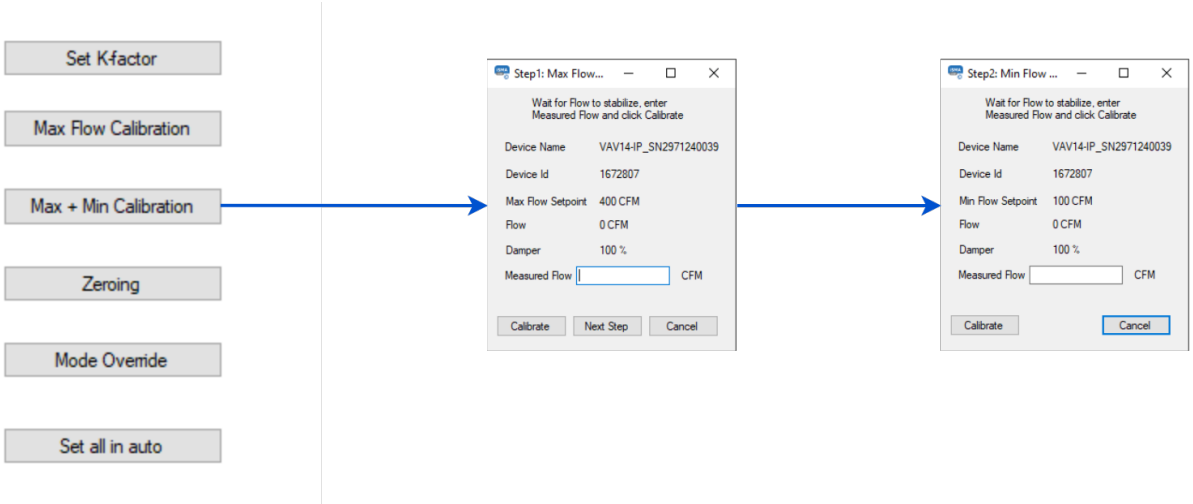


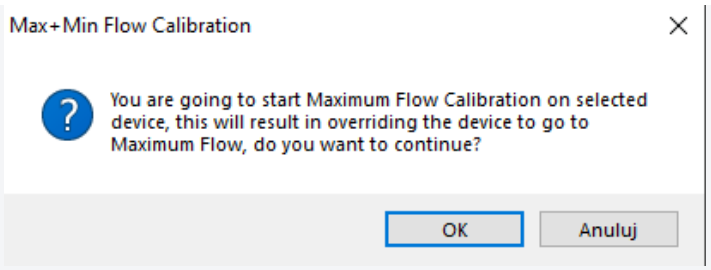
Figure 53. Maximum and minimum airflow calibration pop-ups

In step 1, measure the airflow and enter the measured value. Then, proceed to calibration (using the Calibrate button). The action will return a newly calculated K-factor.

After completing step 1, continue to step 2, the minimum airflow calibration. Enter the measured airflow value and use the Calibrate button.

Note

Invoking this action prompts an override warning pop-up:



Maximum Airflow Calibration

If, for any reason, the minimum airflow calibration is not required, it is possible to perform only the maximum airflow calibration. Measure the airflow and enter the measured value. Then, proceed to calibration (using the Calibrate button). The action will return a newly calculated K-factor.

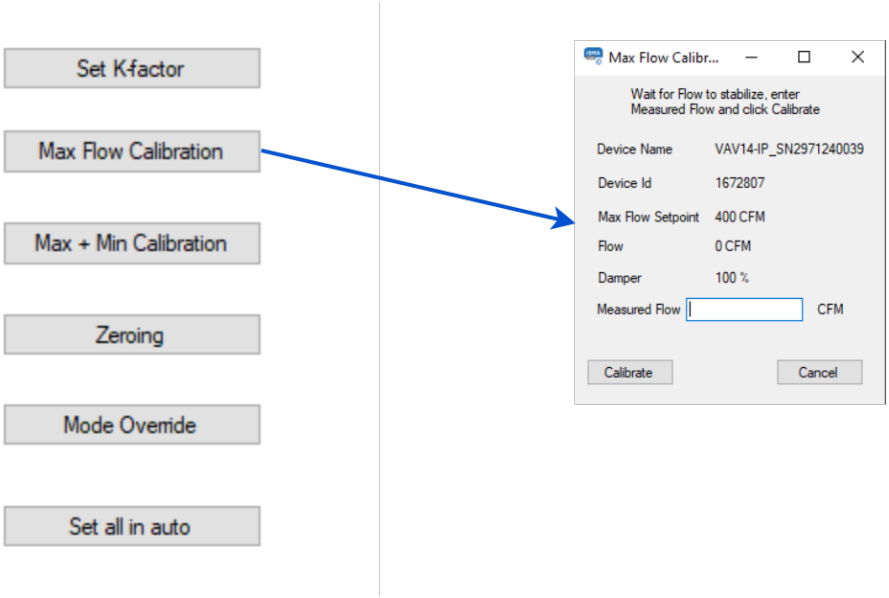
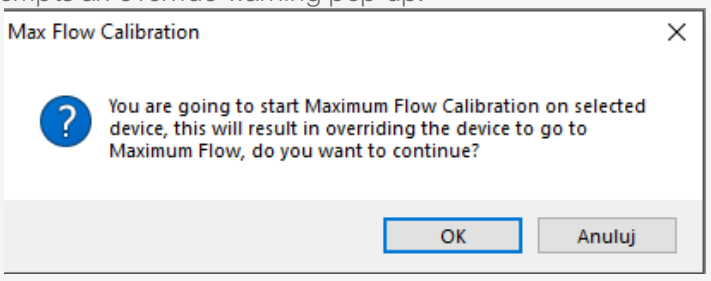


Figure 54. Maximum airflow calibration pop-up

Note

Invoking this action prompts an override warning pop-up:



Other Actions

Mode Override

The Mode Override action allows to force an operating mode to a damper. Available modes are:

- **Auto:** sets auto mode (VAV application logic takes control),
- **Max flow:** goes to a maximum airflow,
- **Min flow:** goes to a minimum airflow,
- **User flow:** goes to a user-set airflow,
- **User position:** goes to a user-set position (% of damper opening),
- **Full open:** damper fully open,

- **Full close:** damper fully closed,
- **Calibrate:** performs a damper calibration (the damper goes to 100%, than to 0%, and goes back to the control loop output).

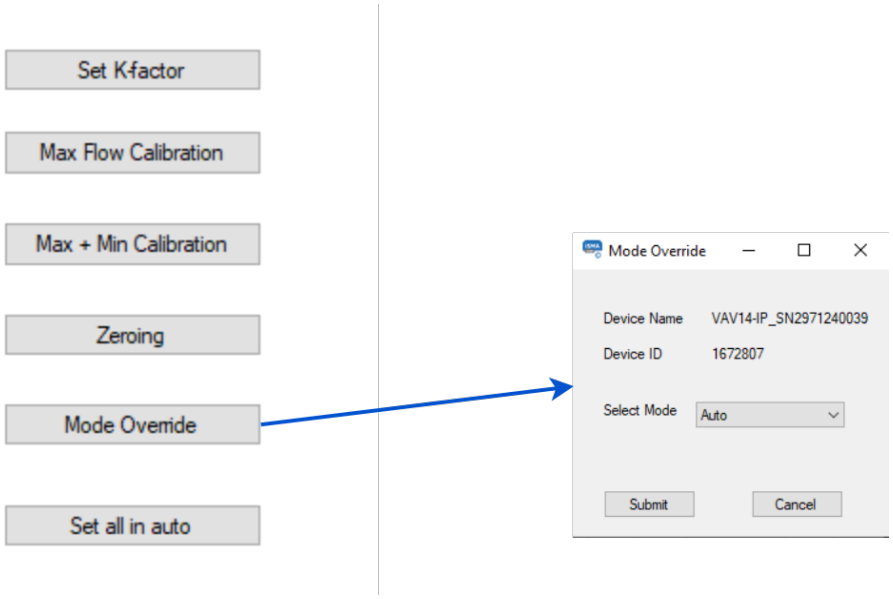


Figure 55. Mode override action pop-up

Set All in Auto

The Set All in Auto action forces all dampers to the Auto operating mode.

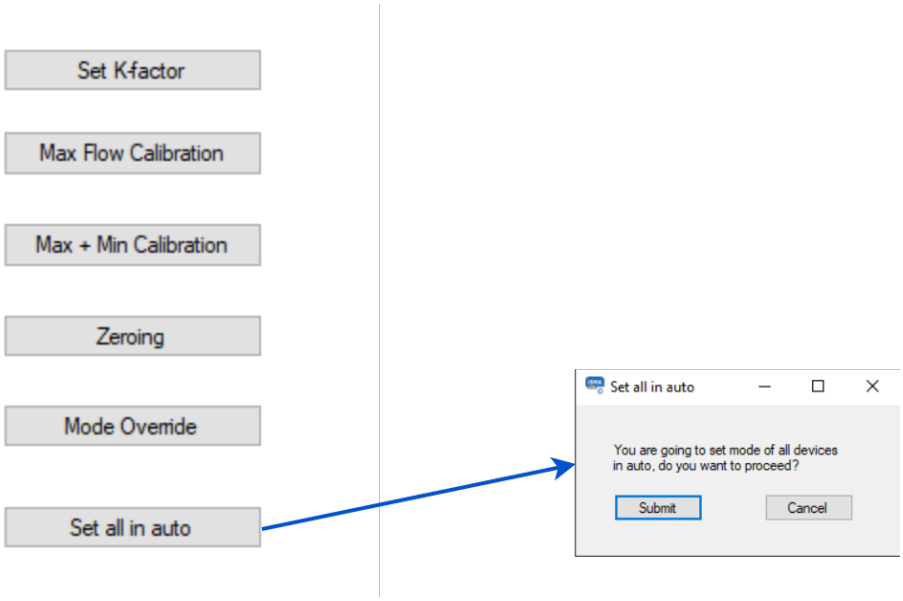


Figure 56. Set all in auto action pop-up

4.2.4 Control Point

Warning!

Before using the Control Point VAV panel with the VAV14-IP controller, make sure to upgrade its firmware to version V2.5.
Instructions how to upgrade firmware: iSMA Configurator.

Balancing Steps

To perform the balancing process and regulate the efficiency point of the ventilation system, follow the below steps. These activities are to performed in the presented order. If required, perform the balancing process with zeroing, however, according to the specific user and accuracy requirements, it is possible to omit the pressure sensor zeroing and/or minimum airflow calibration.

Balancing with zeroing

1. Pressure sensor zeroing.
2. Set the factory K-factor.
3. Measure the airflow and enter the value in the designated field.
4. Perform the maximum airflow calibration.
5. Measure the airflow and enter the value in the designated field.
6. Perform the minimum airflow calibration.

Balancing based on the maximum and minimum airflow calibration

1. Set the factory K-factor.
2. Measure the airflow and enter the value in the designated field.
3. Perform the maximum airflow calibration.
4. Measure the airflow and enter the value in the designated field.
5. Perform the minimum airflow calibration.

Balancing based on the maximum airflow calibration

1. Set the factory K-factor.
2. Measure the airflow and enter the value in the designated field.
3. Perform the maximum airflow calibration.

Therefore, in the Control Point VAV submenu, it is possible to skip the ZERO action under PRES and/or MINF action under FCAL.

KFAC

KFAC: sets the K-factor value to the airflow calculations;

- K-Factor:
  - Modbus register: 40289;
  - BACnet object: AV61, property: Present Value;

FCAL

FCAL: executes a calibration action according to a new maximum and/or minimum airflow value(s);

- Actual Airflow:
  - Modbus register: 30012;
  - BACnet object: AI11, property: Present Value;

Note

If performing the balancing process based on the maximum airflow calibration, it is possible to skip the MINF action here.

Actions	
MAXF	Performs a calibration action according to a new maximum airflow value
MINF (optional)	Performs a calibration action according to a new minimum airflow value

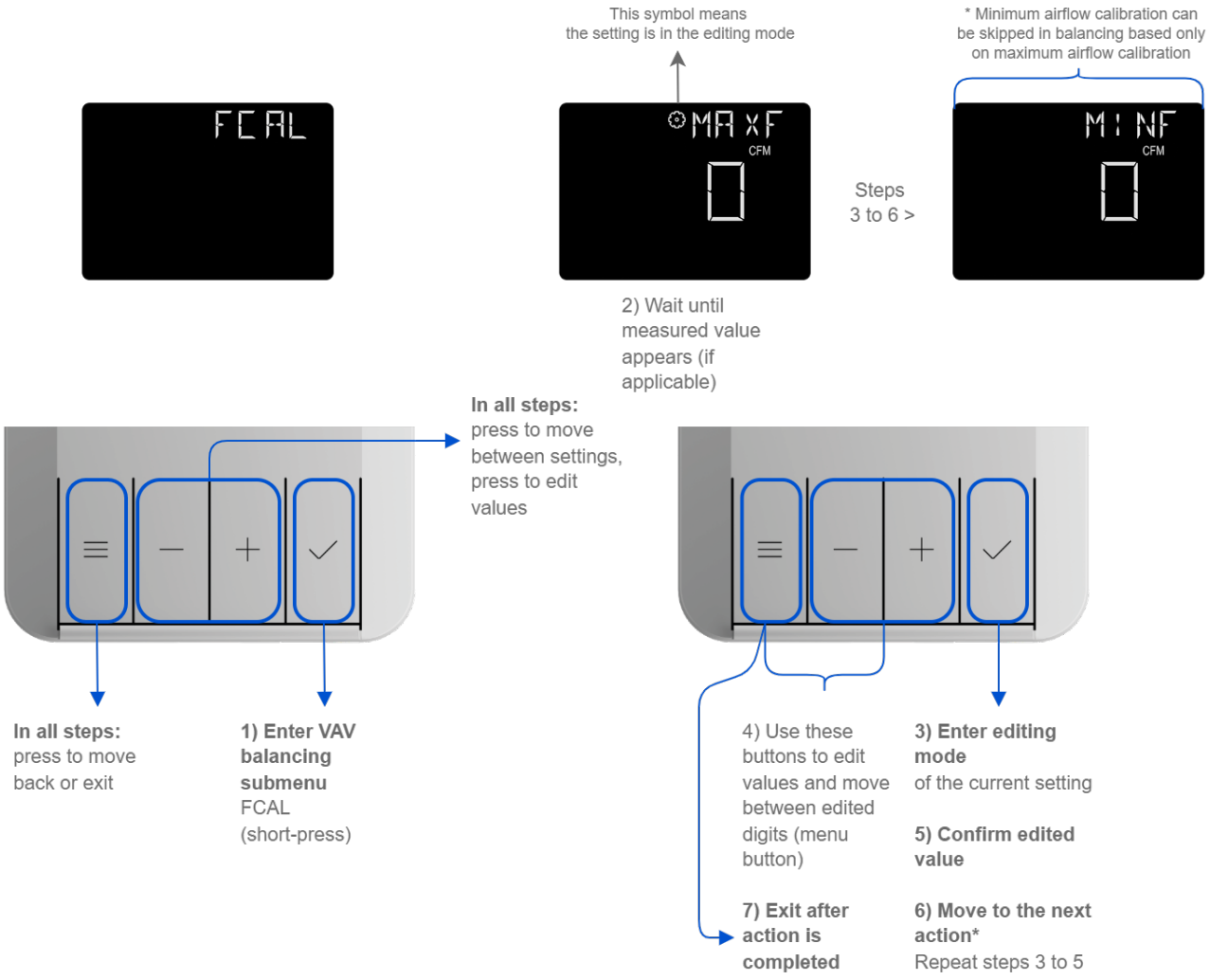


Figure 57. Panel access to FCAL submenu

PRES

PRES: allows to set a differential pressure (initially displayed value measured by the VAV);

- DifferentialPressure:
  - Modbus register: 30016;
  - BACnet object: AI15, property: Present Value;

Note

If performing the balancing process based on the maximum and/or minimum airflow calibration, it is possible to skip the ZERO action here.

Action	
ZERO: • PressureZeroing: ○ Modbus register: 00114; ○ BACnet object: BV13, property: Present Value;	Performs the pressure sensor zeroing action

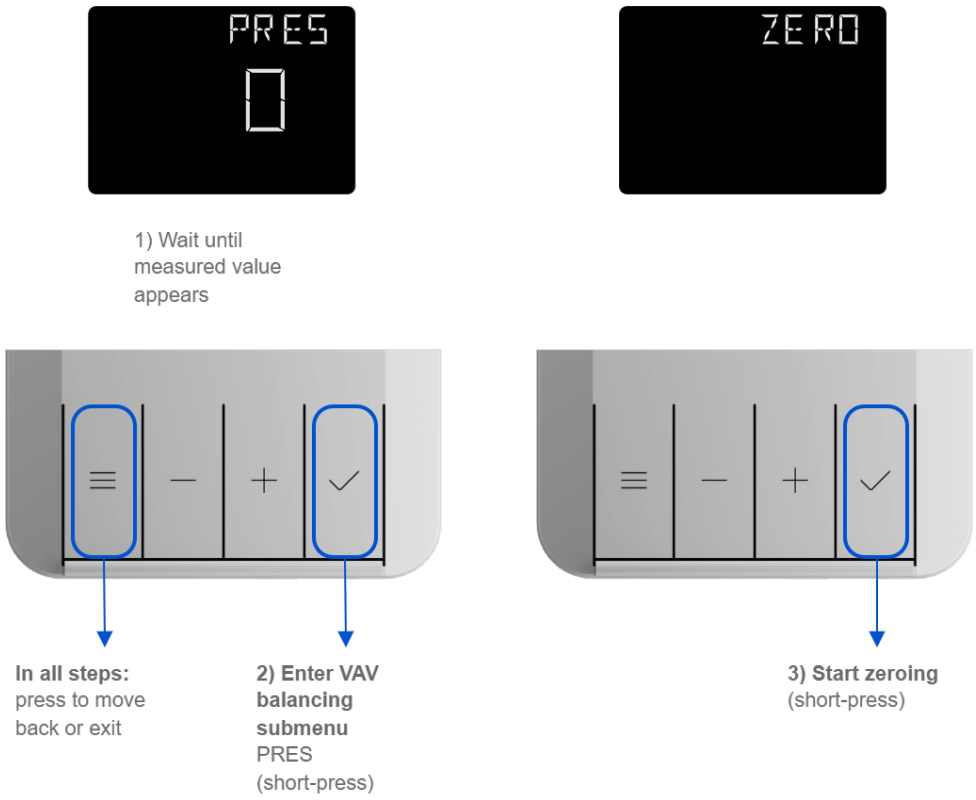


Figure 58. Panel access to PRES submenu

MODE

DAMP: allows to force an operating mode to a damper;

- DamperCommand:
  - Modbus register: 40268;
  - BACnet object: MSV7, property: Present Value;

Damper mode	Action
AUTO	Set auto mode (VAV application logic takes control)
MAXF	Go to a maximum airflow

Damper mode	Action
MINF	Go to a minimum airflow
USRF	Go to a user-set airflow
USRP	Go to a user-set position (% of damper opening)
OPEN	Damper fully open
CLOS	Damper fully closed
CLBR	Performs a damper calibration (the damper goes to 100%, then to 0%, and goes back to the control loop output)

- **UserSetPosition:** user-set opening level (after selecting the USRF opting in the VAV_DAMPER_COMMAND variable);
  - Modbus register: 40292;
  - BACnet object: AV64, property: Present Value;
- **UserSetAirflow:** user-set airflow setpoint (after selecting the USRP opting in the VAV_DAMPER_COMMAND variable);
  - Modbus register: 40293;
  - BACnet object: AV65, property: Present Value;
- **AirflowCalibration:** allows to enter the airflow value measured by the hood for calibration;
  - Modbus register: 40294;
  - BACnet object: AV66, property: Present Value;

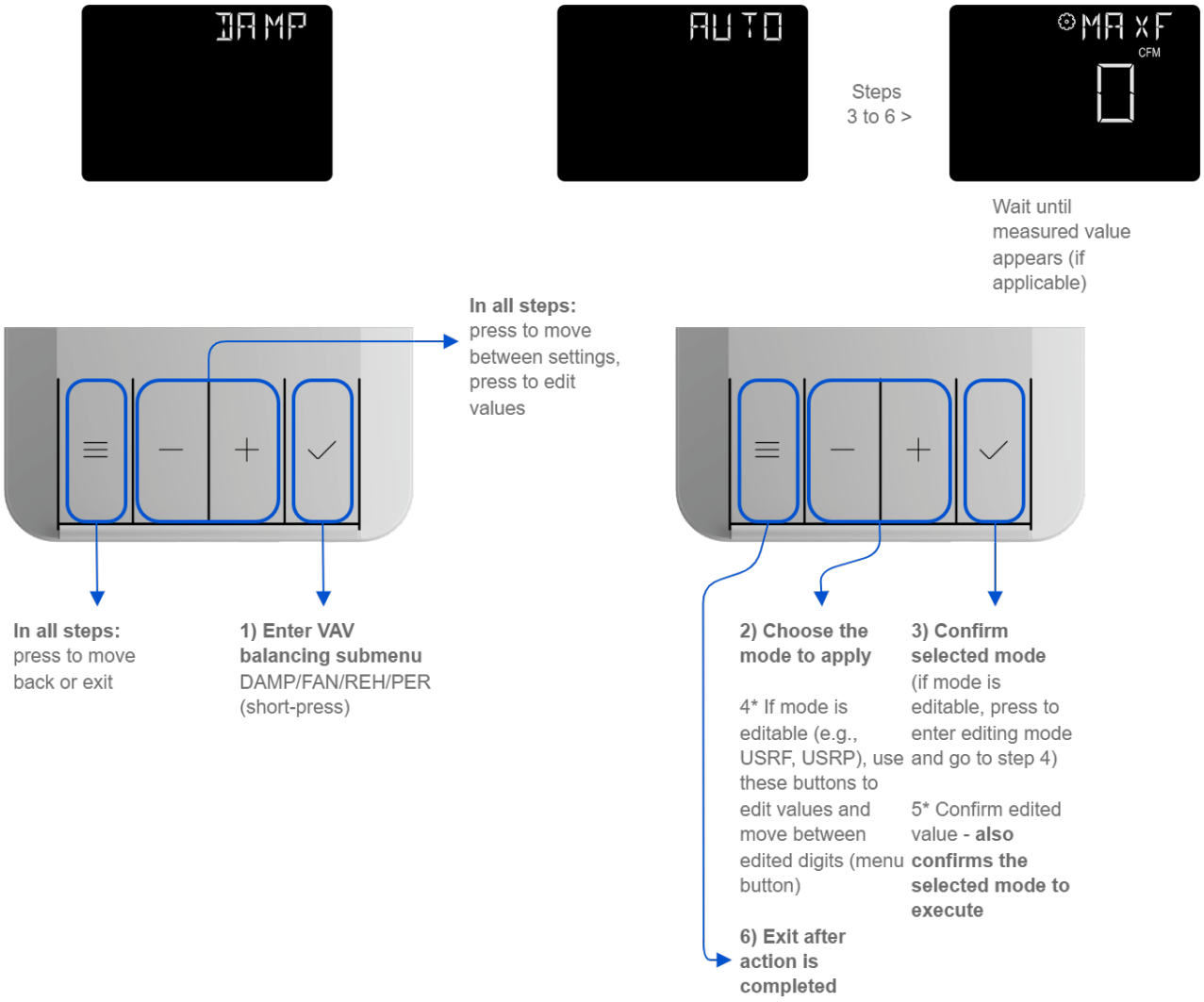


Figure 59. Panel access to MODE submenu

FAN: allows to force an operating mode to a fan;

- FanCommand:
  - Modbus register: 40269;
  - BACnet object: MSV8, property: Present Value;

Fan mode	Action
AUTO	Set an auto mode (VAV application logic takes control)
MAXS	Set a maximum fan speed
MINS	Set a minimum fan speed
STOP	Stop fan

REH: allows to force an operating mode to a reheater;

- ReheaterCommand:
  - Modbus register: 40270;
  - BACnet object: MSV9, property: Present Value;

Reheater mode	Action
AUTO	Set an auto mode (VAV application logic takes control)
OPEN	Fully open
CLOS	Fully closed

PER: allows to force an operating mode to a perimeter;

- **PerimeterCommand:**
  - Modbus register: 40271;
  - BACnet object: MSV10, property: Present Value.

Perimeter mode	Action
AUTO	Set an auto mode (VAV application logic takes control)
OPEN	Fully open
CLOS	Fully closed

4.2.5 iC Device Manager

Installation

Learn how to install and start using the iC Device Manager: iC Device Manager.

The last tab in the iC Device Manager service is the Balancing tab. Here, it is possible to perform actions of the balancing process.

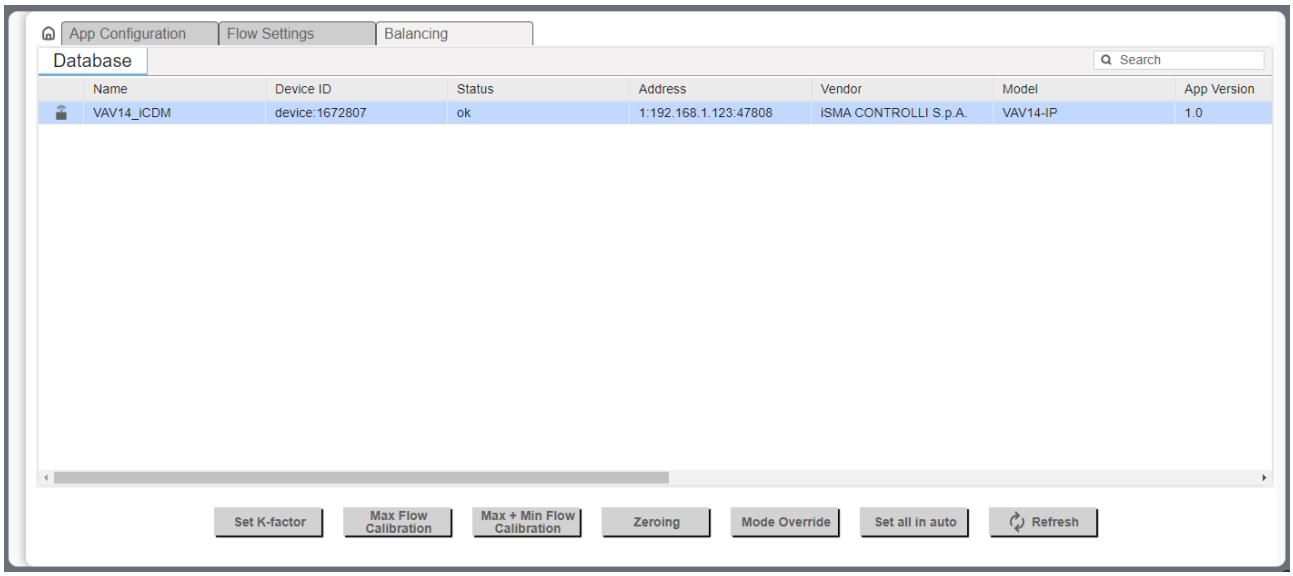


Figure 60. Balancing tab

The main view of the tab shows data read from the device and available actions.

To make sure that the data are up to date, click the Refresh button.

The available actions are:

- **Set K-Factor:** sets the K-factor value to the airflow calculations (AV61, 40289);
- **Max Flow Calibration:** executes a calibration action according to a new maximum airflow value (AV66, 40294);
- **Max + Min Calibration:** executes a calibration action according to a new maximum and minimum airflow values (AV66, 40294);
- **Zeroing:** performs the pressure sensor zeroing action (BV13, 00014);
- **Mode Override:** allows to force the device to operate in one of the available modes (MSV7, 40068):
  - Available settings: Auto, Max. Flow, Min. Flow, User Flow, User Position, Full Open, Full Close, Calibrate;
- **Set all In auto:** forces all dampers to the Auto mode (MSV7, 40068).

Balancing Steps

To perform the balancing process and regulate the efficiency point of the ventilation system, follow the below steps. These activities are to performed in the presented order. If required, perform the balancing process with zeroing, however, according to the specific user and accuracy requirements, it is possible to omit the pressure sensor zeroing and/or minimum airflow calibration.

Balancing with zeroing

1. Pressure sensor zeroing.
2. Set the factory K-factor.
3. Measure the airflow and enter the value in the designated field.

4. Perform the maximum airflow calibration.
5. Measure the airflow and enter the value in the designated field.
6. Perform the minimum airflow calibration.

Balancing based on the maximum and minimum airflow calibration

1. Set the factory K-factor.
2. Measure the airflow and enter the value in the designated field.
3. Perform the maximum airflow calibration.
4. Measure the airflow and enter the value in the designated field.
5. Perform the minimum airflow calibration.

Balancing based on the maximum airflow calibration

1. Set the factory K-factor.
2. Measure the airflow and enter the value in the designated field.
3. Perform the maximum airflow calibration.

Pressure Sensor Zeroing

If required, pressure sensor zeroing is an initial action of the balancing process. According to the specific user and accuracy requirements, this action can be omitted in the balancing process.

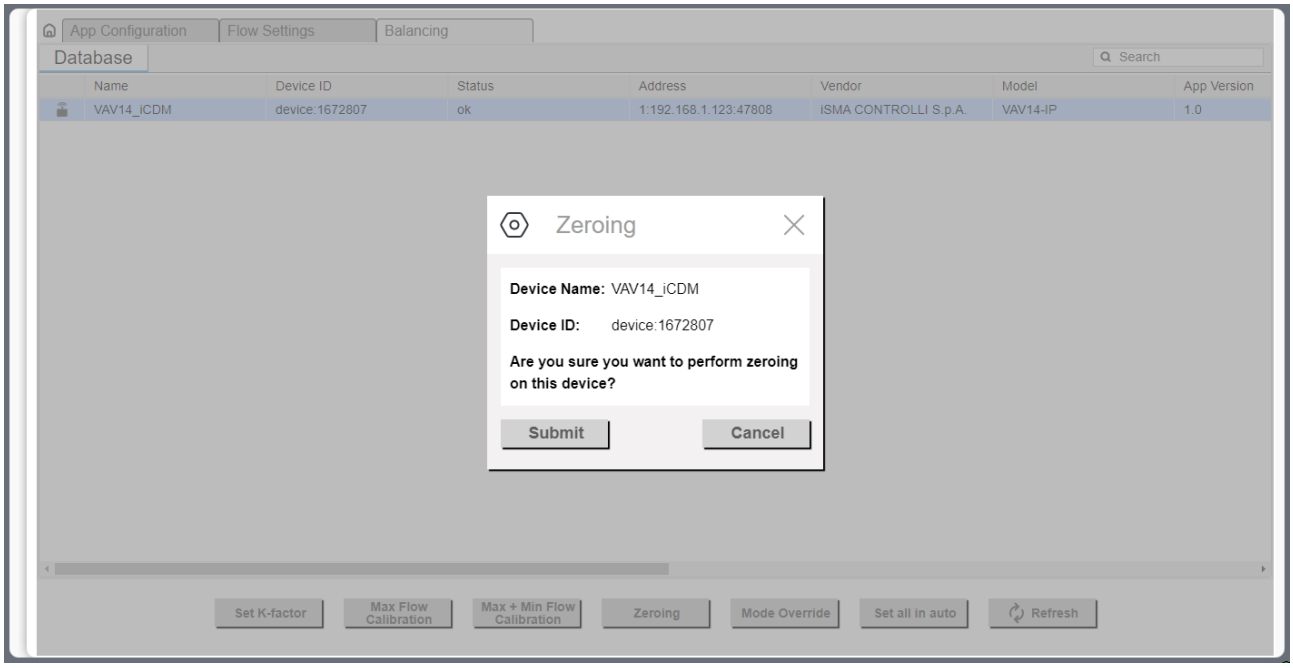


Figure 61. Zeroing action pop-up

Pressure sensor zeroing is a part of the VAV balancing process, which aims at eliminating a constant measurement error of differential pressure on a built-in pressure sensor. The zeroing process involves the following steps:

- Make sure the differential pressure sensor is disconnected from the measuring cross or other measuring method.
- Use a flexible hose of an appropriate diameter to connect the two spigots (+ and -).
- Make sure the hose is well secured and tight to equalize the pressures on both ports.
- Invoke the zeroing action.
- Detach the hose.
- Restore the normal connection of the sensor to the measuring cross or other target circuit, pay attention to the polarity.

Setting K-factor

Setting the K-factor is a mandatory part of the balancing process. It is normally provided by the manufacturer of the VAV box. If for any reason it is unavailable, it is recommended to use a default K-factor from the VAV application (100 l/s /1000 cfm).

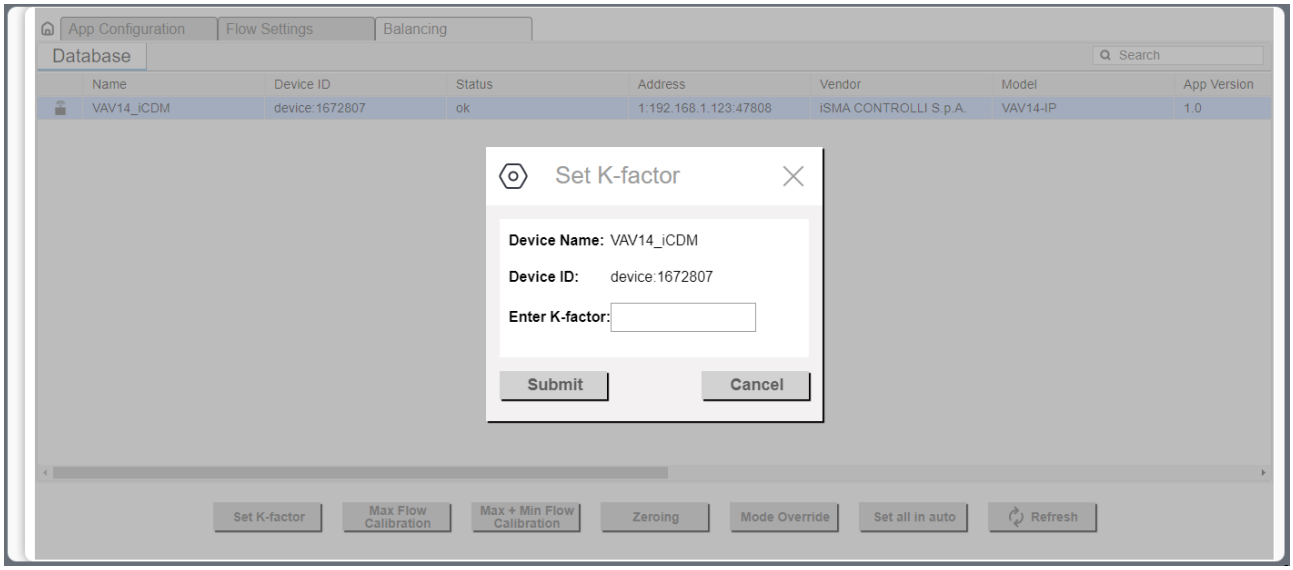


Figure 62. Setting K-factor pop-up

Maximum and Minimum Calibration

To perform the full balancing process, proceed to the maximum and minimum airflow calibration action, under the Max. + Min. Calibration button. The action invokes two pop-up windows, first for the maximum calibration in step 1, and second for the minimum airflow calibration in step 2.

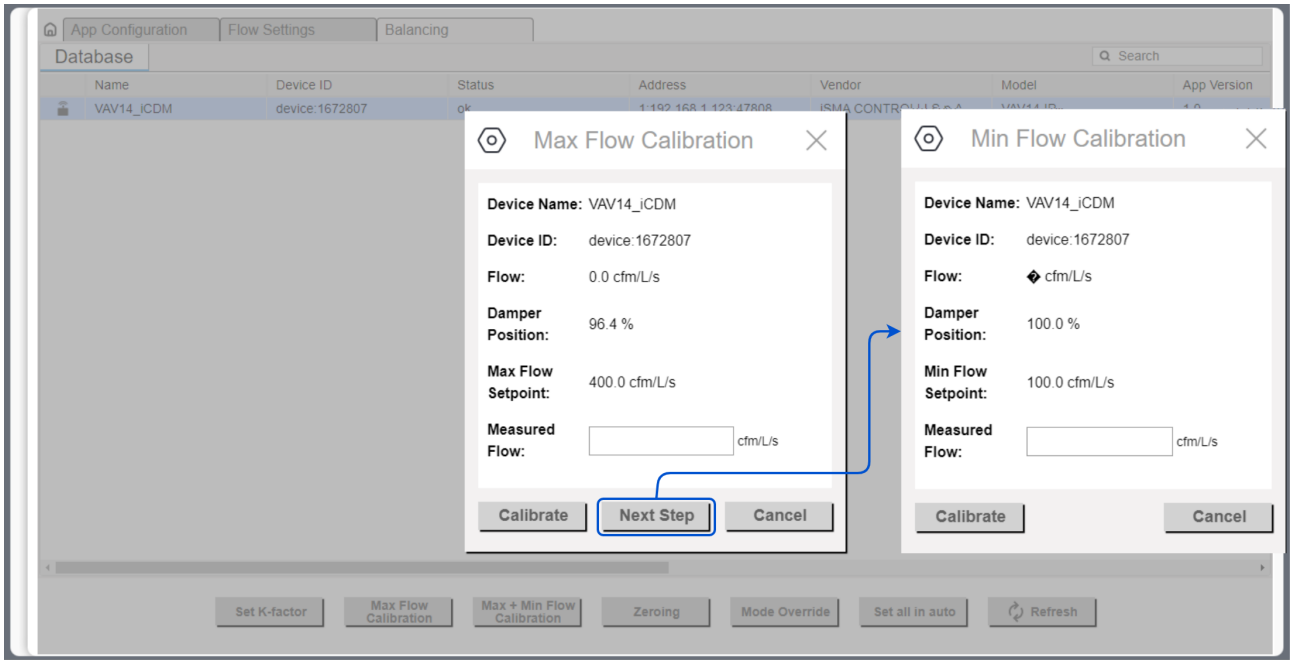


Figure 63. The maximum and minimum airflow calibration pop-ups

In step 1, measure the airflow and enter the measured value. Then, proceed to calibration (using the Calibrate button). The action will return a newly calculated K-factor.

After completing step 1, continue to step 2, the minimum airflow calibration. Enter the measured airflow value and use the Calibrate button.

Maximum Calibration

If, for any reason, the minimum airflow calibration is not required, it is possible to perform only the maximum airflow calibration. Measure the airflow and enter the measured value. Then, proceed to calibration (using the Calibrate button). The action will return a newly calculated K-factor.

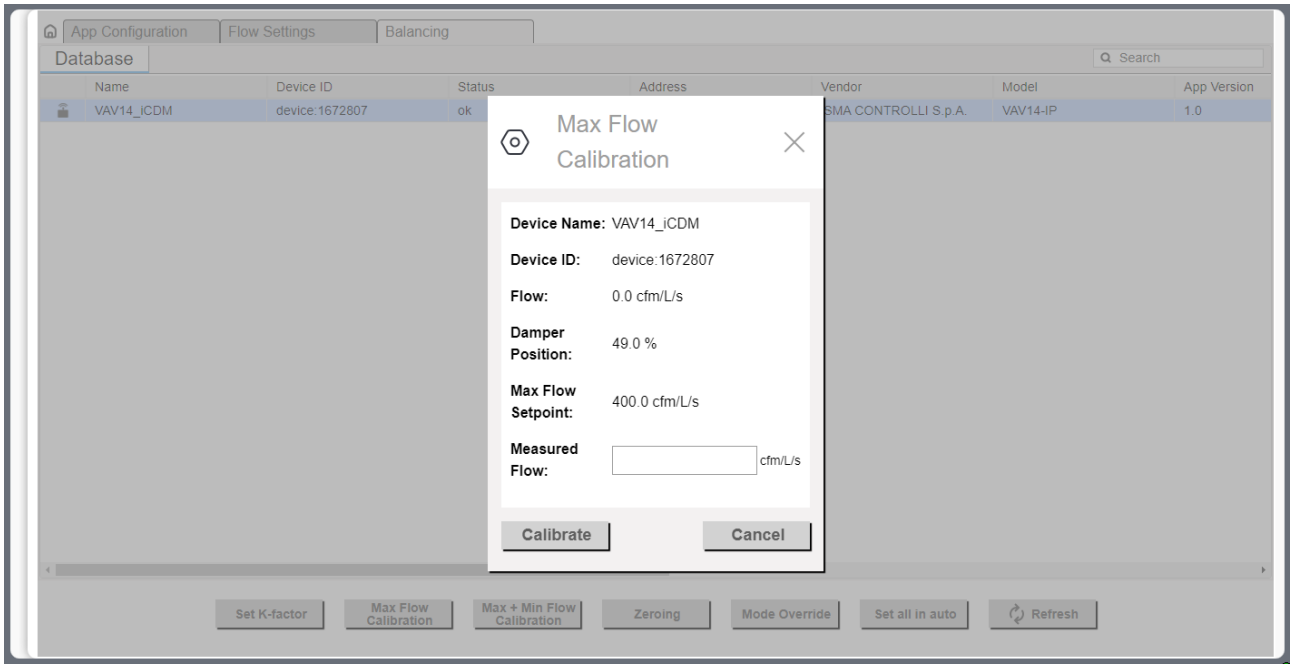


Figure 64. Maximum airflow calibration pop-up

Other Actions

Mode Override

The Mode Override action allows to force an operating mode to a damper. Available modes are:

- **Auto:** sets auto mode (VAV application logic takes control),
- **Max flow:** goes to a maximum airflow,
- **Min flow:** goes to a minimum airflow,
- **User flow:** goes to a user-set airflow,
- **User position:** goes to a user-set position (% of damper opening),
- **Full open:** damper fully open,
- **Full close:** damper fully closed,
- **Calibrate:** performs a damper calibration (the damper goes to 100%, than to 0%, and goes back to the control loop output).

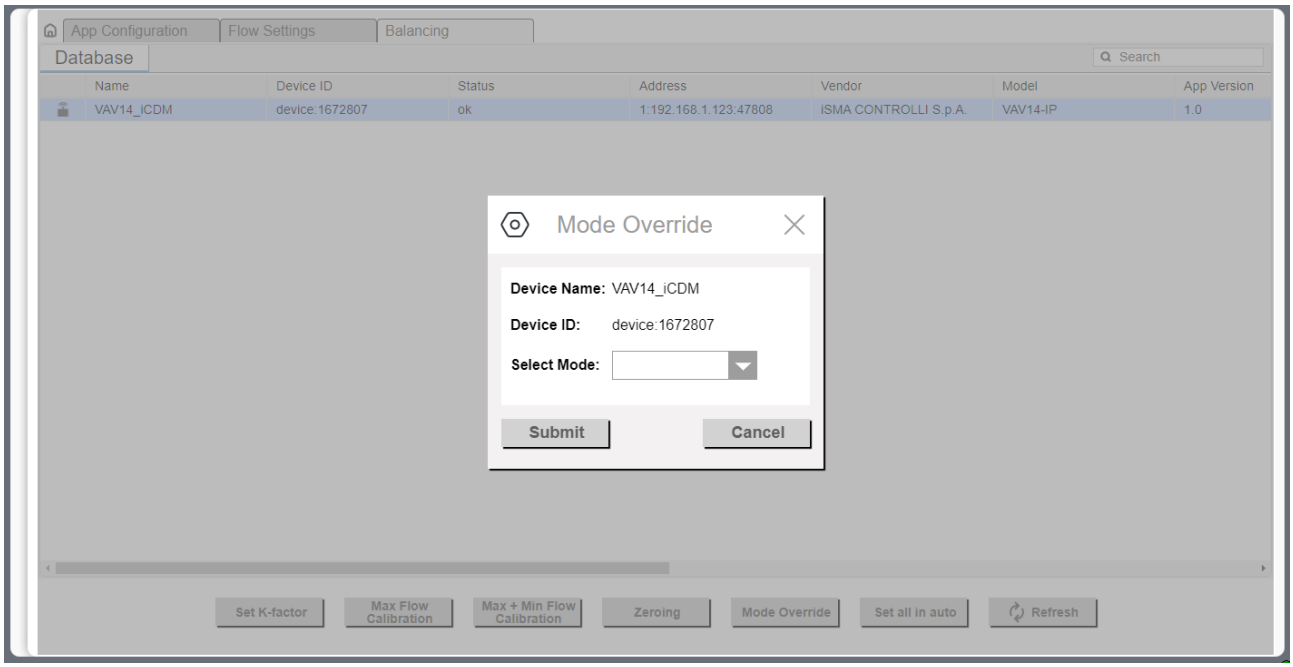


Figure 65. Mode override pop-up

Set All in Auto

The Set All in Auto action forces all dampers to the Auto operating mode.

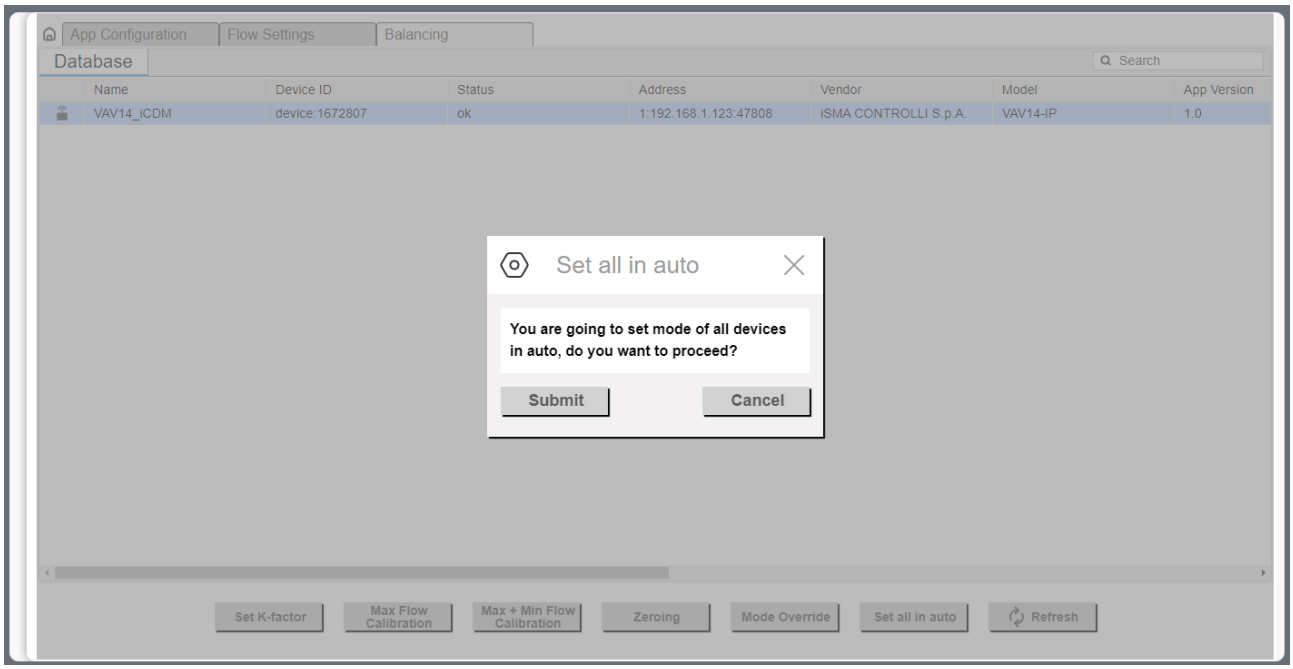


Figure 66. Set all in auto pop-up

Using iC Device Manager

Adding the Module

The iC Device Manager service is a part of the iC Workbench and iC Niagara Expansion Pack (from version 4.14).

Note: For a correct operation of the iC Device Manager service, it is required also to have the latest iClib version.

To start using the iC Device Manager service, go to the Palette window (in iC Workbench or other Niagara tool) and select the Open Palette option.

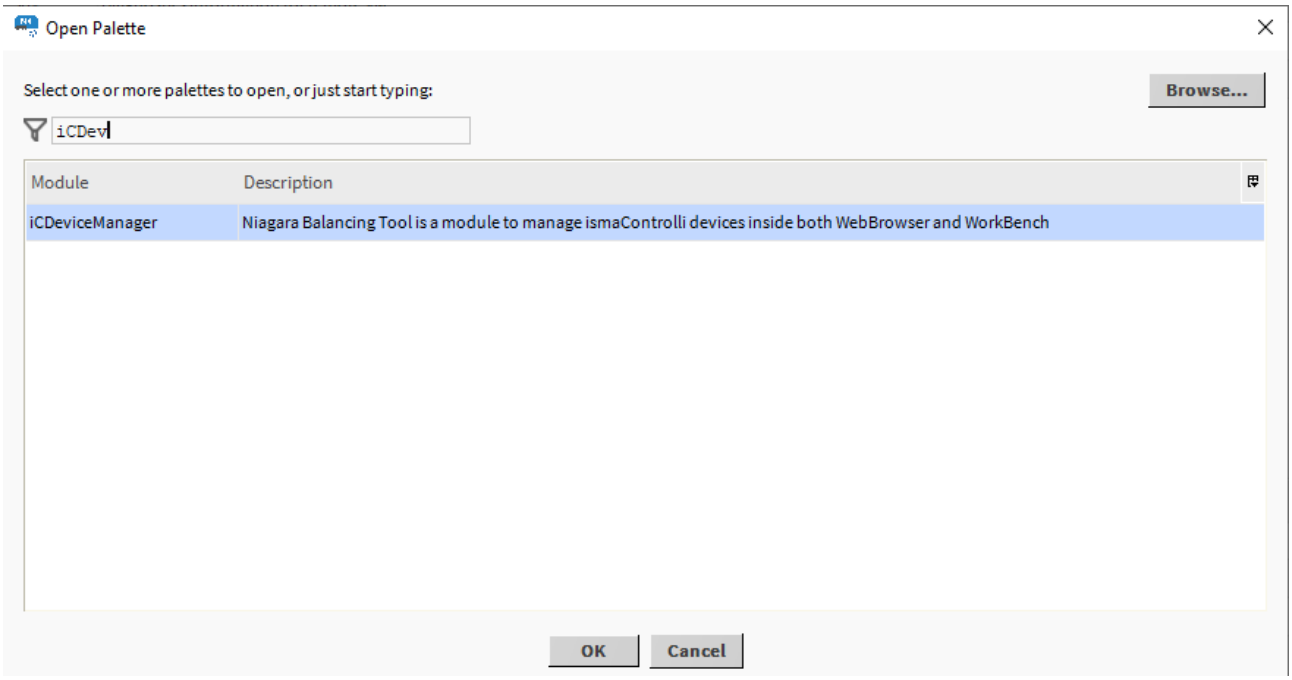


Figure 67. Opening palette

Confirm with OK, the palette is ready to use in the Palette window.

The only location where the service will operate properly is Config → Services.

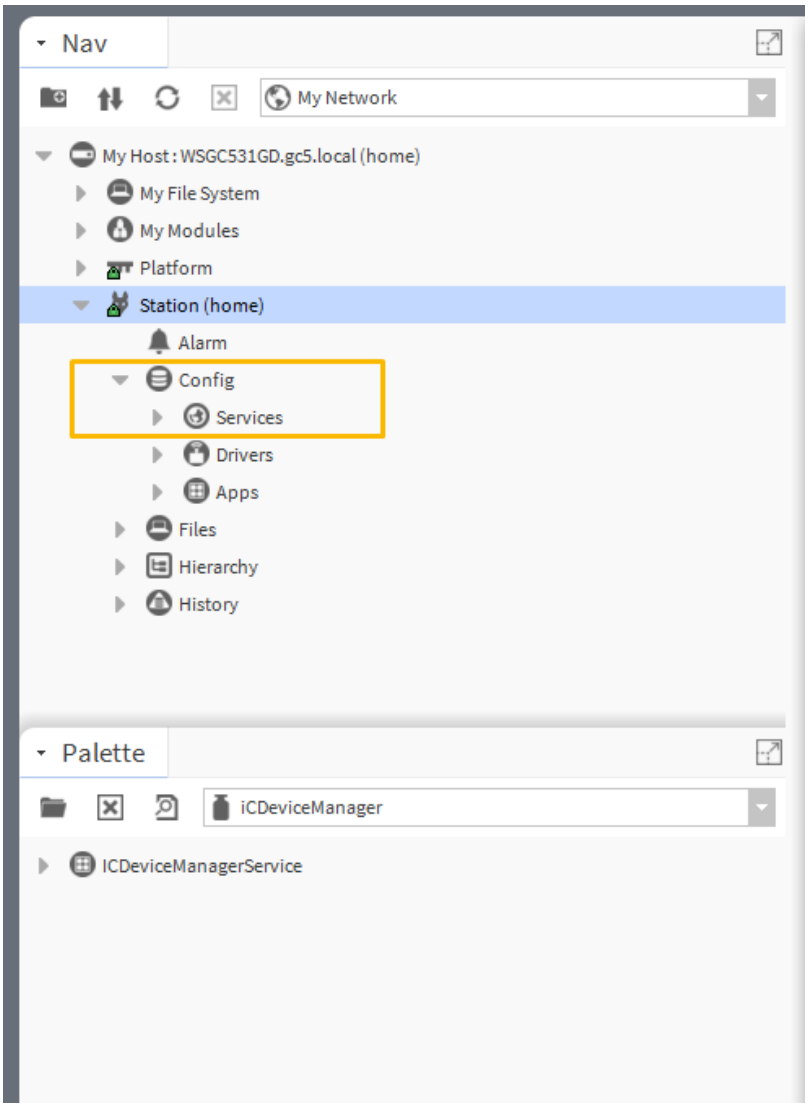
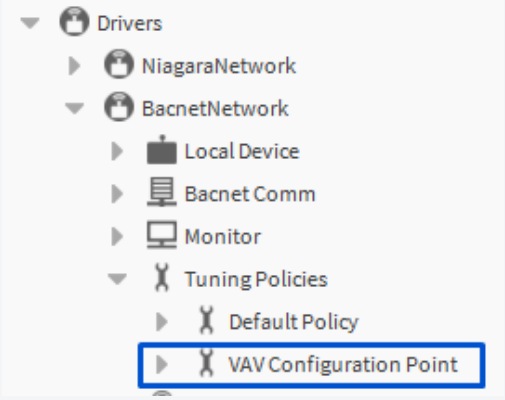


Figure 68. Services location

Drag and drop the iC Device Manager service to Services.

Warning!

For the iC Device Manager service to fully operate, make sure that the VAV14-IP controller is added to the BACnet network in Drivers.
If BACnet has been configured using the default BACnetNetwork module, it is required to add the VAV Configuration Point (from the iSMA_CONTROLLI-Library) to Tuning Policies:



Adding the VAV Device

Offline

1. Add the VAV14-IP device from the iSMA_CONTROLLI_Library (BACnetNetwork → ComfortManagement), according to user requirements: VAV14_CONFIG_POINT or VAV14_PROXY_POINT.
2. Go to BACnetNetwork in the station and invoke the Discover action.
3. Mark the offline device and the device to be matched with it, and confirm with the Match button.

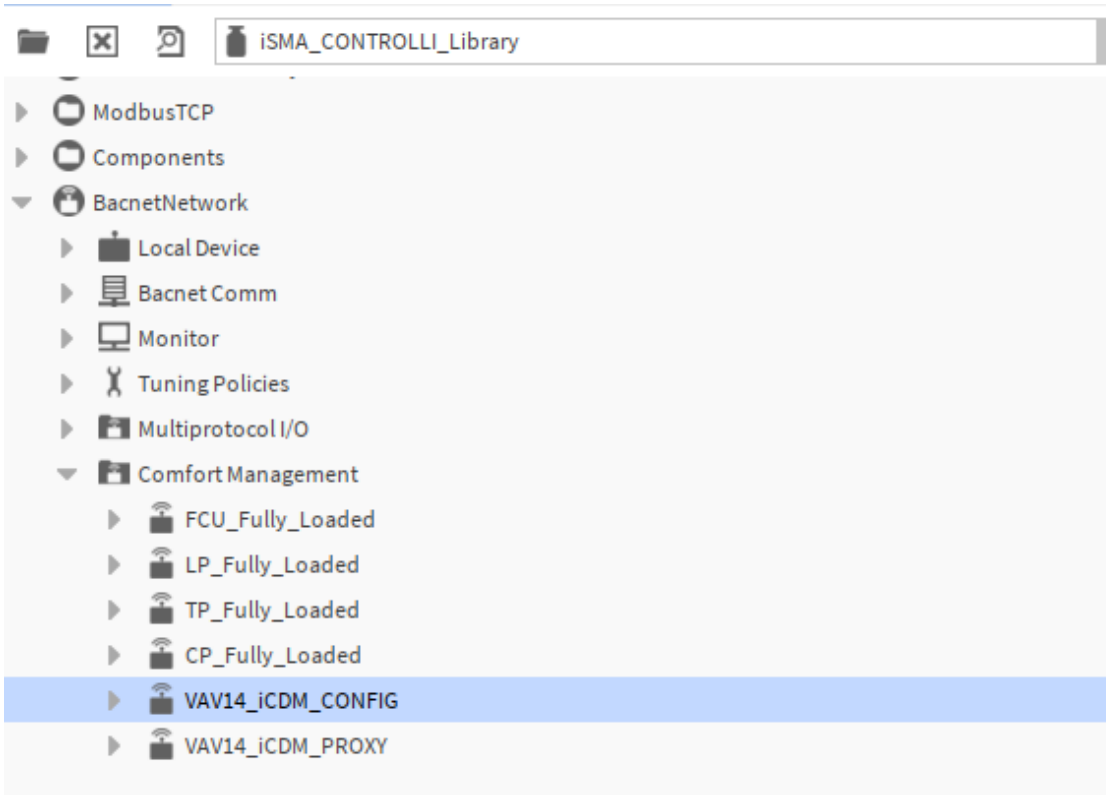


Figure 69. Config and proxy points

Online

1. Go to BACnetNetwork in the station and invoke the Discover action.
2. Add the discovered VAV14-IP device(s).
3. From the iSMA_CONTROLLI_Library select points for devices to be configured as proxy or config points.

Warning!
For a proper recognition of the device, the iC Device Manager verifies its hardware type, firmware version, and application version. From these values, only the application version is read from the AI 100 point. Make sure it is added to the device, otherwise, it may not be visible in the service.

Proxy and config points
Proxy points are BACnet points configured as proxy and placed under the Points folder. These consume Niagara license points.
Config points are BACnet points configured as config and placed under the Config folder. These points do not consume Niagara license points.

5 VAV14-IP as a Freely-programmable Controller

Communication and Credentials

Please refer to this site: Default Communication Settings and Credentials for more information on communication settings and default credentials in the VAV14-IP controller.
See also First Connection and Setting IP Address of the Quick Start-up sections for more information on connecting the controller.

The VAV14-IP controller is factory-equipped with the VAV application, which fits most of the available VAV boxes. However, if the pre-loaded application does not meet user requirements for any reason, there is a possibility to create a custom application. The custom application may be created from scratch (advanced BMS programming skills are required) or by modifying the existing application. Either way, there are dedicated libraries and tools that facilitate creating the user custom application.

- nano EDGE ENGINE Libraries for VAV Application,

Find out more about the nano EDGE ENGINE.

- dedicated tools for creating the custom application.

Important!

After creating the custom application it is strongly recommended to save its backup for future use.
Find out about creating backups in the nano EDGE ENGINE: Backups.

5.1 nano EDGE ENGINE Libraries for VAV Application

There are 2 dedicated applications for creating the custom VAV application:

- VAV Library
- ComfortControl Library
- LocalIO Library

Find out more about other useful nano EDGE ENGINE libraries in the nano EDGE ENGINE Programming user manual.

5.1.1 VAV Library

- AirFlowCalculator
- AirFlowSetpointControl
- DamperControl
- VAVFanControl
- HeaterControl

AirFlowCalculator

Applicable to OS V1.7

The AirFlowCalculator component is used to calculate the airflow based on the measurement of the pressure difference and the K-factor. Additionally, it allows airflow calibration of the calculated airflow by recalculating the initial K-factor value.

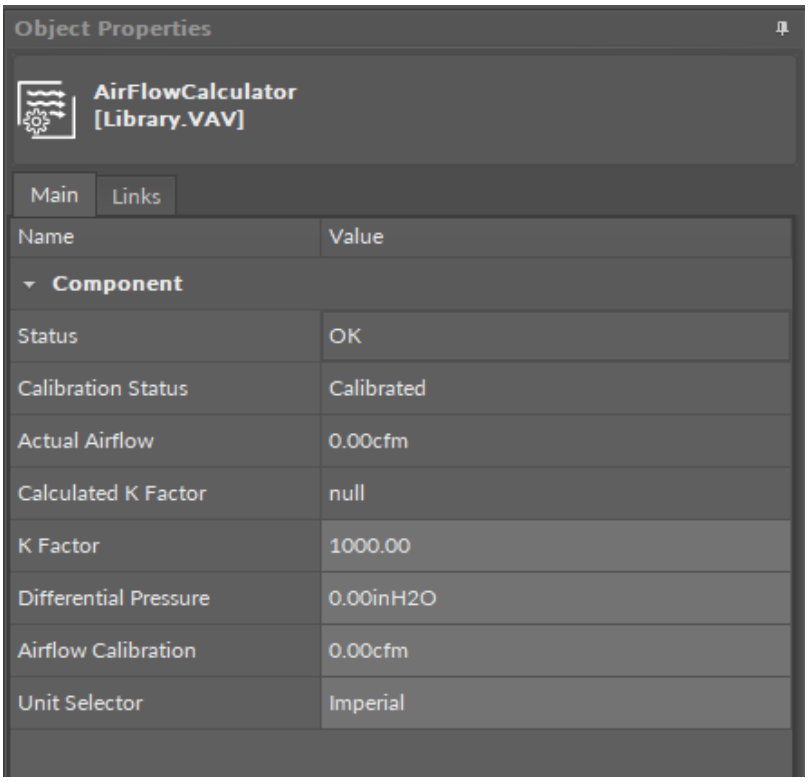


Figure 70. AirFlowCalculator slots

The AirFlowCalculator component has the following slots:

- **Status:** informs about the status of the component; if one of the inputs (K-Factor/Differential Pressure/ Airflow Calibration) has a null value then the status is fault, otherwise it is OK;
    - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;
- **Calibration Status:** shows a calibration status of the airflow; if the calculated K-factor changes, the Calibration Status will indicate that the airflow is calibrated;
- **Actual Airflow:** shows the calculated airflow value based on the differential pressure and the K-factor;
- **Calculated K-Factor:** shows the calculated K-Factor based on the differential pressure and calibrated airflow input;
- **K-Factor:** allows to set a correct value for a specific type of VAV from the BMS or local room panel; can be updated by the Calculated K-Factor value by a trigger action from application;
- **Differential Pressure:** receives a differential pressure value from the pressure transducer;
- **Airflow Calibration:** specifies a measured airflow value during the balancing process from the BMS or local room panel; this value must be reset to zero via link once the airflow calibration is completed;

Note

If the airflow calibration is linked in the application, it is required that once the calibration is completed, the value incoming from the link resets this slot to zero for the calibration status to work correctly.

- **Unit Selector:** allows to set the unit type.

More

More on the airflow control as used in the VAV application: Airflow

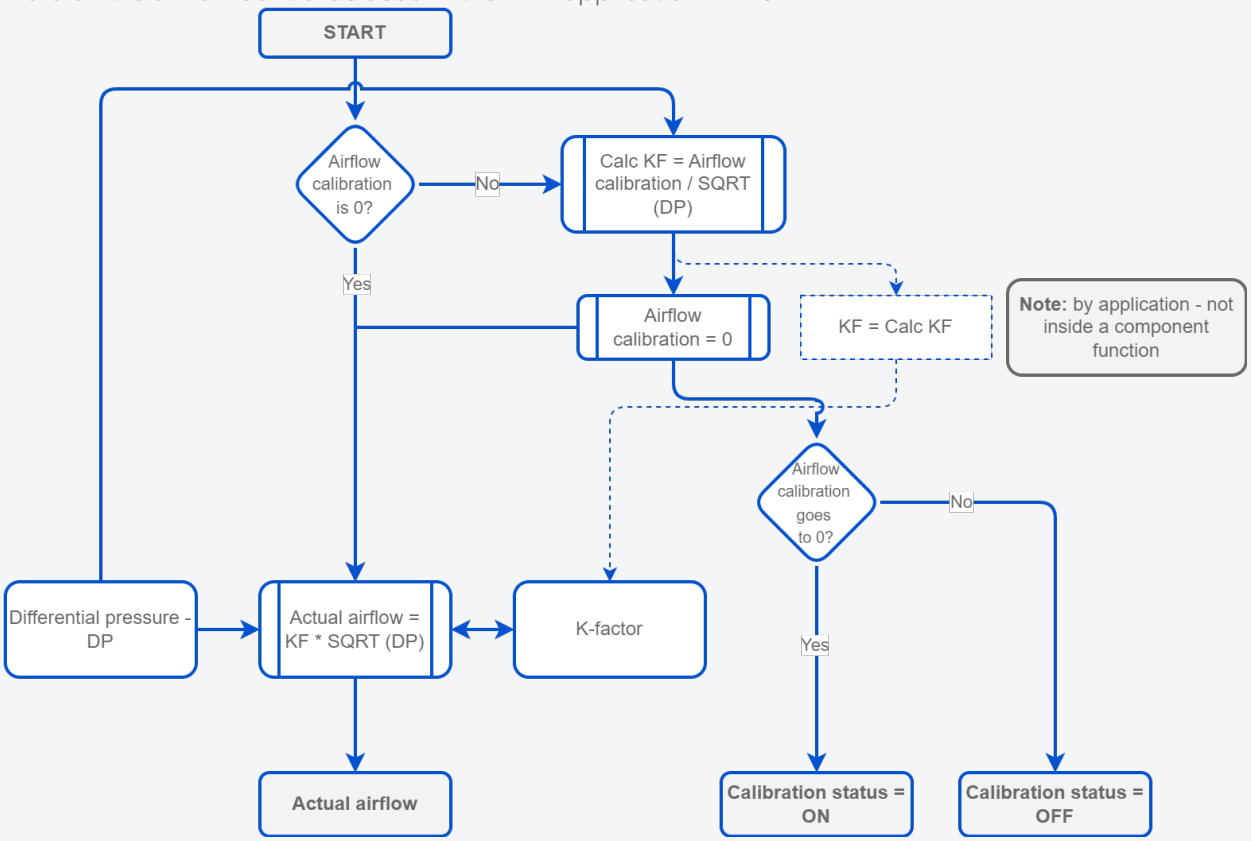


Figure 71. Actual airflow diagram

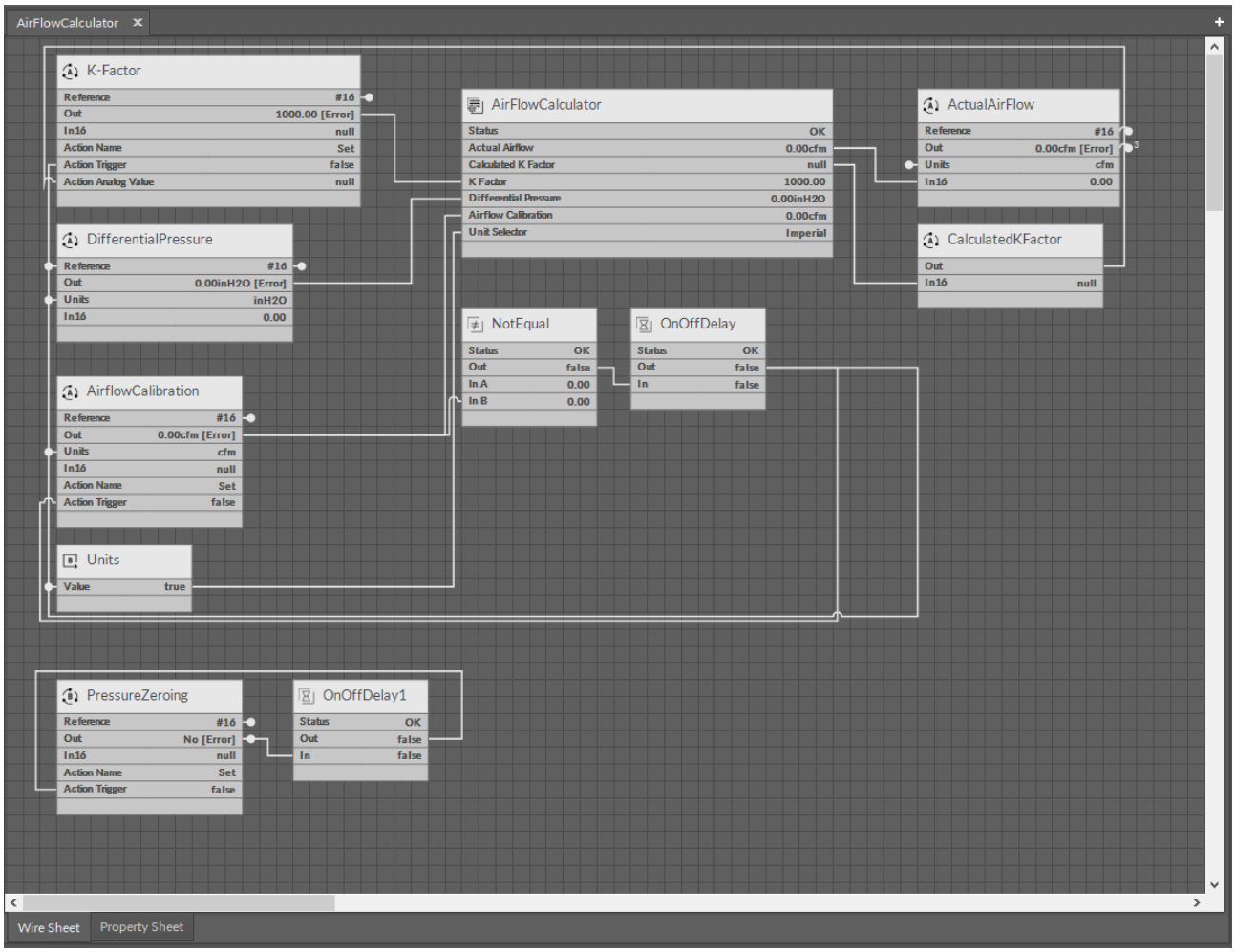


Figure 72. AirFlowCalculator in the application

AirFlowSetpointControl

Applicable to OS V1.7

The AirFlowSetpointControl component determines the effective setpoint for regulating the airflow using the damper, based on the output from the temperature control loop for cooling or heating, the setpoints for the minimum and maximum airflow for cooling and heating, the current occupancy status, operating mode (normal, balancing), and HVAC mode.

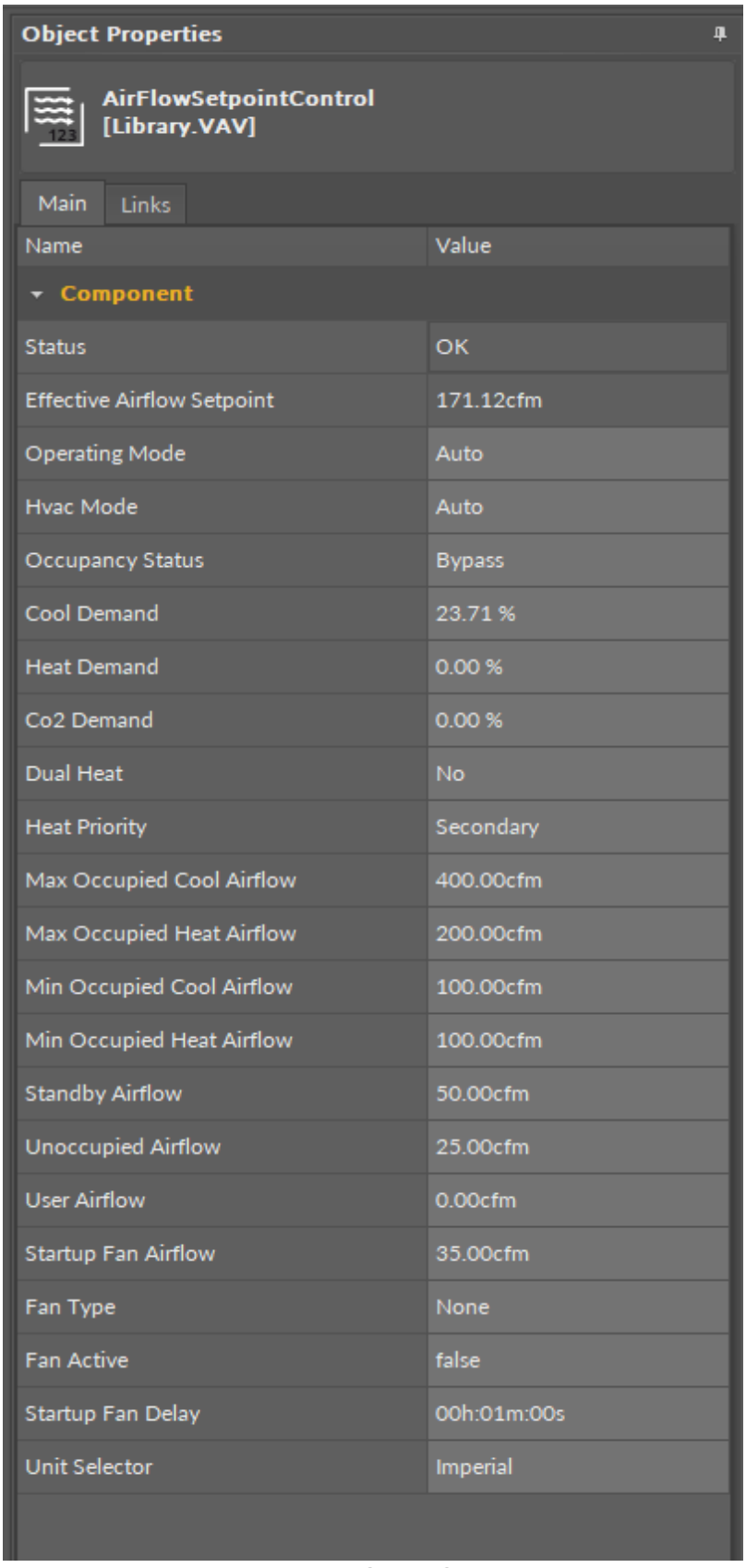


Figure 73. AirFlowSetpointControl slots

The AirFlowSetpointControl component has the following slots:

- **Status:** informs about the status of the component; if the Operating Mode/OccupancyStatus/HVAC Mode slot or one of the demand input slots has a null value then the status is fault, otherwise it is OK;
    - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;

- **Effective Airflow Setpoint:** returns an airflow setpoint calculated based on the output of the temperature control loop (heating or cooling) and taking into account the setpoint limit values for heating or cooling (Max Occupied Cool Airflow, Max Occupied Heat Airflow, Min Occupied Cool Airflow, Min Occupied Heat Airflow), occupancy mode and occupancy setpoints (UnoccAirflowSetpoint, StandbyAirflowSetpoint);
- **Operating Mode:** shows the operating mode from the BMS or panel (from Modbus/BACnet network point) and allows to initiate an airflow setpoint balancing procedure;
    - Available settings: Auto, MaxFlow, MinFlow, UserFlow;
- **HVAC Mode:** receives the HVAC mode setting, which determines the ariflow setting dependencies and impacts functioning of the fan and heaters (if available on the system);
    - Available settings: Auto, Heat, MorningWarmUp, Cool, NightPurge, PreCool, Off, Fire;

HVAC modes

- **Auto:** in the Auto HVAC mode and the operating mode also set to auto, the airflow setpoint is determined by the occupancy status:
    - occupied and bypass:
        - heating: MaxOccHeatAirFlowSetpoint and MinOccHeatAirFlowSetpoint,
        - cooling: MaxOccCoolAirFlowSetpoint and MinOccCoolAirFlowSetpoint;
    - standby (StandbyAirFlowSetpoint) and unoccupied (UnoccAirFlowSetpoint): the setpoint is the same for heating and cooling.

By default, the maximum airflow setpoint for heating is set as half of the value of the airflow setpoint for cooling.

- **Heat:** the airflow setpoint is set depending on the occupancy status:
    - occupied and bypass: between the MaxOccHeatAirFlowSetpoint and MinOccHeatAirFlowSetpoint values - it is a function of the linear heating demand resulting from the heating control loop;
    - unoccupied (UnoccAirFlowSetpoint) and standby (StandbyAirFlowSetpoint), the airflow setpoint is identical to the Auto mode.
- **Cool:** the airflow setpoint is set depending on the occupancy status:
    - occupied and bypass: between the MaxOccCoolAirFlowSetpoint and MinOccCoolAirFlowSetpoint values - it is a function of the linear cooling demand resulting from the cooling control loop;
    - unoccupied (UnoccAirFlowSetpoint) and standby (StandbyAirFlowSetpoint), the airflow setpoint is identical to the Auto mode;
- **MorningWarmUp:** the airflow setpoint is set to the MaxOccHeatAirFlowSetpoint, regardless the occupancy status;
- **PreCool:** the airflow setpoint is set to the MaxOccCoolAirFlowSetpoint, regardless the occupancy status;
- **NightPurge:** the airflow setpoint is set to the MaxOccCoolAirFlowSetpoint, regardless the occupancy status;
- **Fire:** the airflow setpoint is set to the MaxOccCoolAirFlowSetpoint, regardless the occupancy status;
- **Off:** the airflow setpoint is set to 0, regardless of the occupancy status.

- **Occupancy Status:** shows the value coming from the OccupancyCalculator component which is an occupancy status resulting from the values of window contact, present motion, network occupancy, schedule occupancy, airflow detection, and panel button override;
    - Available settings: Occupied, Unoccupied; Standby, Bypass;
- **Cool Demand:** shows the value from the cooling PID loop;
- **Heat Demand:** shows the value from the heating PID loop;
- **Co2 Demand:** shows the value from the CO_2 PID loop;
- **Dual Heat:** provides the component with the DualHeat function status to adjust the effective airflow setpoint calculations in the occupied mode;
    - Available settings:
        - Yes (the DualHeat function active):
            - primary stage (from HeatPriority):

        - for heating demand in the range of 0-50%, scales the effective airflow setpoint between the MinHeatOccAirflowSetpoint and MaxHeatOccAirflowSetpoint values;
        - for heating demand over 50%, effective airflow setpoint equals the MaxHeatOccAirflowSetpoint;
    - secondary stage (from HeatPriority):
        - for heating demand in the range of 0-50%, effective airflow setpoint equals the MinHeatOccAirflowSetpoint;
        - for heating demand over 50%, scales the effective airflow setpoint between the MinHeatOccAirflowSetpoint and MaxHeatOccAirflowSetpoint values;
  - No (the DualHeat function not active);
- **Heat Priority:** provides the component with the heating stages priority to adjust the effective airflow setpoint calculations in the occupied mode as described above;
  - Available settings: primary (true), secondary (false);

DualHeat

The DualHeat function allows to select one or two stages of heating (two stages of heating meaning heating with a reheater and, optionally, perimeter and a damper). The DualHeat function is activated according to the setting of the HeatDamperPriority Data Point (linked to the Heat Priority slot in the DamperControlComponent):

- primary (false): damper in the range of 0-50% (directly from the space temperature heating loop) and heater(s) in the range of 50-100% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop),
- secondary (true): heater(s) in the range of 0-50% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop) and damper in the range of 50-100% (directly from the space temperature heating loop).

Heater(s) in the primary stage can be reheater and/or perimeter. If available in the system and properly configured, they can be arranged in order according to the HeaterPriority variable (ReheaterControl folder):

- reheater/perimeter
- perimeter/reheater
- simultaneous.

If there is no reheater and/or perimeter in the system, the DualHeat function cannot be activated and the damper is the only heating source. If the reheater is available in the system, it can be configured as the primary stage heater and the DualHeat function can be activated. However, if the system is equipped with the reheater and parallel fan, the reheater will always be the secondary:

- parallel fan/reheater.

- **Max Occupied Cool Airflow:** receives a maximum airflow setpoint in the occupied state, set during the balancing process;;
- **Max Occupied Heat Airflow:** receives a maximum airflow setpoint in the occupied state, set during the balancing process;;
- **Min Occupied Cool Airflow:** receives a maximum airflow setpoint in the occupied state, set during the balancing process;;
- **Min Occupied Heat Airflow:** receives a maximum airflow setpoint in the occupied state, set during the balancing process;;
- **Standby Airflow:** receives an airflow setpoint set from the BMS system and, optionally, from the local room panel, when the room is temporarily unoccupied;
- **Unoccupied Airflow:** receives an airflow setpoint set from the BMS system and, optionally, from the local room panel, when the room is unoccupied;
- **User Airflow:** receives an airflow setpoint set for tests in the balancing process;
- **Startup Fan Airflow:** receives a value of minimum airflow starting the parallel fan; after the Startup Fan Delay time has elapsed, normal control starts (airflow control according to the calculated setpoint);
- **Fan Type:** allows to select a preferred fan type as series or parallel; version without fan is None;
- **Fan Active:** the rising state on this slot causes the Startup Fan Airflow setpoint to be used for the time specified in the Startup Fan Delay slot; after the time has elapsed, normal control starts;

- **Startup Fan Delay:** allows to set a startup time with a minimum airflow specified in the Startup Fan Airflow slot;
- **Unit Selector:** allows to set the unit type.

More

More on airflow setpoint as used in the VAV application: Airflow

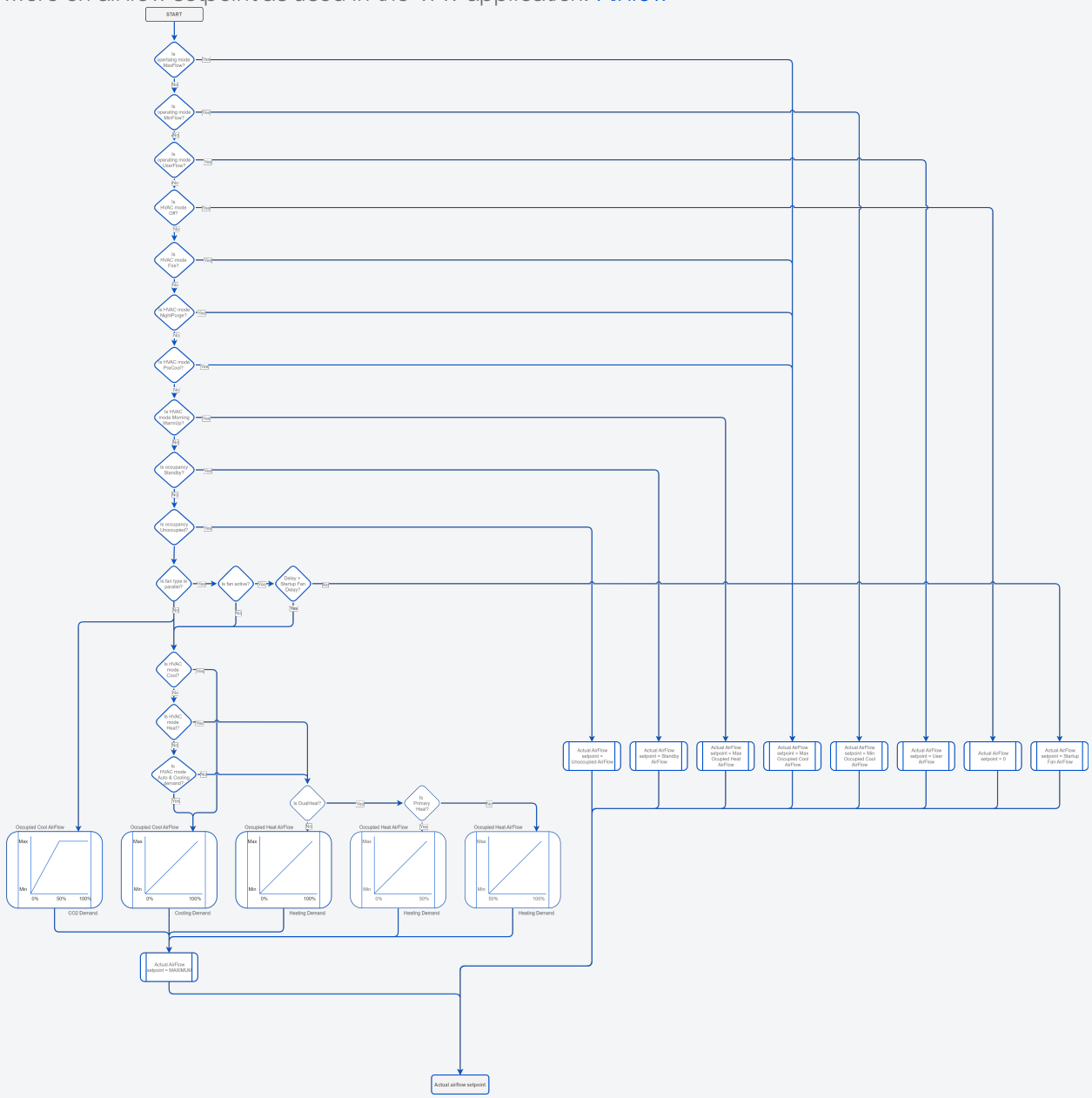
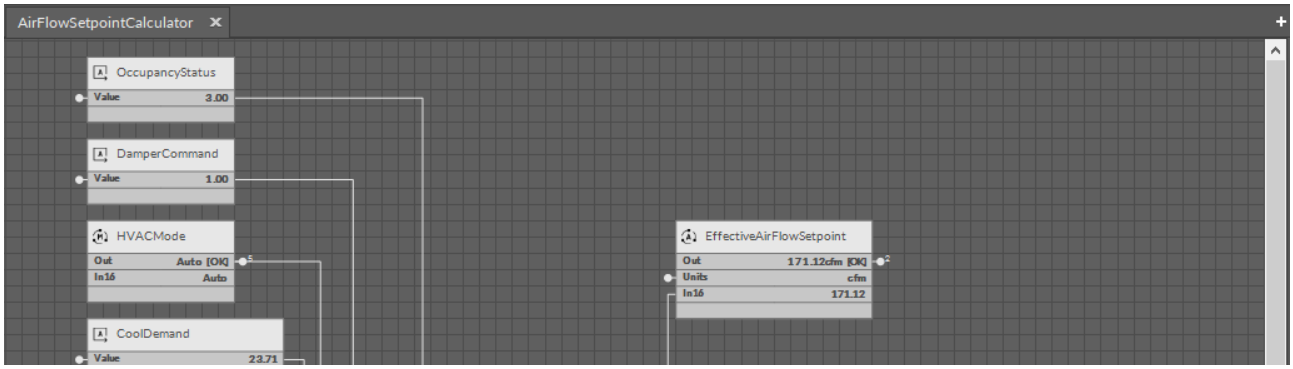


Figure 74. Airflow setpoint diagram



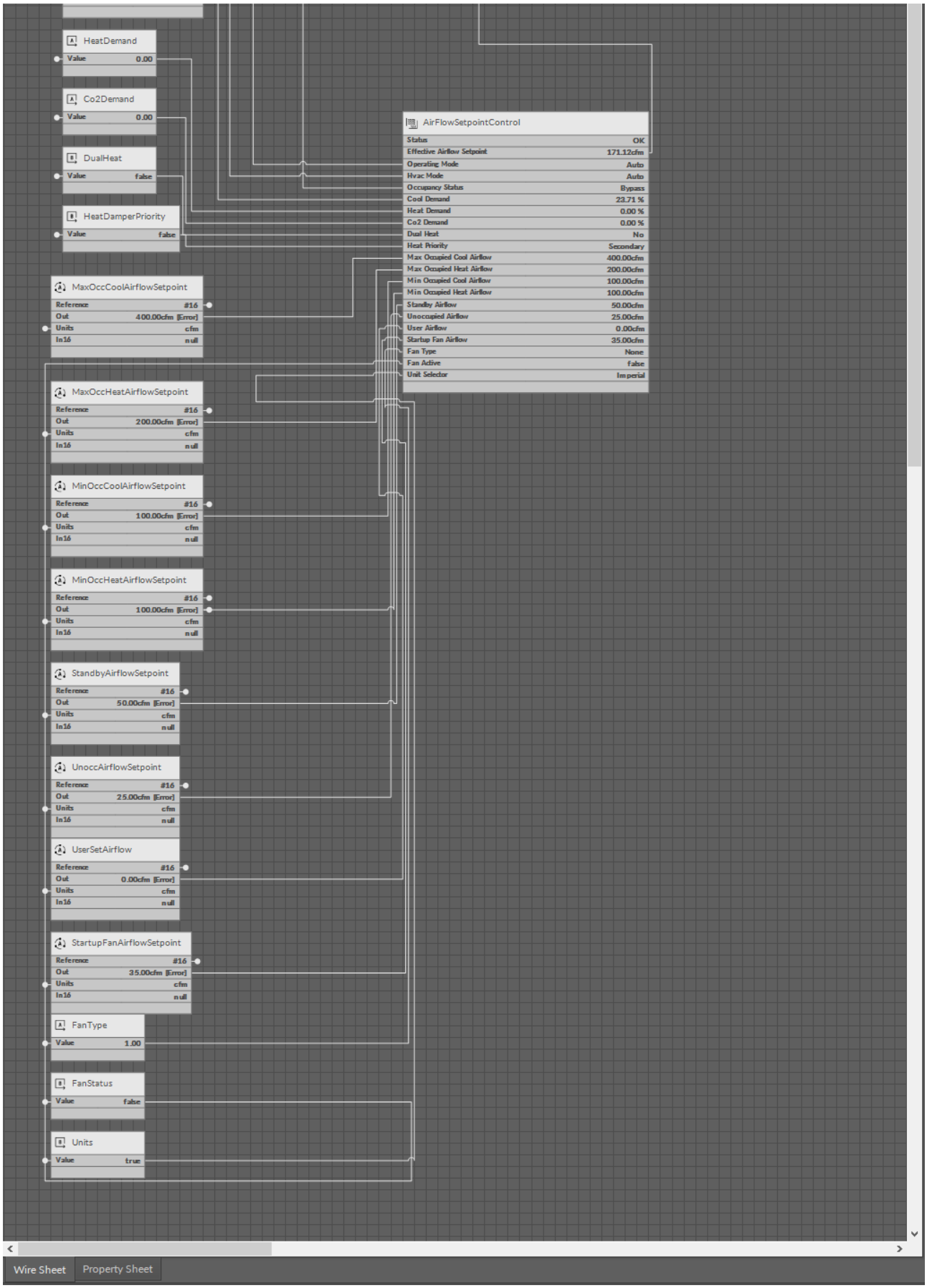


Figure 75. AirFlowSetpointControl in the application

DamperControl

Applicable to OS V1.7

The DamperControl component allows to control of the floating motor of the damper actuator. It allows to set the damper operating mode and it returns the calculated value of the damper opening based on connected PID loops. It also returns the airflow demand for heating or cooling based on the set device type (VAV or VVT).

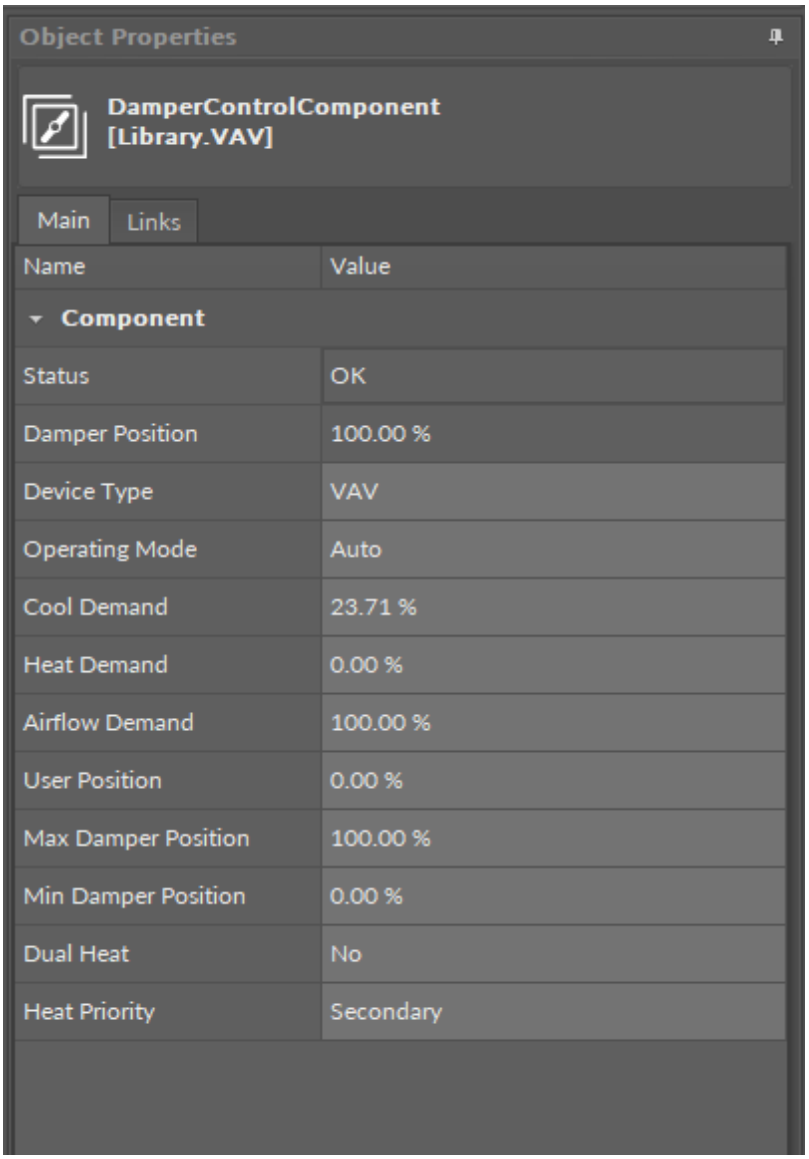


Figure 76. DamperControl slots

The DamperControl component has the following slots:

- **Status:** informs about the status of the component; if the Operating Mode slot or one of the demand input slots has a null value then the status is fault, otherwise it is OK;
    - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;
- **Damper Position:** shows a calculated damper position according to the damper operating mode;

Note

The damper position is calculated according to the operation mode specifications and taking into account the Max Damper Position and Min Damper Position settings. If the slots are set to values different than 100% and 0%, for example, to 90% and 10%, it means than 10-90% becomes a new full range for the damper operation. It can be used, for example, to fit the damper operation range to the size of the VAV.

- **Device Type:** allows to select a preferred device type - VAV for pressure-independent (false) and VVT for pressure-dependent (true);
- **Operating Mode:** shows the operating mode from the BMS or panel (from Modbus/BACnet network point) and allows to initiate a damper balancing procedure;
    - Available settings:
        - Auto: the damper is controlled according to the airflow control loop output (for VAV device type) or temperature control loops (for VVT device type),
        - Full Open: the damper control is forced to 100% open position,
        - Full Close: the damper control is forced to 0% open position (closed), even when the MinDamperPosition value is higher than 0%,
        - User Position: the damper control is forced to the user set position (the UserSetPosition variable),
        - Min Flow: the damper control is forced to fit the set minimum airflow for cooling in the occupied state,
        - Max Flow: the damper control is forced to fit the set maximum airflow for cooling in the occupied state,
        - User Flow: the damper control is forced to fit the set user airflow,
        - Calibrate: forces the damper calibration (the damper goes to 0%, then to 100%, and goes back to the control loop output);
- **Cool Demand:** receives the value from the connected cooling PID loop, which determines the damper control for cooling;
- **Heat Demand:** receives the value from the connected heating PID loop, which determines the damper control for heating;

Note

The control loops for cooling and heating are mutually exclusive, i.e., only one of them regulates the demand, while the other is disabled.

- **Airflow Demand:** receives the value from the connected airflow PID loop, which determines the damper control for a specific airflow demand;
- **User Position:** allows to define a damper opening for testing;
- **Max Damper Position:** allows to set the maximum damper opening - high limit for the range of the damper operation;
- **Min Damper Position:** allows to set the minimum damper opening - low limit for the range of the damper operation;
- **Dual Heat:** allows to enable or disable the DualHeat function, which determines of the damper is used as a heating source in systems with a reheater/perimeter);
- **Heat Priority:** allows to select a priority order for the primary of heating stage for damper as a heating source.

DualHeat

The DualHeat function allows to select one or two stages of heating (two stages of heating meaning heating with a reheater and, optionally, perimeter and a damper). The DualHeat function is activated according to the setting of the HeatDamperPriority Data Point (linked to the Heat Priority slot in the DamperControlComponent):

- primary (false): damper in the range of 0-50% (directly from the space temperature heating loop) and heater(s) in the range of 50-100% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop),

- secondary (true): heater(s) in the range of 0-50% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop) and damper in the range of 50-100% (directly from the space temperature heating loop).

Heater(s) in the primary stage can be reheater and/or perimeter. If available in the system and properly configured, they can be arranged in order according to the HeaterPriority variable (ReheaterControl folder):

- reheater/perimeter
- perimeter/reheater
- simultaneous.

If there is no reheater and/or perimeter in the system, the DualHeat function cannot be activated and the damper is the only heating source. If the reheater is available in the system, it can be configured as the primary stage heater and the DualHeat function can be activated. However, if the system is equipped with the reheater and parallel fan, the reheater will always be the secondary:

- parallel fan/reheater.

More

More on damper control as used in the VAV application: Damper Control

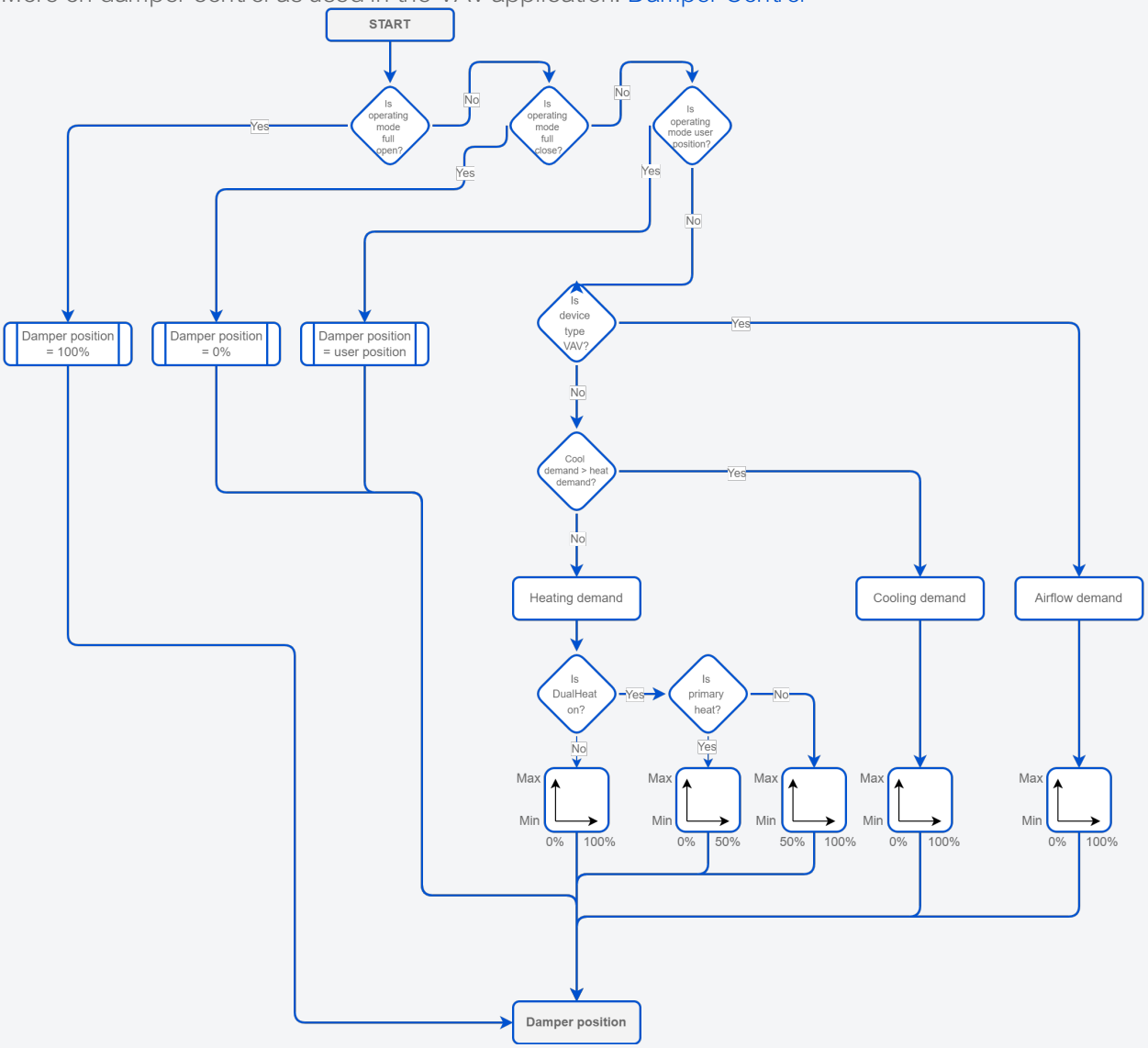


Figure 77. Damper control

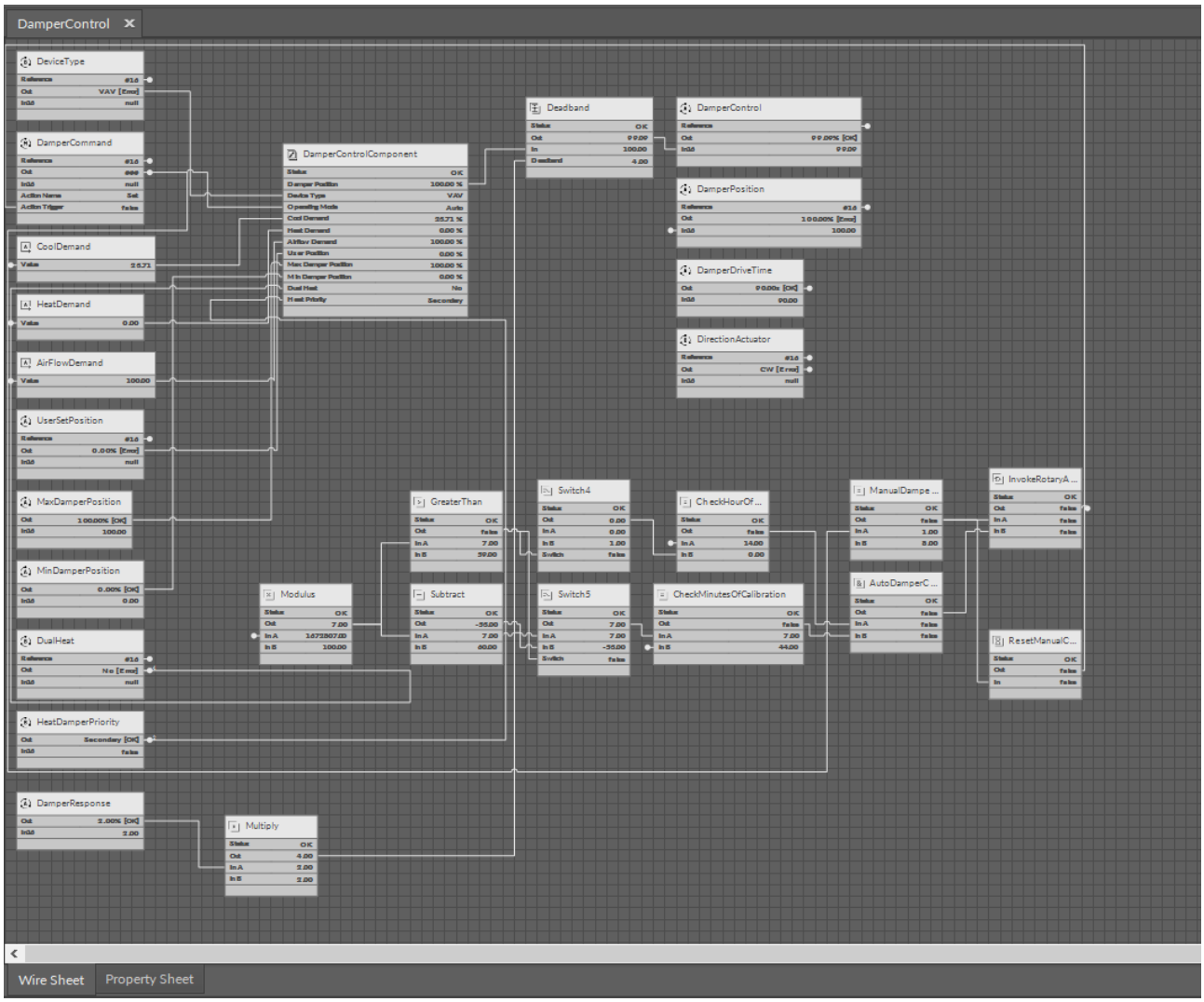


Figure 78. DamperControl in the application

VAVFanControl

Applicable to OS V1.7

The VAVFanControl component allows to control the fan speed (if configured - both analog and digital depending on the fan type), occupancy status, and the demand for heating or cooling. The component determines the effective setpoint for regulating the fan speed, based on the output from the temperature control loop (only for heating in parallel fan case), the setpoints for minimum and maximum fan speed, and the current occupancy status and operating mode (normal, balancing). Additionally, it is possible to use the fan in a selected HVAC mode - such as WarmUp.

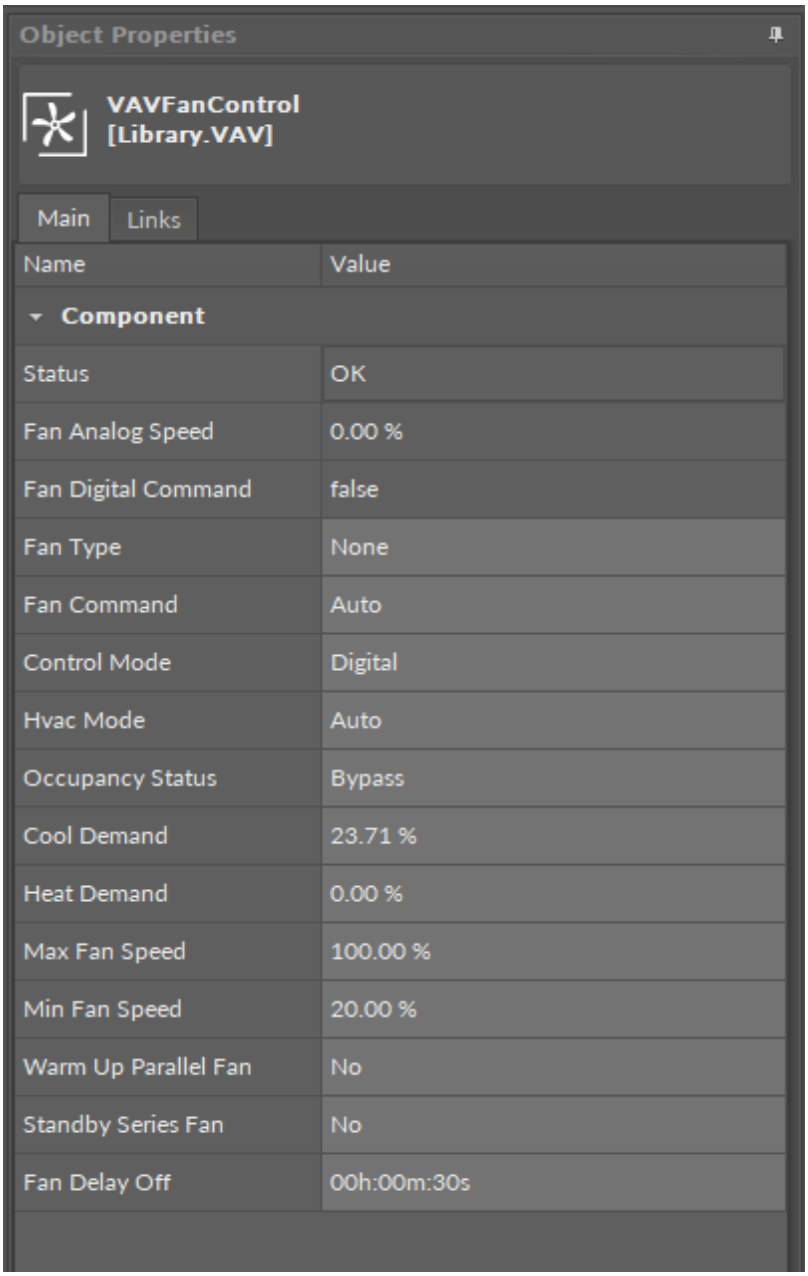


Figure 79. VAVFanControl slots

The VAVFanControl component has the following slots:

- **Status:** informs about the status of the component; if the Fan Type/Fan Commend/Occupancy Status/ HVAC Mode slot or one of the demand input slots has a null value then the status is fault, otherwise it is OK;
  - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;
- **Fan Analog Speed:** returns an analog fan speed resulting from the connected control loops;

- **Fan Digital Command:** returns a calculated digital fan command;

Note

Fan control is calculated in the component based on the fan control mode, type, HVAC mode, occupancy, and active control loops. Fan analog speed also takes into account Max Fan Speed and Min Speed Values, which can limit the range of the fan operation.

- **Fan Type:** specifies a fan type (series or parallel; version without a fan is None);
- **Fan Command:** allows to set a test fan control, if configured in Fan Type for use;
  - Available settings: Auto, Max Speed, Min Speed, Stop;
- **Control Mode:** specifies a control mode for the fan in the analog mode:
  - Available settings:
    - Analog (linearly according to the heating/cooling demand and within limit speeds set in the MinFanSpeed and MaxFanSpeed components),
    - Digital (on/off binary mode available in occupied, bypass and, optionally, standby statuses);
- **Hvac Mode:** receives the HVAC mode set in the HVACMode variable;
  - Available settings: Auto, Heat, Cool, MorningWarmUp, PreCool, NightPurge, Fire, Off;

HVAC modes

- **Auto:**
  - parallel fan: working on heating demand,
  - series fan: working if occupied/bypass or standby (if allowed);
- **Heat:**
  - parallel fan: working on heating demand,
  - series fan: working only on heating demand if occupied/bypass or standby (if allowed);
- **Cool:**
  - parallel fan: inactive,
  - series fan: working only on cooling demand if occupied/bypass or standby (if allowed);
- **MorningWarmUp:**
  - parallel fan: if the WarmUpParallelFan variable is set to true, the parallel fan is active in this mode,
  - series fan: ventilator set to the maximum speed;
- **PreCool:**
  - parallel fan: inactive,
  - series fan: ventilator set to the maximum speed;
- **NightPurge:**
  - parallel fan: inactive,
  - series fan: ventilator set to the maximum speed;
- **Off:** both types of fan disabled;
- **Fire:**
  - parallel fan: inactive,
  - series fan: set to the maximum speed.

- **Occupancy Status:** specifies an occupancy status resulting from the values of window contact, present motion, network occupancy, schedule occupancy, airflow detection, and panel button override;
  - Available settings: Occupied, Unoccupied; Standby, Bypass;
- **Cool Demand:** receives the value from the connected cooling PID loop;
- **Heat Demand:** receives the value from the connected heating PID loop;
- **Max Fan Speed:** allows to set a maximum fan speed - high limit for the range of the fan operation;
- **Min Fan Speed:** allows to set a minimum fan speed - low limit for the range of the fan operation;
- **Warm Up Parallel Fan:** allows to run a parallel fan in the MorningWarmUp HVAC mode;
- **Standby Series Fan:** allows to run a series fan in a standby occupancy mode;
- **Fan Delay Off:** allows to set a delay fan off time after heat demand goes to 0%.

Note

In systems with a water reheater the Fan Delay Off may be set to 0. However, setting the delay is required in systems with an electric reheater.

More

More on fan control as used in the VAV application: Fan Control

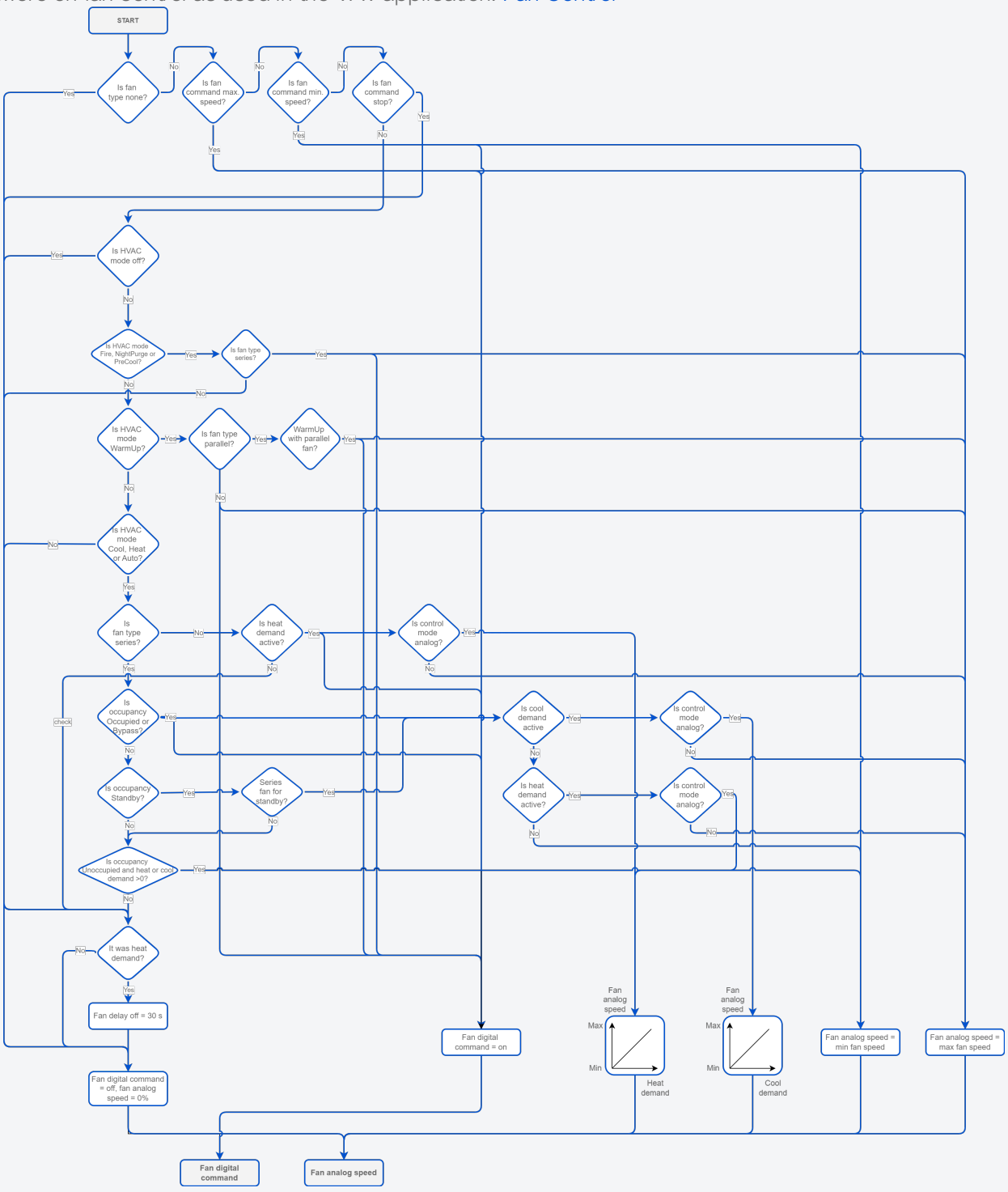


Figure 80. Fan control

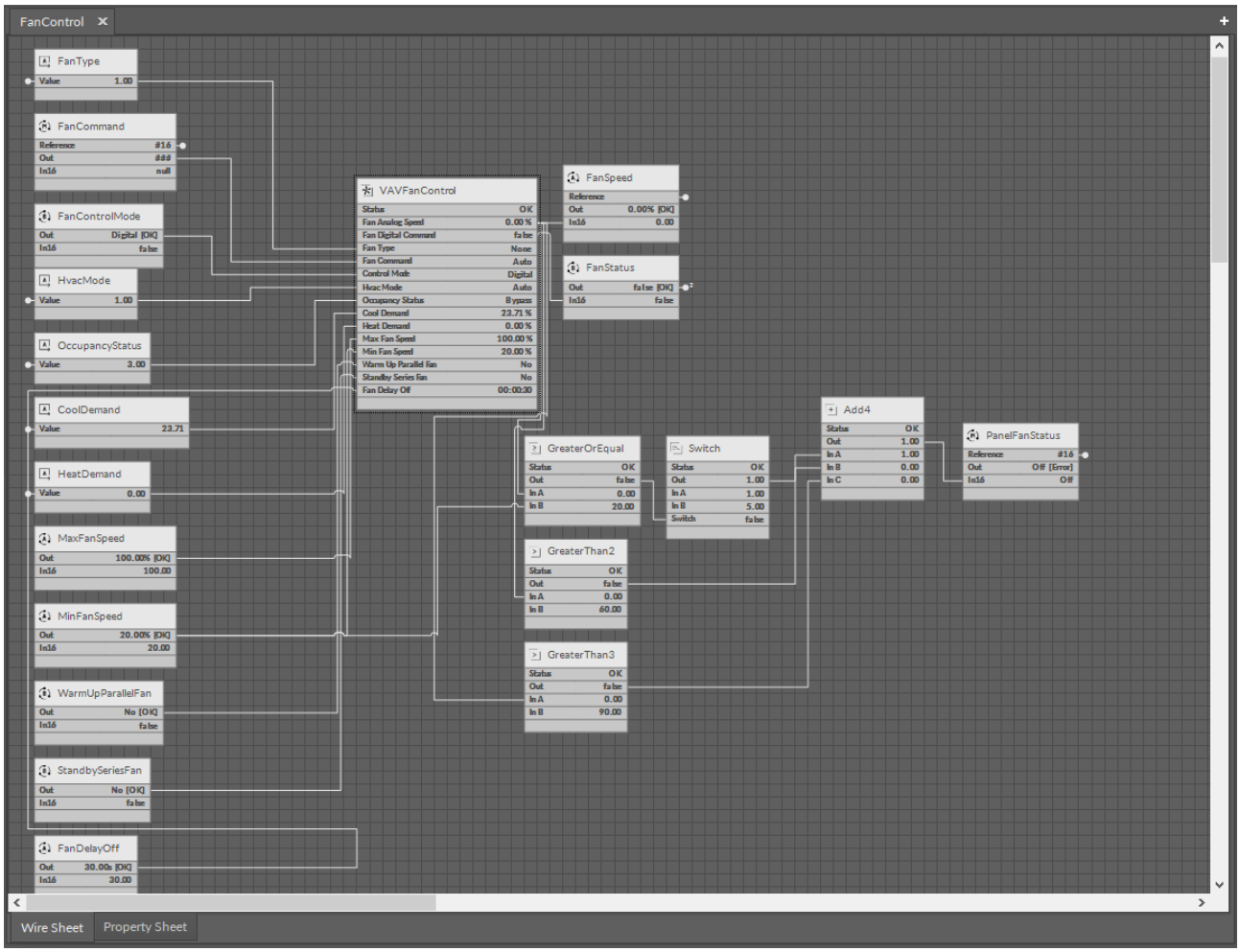


Figure 81. VAVFanControl in the application

HeaterControl

Applicable to OS V1.7

The HeaterControl component allows to control the reheater/perimeter depending on the selected operating mode, if it is configured, both analog and digital (depending on the selected control mode like PWM, staged, floating, or digital). Additionally, it is possible to use the reheater/perimeter in a selected HVAC mode - such as WarmUp, when there are no other operating conditions and when an operating mode other than normal is forced - such as balancing or testing.

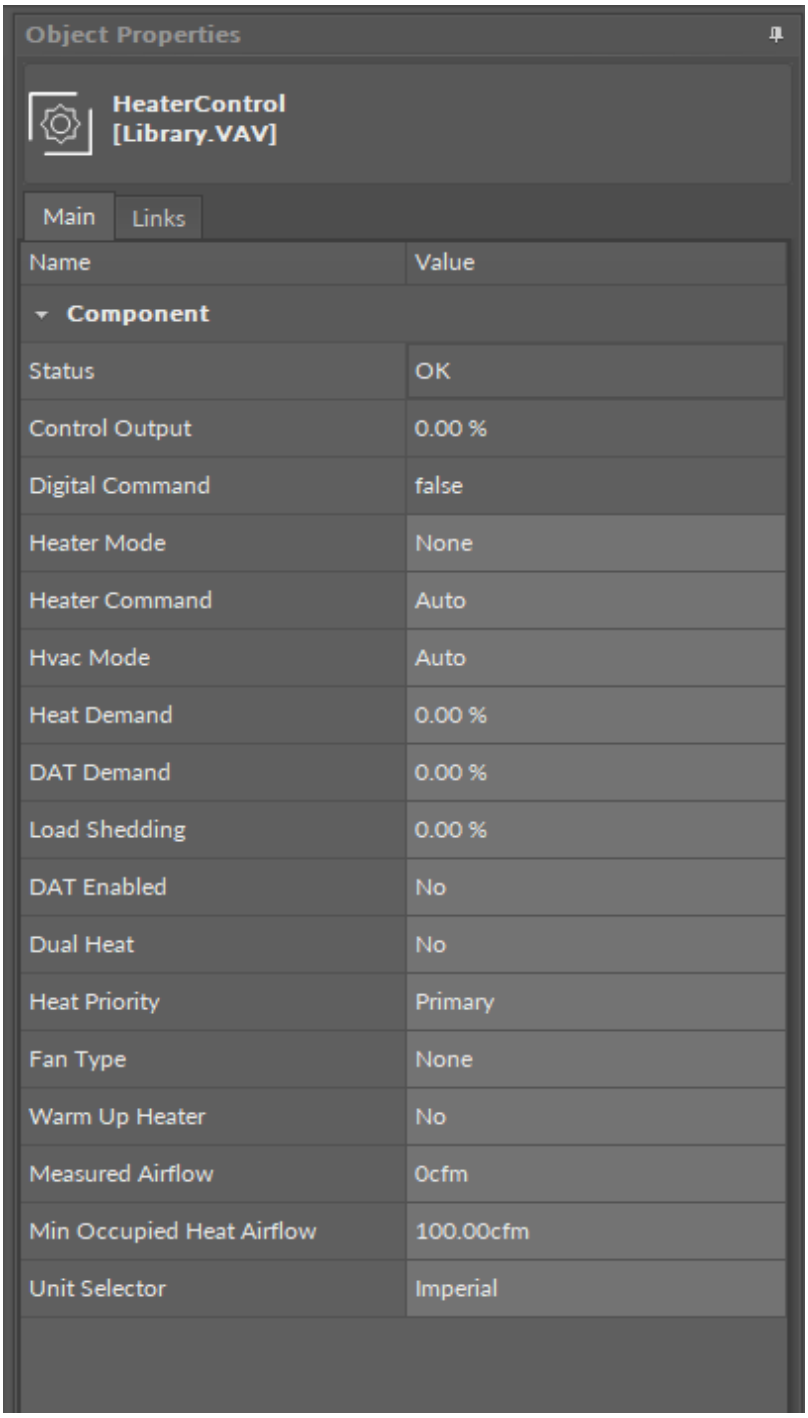


Figure 82. HeaterControl slots

The HeaterControl component has the following slots:

- **Status:** informs about the status of the component; if the Heater Mode/Heater Command/HVAC Mode slot or one of the demand input slots has a null value then the status is fault, otherwise, the status is OK;
  - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;
- **Control Output:** returns a value of the heating control (based on heating demand or DAT demand (if enabled), parameters configured, and override values resulting from the application) for the heater (i.e., analog, floating, PWM actuator, and also as a staged sequence_;
- **Digital Command:** return a value of the heating control for on/off actuator (in the VAV application, used for digital on/off perimeter);
- **Heater Mode**: specifies the type of reheater which the component is used for (reheater or perimeter, none if no heater is available in the system);
- **Heater Command:** allows to set a test heater control, if configured in Heater Type for use;
  - Available settings: Auto, Open/On, Close/Off;
- **Hvac Mode:** receives the HVAC mode set in the AirflowSetpointCalculator variable;
  - Available settings: Auto, Heat, Cool, MorningWarmUp, PreCool, NightPurge, Fire, Off;

HVAC modes

- **Auto:** working according to the heating demand,
- **Heat:** working according to the heating demand,
- **Cool:** switched off,
- **MorningWarmUp:** if the WarmUpHeater variable is set to true, the reheater works according to the heating demand,
- **PreCool:** switched off,
- **NightPurge:** switched off,
- **Off:** switched off,
- **Fire:** switched off.

- **Heat Demand:** receives the value from the connected heating PID loop;
- **DAT Demand:** receives the value from the connected discharge air temperature PID loop;
- **Load Shedding:** specifies the value representing a level of energy load cut-off, which decreases the value of the reheater control output (RH control output = (100% - LoadShedding)*HeatDemand);
- **DAT Enabled:** allows to enable heater control with DAT demand if the discharge air temperature sensor is connected and provides temperature measurements for the control loop;
- **Dual Heat:** informs if the DualHeat function is activated or not;
  - Available settings: No, Yes;

DualHeat

The DualHeat function allows to select one or two stages of heating (two stages of heating meaning heating with a reheater and, optionally, perimeter and a damper). The DualHeat function is activated according to the setting of the HeatDamperPriority Data Point (linked to the Heat Priority slot in the DamperControlComponent):

- primary (false): damper in the range of 0-50% (directly from the space temperature heating loop) and heater(s) in the range of 50-100% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop),
- secondary (true): heater(s) in the range of 0-50% (indirectly from the discharge temperature heating loop and directly from the space temperature heating loop) and damper in the range of 50-100% (directly from the space temperature heating loop).

Heater(s) in the primary stage can be reheater and/or perimeter. If available in the system and properly configured, they can be arranged in order according to the HeaterPriority variable (ReheaterControl folder):

- reheater/perimeter
- perimeter/reheater
- simultaneous.

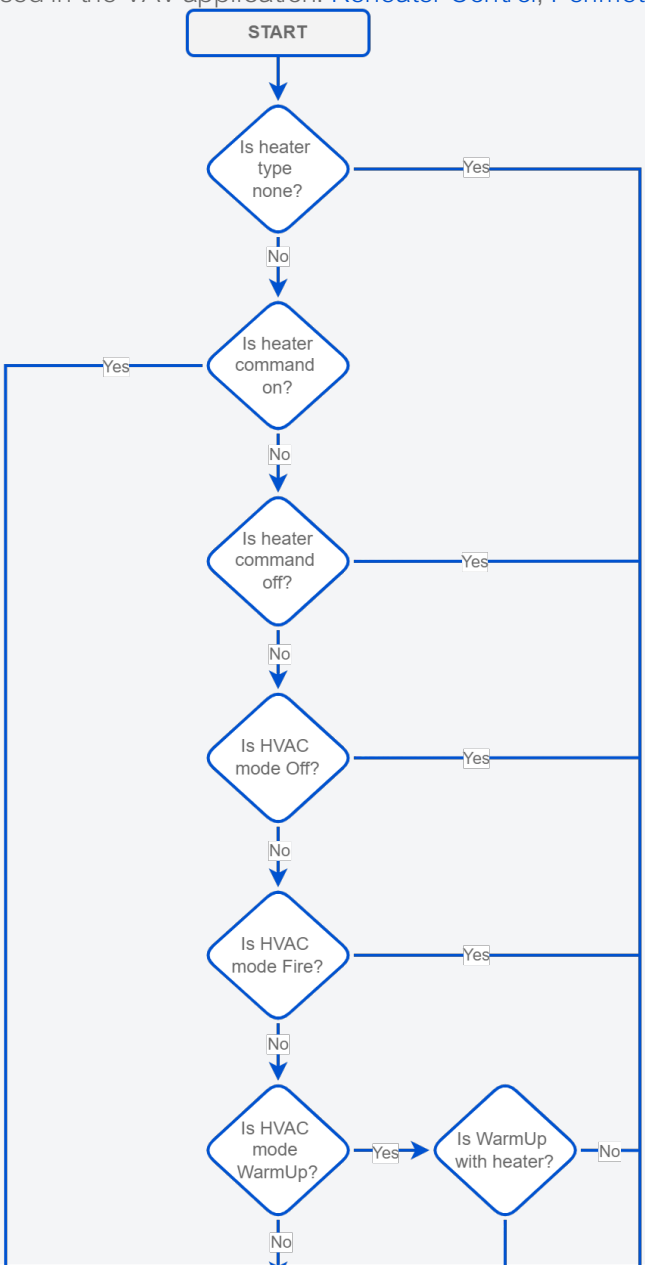
If there is no reheater and/or perimeter in the system, the DualHeat function cannot be activated and the damper is the only heating source. If the reheater is available in the system, it can be configured as the primary stage heater and the DualHeat function can be activated. However, if the system is equipped with the reheater and parallel fan, the reheater will always be the secondary:

- parallel fan/reheater.

- **Heat Priority:** allows to determine if the heating source, which this component is used for (reheater or perimeter), is a first or second priority heating source for the primary stage of the DualHeat function;
- **Fan Type:** specifies a fan type (series or parallel; version without a fan is None);
- **WarmUp Heater:** allows to use the heater in the Morning WarmUp HVAC mode (as described in the Hvac Mode slot);
- **Measured Airflow:** receives a value of the measured airflow from the AirflowCalculator component;
- **Min Occupied Heat Airflow:** specifies a value of the minimum airflow setpoint for heating in the occupied status; it is a minimum airflow value for the reheater to start working (for perimeter, may be set to 0);
- **Unit Selector:** allows to set the unit type.

More

More on the heaters control as used in the VAV application: Reheater Control, Perimeter Control.



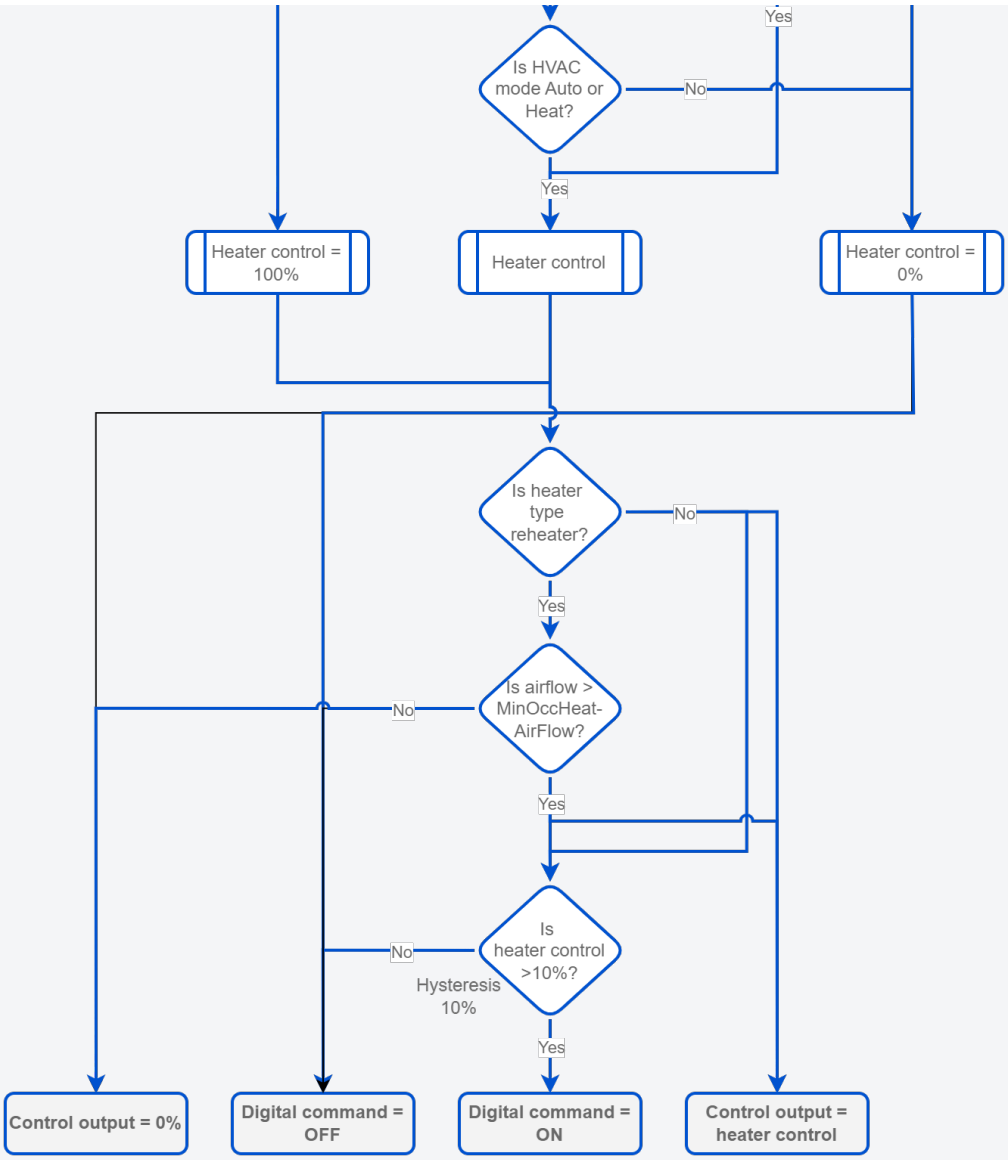


Figure 83. Reheater control

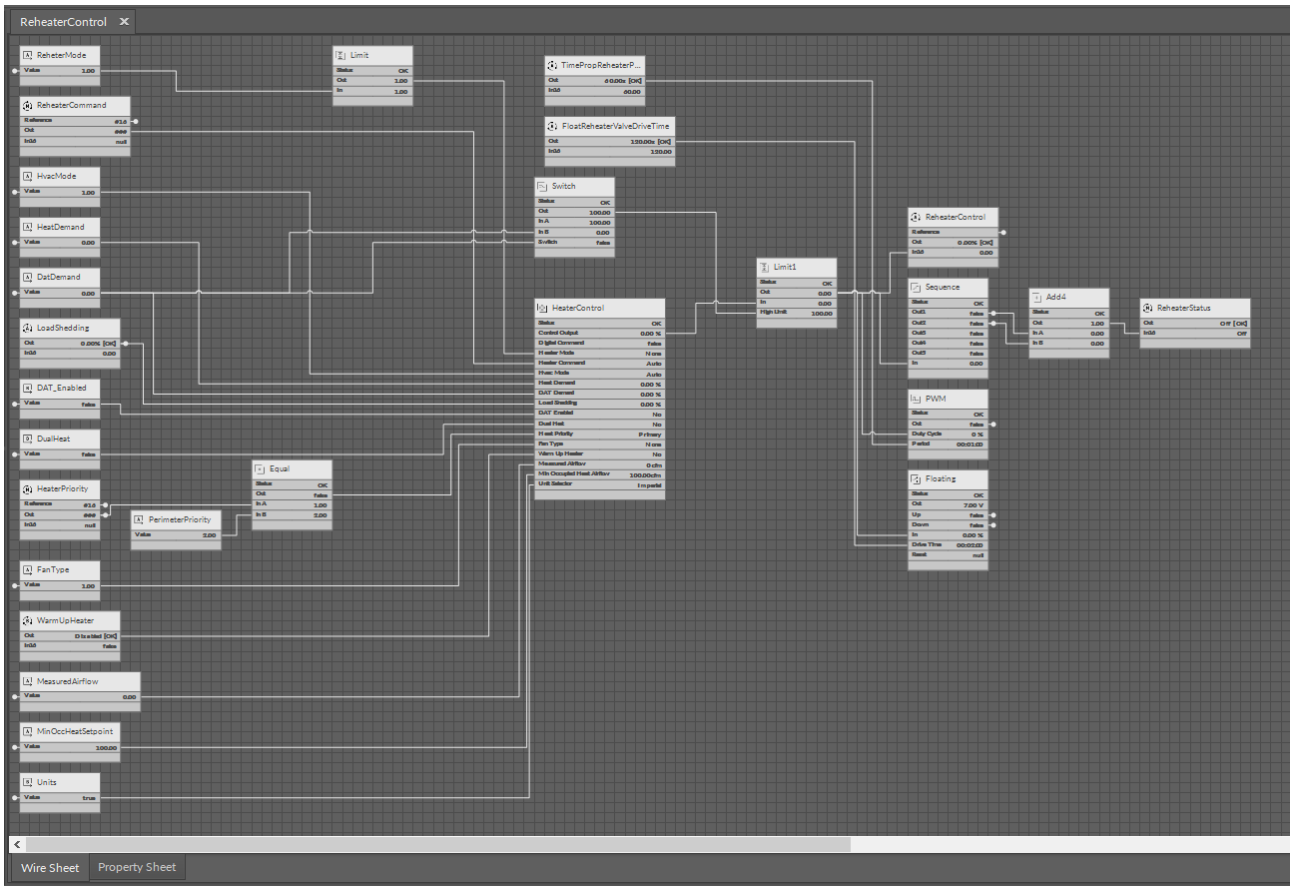


Figure 84. HeaterControl in the application

5.1.2 ComfortControl Library

- AdvancedOccupancyCalculator
- RoomCo2Selector
- TemperatureSensorSelector
- TemperatureSetpointSelector

AdvancedOccupancyCalculator

Applicable to OS V1.7

The AdvancedOccupancyCalculator component determines the room occupancy status based on input signals from the window contact, presence detector, local schedule, BMS network variable, and airflow detection. The component also allows the auto occupancy function, which determines the occupancy status (occupied/unoccupied) based on the comparison of the measured airflow and the effective airflow setpoint.

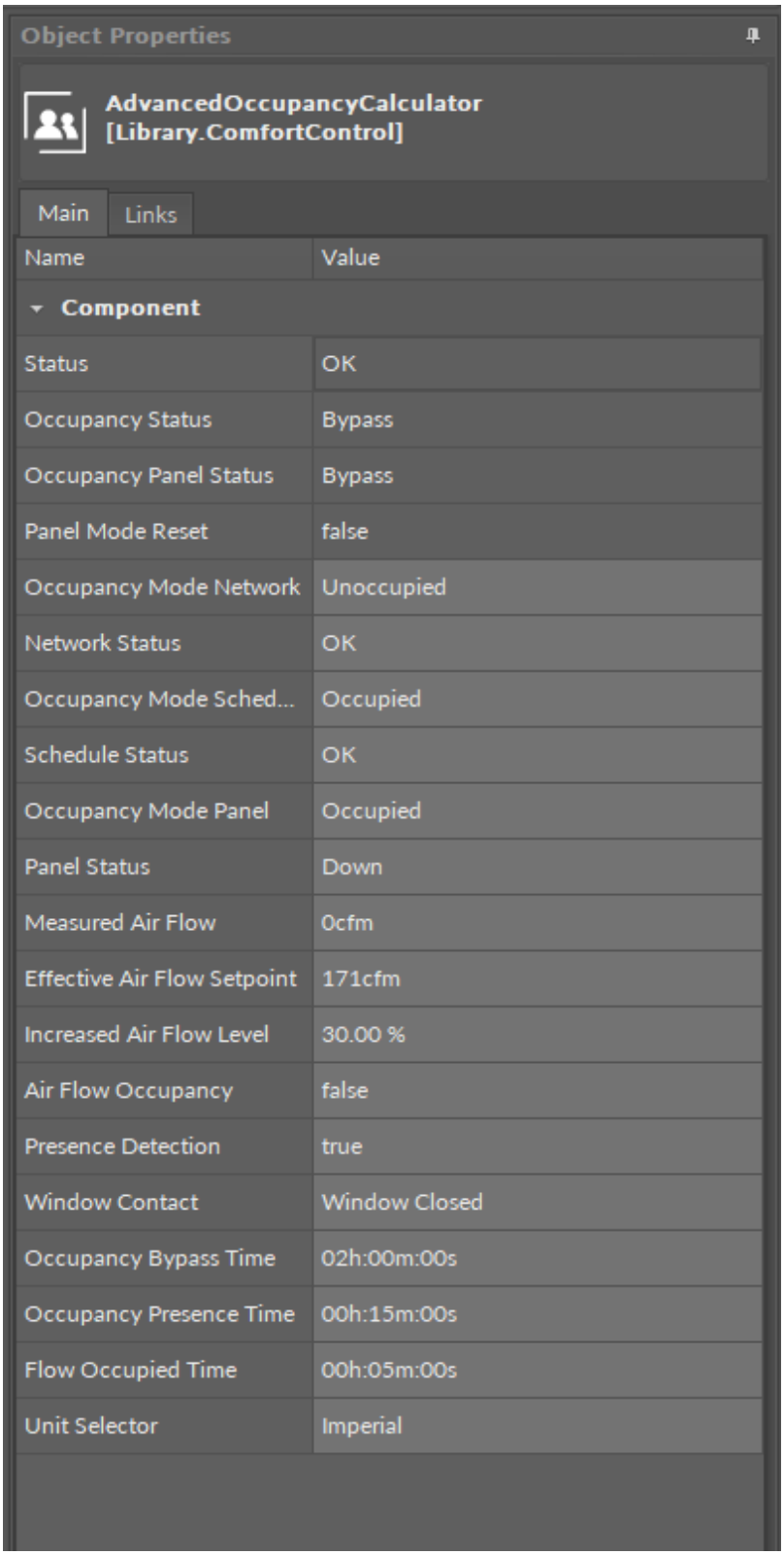


Figure 85. AdvancedOccupancyCalculator slots

The AdvancedOccupancyCalculator component has the following slots:

- **Status:** informs about the status of the component; if one of the Status-type slots in the component (Network Status/Schedule Status/Panel Status) or Presence Detection/Window Contact slot has a null value, then the status is fault;
    - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;

- **Occupancy Status:** calculates the occupancy status resulting from the values of window contact, present motion, network occupancy, schedule occupancy, airflow detection, and panel button override;
- **Occupancy Panel Status:** (only for panel) calculates the occupancy status resulting from the values of window contact, present motion, network occupancy, schedule occupancy, airflow detection, panel button override;
- **Panel Mode Reset:** resetting the panel's occupancy mode after deactivating the occupancy mode which was forced by the room panel;
- **Occupancy Mode Network:** receives the occupancy status from the BMS or AHU system;
- **Network Status:** shows the status of the BMS or AHU system; if it is OK then the Occupancy Mode Network is used, otherwise, it is ignored;
- **Occupancy Mode Schedule:** receives the occupancy status from the local schedule;
- **Schedule Status:** receives the status of the local schedule; if it is OK then the Occupancy Mode Schedule is used, otherwise, it is ignored;
- **Occupancy Mode Panel:** receives the occupancy status from the local panel, activates the Bypass mode (temporarily occupied); overridden function works if the Occupancy status is other than occupied;
- **Panel Status:** shows the status of the local panel; if it is OK then the Occupancy Mode Panel is used, otherwise, it is ignored;
- **Measured Air Flow:** receives the currently measured airflow value;
- **Effective Air Flow Setpoint:** receives the value of the effective airflow setpoint;
- **Increased Air Flow Level:** specifies the value by which the airflow must increase compared to the setpoint value to change the occupancy mode to occupied in the auto occupancy function;
- **Air Flow Occupancy:** if set to true, allows enabling the auto occupancy function;

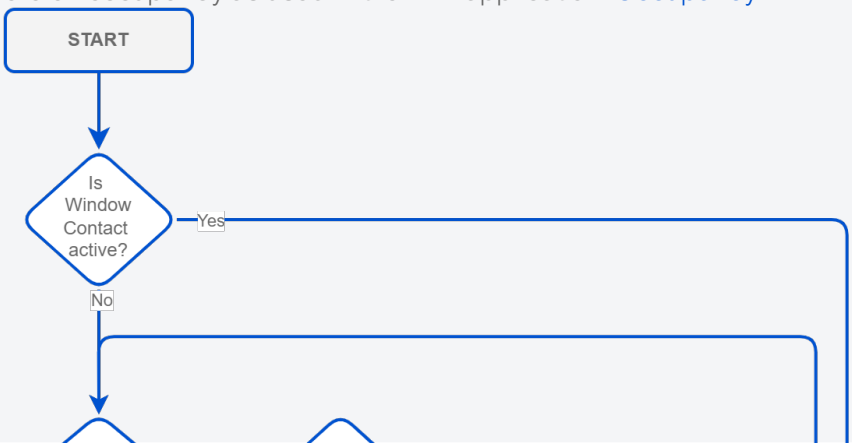
Auto Occupancy

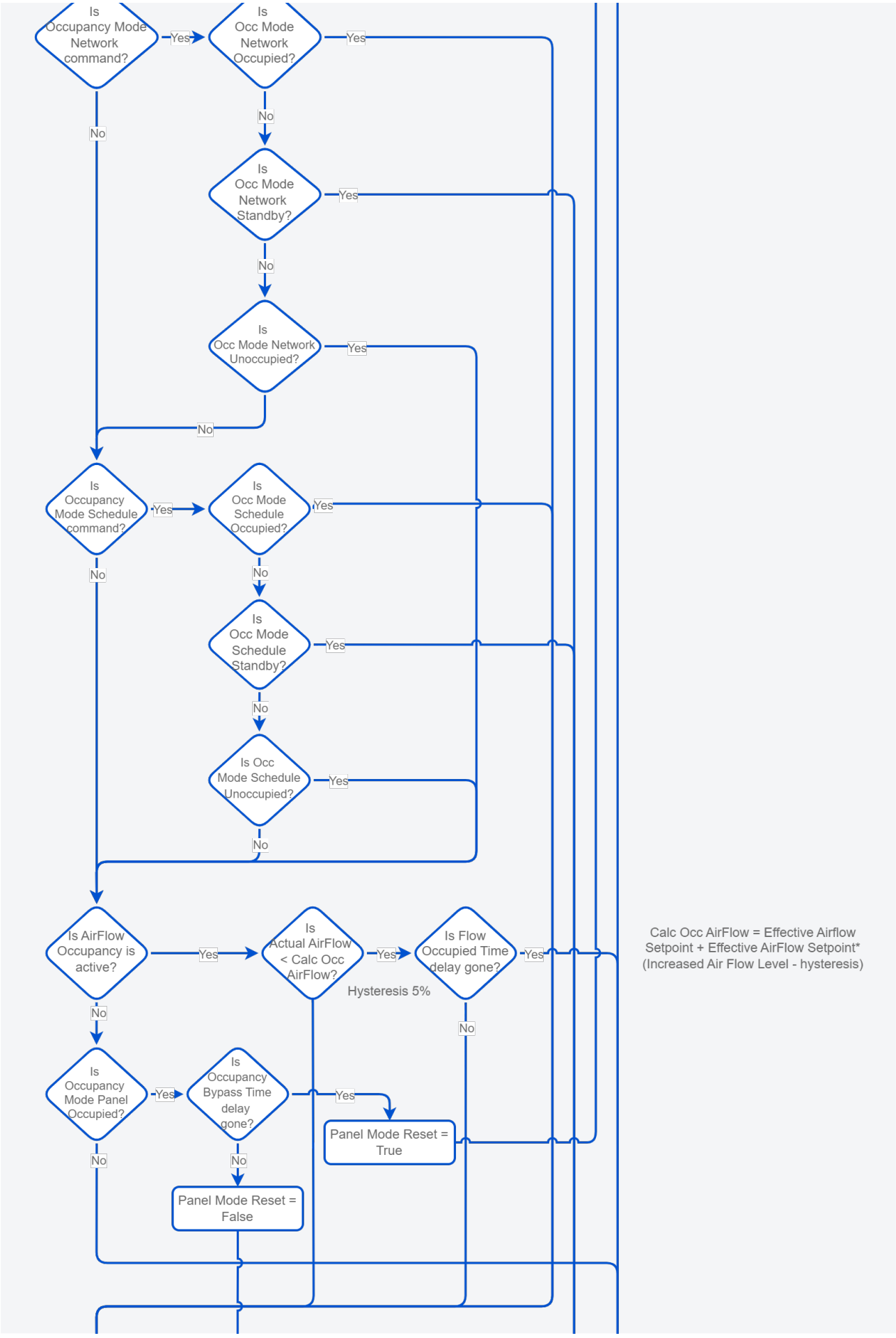
The auto occupancy function changes the occupancy status between occupied and unoccupied by comparing the Measured Airflow value with the Effective Airflow Setpoint value. If the measured airflow is higher than the effective airflow setpoint by the Increased Airflow Level value.

- **Presence Detection:** specifies if the presence sensor is available and active;
- **Window Contact:** receives the window open/close state from the connected contact switch (if available);
- **Occupancy Bypass Time:** specifies the time to maintain the Bypass mode (temporarily occupied); after the time expires, the occupancy mode returns to standby;
- **Occupancy Presence Time:** specifies the time to maintain the Occupied mode after motion has been detected; after the time expires and no motion is detected, the occupancy mode returns to a previous state;
- **Flow Occupied Time:** specifies the delay time to change from Occupied to Unoccupied state after reducing airflow;
- **Unit Selector:** allows to set the unit type.

More

More on occupancy as used in the VAV application: Occupancy




Is Occupancy Mode Network command?
Yes
Is Occ Mode Network Occupied?
Yes
No
No
Is Occ Mode Network Standby?
Yes
No
Is Occ Mode Network Unoccupied?
Yes
No
Is Occupancy Mode Schedule command?
Yes
Is Occ Mode Schedule Occupied?
Yes
No
No
Is Occ Mode Schedule Standby?
Yes
No
Is Occ Mode Schedule Unoccupied?
Yes
No
Is AirFlow Occupancy is active?
Yes
Is Actual AirFlow < Calc Occ AirFlow?
Yes
Is Flow Occupied Time delay gone?
Yes
Hysteresis 5%
No
No
Is Occupancy Mode Panel Occupied?
Yes
Is Occupancy Bypass Time delay gone?
Yes
Panel Mode Reset = True
No
No
Panel Mode Reset = False
Calc Occ AirFlow = Effective Airflow Setpoint + Effective AirFlow Setpoint* (Increased Air Flow Level - hysteresis)

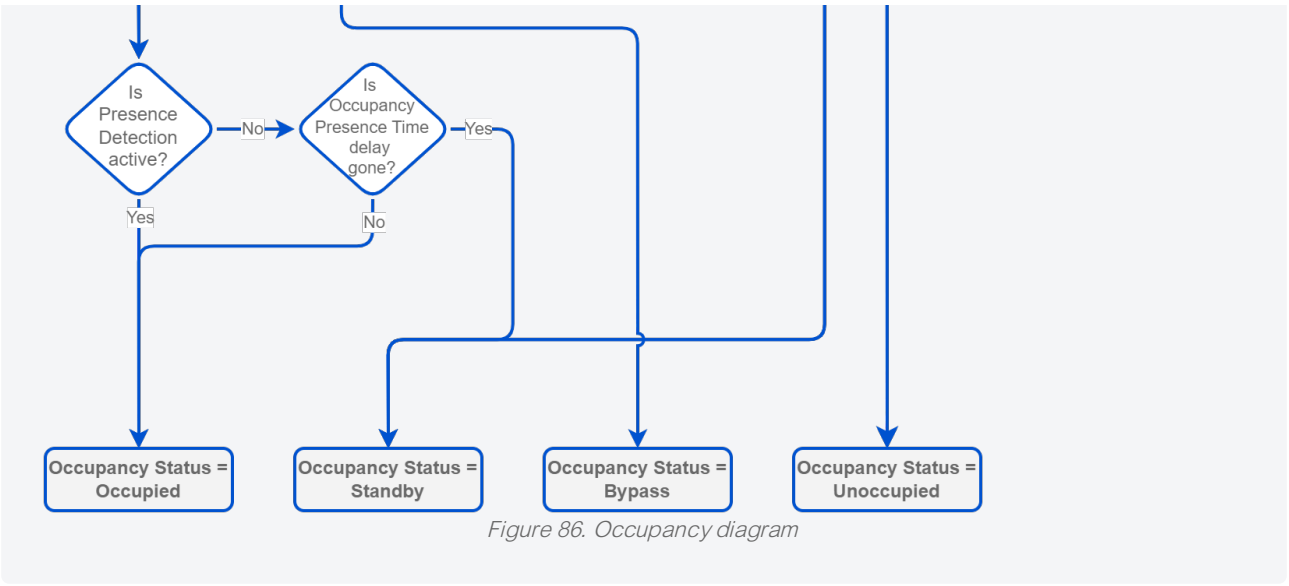


Figure 86. Occupancy diagram

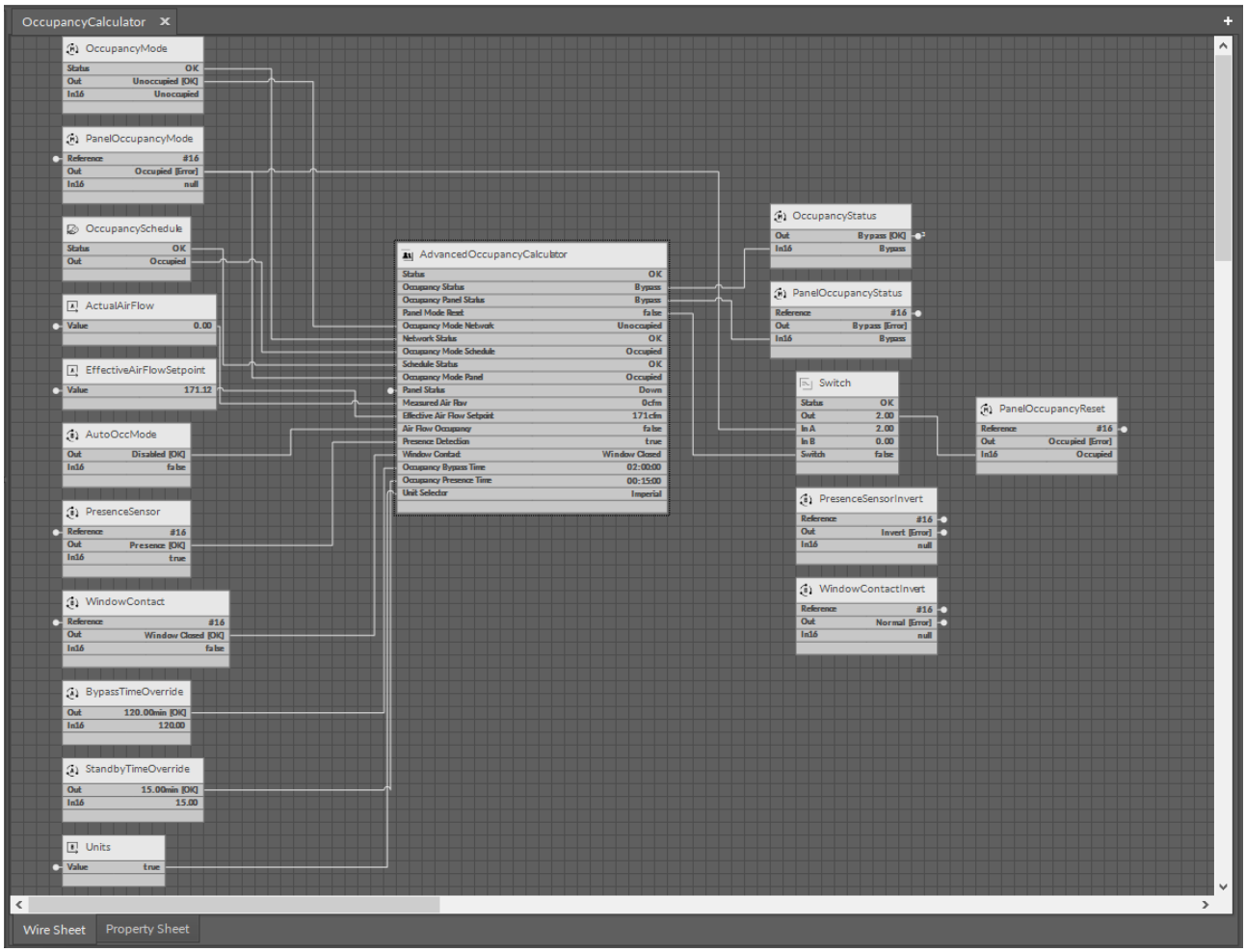


Figure 87. AdvancedOccupancyCalculator in the application

RoomCo2Selector

Applicable to OS V1.7

The RoomCO2Selector component is used to determine the CO_2 signal from the priority source, starting from the room panel, then the input, network, and constant value.

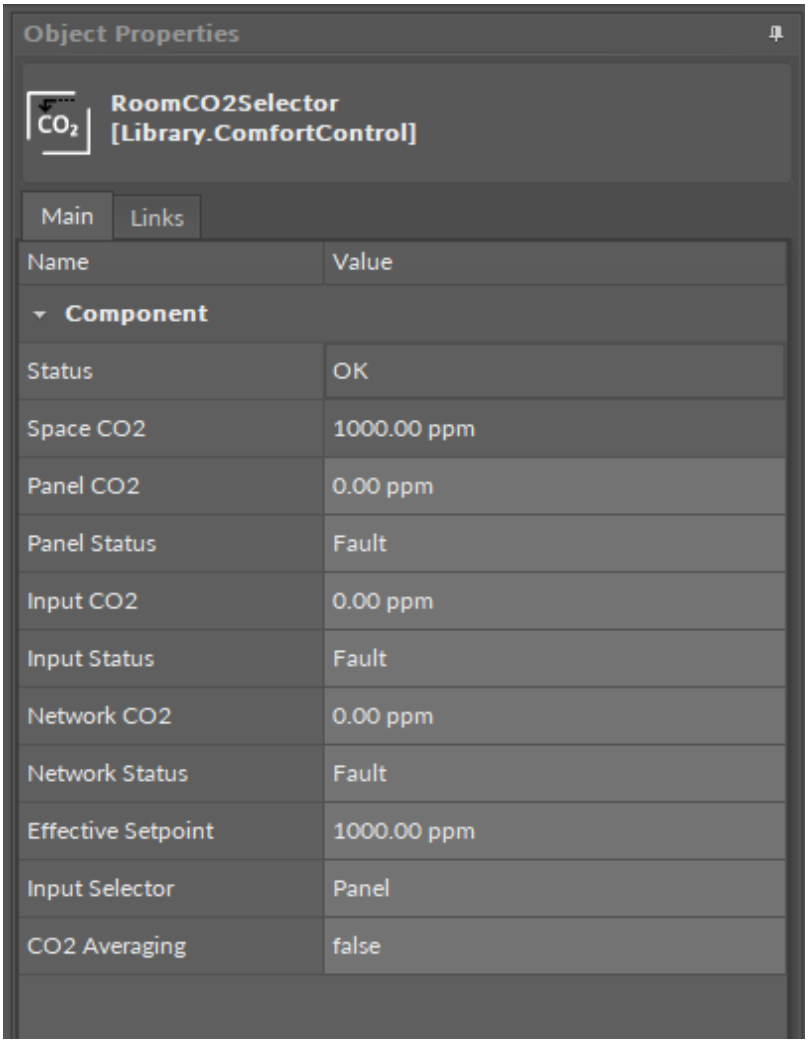


Figure 88. RoomCO2Selector slots

The RoomCO2Selector component has the following slots:

- **Status:** informs about the status of the component; if one of the Status-type slots in the component (Panel Status/Input Status/Network Status) has a null value, then the status is fault;
  - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;
- **Space CO2:** calculates space CO_2 based on selected and available CO_2 inputs (or using Effective Setpoint if not);
- **Panel CO2:** receives the CO_2 concentration read from the local room panel;
- **Panel Status:** shows the status of the local panel; if it is OK then the Panel CO2 is used, otherwise, it is ignored;
- **Input CO2:** receives the CO_2 concentration read from the universal input of the VAV controller;
- **Input Status:** shows the status of the universal input of the VAV controller; if it is OK then the Input CO2 is used, otherwise, it is ignored;
- **Network CO2:** receives the CO_2 concentration read from the network controller;
- **Network Status:** shows the status of the network controller; if it is OK then the Network CO2 is used, otherwise, it is ignored;

- **Effective Setpoint:** allows to link or set the effective setpoint value to be used if no CO_2 measurement is available;
- **Input Selector:** allows to select a preferred input signal of CO_2 concentration - panel, input, network;
- **CO2 Averaging:** allows to set the CO_2 concentration as a calculation of an average of available CO_2 inputs.

More

More on CO_2 control as used in the VAV application: CO2.

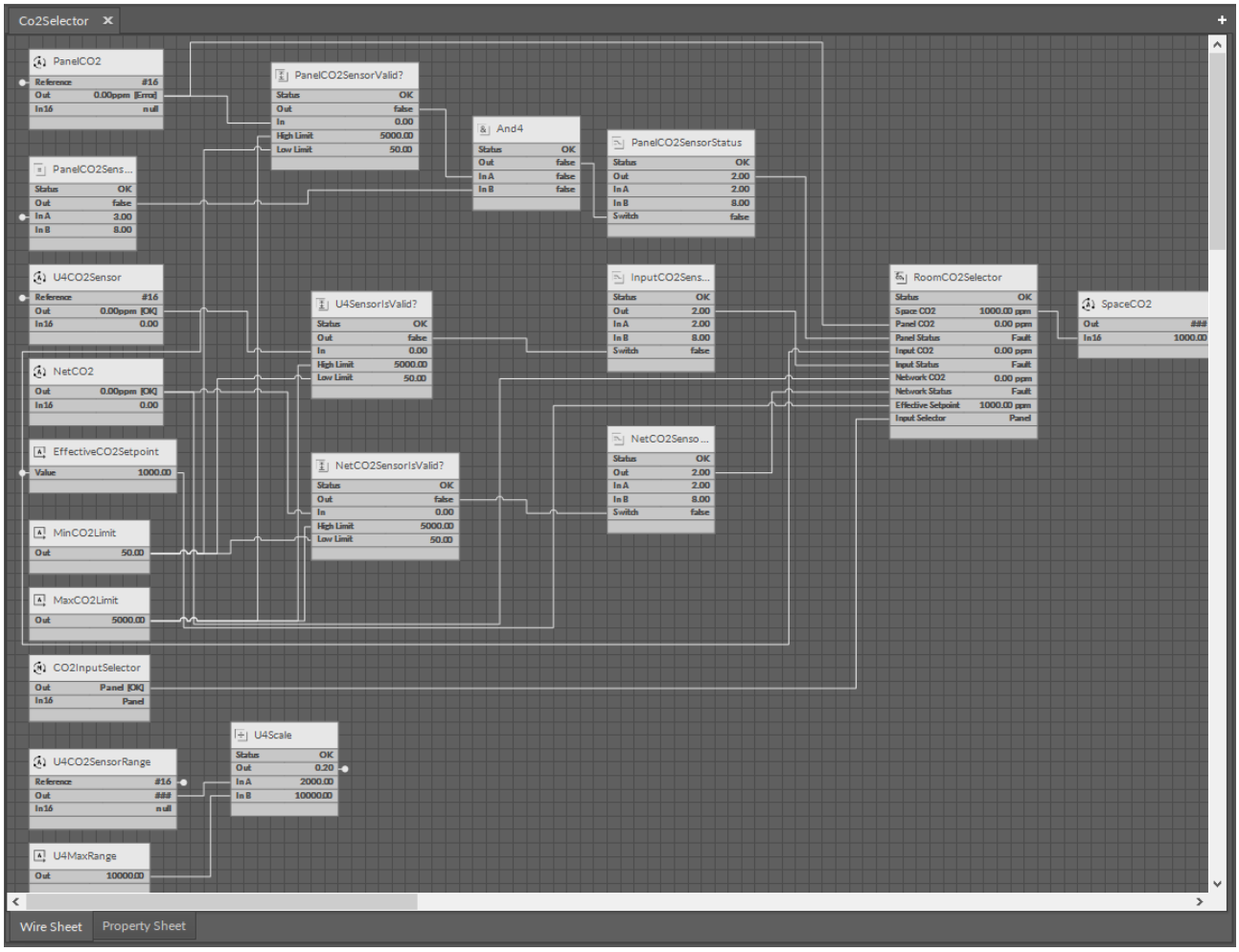


Figure 89. RoomCO2Selector in the application

TemperatureSensorSelector

Applicable to OS V1.7

The TemperatureSensorSelector component is used to determine the temperature signal from the priority source, starting from the room panel, then the input, network, and constant value.

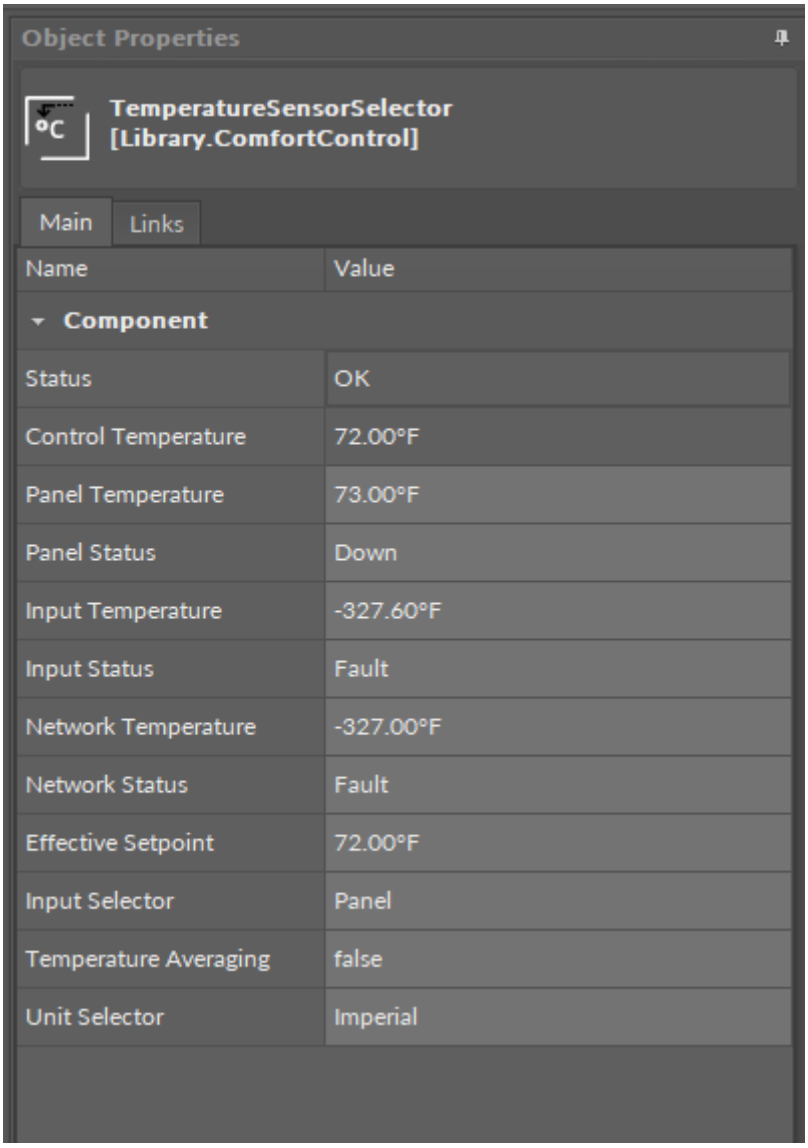


Figure 90. TemperatureSensorSelector slots

The TemperatureSensorSelector component has the following slots:

- **Status:** informs about the status of the component, if one of the Status-type slots in the component (Panel Status/Input Status/Network Status) has a null value, then the status is fault;
    - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;
- **Control Temperature:** calculates the control temperature based on selected and available temperature inputs (or using effective setpoint if not);
- **Panel Temperature:** receives the temperature read from the local room panel;
- **Panel Status:** shows the status of the local panel; if it is OK then the Panel Temperature is used, otherwise, it is ignored;
- **Input Temperature:** receives the temperature read from the universal input of the VAV controller;
- **Input Status:** shows the status of the universal input of the VAV controller; if it is OK then the Input Temperature is used, otherwise, it is ignored;
- **Network Temperature:** receives the temperature read from the network controller;

- **Network Status:** shows the status of the network controller; if it is OK then the Network Temperature is used, otherwise, it is ignored;
- **Effective Setpoint:** allows to link or set the effective setpoint value to be used if no temperature measurement is available;
- **Input Selector:** allows to select a preferred input signal of the temperature - panel, input, network;
- **Temperature Averaging:** allows to set the temperature as a calculation of an average of available temperature inputs;
- **Unit Selector:** allows to set the unit type.

More

More on temperature control as used in the VAV application: Temperature.

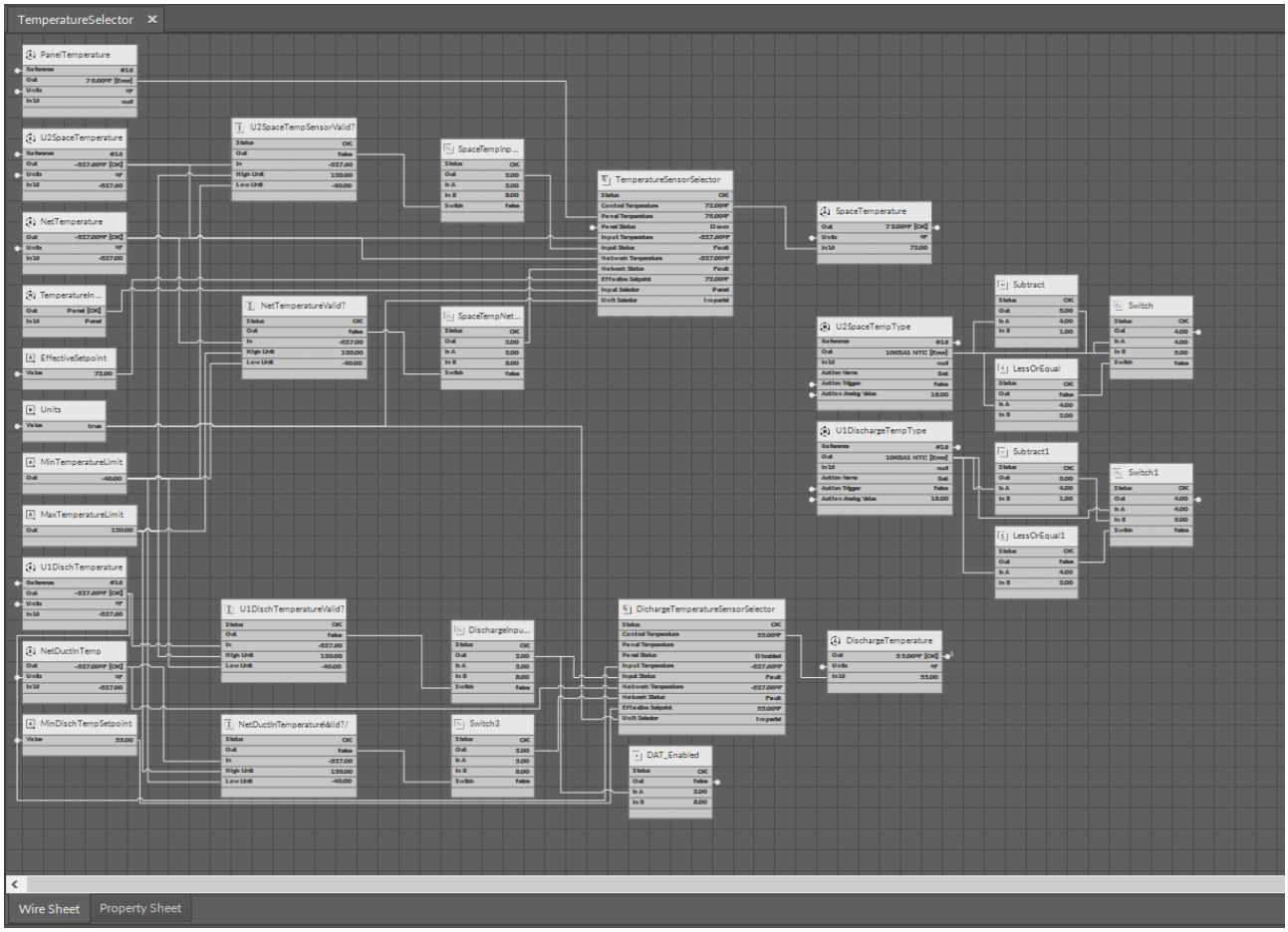


Figure 91. TemperatureSensorSelector in the application

TemperatureSetpointSelector

Applicable to OS V1.7

The TemperatureSetpointSelector component determines the effective setpoints for heating, cooling, and central (average value) based on the input occupancy state, the setpoints set for heating and cooling for each occupancy state, HVAC mode, and the set offset.

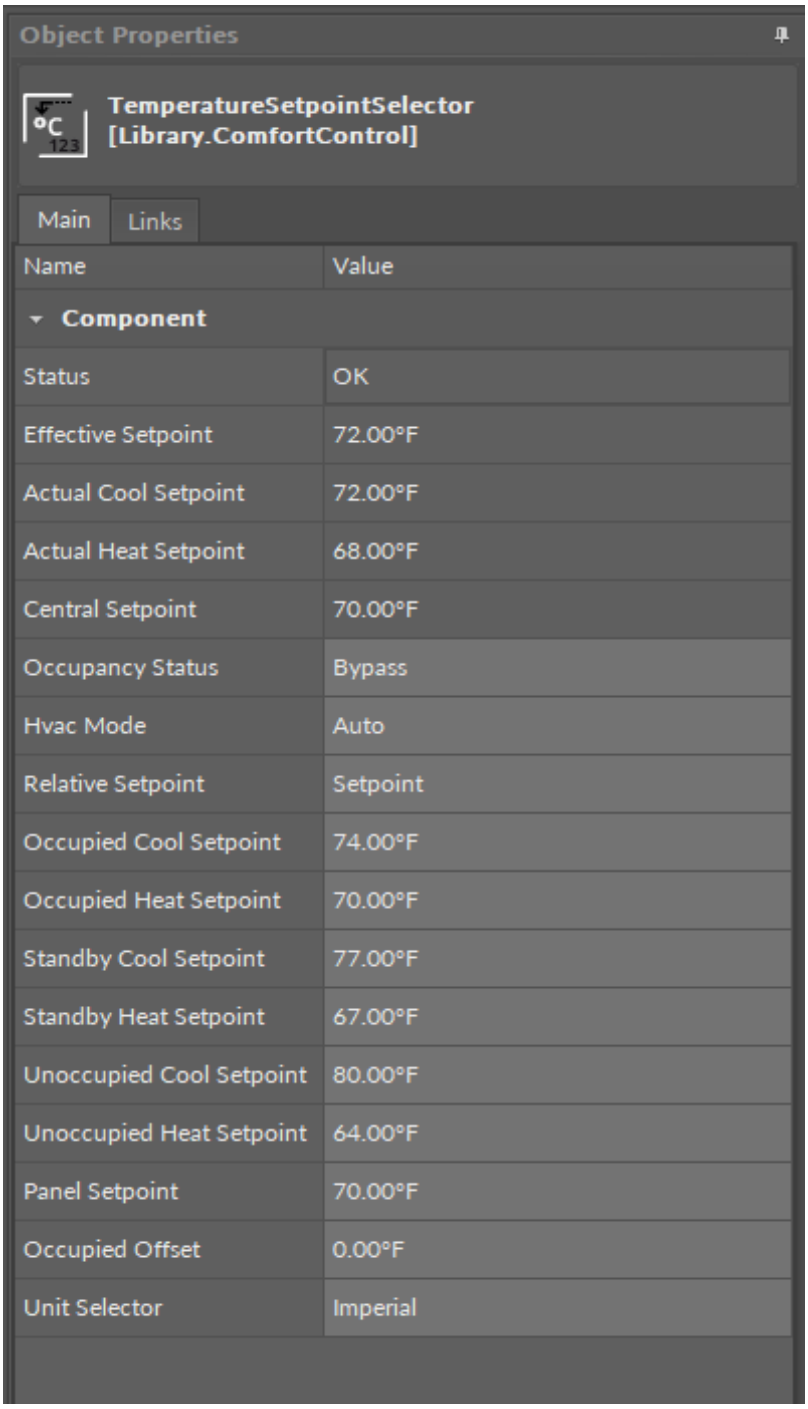


Figure 92. TemperatureSetpointSelector slots

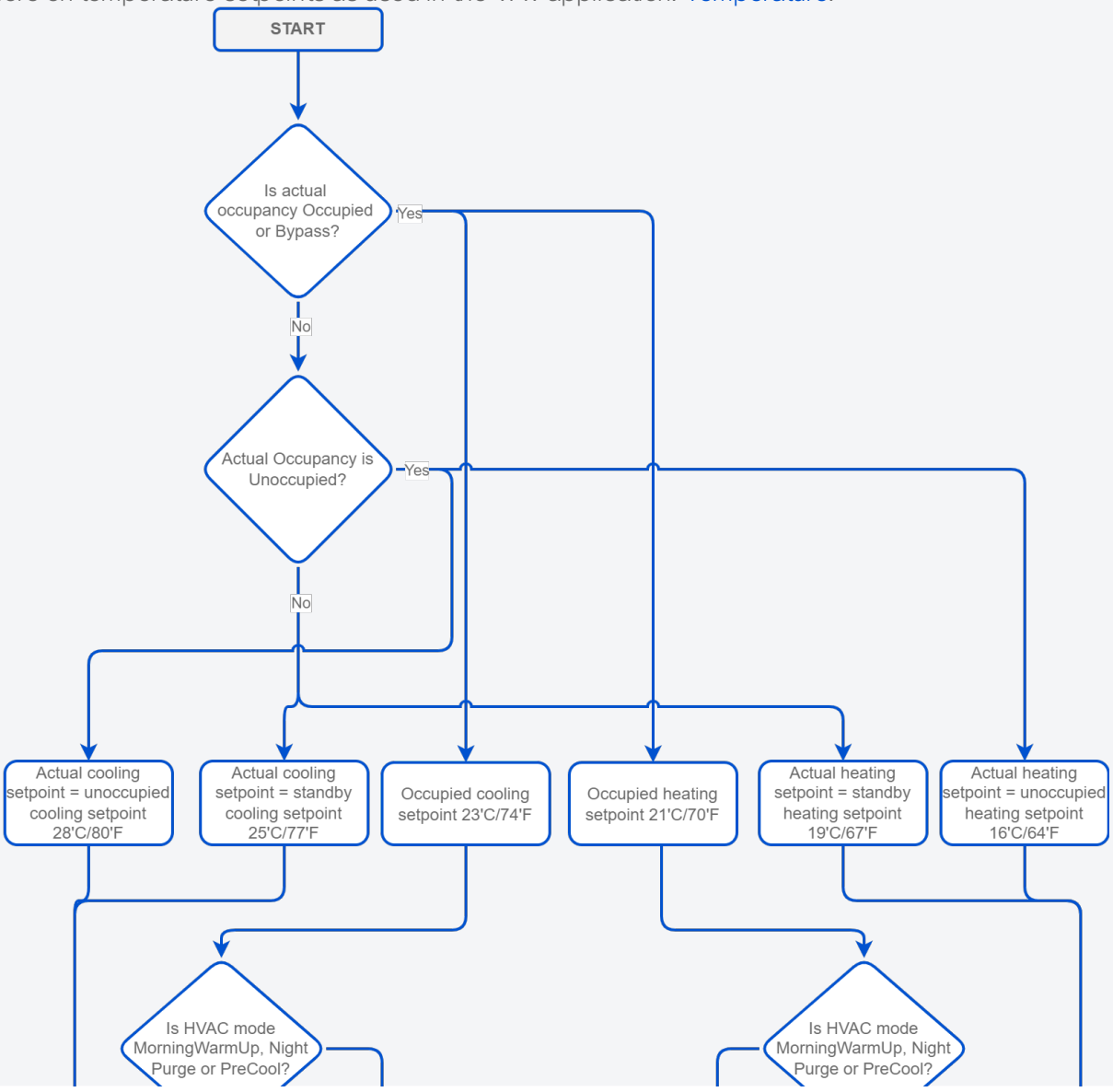
The TemperatureSetpointSelector component has the following slots:

- **Status:** informs about the status of the component, if one of the slots, Hvac Mode or Occupancy Status, has a null value, then the status is fault;
    - Available statues: OK, Fault, Disabled, Unlicensed, Down, Error;
- **Effective Setpoint:** shows the calculated effective setpoint value;

- **Actual Cool Setpoint:** shows the calculated setpoint value for cooling;
- **Actual Heat Setpoint:** shows the calculated setpoint value for heating;
- **Central Setpoint:** shows the average of both actual setpoints;
- **Occupancy Status:** receives the occupancy status resulting from the values of window contact, present motion, network occupancy, schedule occupancy, airflow detection, and panel button override (may be linked from the AdvancedOccupancyCalculator);
- **HVAC Mode:** receives the HVAC mode from the BMS or AHU system (from Modbus/BACnet network point);
- **Relative Setpoint:** allows to select a type of setpoint: false - relative (offset), true - real setpoint;
- **Occupied Cool Setpoint:** receives the setpoint for cooling in the occupied state, set from the BMS;
- **Occupied Heat Setpoint:** receives the setpoint for heating in the occupied state, set from the BMS;
- **Standby Cool Setpoint:** receives the setpoint for cooling in the standby state, set from the BMS;
- **Standby Heat Setpoint:** receives the setpoint for heating in the standby state, set from the BMS;
- **Unoccupied Cool Setpoint:** receives the setpoint for cooling in the unoccupied state, set from the BMS;
- **Unoccupied Heat Setpoint:** receives the setpoint for heating in the unoccupied state, set from the BMS;
- **Panel Setpoint:** receives the setpoint from the local room panel;
- **Occupied Offset:** receives the offset value from the local room panel;
- **Unit Selector:** allows to set the unit type.

More

More on temperature setpoints as used in the VAV application: Temperature.



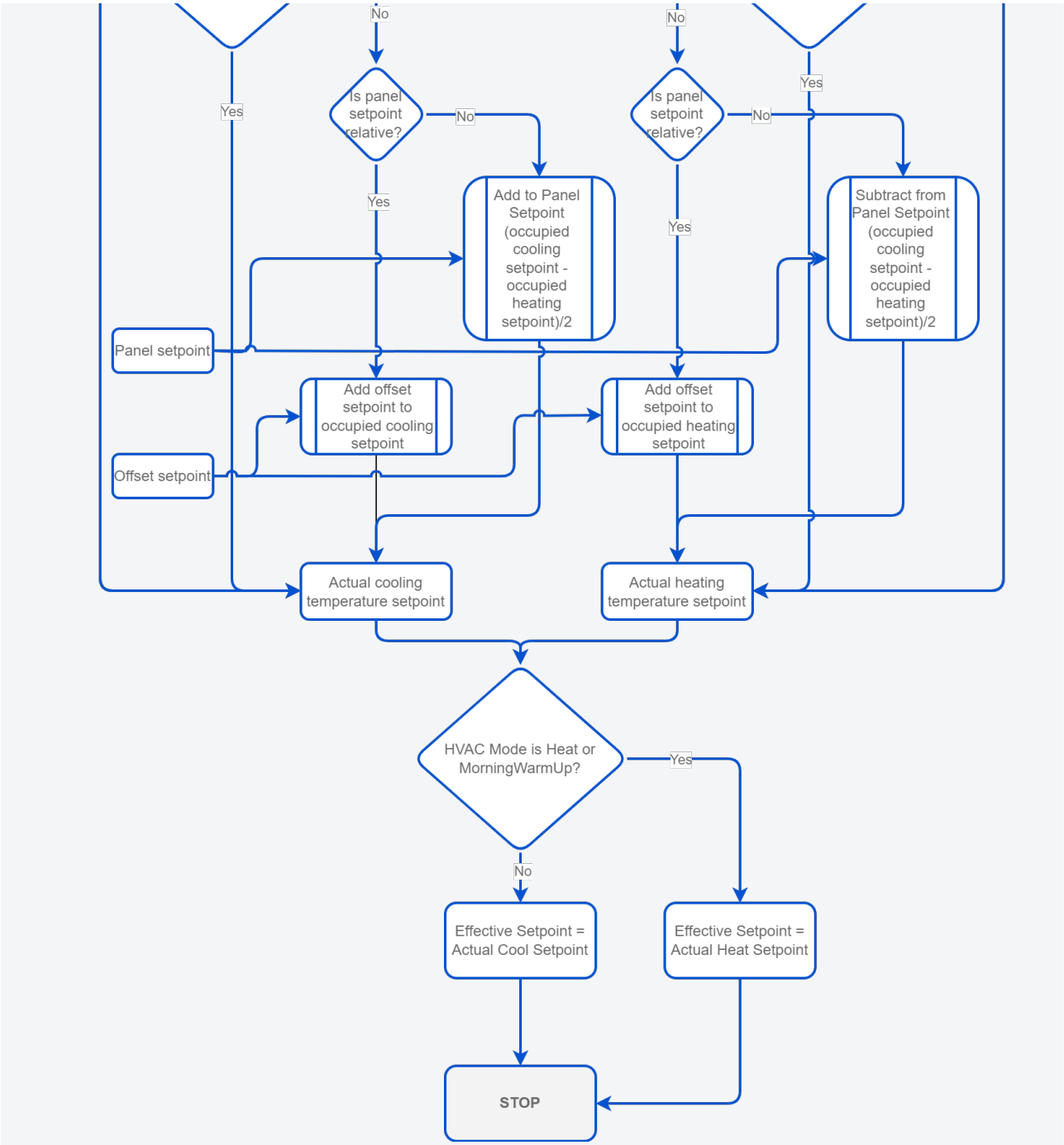
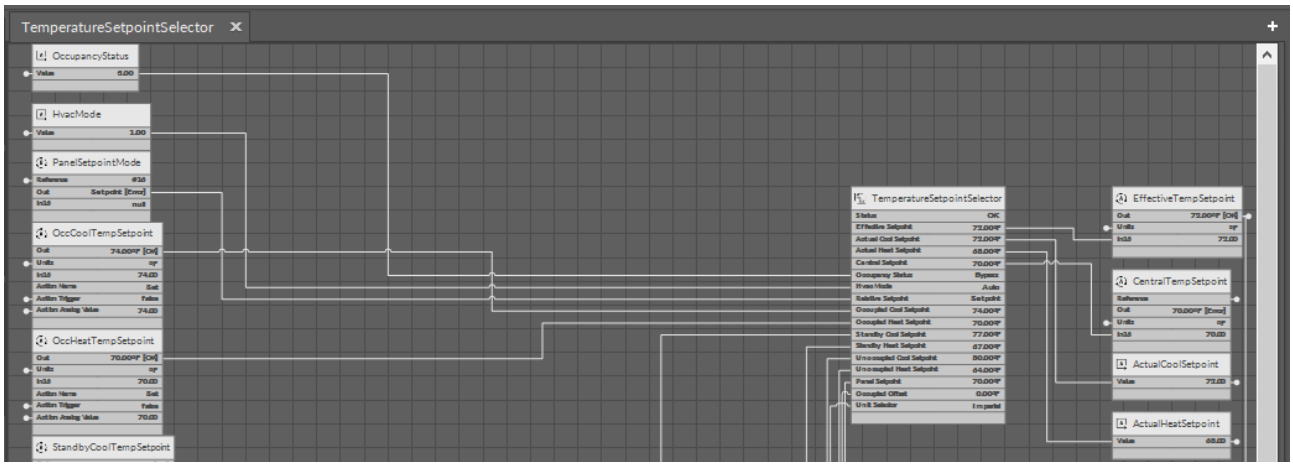


Figure 93. Temperature setpoint



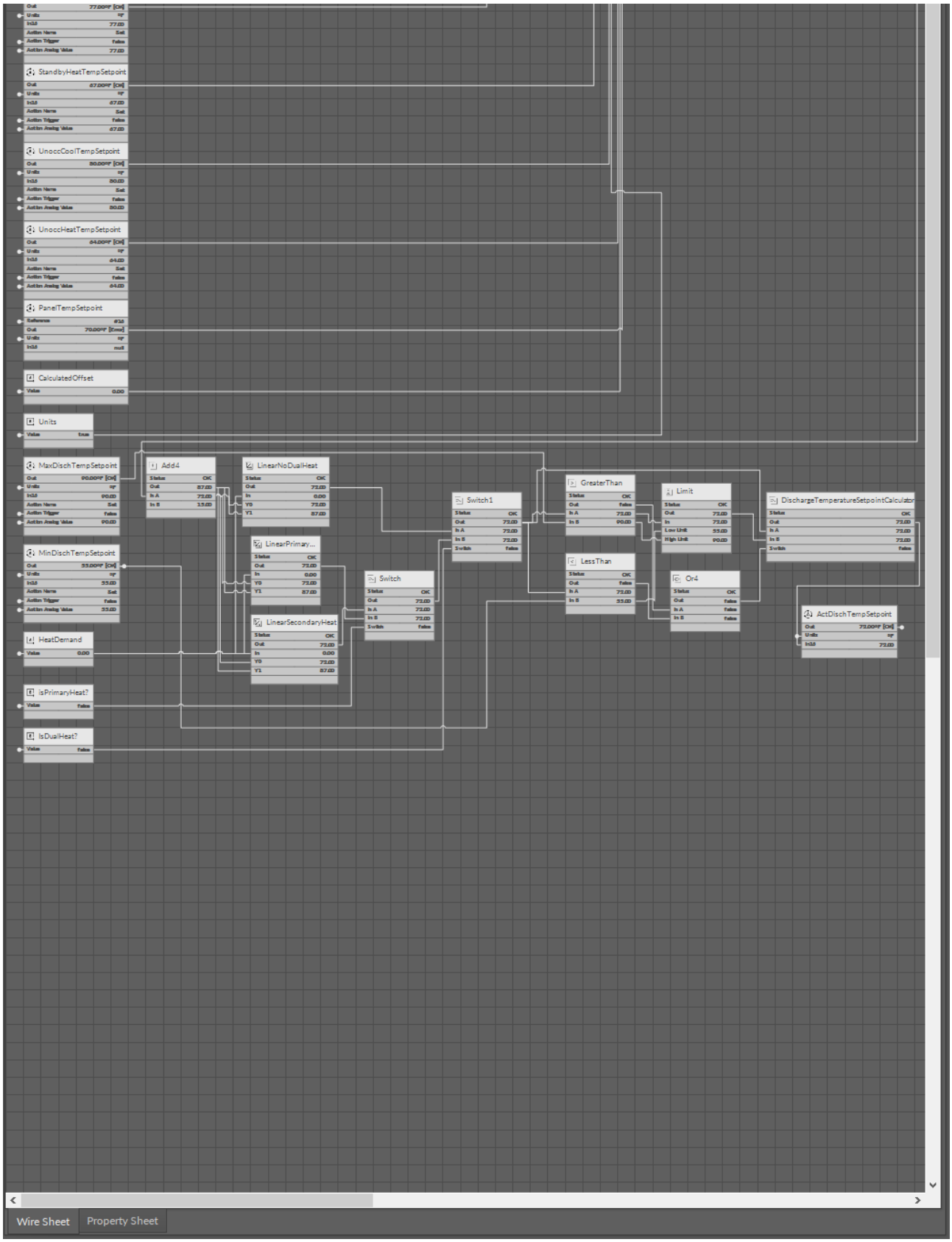


Figure 94. TemperatureSetpointSelector in the application

5.1.3 LocalIO Library

The IO library consists of nine network point components, dedicated to read and control the inputs and outputs of the actual device. The IO components are defined for various type of physical inputs and outputs.

In order to operate properly, the IO components must be placed in the Network container, under their superior component, the LocalIO.

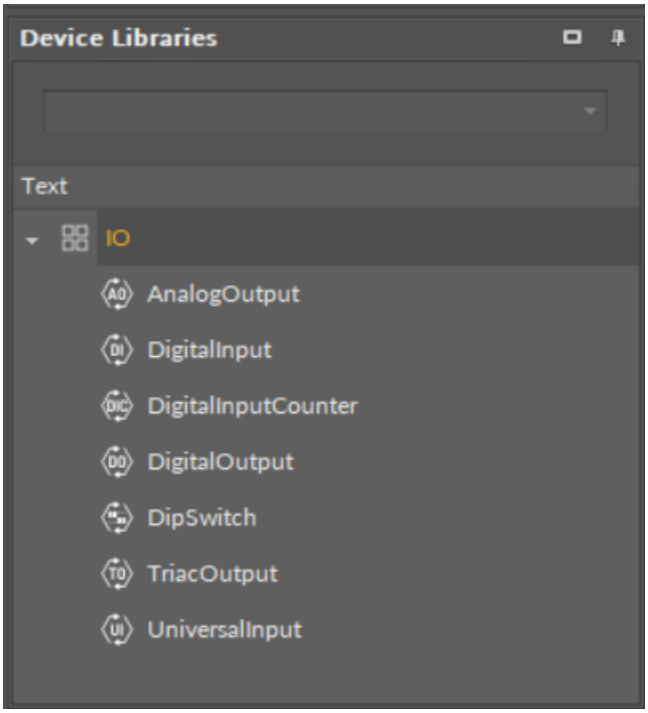


Figure 95. The IO library

Local IO Component

The Local IO component is a component that manages the communication with physical inputs and outputs of the device. It allows to configure communication for seven network point class components, each dedicated to service different type of input or output.

In the Property Sheet the Local IO component shows its status and its inferior components. It is possible to expand each of these components and control them from the Local IO's Property Sheet view.

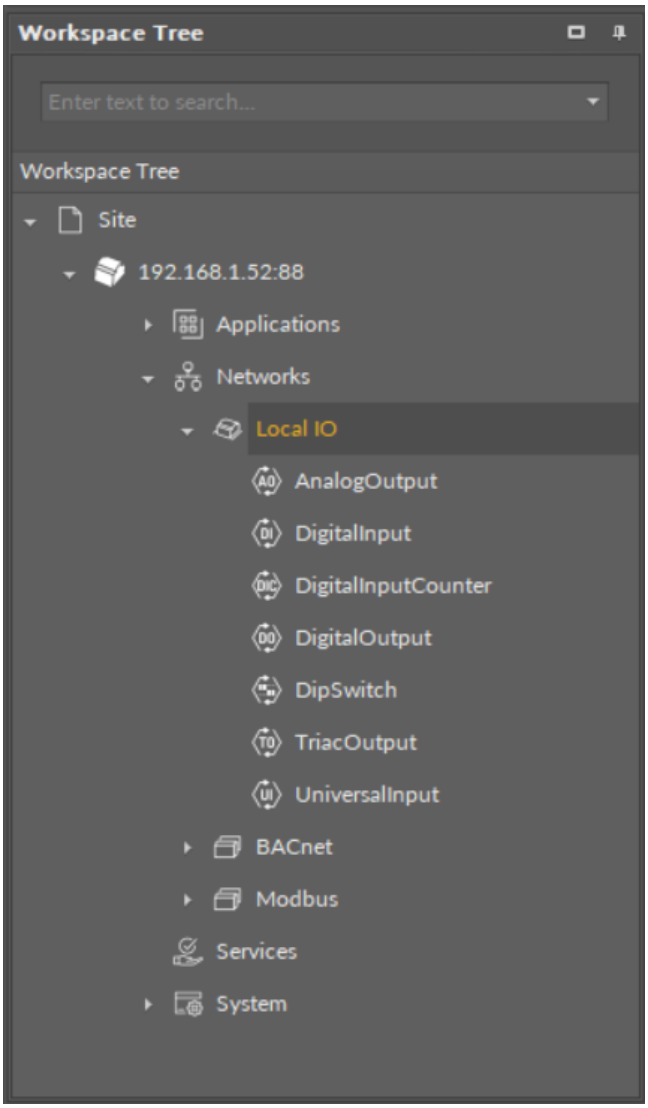


Figure 96. The LocalIO component

Slots

The Local IO component has the following slots:

- **Status:** indicates the current status of the component. If the component works, its status is OK; the component's status becomes Disabled if it has been stopped by a manual action.
  - Available information: Disabled, OK.
- **Info:** provides a detailed information about the Disabled status of the component;
  - Available information: Device disabled - the Enabled slot in the component is set to false;
- **Enabled:** allows to enable or disable the component and its inferior components (even if their Enabled slots are set to true):
  - Available settings: true (enabled), false (disabled);
- **Digital Inputs:** indicates the number of physical digital inputs in the device;
- **Digital Outputs:** indicates the number of physical digital outputs in the device;
- **Universal Inputs:** indicates the number of physical universal inputs in the device;
- **Analog Outputs:** indicates the number of physical analog outputs in the device;
- **Triac Outputs:** indicates the number of physical triac outputs in the device;
- **Fast Poll Frequency:** sets the time between requests for the point's value sent in the fast mode;
- **Normal Poll Frequency:** sets the time between requests for the point's value sent in the normal mode;
- **Normal Poll Frequency:** sets the time between requests for the point's value sent in the normal mode.

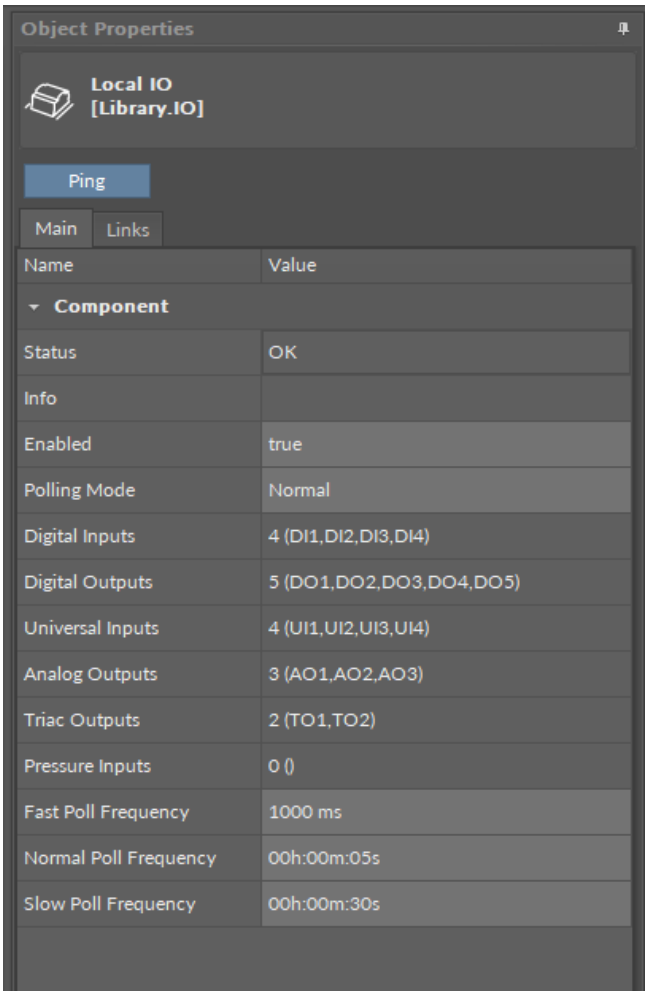


Figure 97. The LocalIO component's slots

UniversalInput

Applicable to OS V1.7

The UniversalInput component is an IO point (network point class) component that retrieves data from the physical universal input of a device. The component allows to configure the universal input according to its purpose—the universal input may serve as a voltage, current, digital, resistance, or temperature input—in order to communicate properly with the linked Data Point. The component allows to change the type of sensor, its resolution, and to apply an input signal filtering.

The component can pass data to the Data Point class component by linking Reference slots; the UniversalInput component may be linked to an Analog Data Point. In order to operate properly, the UniversalInput component must be located under the LocalIO component in the Networks container and have its unique number assigned in the Address slot.

Note: Before using the component, make sure that its individual address is assigned and its Status is OK.

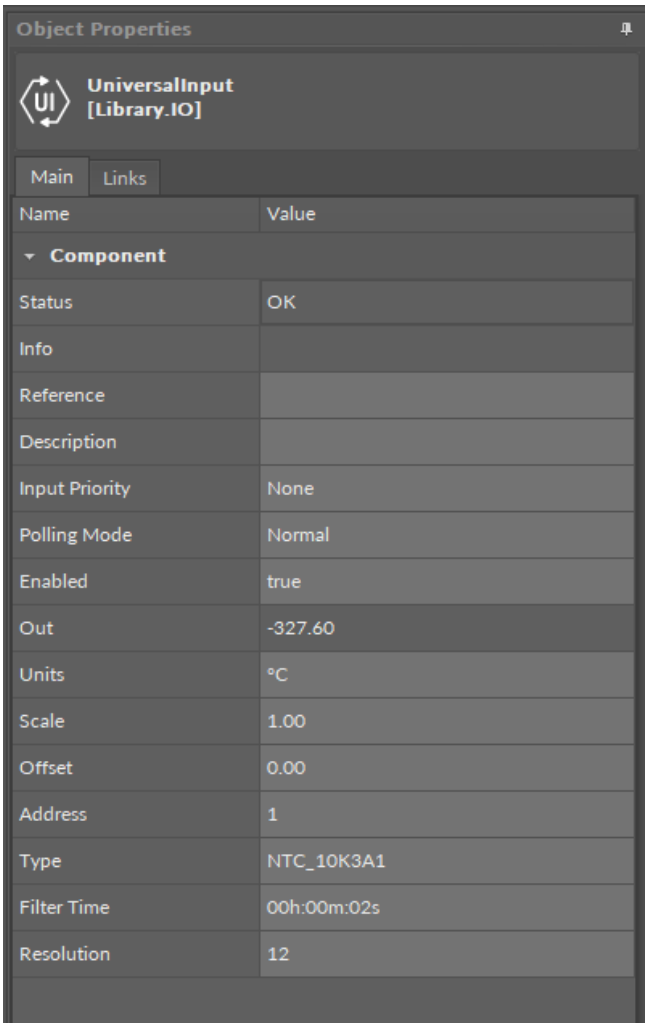


Figure 98. The UniversalInput component

The UniversalInput component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false. The component's status is Fault, once the Address slot is null, 0, or exceeding an available range;
  - Available information: Down, Disabled, Fault, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
  - Available information:

    - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);
    - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
    - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
    - Point disabled (status Disabled): the Enabled slot in the component is set to false;
- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

By default, once the Reference link is created from the network point to the Data Point it sets the input priority to 16, which later can be changed manually.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, or any other information the user finds applicable.
- **InputPriority:** allows to select the input number in the Data Point, which the value from the network point class component's output is sent to; by default, the priority is None and sets to 16 after linking with a Data Point (can be changed manually).
    - Available settings: none, 1-16.

Note: The Reference link from the network point to the Data Point cannot be changed to a 17th, default, priority.

- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
    - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to read values from the physical input; by default, the component is enabled.
    - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** a real value read from the physical input of the address set in the Address slot.

Note: If the component's Status is fault (e.g., an invalid value in the Address slot), the Out value is null.

Note: The Out slot value depends on the input type defined in the Type slot.

- **Units:** defines a unit of the Out slot value, depending on the sensor type set in the Type slot—the unit is automatically set once the sensor type is selected in the Type slot; however, it can be manually adjusted by the user;
    - Available units: according to BACnet units;
- **Scale:** sets a fixed scaling factor for output linearization; the Out value is calculated according to the linear function formula (y=ax+b), and the Scale slot set the a value of the formula;
- **Offset:** sets a fixed offset value to the output value; the Out value is calculated according to the linear function formula (y=ax+b), and the Offset slot set the b value of the formula;
- **Address:** allows setting an address of a physical input of the device; once the component has been added, the slot's default value is null—for the component to operate properly, the unique address value must be set in this slot.
    - Available settings: 1-n, where "n" stands for the number of actual inputs in the device.
- **Type:** defines a sensor type for proper reading of values.
    - Available settings: Voltage, Current, Digital, Resistance, Temperature (supported temperature sensors:

Celsius degrees: NTC 10K3A1, NTC 10K4A1, NTC 10K Carel, NTC 20K6A1, NTC 2,2K3A1, NTC 3K3A1, NTC 30K6A1, NTC SIE1, NTC TAC1, NTC SAT1, PT1000, NI1000, NI1000 21C, NI1000 LG,

Fahrenheit degrees: NTC 10K Type2, NTC 10K Type3, NTC 20K NTC, NTC 3K, PT1000, NI1000 32F, NI1000 70F).

Note: Choosing one input type disables the rest—the universal input works only as a type defined in the Type slot, and it will not adjust automatically if the incoming value changes its type. By default, the NTC 10K3A1 temperature sensor is set as an input type; if any other input type is relevant for the particular universal input, it needs to be adjusted manually.

- **Filter Time:** sets the filtering period in order to avoid the read values peaks.
  - Available settings: 0-60 s (by default, the filtering time is set to 2 s).
- **Resolution:** defines a precision of data conversion from analog data to bits for further data processing in the controller's software.
  - Available settings: 12 or 16-bit (12-bit resolution provides fast data processing, whereas 16-bit resolution provides more precise data; however, it may hamper the speed of data processing).

Note: As the default value of the Resolution slot is 12-bit, it needs to be emphasized that PT1000 and NI1000 temperature sensors work correctly only with the 16-bit resolution; therefore, if these types of sensors are set in the Type slot, the Resolution slot is automatically adjusted to a 16-bit value.

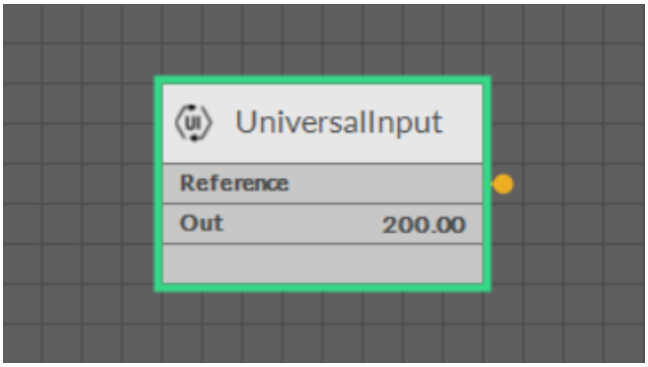


Figure 99. The UniversalInput component linked

DigitalInput

Applicable to OS V1.7

The DigitalInput component is an I/O point (network point class) component that retrieves data from a physical digital input of a device. The component allows to configure the digital input in order to communicate properly with the linked Data Point. The component can pass data to the Data Point class component by linking Reference slots; the DigitalInput component may be linked to a Binary Data Point. In order to operate properly, the DigitalInput component must be located under the LocalIO component in the Networks container and it have a unique number assigned in the Address slot.

Note: Before using the component, make sure that its individual address is assigned and its Status is OK.

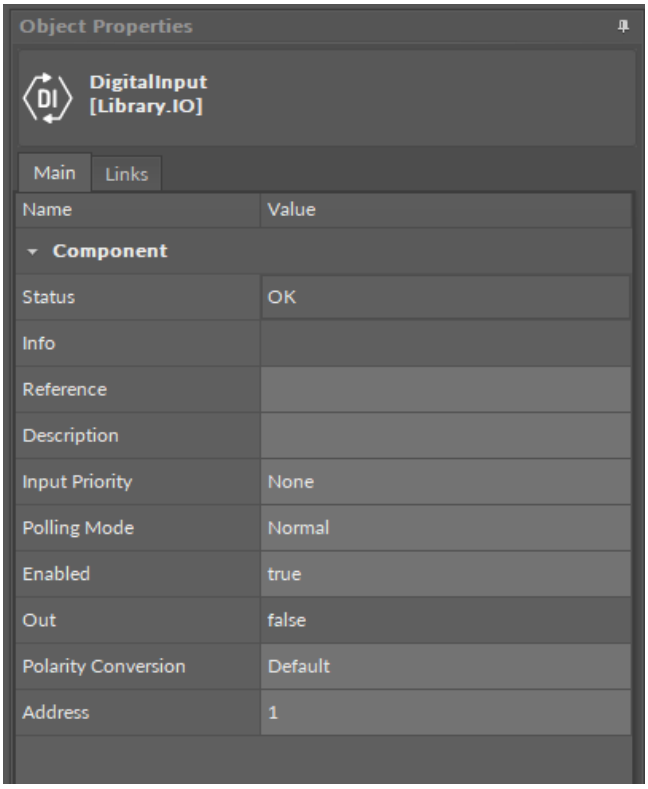


Figure 100. The DigitalInput component

The DigitalInput component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false. The component's status is Fault, once the Address slot is null, 0, or exceeding an available range;
  - Available information: Disabled, Fault, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
  - Available information:
    - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);
    - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
    - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
    - Point disabled (status Disabled): the Enabled slot in the component is set to false;
- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

By default, once the Reference link is created from the network point to the Data Point it sets the input priority to 16, which later can be changed manually.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, or any other information the user finds applicable.
- **InputPriority:** allows to select the input number in the Data Point, which the value from the network point class component's output is sent to; by default, the priority is None and sets to 16 after linking with a Data Point (can be changed manually).
    - Available settings: none, 1-16.

Note: The Reference link from the network point to the Data Point cannot be changed to a 17th, default, priority.

- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
    - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to read values from the physical input; by default, the component is enabled.
    - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** a real value read from the physical input of the address set in the Address slot.

Note: If the component's Status is fault (e.g., an invalid value in the Address slot), the Out value is null.

- **Address:** allows setting an address of a physical input of the device; once the component has been added, the slot's default value is null—for the component to operate properly, the unique address value must be set in this slot.
    - Available settings: 1-n, where "n" stands for the number of actual inputs in the device.

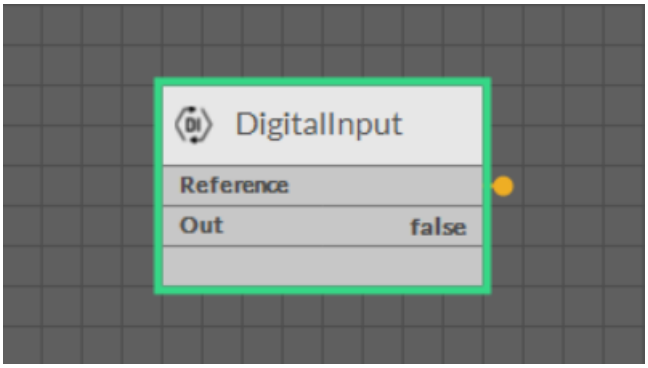


Figure 101. The DigitalInput component linked

DigitalInputCounter

Applicable to OS V1.7

The DigitalInputCounter component counts the pulses on rising edge in a physical digital input of the device. The counter may be employed, for example, to add up a number of pulses for used water meters or similar purposes. Before the DigitalInputCounter component is addressed to a physical input, the counter's default value is 0 and it changes once the slot starts reading values from a connected meter or sensor.

Note: If the component's Status is fault (e.g., an invalid value in the Address slot), the counter's value is null.

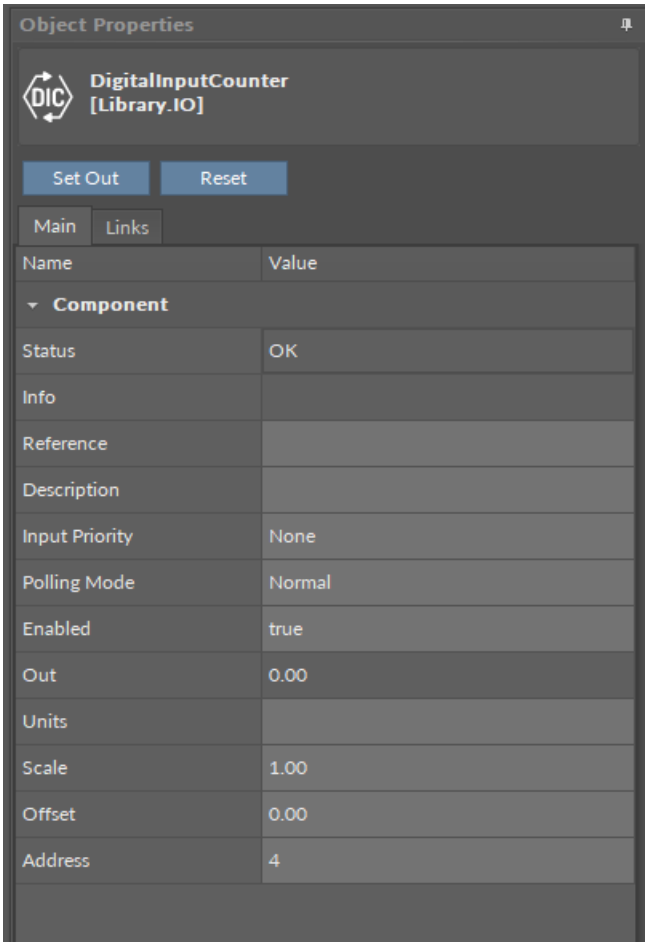


Figure 102. The DigitalInputCounter component

The DigitalInputCounter component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false. The component's status is Fault, once the Address slot is null, 0, or exceeding an available range;
    - Available information: Disabled, Fault, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
    - Available information:
        - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);
        - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
        - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
        - Point disabled (status Disabled): the Enabled slot in the component is set to false;

- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

By default, once the Reference link is created from the network point to the Data Point it sets the input priority to 16, which later can be changed manually.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, or any other information the user finds applicable.
- **InputPriority:** allows to select the input number in the Data Point, which the value from the network point class component's output is sent to; by default, the priority is None and sets to 16 after linking with a Data Point (can be changed manually).
    - Available settings: none, 1-16.

Note: The Reference link from the network point to the Data Point cannot be changed to a 17th, default, priority.

- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
    - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to read values from the physical input; by default, the component is enabled.
    - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** shows a number of rising edges from the physical digital input of the address set in the Address slot.

Note: If the component's Status is fault (e.g., an invalid value in the Address slot), the Out value is null.

- **Units:** defines a unit of the Out slot value; no unit is set by default;
- **Scale:** sets a fixed scaling factor for output linearization; the Out value is calculated according to the linear function formula (y=ax+b), and the Scale slot set the a value of the formula;
- **Offset:** sets a fixed offset value to the output value; the Out value is calculated according to the linear function formula (y=ax+b), and the Offset slot set the b value of the formula;
- **Address:** allows setting an address of a physical input of the device; once the component has been added, the slot's default value is null—for the component to operate properly, the unique address value must be set in this slot.
    - Available settings: 1-n, where "n" stands for the number of actual inputs in the device.

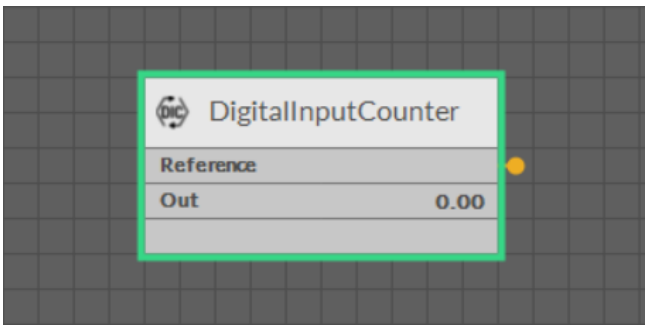


Figure 103. The DigitalInputCounter component linked

The DigitalInputCounter has the following actions:

- **SetOut:** sets the Out slot to a specific value;
- **Reset:** sets the Out slot to the 0 value.

ActionTrigger Extension

The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- **Action Name:** allows to select an action to invoke;
- **Action Trigger:** triggers an action selected in the Action Name slot;
- **Action Analog Value/Action Binary Value/Action String Value:** a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).

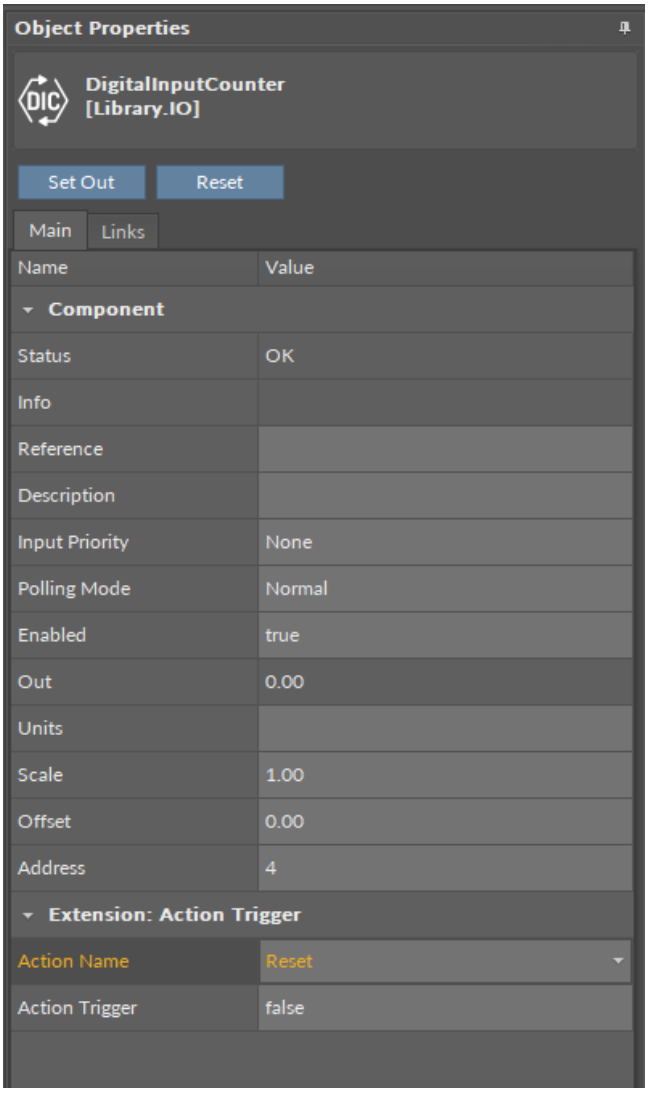


Figure 104. The ActionTrigger extension in the DigitalInputCounter component

PressureInput

Applicable to OS V1.7

The PressureInput component is an I/O point (network point class) component, which allows to read data from a built-in pressure sensor operating in range of -500 to 500 Pa.

In order to operate properly, the PressureInput component must be located under the Local IO component in the Networks container.

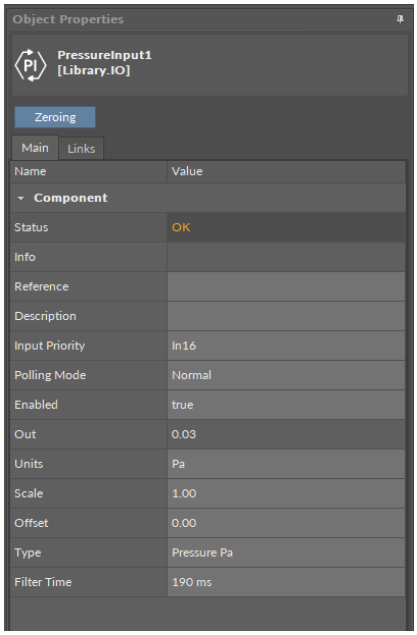


Figure 105. The PressureInput component slots

The PressureInput component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false;
  - Available information: Disabled, Fault, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
  - Available information:
    - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);
    - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
    - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
    - Point disabled (status Disabled): the Enabled slot in the component is set to false;
- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

By default, once the Reference link is created from the network point to the Data Point it sets the input priority to 16, which later can be changed manually.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, or any other information the user finds applicable.
- **InputPriority:** allows to select the input number in the Data Point, which the value from the network point class component's output is sent to; by default, the priority is None and sets to 16 after linking with a Data Point (can be changed manually).
  - Available settings: none, 1-16.

Note: The Reference link from the network point to the Data Point cannot be changed to a 17th, default, priority.

- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
  - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to read values from the physical input; by default, the component is enabled.
  - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** a real value read from the differential pressure sensor;

Note: If the component's Status is fault, the Out value is null.

Note: The Out slot value depends on the input type defined in the Type slot.

- **Units:** defines a unit of the Out slot value, depending on the sensor type set in the Type slot—the unit is automatically set once the sensor type is selected in the Type slot; however, it can be manually adjusted by the user;
  - Available units: according to BACnet units;
- **Scale:** sets a fixed scaling factor for output linearization; the Out value is calculated according to the linear function formula (y=ax+b), and the Scale slot set the a value of the formula;
- **Offset:** sets a fixed offset value to the output value; the Out value is calculated according to the linear function formula (y=ax+b), and the Offset slot set the b value of the formula;
- **Type:** defines a sensor type for proper reading of values;
  - Available settings: pressure Pa, pressure inH2O;

Note: Choosing one input type disables the other—the component works only as a type defined in the Type slot, and it will not adjust automatically if the incoming value changes its type.

- **Filter Time:** sets the filtering period in order to avoid the read values peaks.
  - Available settings: 0-60 ms (by default, the filtering time is set to 20 ms).

The PressureInput component has one available action:

- **Zeroing:** sets the current Out value to zero. If the Offset slot is set to any value other than 0, the zeroing action will eliminate any pressure difference resulting from the measurement error, but the Offset value will not be erased. In effect, for example, if the Out slot is 12 and the Offset slot is set to 10, after executing the zeroing action, the output will be 10.

Pressure Sensor Zeroing

Pressure sensor zeroing is a part of the VAV balancing process, which aims at eliminating a constant measurement error of differential pressure on a built-in pressure sensor. The zeroing process involves the following steps:

- Make sure the differential pressure sensor is disconnected from the measuring cross or other measuring method.
- Use a flexible hose of an appropriate diameter to connect the two spigots (+ and -).
- Make sure the hose is well secured and tight to equalize the pressures on both ports.
- Invoke the zeroing action using one of the methods:
  - using the action in the PressureInput component (iC Tool)
  - in the Balancing tab available in one of the tools (iSMA Configurator, iC Device Manager),
  - writing a value to the PressureZeroing variable (BACnet object: BV13, Modbus address: 13),
  - from the Control Point VAV panel.
- Detach the hose.

- Restore the normal connection of the sensor to the measuring cross or other target circuit, pay attention to the polarity.

ActionTrigger Extenstion

The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- **Action Name:** allows to select an action to invoke;
- **Action Trigger:** triggers an action selected in the Action Name slot;
- **Action Analog Value/Action Binary Value/Action String Value:** a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).

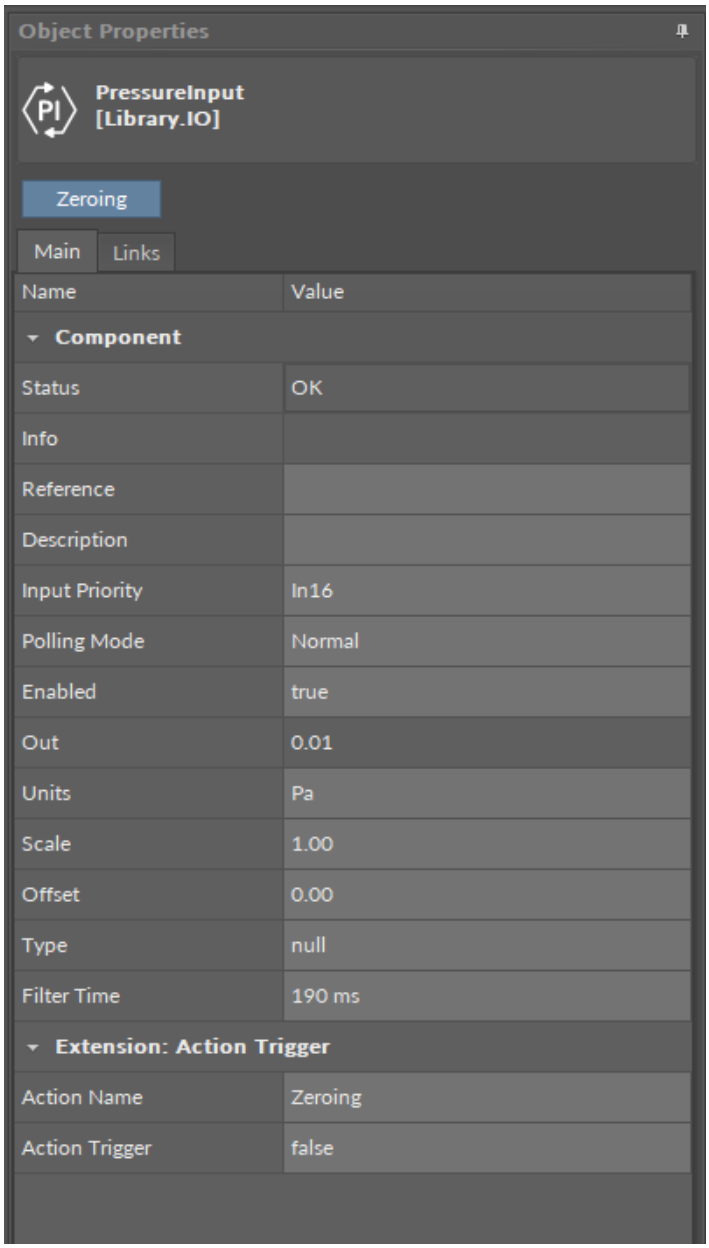


Figure 106. ActionTrigger extension in the PressureInput component

AnalogOutput

Applicable to OS V1.7

The AnalogOutput component is an IO point (network point class) component that transfers the value to a physical analog output of a device. The component allows to transfer data received from the linked Data Point in order to control the physical analog output of the device, and allows to set the polling mode of the point. The component can pass data from the Data Point class component by linking Reference slots; the AnalogOutput component may be linked from an Analog Data Point.

In order to operate properly, the AnalogOutput component must be located under the Local IO component in the Networks container.

Note: Before using the component, make sure that its individual address is assigned and its Status is OK.

The AnalogOutput component includes an action that allows to set its Out value in case no Data Point class component is linked to it.

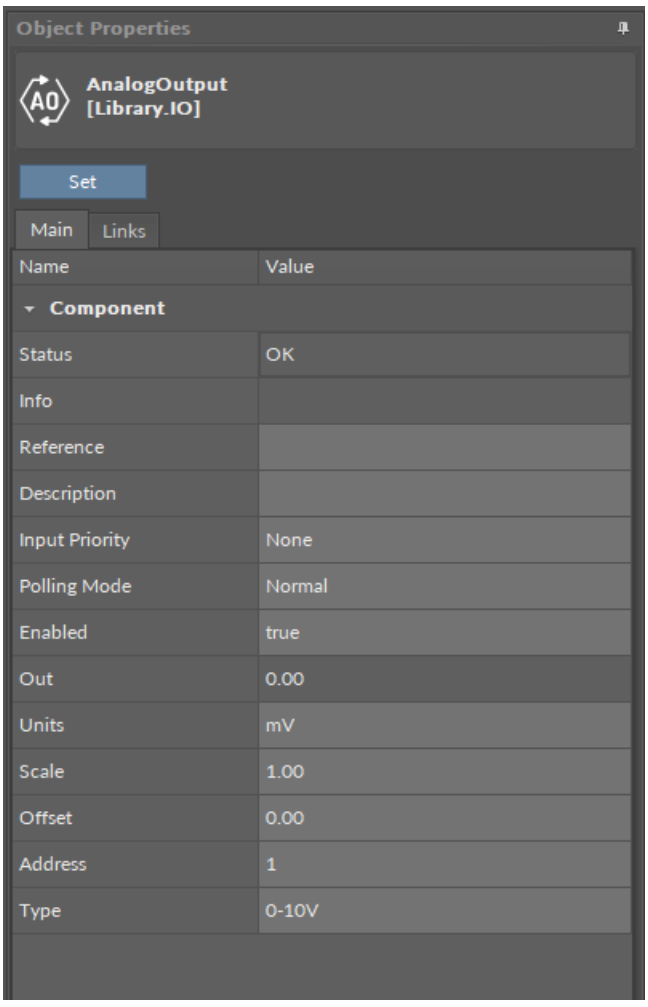


Figure 107. The AnalogOutput component

The Analog Output component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false. The component's status is Fault, once the Address slot is null, 0, or exceeding an available range;
  - Available information: Disabled, Fault, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
  - Available information:

    - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);
    - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
    - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
    - Point disabled (status Disabled): the Enabled slot in the component is set to false;
- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

Note: Reference links from Data Points to network points also transfer values in the opposite direction, in a link-back-from process: having received a value by the Reference link, the network point transfers it back to the Data Point to whichever input priority from 1 to 16 is set in the network point.

Note: The Reference links work on a change-of-value. If a value in the Data Point's Out slot changes, it is immediately transferred to a network point linked by the Reference link–such change is not dependent on an application cycle.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, meter's or sensor's location, or any other information the user finds applicable.
- **InputPriority:** allows to indicate the input number in the Data Point, which the network point class component's output value is sent to, in case the network point detects the change on its Out slot; by default, the priority is none.
    - Available settings: none, 1-16.
- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
    - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to transfer values to the physical output; by default, the component is enabled.
    - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** displays a value transferred to the output of the address set in the Address slot.

Note: If the component's Status is fault (e.g., an invalid value in the Address slot), the Out value is null.

- **Units:** defines a unit of the Out slot value, depending on the selected mode of operation—the unit is automatically set once the mode type is selected in the Type slot, however, it can be manually adjusted by the user;
    - Available units: according to BACnet units;
- **Scale:** sets a fixed scaling factor for output linearization; the value transferred to a physical output is calculated according to the inversed linear function y=(x-b)/a, and recalculated according to the linear function y=ax+b to the Out slot; the Scale slot sets the a value of the formula;
- **Offset:** sets a fixed offset value to the output value; the value transferred to a physical output is calculated according to the inversed linear function y=(x-b)/a, and recalculated according to the linear function y=ax+b to the Out slot; the Offset slot sets the b value of the formula;

Example

The output linearization allows to adjust an output value to the requirements of a controlled device/equipment. For example:

- if an analog output controls an actuator, which operates within a 2-10 V range, the AnalogOutput component can be scaled to produce output fitting the required range:
    - the Type slot set to 0-10 V;
    - the Scale slot value set to **1.25** (10000 mV/8000 mV);
    - the Offset slot value scaled to **-2500** (redefining the output's range to **2-10 V**: 2000 mV/0.8 range).

Note: Though the Type slot is set to 0-10 V, the actual value is controlled in mV. Accordingly, the offset must be set in mV.

- to scale the output to control a device, which operates within a 0-5 V range, the AnalogOutput parameters have to be set as follows:
  - the Type slot set to 0-10 V;
  - the Scale slot value set to 2;
  - the Offset slot value scaled to 0.
- to translate 0-100% to 0-10000 mV, the AnalogOutput component can be scaled to produce output fitting the required range:
  - the Scale slot value set to 0.01 (100%/10000 mV);
  - the Offset slot value scaled to 0.

Note: Incorrect scaling of the results in the calculated values (exceeding the available range for the device, e.g., 0-10 V), results in indicating a different value on the input from Reference link and on the Out slot. If the scaling is correct, the values are identical.

- **Address:** allows setting an address of a physical output of the device; once the component has been added, the slot's default value is null—for the component to operate properly, the unique address value must be set in this slot.
  - Available settings: 1-n, where "n" stands for the number of actual outputs in the device.
- **Type:** defines the mode of operation—the component may operate as a voltage or digital output, or in pulse width modulation mode; once the component has been added, the component's mode is set to Voltage;
  - Available modes: Voltage, Digital, PWM001Hz, PWM01Hz, PWM1Hz, PWM10Hz, PWM100Hz.

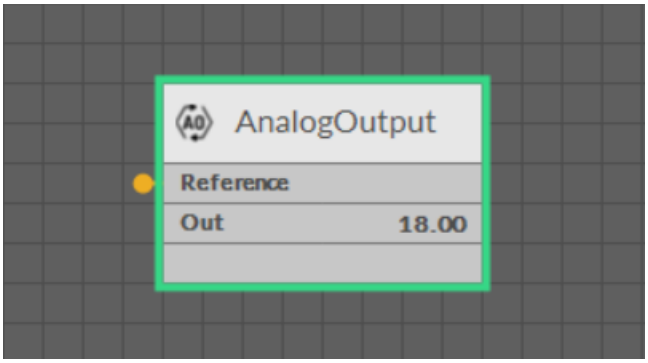


Figure 108. The AnalogOutput component linked

The AnalogOutput component has the following action:

- **Set:** sets a value to the Out slot—in case no Data Point is linked to the output network point, it is possible to set its Out value with this action.

ActionTrigger Extension

The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- **Action Name:** allows to select an action to invoke;
- **Action Trigger:** triggers an action selected in the Action Name slot;

- **Action Analog Value/Action Binary Value/Action String Value:** a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).

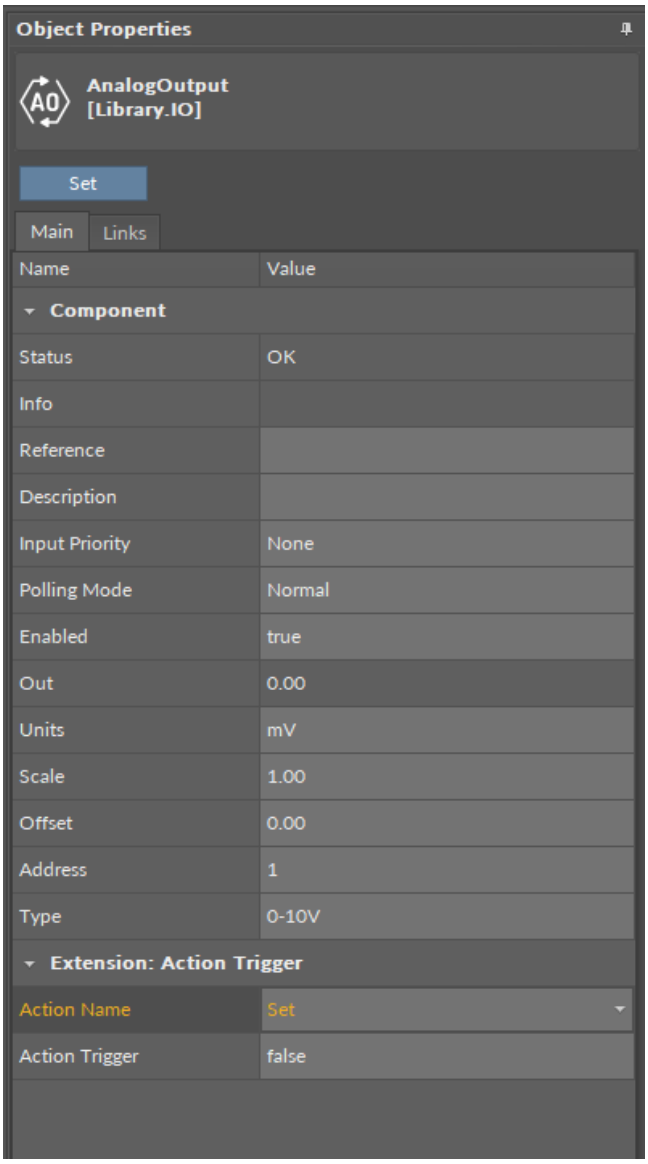


Figure 109. The ActionTrigger extension in the AnalogOutput component

DigitalOutput

Applicable to OS V1.7

The DigitalOutput component is an IO point (network point class) component that transfers the value to a physical digital output of a device. The component allows to transfer data received from the linked Data Point in order to control the physical digital output of the device. The component can pass data from the Data Point class component by linking Reference slots; the DigitalOutput component may be linked from a Binary Data Point. In order to operate properly, the DigitalOutput component must be located under the Local IO component in the Networks container.

Note: Before using the component, make sure that its individual address is assigned and its Status is OK.

The DigitalOutput component includes an action that allows to set its Out value in case no Data Point class component is linked to it.

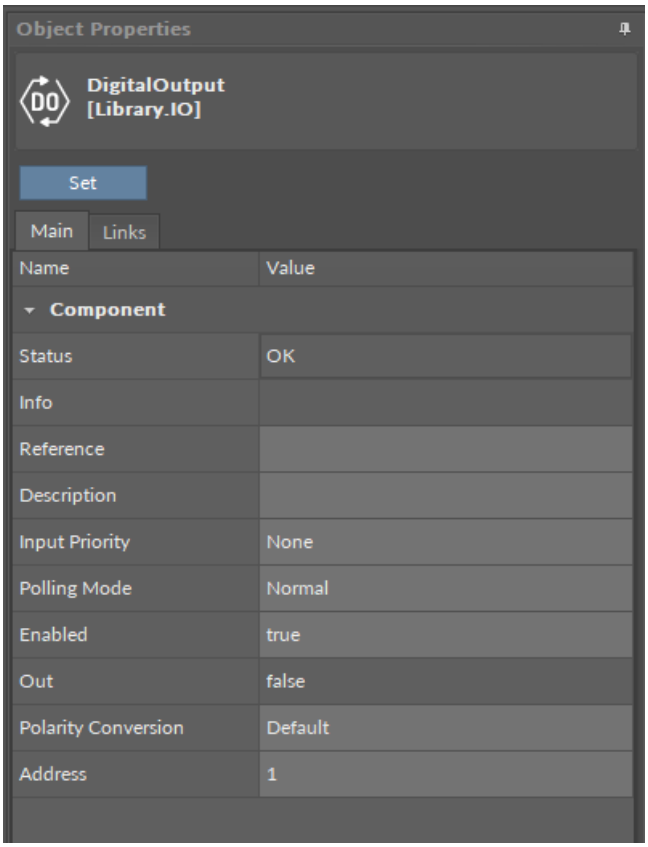


Figure 110. The DigitalOutput component

The DigitalOutput component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false. The component's status is Fault, once the Address slot is null, 0, or exceeding an available range;
    - Available information: Disabled, Fault, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
    - Available information:
        - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);
        - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
        - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
        - Point disabled (status Disabled): the Enabled slot in the component is set to false;

- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

Note: Reference links from Data Points to network points also transfer values in the opposite direction, in a link-back-from process: having received a value by the Reference link, the network point transfers it back to the Data Point to whichever input priority from 1 to 16 is set in the network point.

Note: The Reference links work on a change-of-value. If a value in the Data Point's Out slot changes, it is immediately transferred to a network point linked by the Reference link–such change is not dependent on an application cycle.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, meter's or sensor's location, or any other information the user finds applicable.
- **InputPriority:** allows to indicate the input number in the Data Point, which the network point class component's output value is sent to, in case the network point detects the change on its Out slot; by default, the priority is set to none.
  - Available settings: none, 1-16.
- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
  - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to transfer values to the physical output; by default, the component is enabled.
  - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** displays a value transferred to the output of the address set in the Address slot.

Note: If the component's Status is fault (e.g., an invalid value in the Address slot), the Out value is null.

- **Address:** allows setting an address of a physical output of the device; once the component has been added, the slot's default value is null—for the component to operate properly, the unique address value must be set in this slot.
  - Available settings: 1-n, where "n" stands for the number of actual outputs in the device.

The DigitalOutput component has the following action:

- **Set:** sets a value to the Out slot—in case no Data Point is linked to the output network point, it is possible to set its Out value with this action.

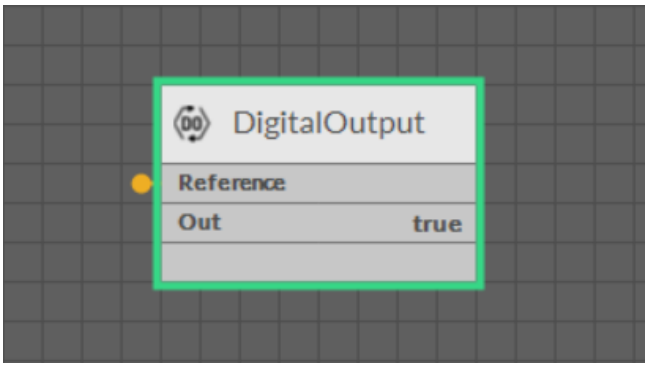


Figure 111. The DigitalOutput component linked

ActionTrigger Extension

The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- **Action Name:** allows to select an action to invoke;
- **Action Trigger:** triggers an action selected in the Action Name slot;
- **Action Analog Value/Action Binary Value/Action String Value:** a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).

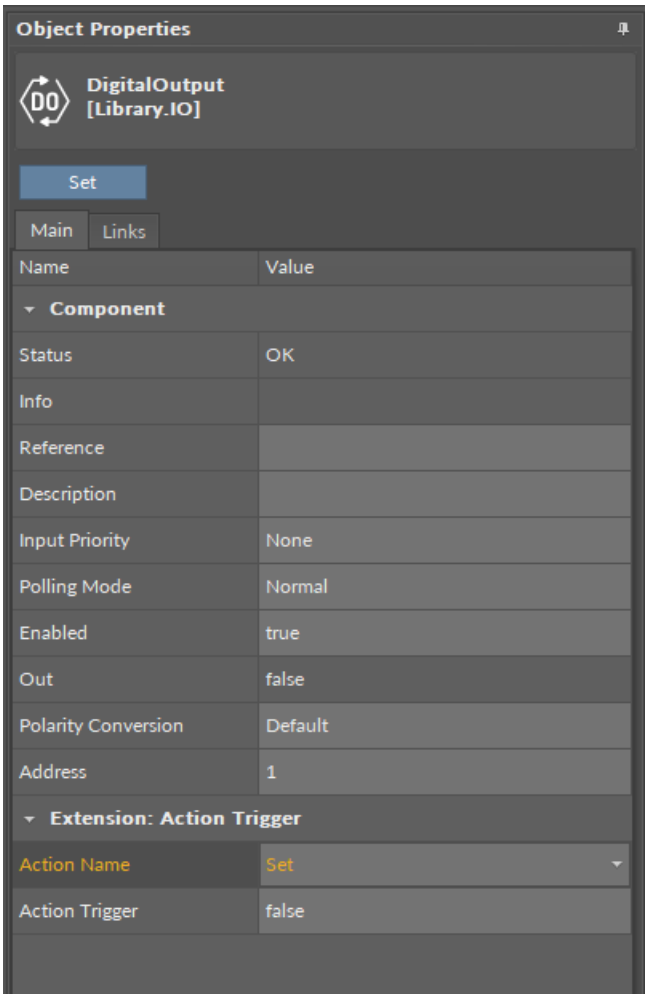


Figure 112. The ActionTrigger extension in the DigitalOutput component

TriacOutput

Applicable to OS V1.7

The TriacOutput component is an I/O point (network point class) component that transfers the value to a physical triac output of a device. The component allows to transfer data received from the linked Data Point in order to control the physical triac output of the device. The component can pass data from the Data Point class component by linking Reference slots; the TriacOutput component may be linked from an Analog Data Point. In order to operate properly, the TriacOutput component must be located under the Local IO component in the Networks container.

Note: Before using the component, make sure that its individual address is assigned and its Status is OK.

The TriacOutput component includes an action that allows to set its Out value in case no Data Point class component is linked to it.

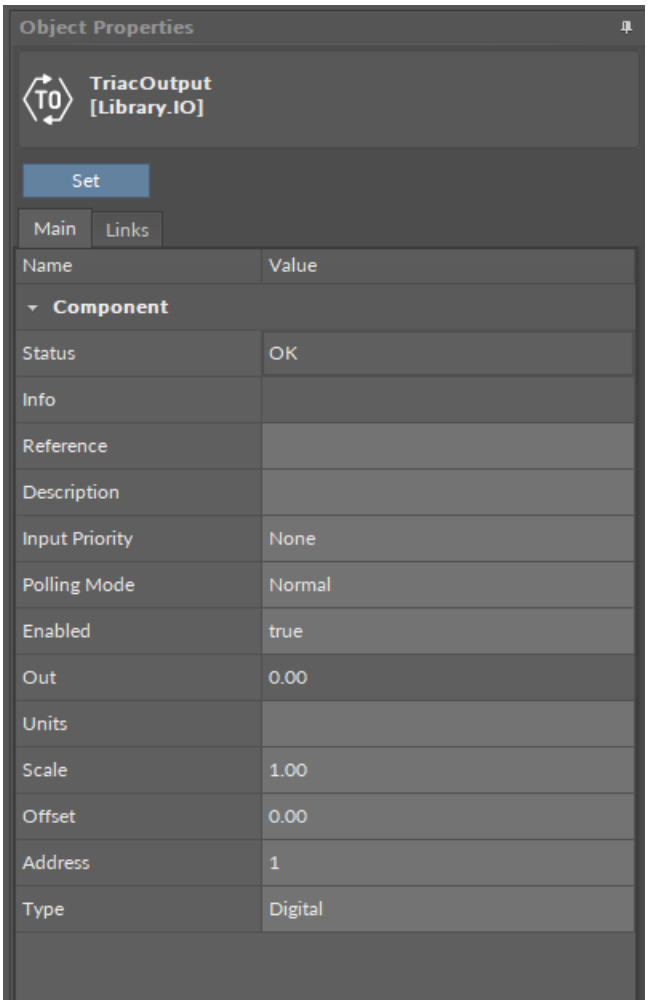


Figure 113. The TriacOutput component

The TriacOutput component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false. The component's status is Fault, once the Address slot is null, 0, or exceeding an available range;
  - Available information: Disabled, Fault, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
  - Available information:
    - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);

    - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
    - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
    - Point disabled (status Disabled): the Enabled slot in the component is set to false;
- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

Note: Reference links from Data Points to network points also transfer values in the opposite direction, in a link-back-from process: having received a value by the Reference link, the network point transfers it back to the Data Point to whichever input priority from 1 to 16 is set in the network point.

Note: The Reference links work on a change-of-value. If a value in the Data Point's Out slot changes, it is immediately transferred to a network point linked by the Reference link–such change is not dependent on an application cycle.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, meter's or sensor's location, or any other information the user finds applicable.
- **InputPriority:** allows to indicate the input number in the Data Point, which the network point class component's output value is sent to, in case the network point detects the change on its Out slot; by default, the priority is set to none.
    - Available settings: none, 1-16.
- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
    - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to transfer values to the physical output; by default, the component is enabled.
    - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** displays a value transferred to the output of the address set in the Address slot.

Note: If the component's Status is fault (e.g., an invalid value in the Address slot), the Out value is null.

- **Units:** defines a unit of the Out slot value, depending on the selected mode of operation—the unit is automatically set once the mode type is selected in the Type slot, however, it can be manually adjusted by the user;
    - Available units: according to BACnet units;
- **Scale:** sets a fixed scaling factor for output linearization; the value transferred to a physical output is calculated according to the inversed linear function y=(x-b)/a, and recalculated according to the linear function y=ax+b to the Out slot; the Scale slot sets the a value of the formula;
- **Offset:** sets a fixed offset value to the output value; the value transferred to a physical output is calculated according to the inversed linear function y=(x-b)/a, and recalculated according to the linear function y=ax+b to the Out slot; the Offset slot sets the b value of the formula;

Note: Incorrect scaling of the results in the calculated values (exceeding the available range for the device, e.g., 0-10 V), results in indicating a different value on the input from Reference link and on the Out slot. If the scaling is correct, the values are identical.

- **Address:** allows setting an address of a physical output of the device; once the component has been added, the slot's default value is null—for the component to operate properly, the unique address value must be set in this slot.
    - Available settings: 1-n, where "n" stands for the number of actual outputs in the device.
- **Type:** defines the mode of operation—the component may operate as a digital output, or in pulse width modulation mode; once the component has been added, the component's mode is set to Digital;
    - Available modes: Digital, PWM001Hz, PWM01Hz, PWM1Hz, PWM10Hz.

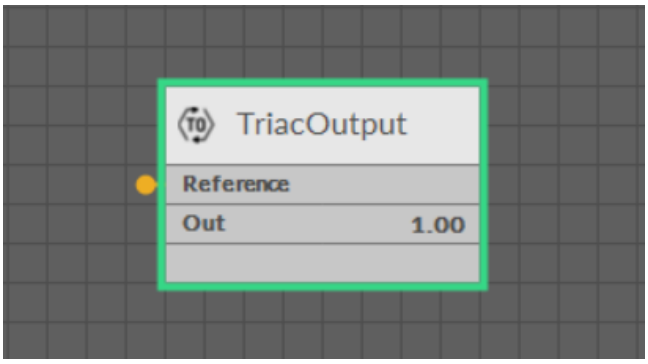


Figure 114. The TriacOutput component linked

The TriacOutput has the following action:

- **Set:** sets a value to the Out slot—in case no Data Point is linked to the output network point, it is possible to set its Out value with this action.

ActionTrigger Extension

The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- **Action Name:** allows to select an action to invoke;
- **Action Trigger:** triggers an action selected in the Action Name slot;
- **Action Analog Value/Action Binary Value/Action String Value:** a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).

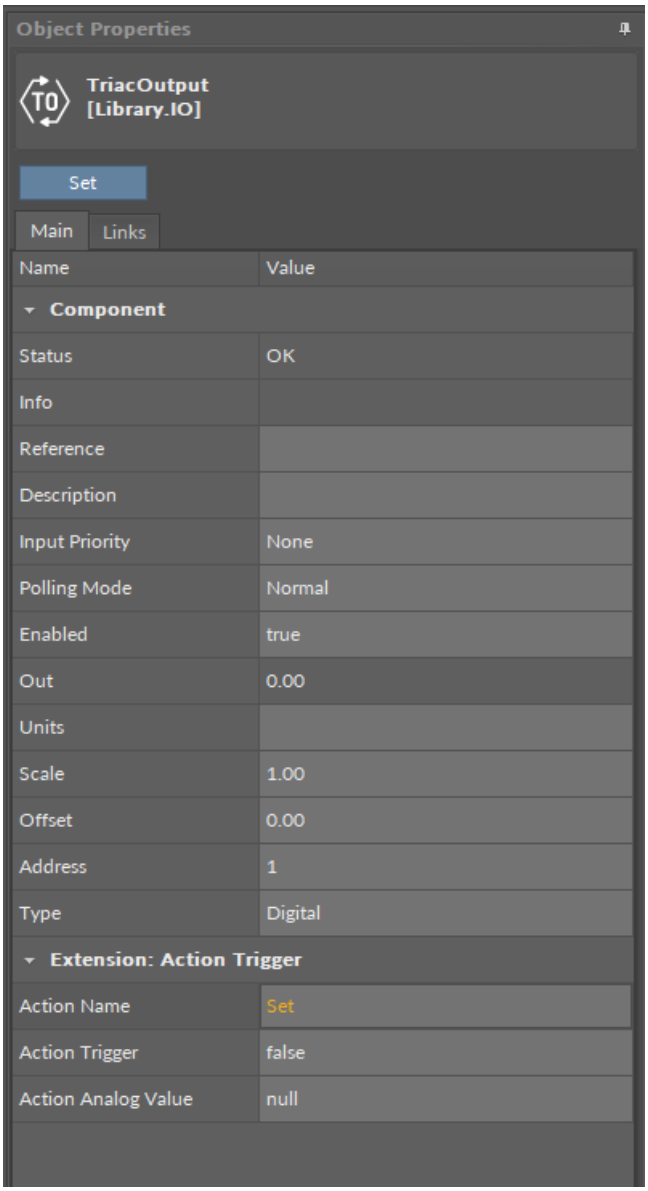


Figure 115. The ActionTrigger extension in the TriacOutput component

RotaryActuator

Applicable to OS V1.7

The RotaryActuator component is an IO point (network point class) component that allows to control damper actuators. The component allows to transfer data received from the linked Data Point in order to control the damper actuator and allows to set the polling mode of the point. The component can pass data from the Data Point class component by linking Reference slots. The RotaryActuator component may be linked from an Analog Data Point.

In order to operate properly, the RotaryActuator component must be located under the Local IO component in the Networks container.

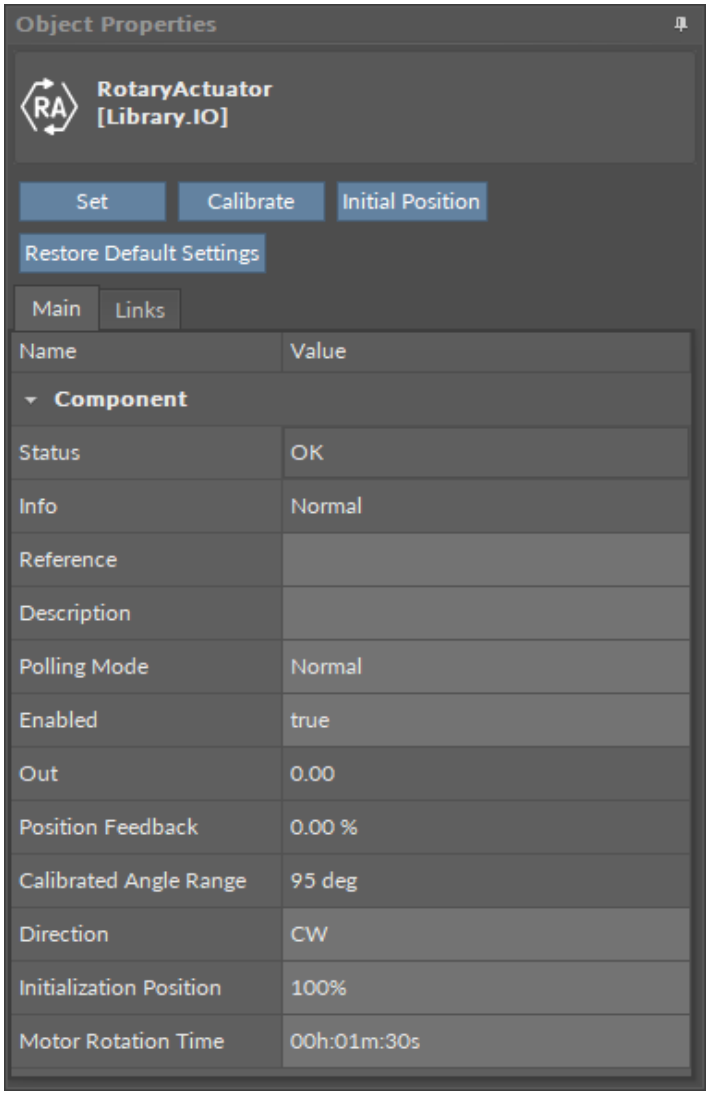


Figure 116. RotaryActuator component

The RotaryActuator component has the following slots:

- Status: indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false;
  - Available information: Disabled, Fault, OK;
- Info: provides a detailed information about a fault status of the component;
  - Available information:
    - Normal (status OK): the motor is in a normal state;
    - ManualOverride (status OK): the manual override button has been pressed on the device;

  - Calibrating (status OK): the motor is in a calibration process;
  - SettingPosition (status OK): the motor is in a setting position process;
  - Going to initial position (status OK): the motor is going to an initial position;
  - StuckAlarm (status OK): the VAV motor is mechanically blocked;
  - CalibrationAlarm (status OK): the calibration process has been invoked when the motor is blocked in both directions;
  - Low Voltage (status OK): the 24 V power supply is unplugged;
  - Driver Alarm (status OK): the motor driver is in a faulty condition such as over-current protection (OCP) or over-temperature protection (OTP);
  - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);
  - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
  - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
  - Point disabled (status Disabled): the Enabled slot in the component is set to false;

- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

Notes

Reference links from Data Points to network points also transfer values in the opposite direction, in a link-back-from process: having received a value by the Reference link, the network point transfers it back to the Data Point to whichever input priority from 1 to 16 is set in the network point.
The Reference links work on a change-of-value. If a value in the Data Point's Out slot changes, it is immediately transferred to a network point linked by the Reference link–such change is not dependent on an application cycle.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, meter's or sensor's location, or any other information the user finds applicable.
- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
  - Available settings: fast, normal (default), slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to transfer values to the damper actuator; by default, the component is enabled.
  - Available settings: true (enabled - default), false (disabled).

Note

If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** displays a value transferred to the damper actuator;

Note

If the component's Status is fault (e.g., an invalid value in the Address slot), the Out value is null.

- **Position Feedback:** shows current position of the motor;
- **Calibrated Angle Range:** shows a calibration angle degree;
  - Available information:
    - 0 – when not calibrated,
    - >0 – angle between limiters,
    - 95 – calibration with default limiters;
- **Direction:** allows to set the damper actuator's movement direction to 100% open position;
  - Available settings: CW (clockwise - default), CCW (counterclockwise);
- **Initialization Position:** allows to set an initial position for the damper actuator (by default, 100%);
- **Motor Rotation Time:** allows to set the time for a 90 degrees stroke of the motor (by default, 1 min 30 s).

The RotaryActuator component has the following actions:

- **Set:** allows to set a specific position for the actuator to reach upon invoking the action;
- **Calibrate:** allows to initialize the calibration process, in which the damper actuator goes to 100%, then to 0%, and goes back to the value set in the Out slot. This action is triggered after each restart of the VAV device, as long as it is not calibrated. It is also triggered after adding RA component onto the wire sheet.
- **Initial Position:** sets the damper actuator in an initial position, which is defined in the Initialization Position slot; afterwards, sets the actuator to the value set in the Out slot. This action is triggered after each restart of VAV device, as long as it is calibrated. It is also triggered after releasing the clutch;
- **Restore Default Settings:** sets default settings to the component.

ActionTrigger Extension

The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- **Action Name:** allows to select an action to invoke;
- **Action Trigger:** triggers an action selected in the Action Name slot;
- **Action Analog Value/Action Binary Value/Action String Value:** a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).

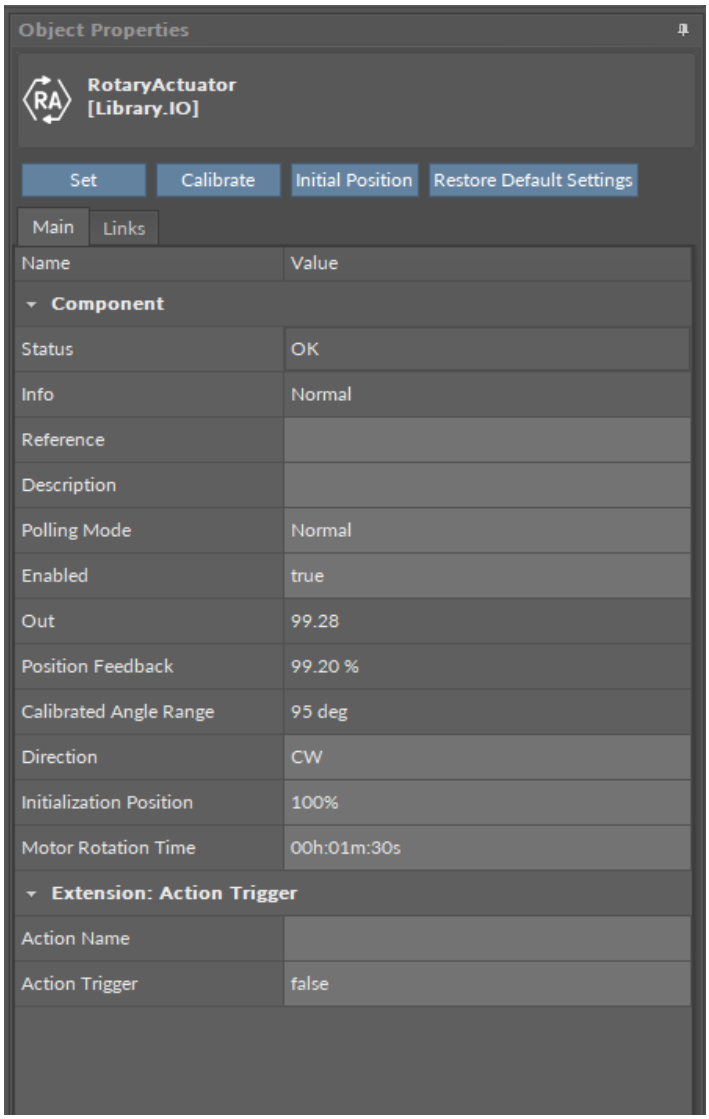


Figure 117. ActionTrigger extension in the RotaryActuator component

DipSwitch

Applicable to OS V1.7

The DipSwitch component is an I/O point (network point class) component, which allows to read the bit states of DIP switches installed in the front panel of the controller. There are three DIP switches installed in the front panel: one 6-position (S1) and two 8-position (S2, S3). The DipSwitch component values are updated in every execution cycle of the Local IO component.

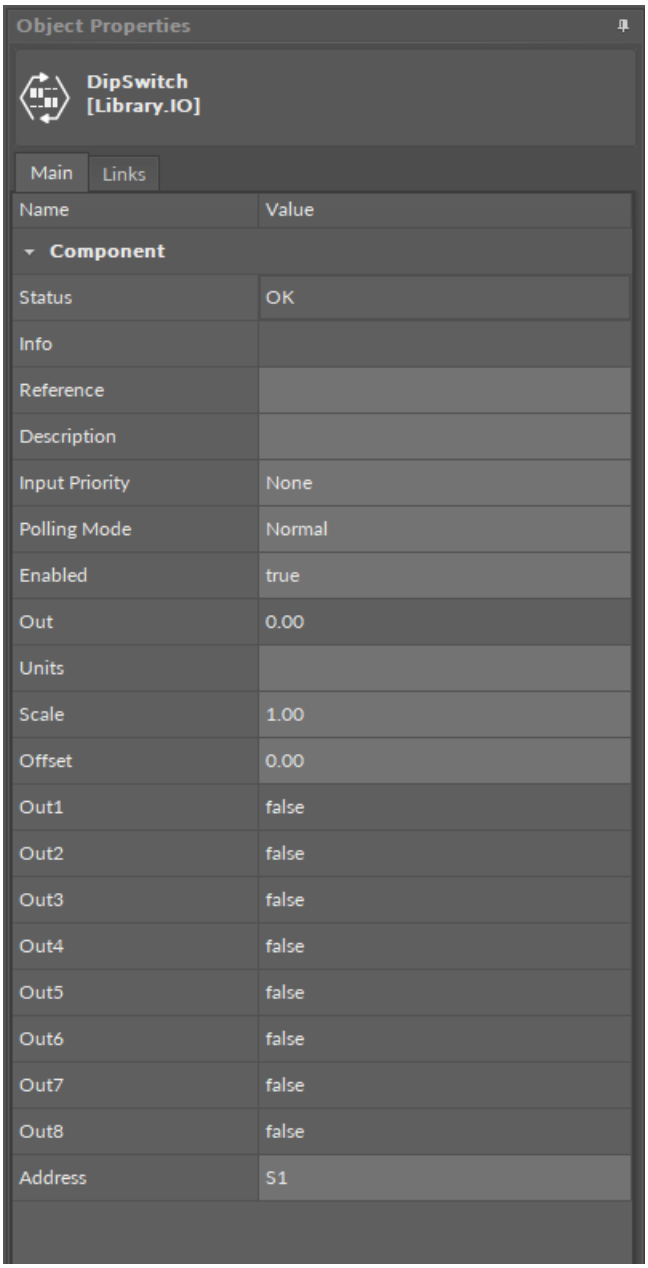


Figure 118. The DipSwitch component slots

The DipSwitch component has the following slots:

- **Status:** indicates the current status of the component; if the component works properly, its status is OK. The component becomes Disabled, once the Enabled slot is in false;
  - Available information: Disabled, OK;
- **Info:** provides a detailed information about non-OK statuses of the component;
  - Available information:
    - Bad request (status Down): occurs when point could not be polled successfully, most likely because its parameters are incorrect (e.g., object ID outside of range);

  - Incorrect configuration (status Fault): occurs when a parameter is missing or wrongly configured (e.g., the Address slot is null) or when the component is added to a device which does not support this particular type of network point;
  - Device disabled (status Disabled): the Enabled slot in the LocalIO is set to false;
  - Point disabled (status Disabled): the Enabled slot in the component is set to false;
- **Reference:** a special slot allowing to connect network point class components with Data Point class components. It allows to transfer the Out slot value along with the component's status.

By default, once the Reference link is created from the network point to the Data Point it sets the input priority to 16, which later can be changed manually.

- **Description:** an additional detailed information about a component that may be freely described by the user; the description may contain individual coding, defined in the user's system documentation, or any other information the user finds applicable.
- **InputPriority:** allows to select the input number in the Data Point, which the value from the network point class component's output is sent to; by default, the priority is none and sets to 16 after linking with a Data Point;
  - Available settings: none, 1-16.

Note: The Reference link from the network point to the Data Point cannot be changed to a 17th, default, priority.

- **Polling Mode:** allows to set the frequency of sending polling requests for the point's value—by default, the polling mode is set to normal;
  - Available settings: fast, normal, slow;
- **Enabled:** change of the slot's value enables or disables the component—if the component becomes disabled, it stops to read values from the physical input; by default, the component is enabled.
  - Available settings: true (enabled), false (disabled).

Note: If the Enabled slot is in false (meaning the component is disabled), the Status slot becomes Disabled.

- **Out:** displays the binary sum of bits (bit no. 1 = 2^0, + 2^1 for bit no. 2, etc.);
- **Units:** defines a unit of the Out slot value; no unit is used here by default;
- **Scale:** sets a fixed scaling factor for output linearization; the Out value is calculated according to the linear function formula (y=ax+b), and the Scale slot set the a value of the formula;
- **Offset:** sets a fixed offset value to the output value; the Out value is calculated according to the linear function formula (y=ax+b), and the Offset slot set the b value of the formula;
- **Out1-Out8:** shows the binary value of the respective bit on the addressed DIP switch;
  - Available states: true or false;

Worth to Notice

In case of the 6-position DIP switch, the Out7 and Out8 slots are permanently false. The 6th switch in the true state restores factory settings in the controller.

- **Address:** allows setting the address of the DIP switch, which bits are read in slots Out1-Out8;
  - Available settings: S1 (6-position DIP switch), S2 (8-position DIP switch), S3 (8-position DIP switch).

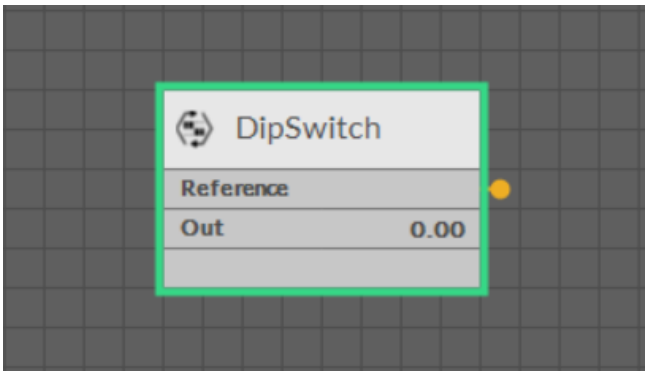


Figure 119. The DipSwitch component linked

5.2 Dedicated Tools

Tools that enable free programming of a user application are:

- iC Tool
- iC Workbench with nE2 Link Module

5.2.1 iC Tool

The iC Tool is a free computer program provided by the iSMA CONTROLLI.

As a significant part of an end-to-end solution, the iC Tool is a native programming tool for the nano EDGE ENGINE devices. The iC Tool covers all requirements to create and manage applications: it has a wire sheet for convenient visual programming and property sheets for details. It offers libraries management, real-time monitoring of system states and slots values, logs, deployment, and backup.

iC Tool Version for VAV14-IP Programming

For successful programming, please make sure that V1.7.0 version or higher of the iC Tool is used.

Find out more about iC Tool in the user manual.

5.2.2 iC Workbench with nE2 Link Module

iC Workbench is a Niagara 4 Workbench tool available to download in the iC Connect tool.

nE2 Link

The nE2 Link module V1.1 is essentially required for a successful operating of the nano EDGE ENGINE libraries in the Niagara environment.

The nE2 Link for Niagara is a comprehensive solution designed to enhance the Niagara Framework by enabling seamless commissioning, programming, and control of nano EDGE ENGINE devices.

The module is addressed to current and future Niagara Framework users who want to comprehensively manage, program, and integrate nano EDGE ENGINE devices directly into the Niagara 4 environment.

Using the built-in functionalities in Niagara, nE2 Link extends its capabilities to include nano EDGE ENGINE functions natively. The extension greatly extends the reach and usability of nano EDGE ENGINE devices, making it easier for users to integrate and manage building systems directly into Niagara 4, without the need for third-party tools.

The purpose of this document is to describe how to correctly install and start using nano EDGE ENGINE devices in a native Niagara environment.

Follow these step to start using the nE2 Link:

- Step 1: Installation
- Step 2: Configuration
- Step 3: Programming
- Step 4: Integration to Niagara

Step 1: Installation

Supported Niagara Versions

nE2 Link is dedicated to Niagara 4 and supports Niagara 4.11 and higher.

Required Modules

In order to work properly, the nE2 Link requires the following modules:

- nE2Link-rt.jar,
- nE2Link-ux.jar,
- nE2Link-wb.jar.

Contact the authorized iSMA CONTROLLI distributor to get the latest modules.

Installation

Niagara Workbench (Recommended)

> nE2 Link is dedicated to work with Niagara Workbench.

nE2 Link can work directly on local Workbench stations. In order to use it correctly, follow the steps below:

1. Close the Workbench.
2. Copy the `nE2link-rt.jar`, `nE2link-ux.jar`, and `nE2link-wb.jar` files to the Niagara modules directory. Default Path: `C:\Niagara\Niagara-4.x.x\modules`.
3. Reopen the Workbench.
4. Connect to the local Platform using Workbench and restart the station.

Niagara Controller

> In exceptional situations, it is possible to use nE2 Link directly on a Niagara controller with a limitation of opening maximum 2 connections at a time using nE2deviceExt in Niagara controllers. **For effective and seamless work, it is recommended to use nE2 Link with Niagara Workbench.**

It is possible to use the nE2 Link directly on a Niagara controller, such as MAC36 or JACE.

> Note: Before proceeding, make sure the module is correctly installed in the local Workbench.

1. In Workbench, connect to the controller's Platform.
2. Open the Software Manager.
3. Locate the `nE2link-rt` and `nE2link-ux` modules in the list and select them.
4. Click the `Install` button at the bottom of the window.
5. Click the `Commit` button at the bottom of the window.
6. The modules will be installed on the station, and the list will reload when the process is complete.
7. Verify the `nE2link` modules are marked as “Up to Date”.
8. Restart the Niagara station.

By following these steps, the nE2 Link modules will be successfully installed, enabling to leverage the full capabilities of the nano EDGE ENGINE devices within the Niagara 4 environment.

Step 2: Configuration

Adding nE2DeviceExt in Niagara

(a) In Workbench, navigate to the **nE2Link** module in the Palette window, search and open the **nE2Link** module.

The module palette contains the Programming folder.

The **nE2DeviceExt** is a network device extension located in the Programming folder. The nE2DeviceExt functions as a device extension inside Niagara networks, the BACnetNetwork or ModbusTcpNetwork, it must be dropped under the proper network device.

(b) Locate the **nE2DeviceExt** extension within the **Programming** folder.

Figure 120. nE2Device extension in the nE2Link module

The nE2DeviceExt extension consist of:

- **Poll Scheduler:** manages communication between the Niagara Framework and the nE2 controller;
- **Software Manager:** allows for managing libraries on the controller.
- **Libraries:** by default, the folder is empty and requires a real-time connection to upload the libraries available on the device. Once connected, the device's library will be populated with data from the device and load all the libraries available on the nano EDGE ENGINE device.

Figure 121. nE2DeviceExt contents

(c) Add nE2DeviceExt to BACnet Device:

- Make sure that the BACnet network is set up in the station.

Note: In nano EDGE ENGINE devices, such as the RAC18-IP, the native BACnet support guarantees that it can be discovered on the BACnet IP network out of the box.

- Drag the **nE2DeviceExt** extension from the **nE2Link** palette and drop it onto the BACnet device in the Niagara station.

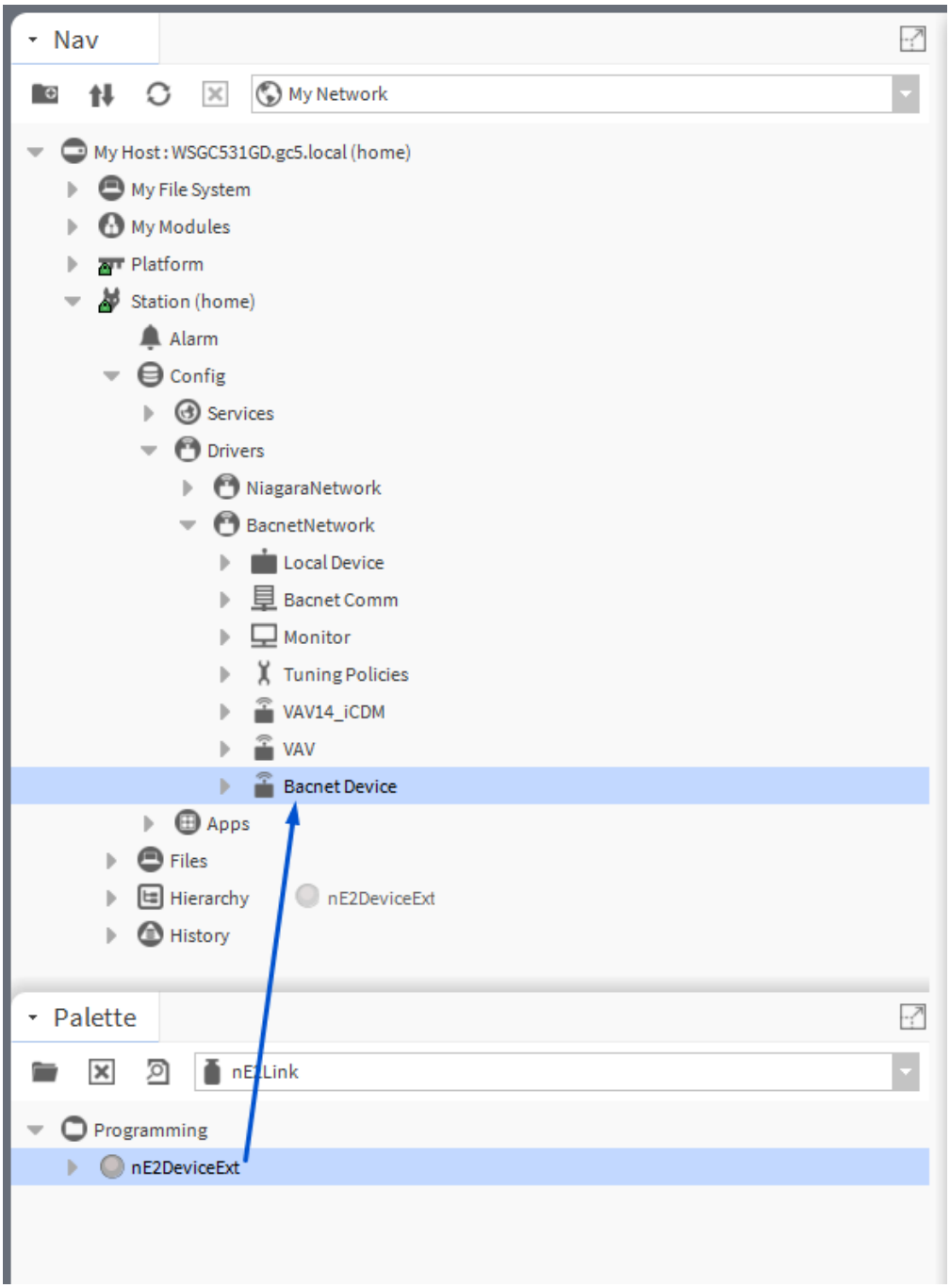


Figure 122. Adding nE2Device Ext to the BACnet device

Establishing a Connection with the nano EDGE ENGINE Device

(a) Once the extension is added to the device, right-click on the `nE2DeviceExt`, go to Actions → Connect.

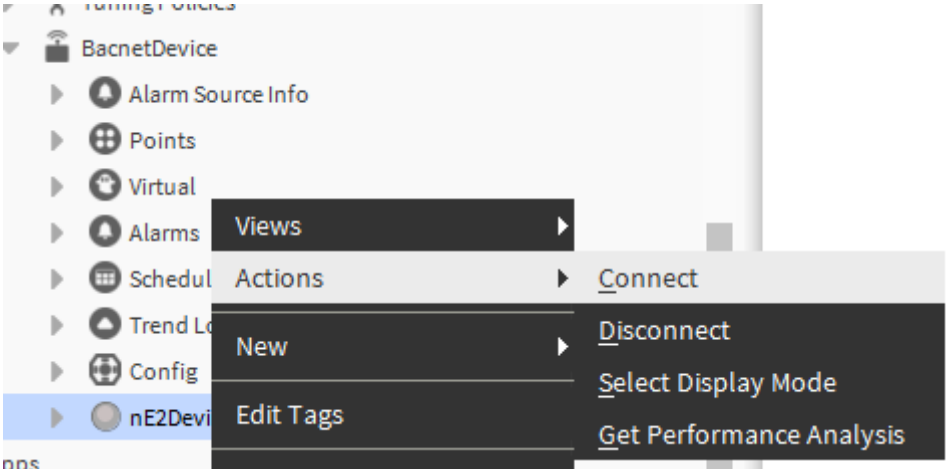


Figure 123. Connect option

(b) A pop-up connection window will open. Fill in the following slots:

- **IP Address:** the nano EDGE ENGINE device address;
- **Port:** iFnet port (by default, 88);
- **User Name:** nano EDGE ENGINE username (by default, admin);
- **Password:** nano EDGE ENGINE user password (by default, admin).

Note: Password must be changed after the first connection to the device, see First Connection and Password Setup.

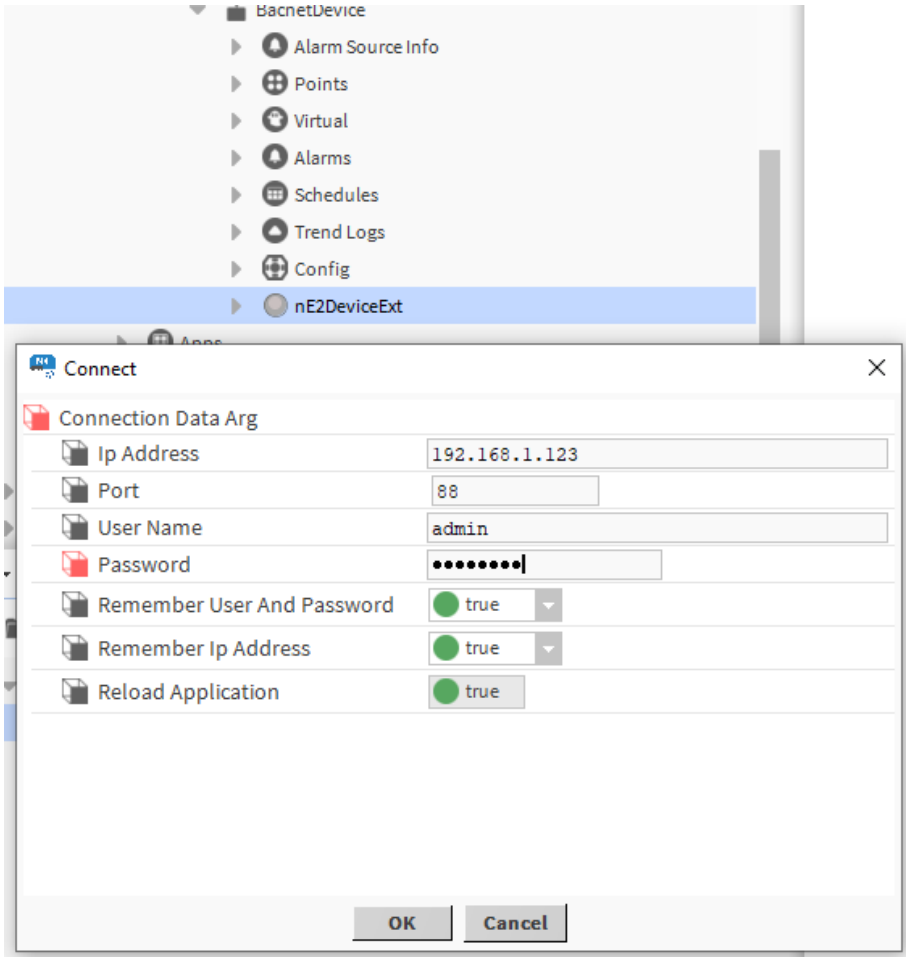


Figure 124. Filling in connection credentials

(c) Click OK to establish connection with the device.

nE2DeviceExt Interactive Connection LED

The **`nE2DeviceExt`** icon includes a status LED indicator that visually represents the device's connection state to ensure easy monitoring for users:

- gray LED: the device is disconnected,
- orange LED: the device is connecting,
- yellow LED: the device is connected and the application is loading,
- green LED: the device is successfully connected and the application has finished loading.

These color-coded LED statuses provide clear feedback to the customer about the current state of the device.

Once the connection with the device has been established (the green LED indicator is displayed), the following start screen is displayed:

Communication Status: Connected (Ready)	Status: Running
Device Name: RAC18-IP_SN27640513	Device Model: RAC18-IP
Serial Number: 27640513	OS Version: 1.6.0.8576
I/O: AO: 3, DO: 5, TO: 2, UI: 4, DI: 4	Interfaces: Serial: 1, Ethernet: 1
Current Time: 2024.12.13 09:28:30 (ok)	Uptime: 1:16:33:33
CPU Load: 27%	Available Datapoints: 101

Figure 125. nE2 Link start screen

The start screen shows the following information:

- communication status,
- device status,
- device name,
- device model,
- serial number,
- OS version,
- list of I/Os,
- interfaces,
- current time,
- uptime,
- CPU load,
- available Data Points.

Worth to notice:

If the connection is established for the first time or the extension gets disconnected, the following home screen is displayed:

The screen provides information such as:

- version of the module;
- copyrights;
- support;
- contact information.

Emergency Mode

The system and application(s) of the nano EDGE ENGINE controllers are stored on an SD card. If the SD card is not detected in the device or the device detects frequent reboots (at least 5 times in 6 minutes), which prevent correct operation, the device enters an emergency mode.

What Causes the Emergency Mode?

- No SD card is detected in the device.
- The diagnostic process reveals error in I/Os.
- Storage limit is exceeded.
- Required files are missing during a start-up of the device.
- Libraries or files are corrupted.

Operation in Emergency Mode

In the emergency mode, the device operation is limited:

- libraries are not loaded;
- the SD card configuration is not loaded;
- only the System container with limited options (only Logs and Platform components) is displayed in the tree;

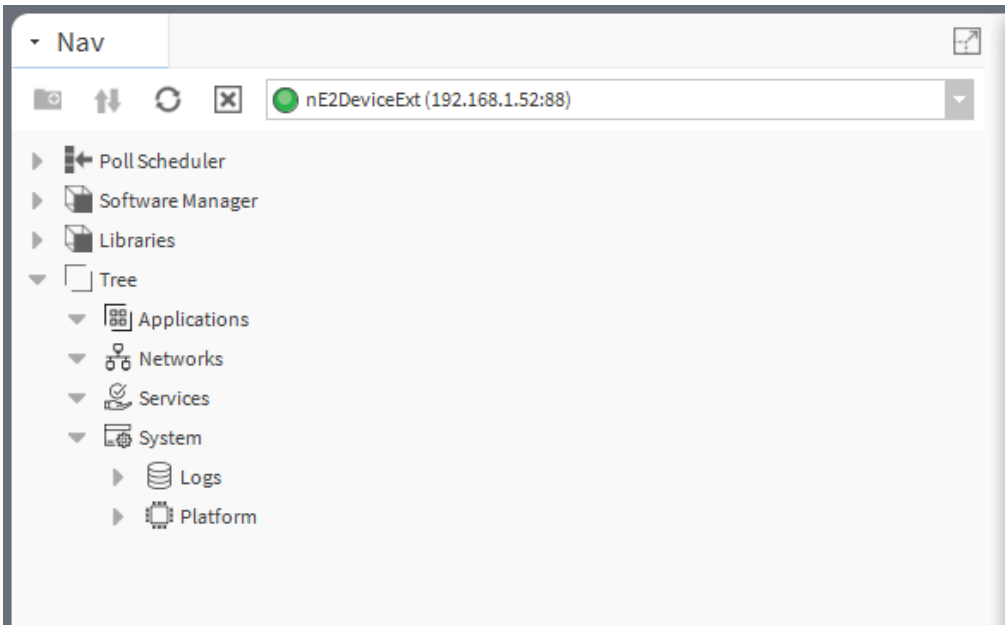


Figure 126. Device tree in the emergency mode

- the ALM LED is lit continuously;
- the iFnet runs with an IP/port taken from a flash storage;

Note: The flash storage must be synchronized to configuration slots when available.

- no authorization or credentials are taken from the flash storage (like IP/port).

Communication Status: Connected (Ready)	Status: EmergencyMode
Device Name:	Device Model: RAC18-IP
Serial Number: 4	OS Version: 1.7.0.9562
I/O: AO: 3, DO: 5, TO: 2, UI: 4, DI: 4	Interfaces: Serial: 1, Ethernet: 1
Current Time: --	Uptime: --:--:--:--
CPU Load: --%	Available Datapoints: 0

Figure 127. Start screen for a connected device in emergency mode

Possible Actions

When the device enters the emergency mode, take one of a few possible actions:

1. read logs from the SD card if available;
2. reboot;
3. restore to defaults (using the Restore in the System context menu): remove files from the SD card (if available and formatted) excluding only files with IP, port, and credentials (libraries must be also removed);

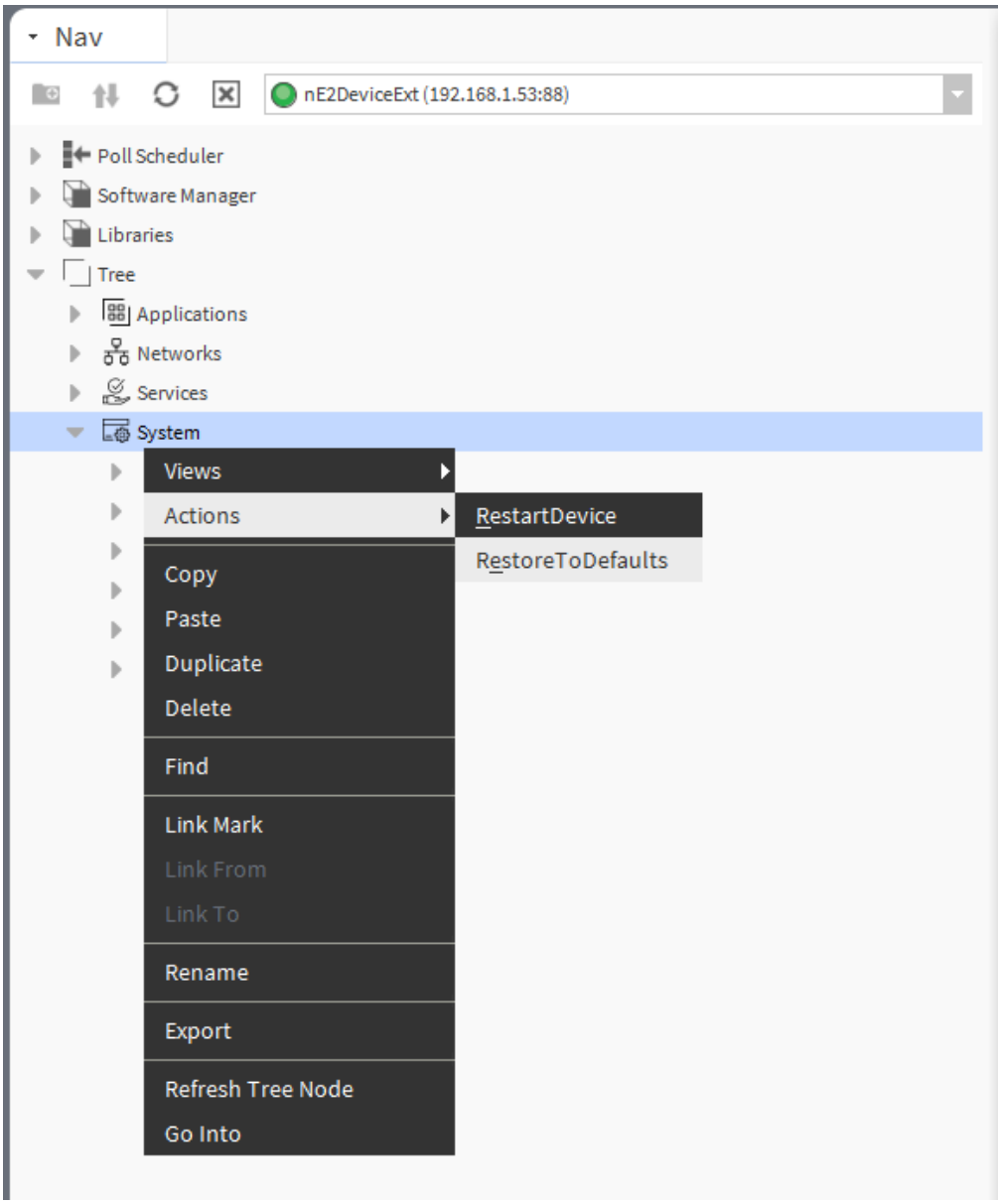


Figure 128. RestoreToDefaults action

4. restore to factory defaults (restoring with S1 6th DIP switch): format the SD card (if available), restore default credentials, IP, mask, gateway, iFnet port.

First Connection and Password Setup

When the connection is established correctly, the `nE2DeviceExt` icon will go from gray to green.

During the first connection to the device using a default password, a message will be displayed requesting to change the password.

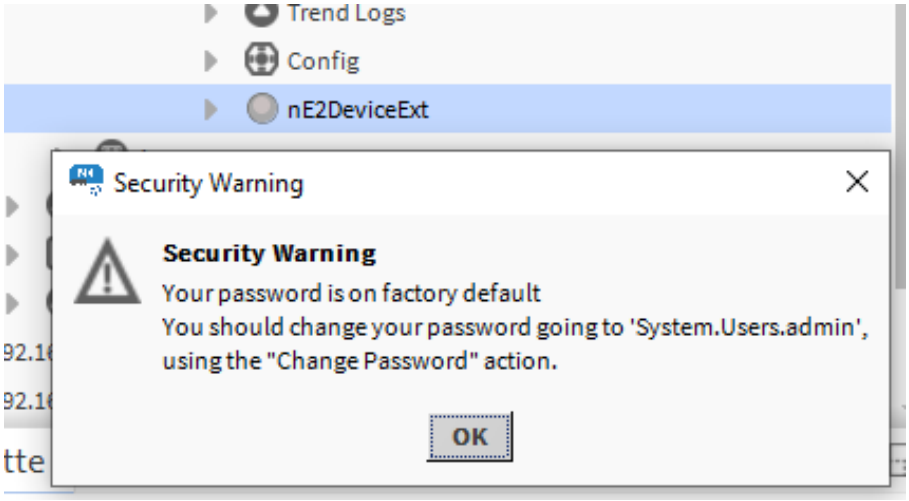


Figure 129. Change password prompt

Changing password is obligatory when first connecting to the device!

To change the password:

- expand the System container;
- expand Users;
- right-click the admin user;
- go to Actions → ChangePassword.

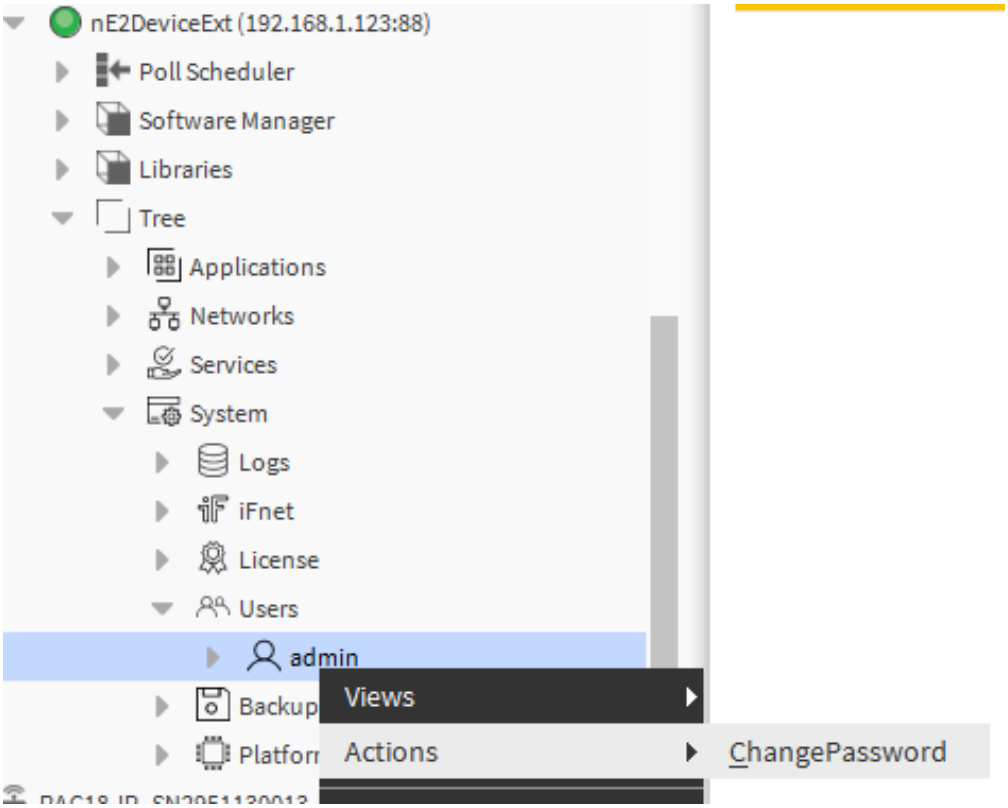


Figure 130. Change password action

A ChangePassword pop-up window will appear on the screen.

- Enter and confirm the new password according to the required length and the number and type of characters.

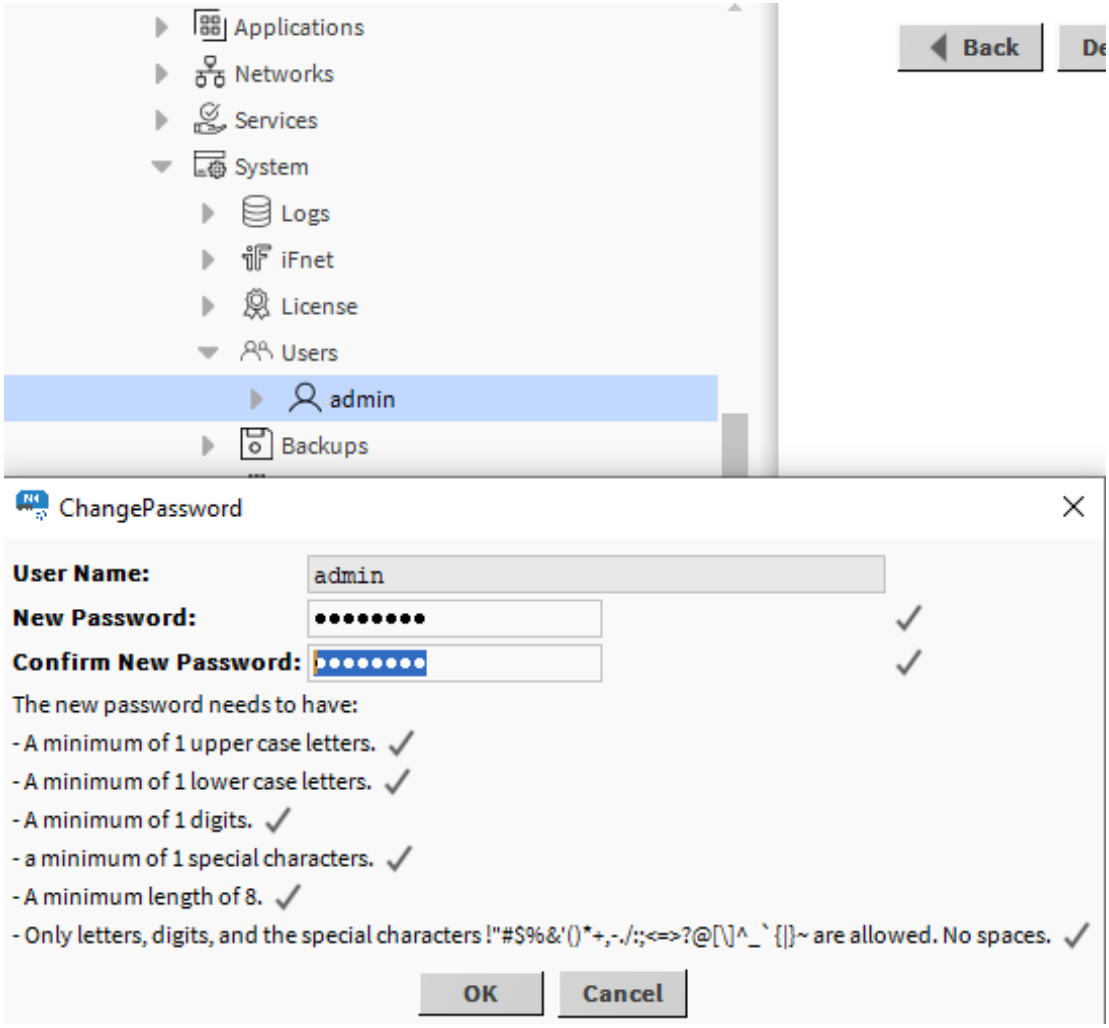


Figure 131. ChangePassword dialog window

- After successfully changing the password, right-click on the **`nE2DeviceExt`**; go to Actions → Disconnect.

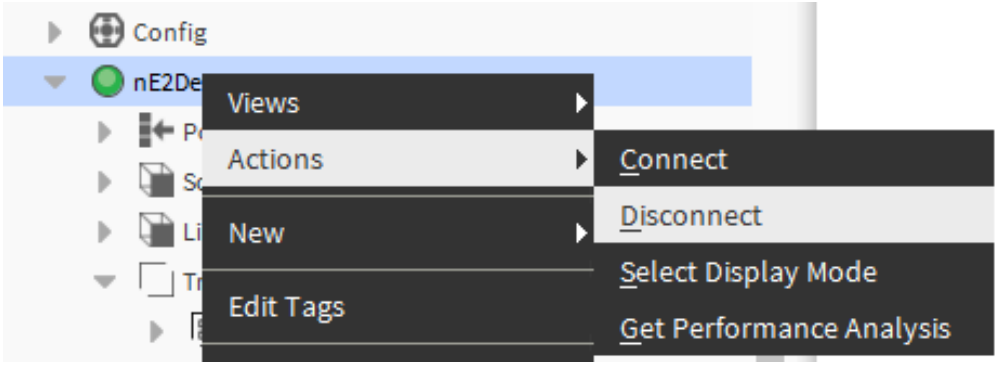


Figure 132. Disconnect option

The device icon will change from green to gray

- Right-click on the **`nE2DeviceExt`** again; go to Actions → Connect and enter the new password in the password field.

Note

If the typed password is incorrect, a pop-up will appear. Repeat the previous step and input the correct password.

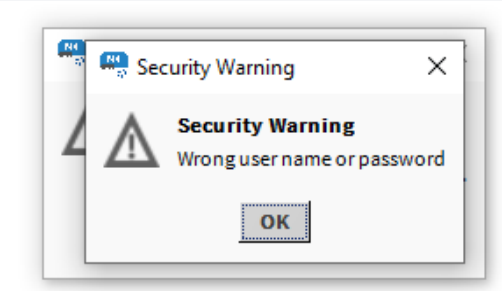


Figure 133. Wrong password notification

Once properly connected, expand the **`nE2DeviceExt`**. A new tree component is visible.

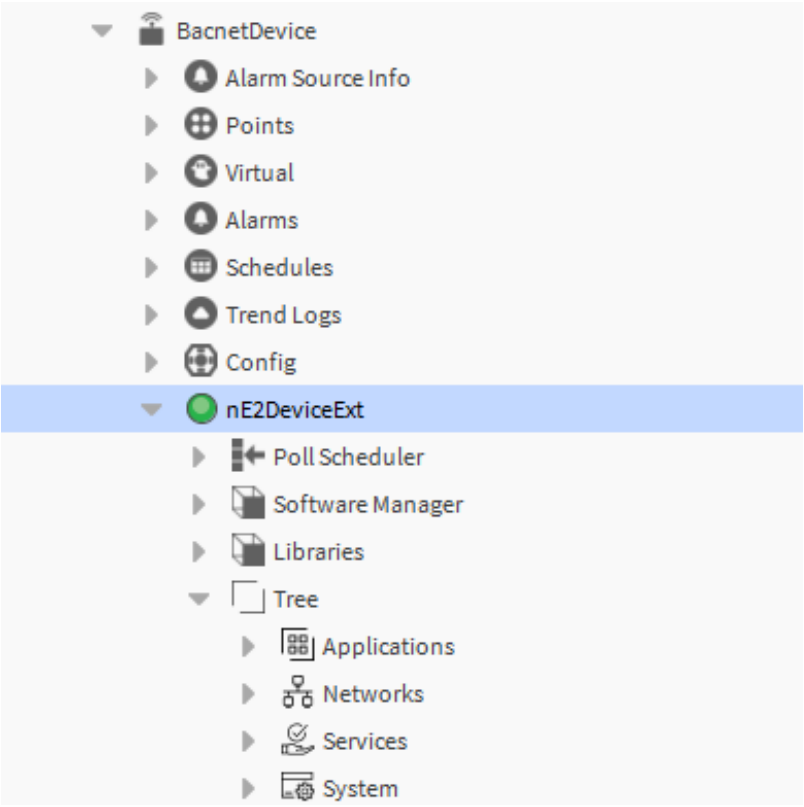


Figure 134. The nE2DeviceExt structure

Once expanded, the default available containers will appear in the device tree structure:

- Applications;
- Networks;
- Services;
- System..

The first connection with the device is fully established.

To learn more about the nano EDGE ENGINE architecture, please refer to the nano EDGE ENGINE Programming user manual.

IP Network Configuration

Change IP Network Settings

The IP address and other network settings are part of the Ethernet configuration in the Platform component in the System container.

To learn more about the System container, please refer to the nano EDGE ENGINE Programming user manual.

To change the network Settings:

- navigate to the Platform component in the System container;
- double-click on the Ethernet1 component.

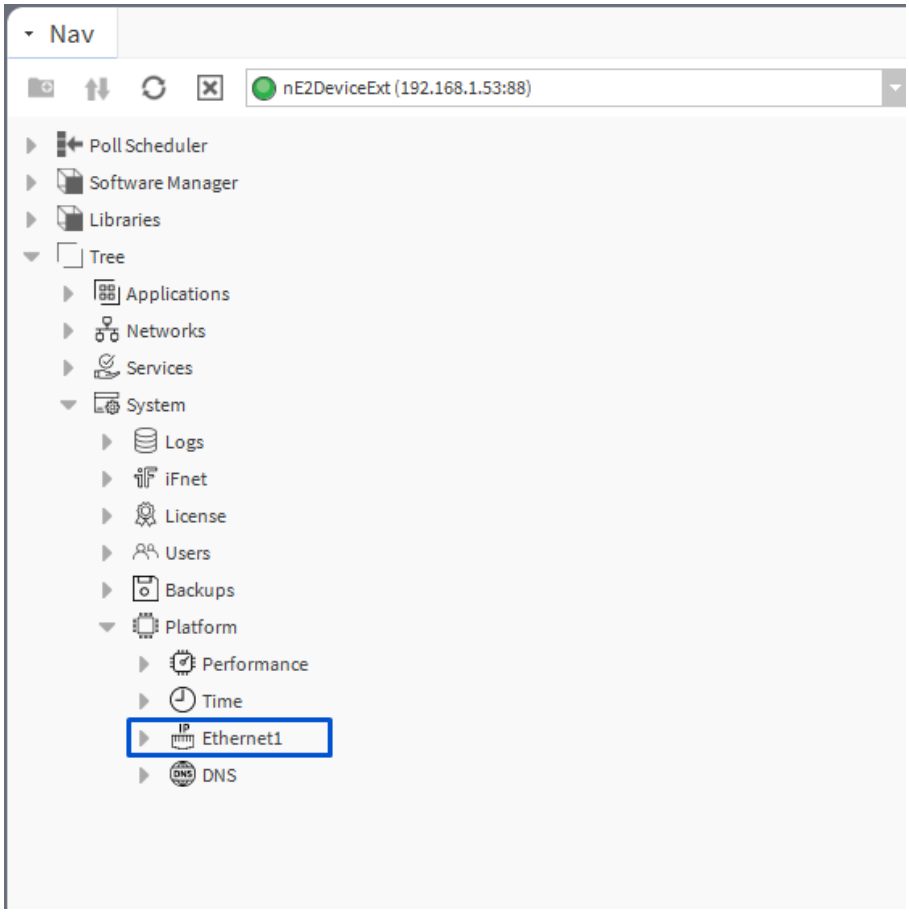


Figure 135. The Ethernet1 component in the tree

The Ethernet1 properties sheet will open on the main screen.

- Expand the IPAddress slot and type the new IP address, mask and gateway, or enable the DHCP mode.
- Confirm the new configuration with the Save button.

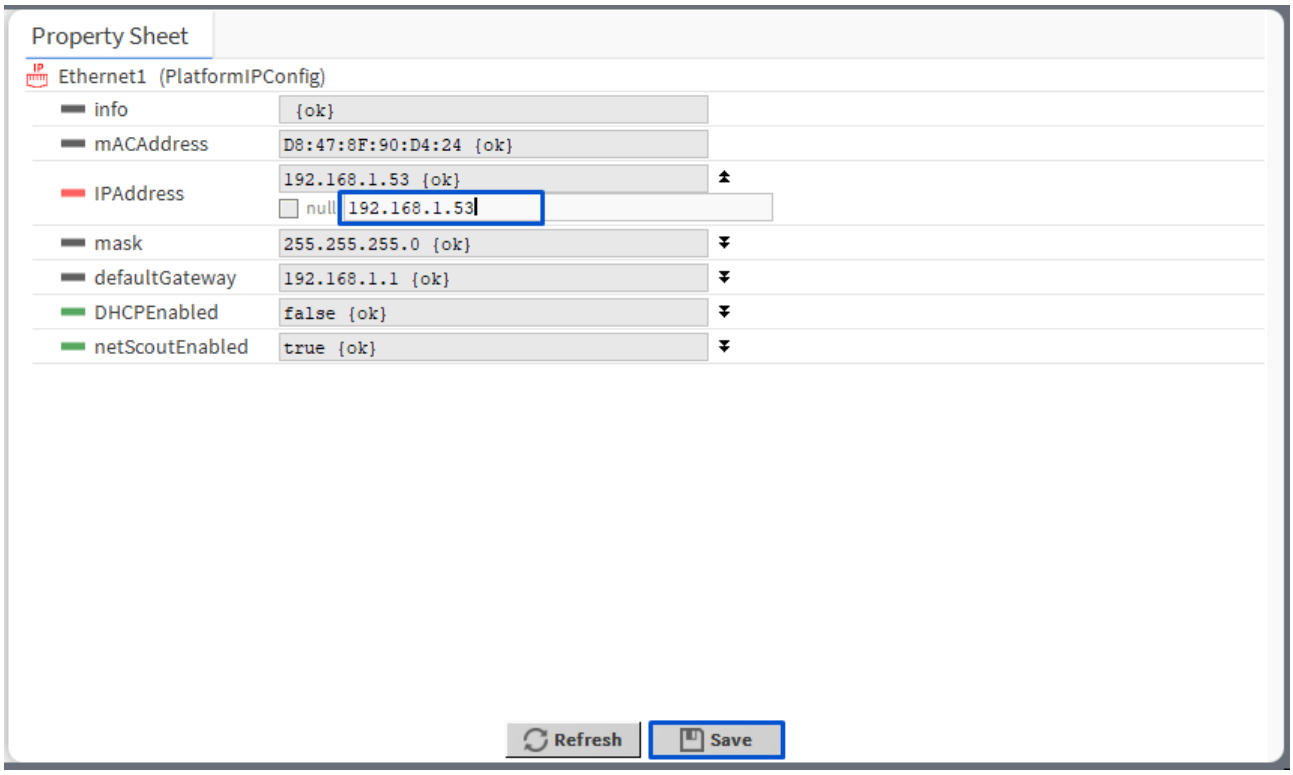


Figure 136. Ethernet1 properties

To learn more about Ethernet1, please refer to the nano EDGE ENGINE Programming user manual.

- After changing the device address, right-click on Ethernet1 and go to Actions -> RestartDevice;

The device will be rebooted after confirming the RestartDevice prompt:

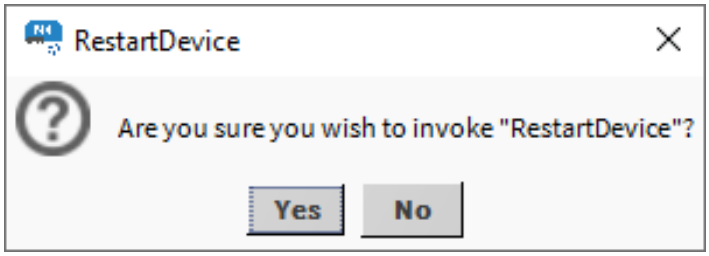


Figure 137. RestartDevcie prompt

- Reconnect to the device by changing the IP address in the nE2DeviceExt using the Connect action.

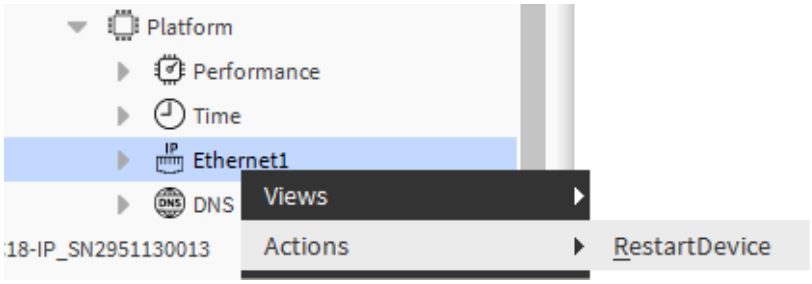


Figure 138. RestartDevice action in the Ethernet1 component

Time Settings

The time setting function allows users to configure the time settings of the controller directly from the Niagara station or to set a custom time. This feature is a part of the Platform component.

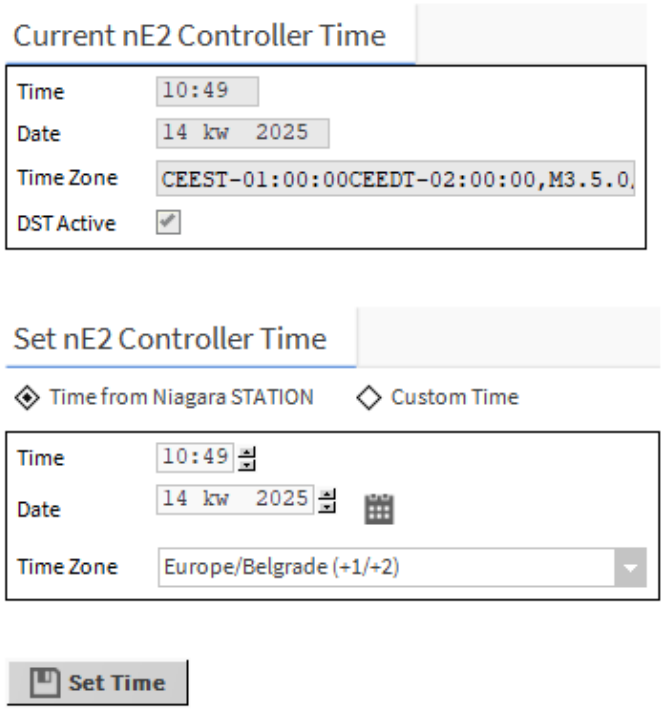


Figure 139. Time settings

To set the time:

- navigate to System>Platform;
- double-click the Time component.

The dialog window will display the following:

- **Current Nano Controller Time:** shows the currently set time, date, and time zone as well as indicates whether the Daylight Saving Time is currently active;

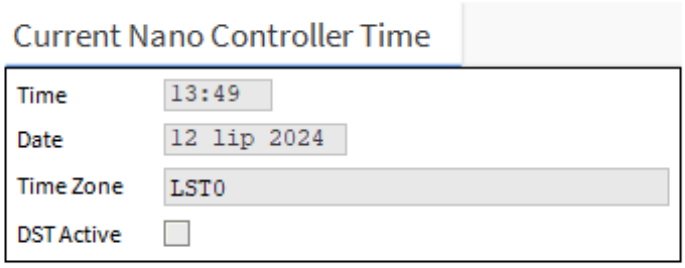
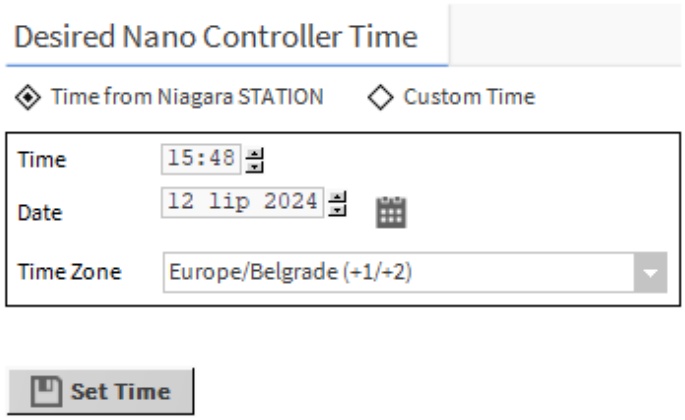


Figure 140. Current nano EDGE ENGINE device time

- **Desired Nano Controller Time:** allows the user to set the time on the controller directly from the Niagara station or to set a custom time.



To set the nano EDGE ENGINE device time based on the Niagara station time:

- confirm that the Time from Niagara Station option is selected;

In this configuration, the displayed time, date, and time zone are in read-only mode.

- click Set Time to configure the time on the nano EDGE ENGINE device as in the station;
- a pop-up asking to restart the device will be displayed;
- click Yes to confirm, the device will be restarted;

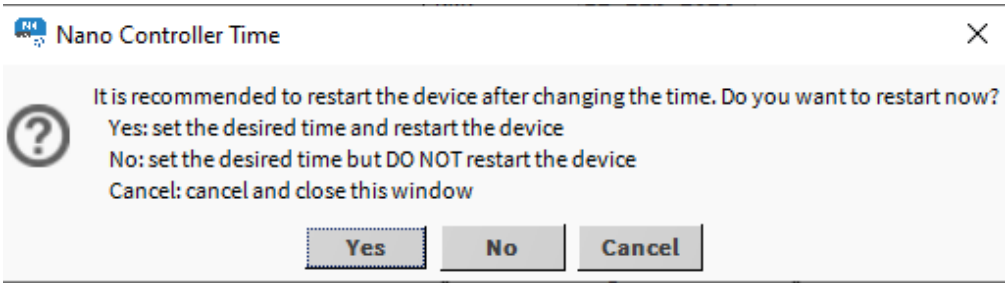


Figure 141. Set the time from the station dialog window

- reconnect with the device using the Connect action in the nE2DeviceExt.

To set a custom time in the controller:

- select the Custom Time option;
- the Time Setting dialog window can now be edited: set the time, date, and time zone;
- click Set Time to confirm;
- a pop-up asking to restart the device will be displayed, click Yes to confirm. The device will be restarted;
- reconnect with the device using the Connect action in the nE2DeviceExt.

To learn more about time settings and configurations, please refer to the nano EDGE ENGINE Programming user manual.

Software Manager

The Software Manager is synchronized with the Workbench or the Niagara controller shared folder of the station. By default, the Software Manager displays the default nano EDGE ENGINE libraries supported by the version of the module. All third party libraries must be added to the shared folder.
To navigate, double click the Software Manager component in the device tree.

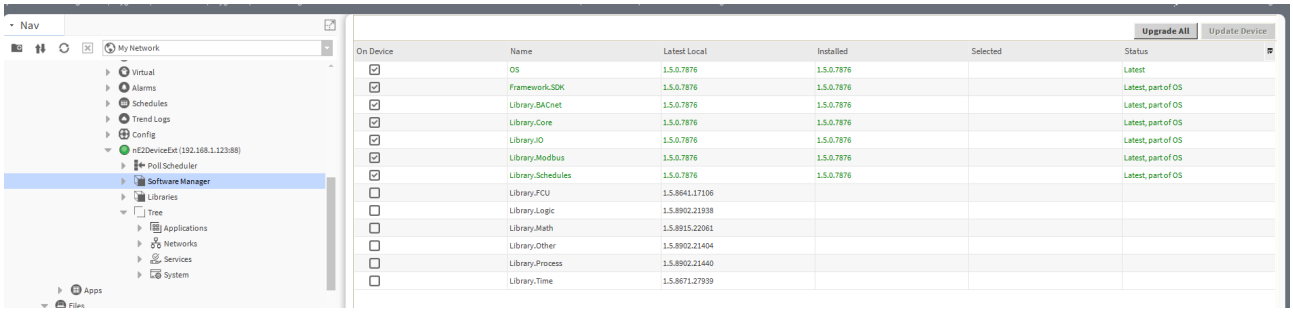


Figure 142. Software Manager view

Using Software Manager

The Software Manager view lists the OS and libraries available locally. The view highlights each row (OS or libraries) according to its status:

- **green:** the element is up to date, and requires no action;
- **orange:** the element is out of date, and can be updated;
- **blue:** the action is about to be taken on the element.

The Software Manager table contains the following columns:

- **On Device:** indicates, whether a given element is already installed on the device.
- **Name:** shows the name of the element.
- **Latest Local:** shows the latest version available locally to be installed on the device.
- **Installed:** shows the version of the element installed on the device.
- **Selected:** opens a dropdown list with all versions available locally for a selected element.
- **Status:** indicates, which action is to be performed on the element, once a specific version has been selected in the Action column.
    - Available information: Latest, Out of Date, Upgrade, Downgrade, Install, Uninstall, none (the selected version is the same as the one installed on the device).

In order to upgrade or downgrade the selected element, choose the desired version of the element in the Selected column, and press the Update Device option (highlighted in blue in the right upper corner of the Software Manager). This option executes all actions indicated in the Status column.

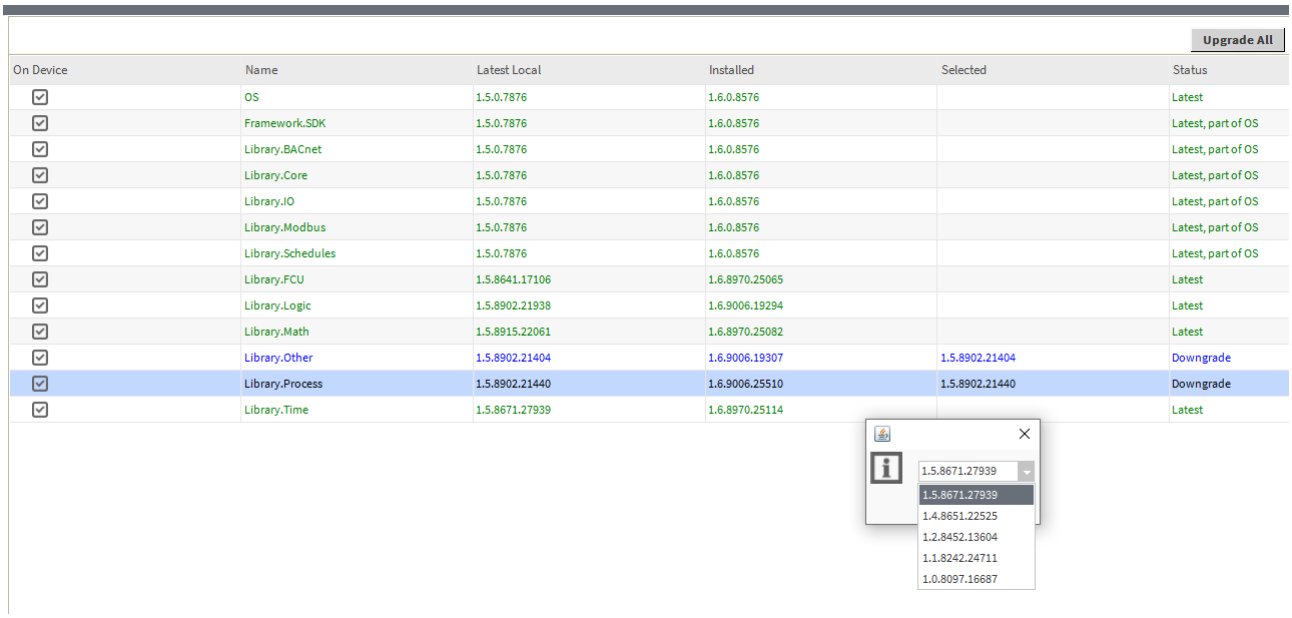


Figure 143. Selecting a library's version

Unless the user intends to manually select the versions to be installed, there is also the option to automatically select all newest versions for all out of date elements using the Upgrade All button.

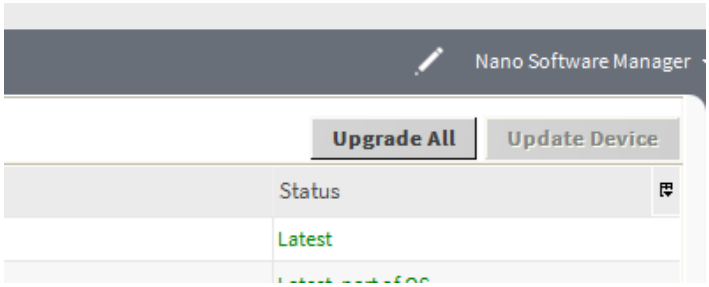


Figure 144. Upgrade All button

Regardless of the fact whether the user intends to add or remove the library available in the Software Manager, each operation requires performing three steps:

- check the box in the On Device column to install the library, or uncheck it to uninstall the library;
- provided the library is to be installed on the device, check its preferred version in the Selected column–by default, the newest version available locally is indicated to be installed;
- once selection of all libraries to be installed or uninstalled on the device is complete, hit the Update Device command.

Warning!
The OS cannot be removed from the device; it is preinstalled on the device's SD card, and the only operations, which can be performed on this element, are upgrading or downgrading it.

Uploading New Libraries

With the nE2Link, it is possible to upload libraries and/or OS files to the controller, which were not pre-loaded in the nE2DeviceExt. New libraries have to be saved on a local PC and then copied to the nanoEdgeEngine folder on the station:

- save new libraries in a folder selected location on the local PC;
- copy the proper nano EDGE ENGINE libraries and/or OS files from the local PC by navigating to the proper location on My Host (local PC); (it is possible to copy the whole folder with proper contents but please make sure then that the folder is named ‘nanoEdgeEngine’);

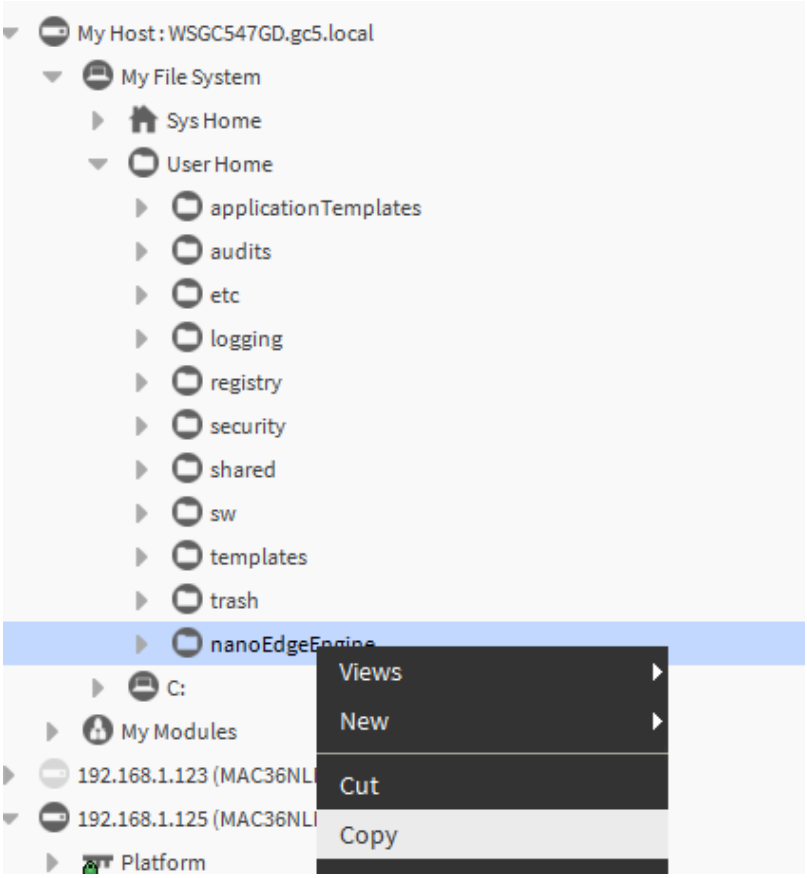


Figure 145. Copying the nanoEdgeEngine folder

- navigate to Station → Files → nE2Link →nanoEdgeEngine folder;

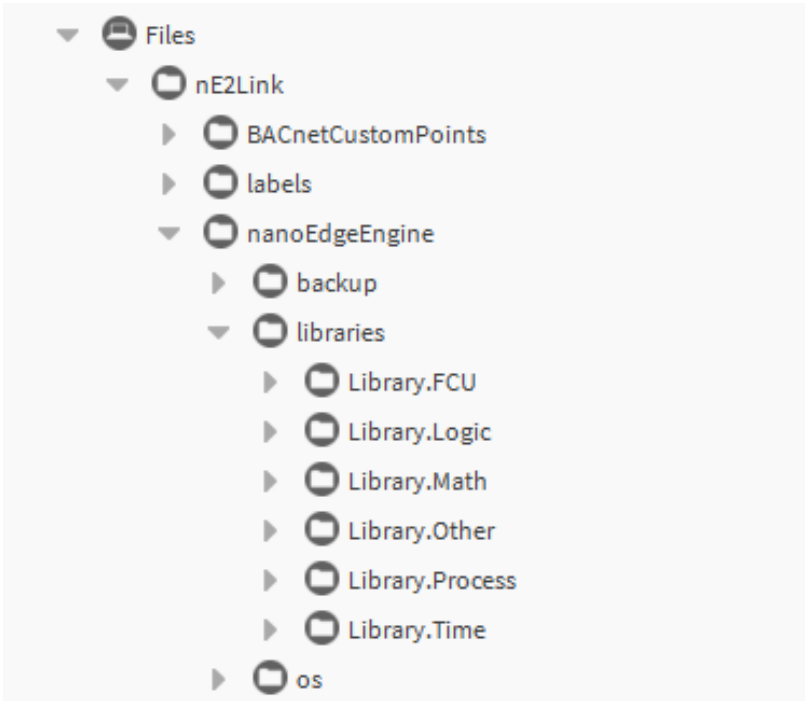


Figure 146. Pasted folder

- paste it under the Files container in the local station.

Note: Please make sure to preserve a proper structure of subfolders: new libraries files have to be stored in the libraries subfolder, OS files in the os subfolder, and backup files in the backup subfolder.

Once the libraries are added, they will become visible in the Software Manager.

- Select the libraries and OS version to be installed, upgraded or downgraded on the controller, or select the Upgrade all option;
- once all necessary software is selected, click Update device;
- a pop-up will be shown asking to confirm the action. Click Yes to load the new OS and libraries.

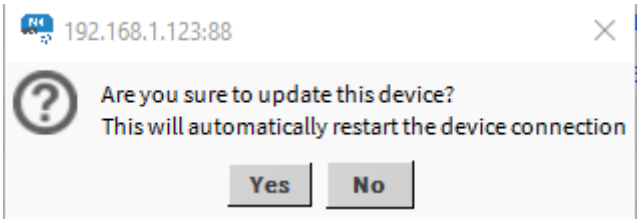


Figure 147. Update device prompt

The device will be restarted automatically.

- Once the device has restarted successfully, right-click on the nE2DeviceExt and connect to the device;
- after reconnection, confirm that selected software was successfully installed on the device.

Upgrade All | Update Device

On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.4.1.7340	1.4.1.7340		Latest
☑	Framework.SDK	1.4.8585.25238	1.4.8585.25238		Latest, part of OS
☑	Library.BACnet	1.4.8655.31468	1.4.8655.31468		Latest, part of OS
☑	Library.Core	1.4.8655.31015	1.4.8655.31015		Latest, part of OS
☑	Library.IO	1.4.8655.31292	1.4.8655.31292		Latest, part of OS
☑	Library.Modbus	1.4.8704.26958	1.4.8704.26958		Latest, part of OS
☑	Library.Schedules	1.4.8655.31332	1.4.8655.31332		Latest, part of OS
☑	Library.FCU	1.4.8585.25488	1.4.8585.25488		Latest
☑	Library.Logic	1.4.8585.25337	1.4.8585.25337		Latest
☑	Library.Math	1.4.8621.25370	1.4.8621.25370		Latest
☑	Library.Other	1.4.8641.19349	1.4.8641.19349		Latest
☑	Library.Process	1.4.8585.25565	1.4.8585.25565		Latest
☑	Library.Time	1.4.8651.22525	1.4.8651.22525		Latest

Figure 148. Updated Software Manager view

Library Not Loaded

The Software Manager has a mechanism informing the user about the libraries compatibility. For a proper operation, libraries versions have to be supported by the OS. If the library installed on the device has a version, which is not supported by the OS, Software Manager displays an error prompt and marks the library as 'not loaded'. In such case, it is required to upgrade the library.

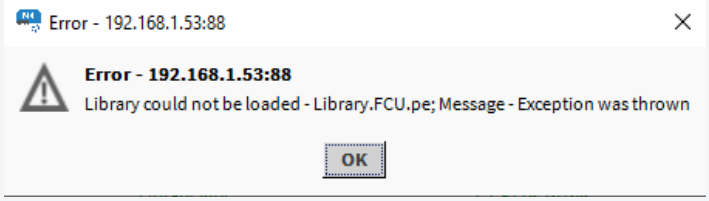
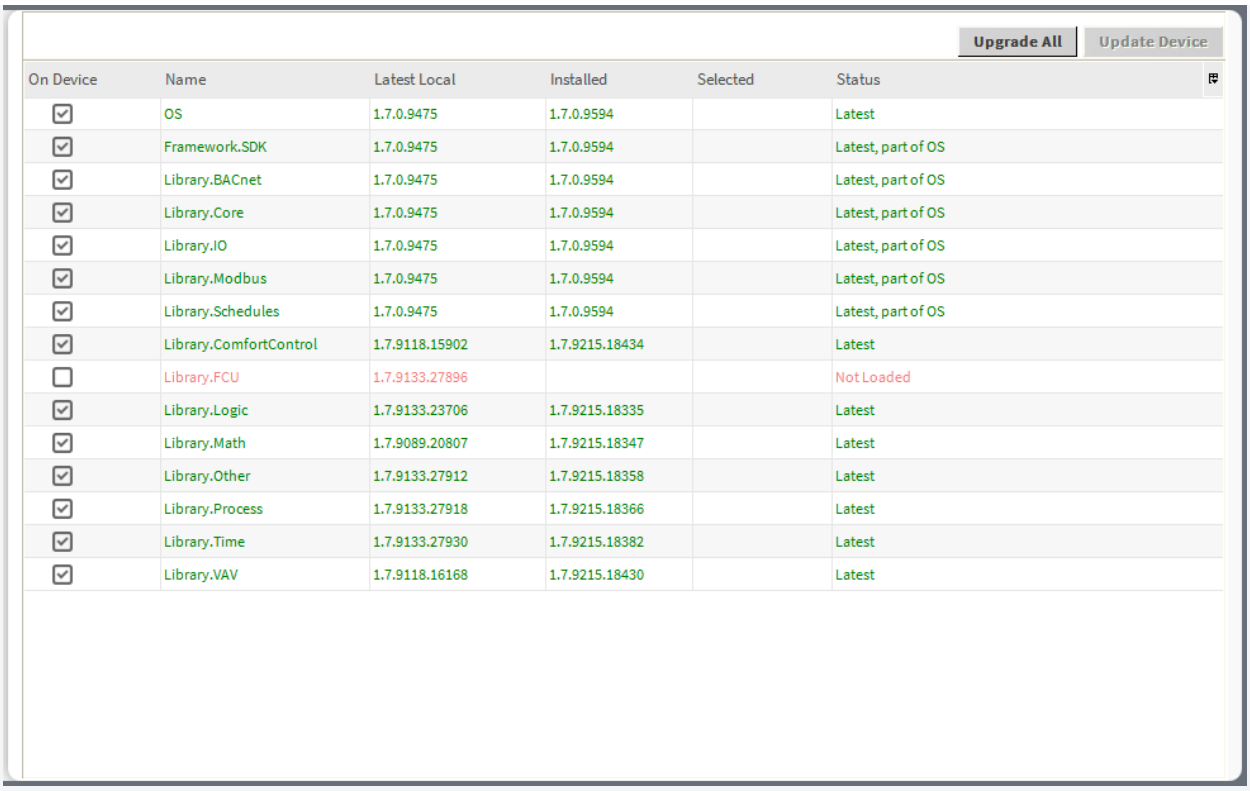


Figure 149. Error prompt



On Device	Name	Latest Local	Installed	Selected	Status
☑	OS	1.7.0.9475	1.7.0.9594		Latest
☑	Framework.SDK	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.BACnet	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.Core	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.IO	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.Modbus	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.Schedules	1.7.0.9475	1.7.0.9594		Latest, part of OS
☑	Library.ComfortControl	1.7.9118.15902	1.7.9215.18434		Latest
☐	Library.FCU	1.7.9133.27896			Not Loaded
☑	Library.Logic	1.7.9133.23706	1.7.9215.18335		Latest
☑	Library.Math	1.7.9089.20807	1.7.9215.18347		Latest
☑	Library.Other	1.7.9133.27912	1.7.9215.18358		Latest
☑	Library.Process	1.7.9133.27918	1.7.9215.18366		Latest
☑	Library.Time	1.7.9133.27930	1.7.9215.18382		Latest
☑	Library.VAV	1.7.9118.16168	1.7.9215.18430		Latest

Figure 150. Library not loaded

Backups

nE2DeviceExt allows users to invoke the device's backup function. The local backups are saved directly into the station's shared folder.

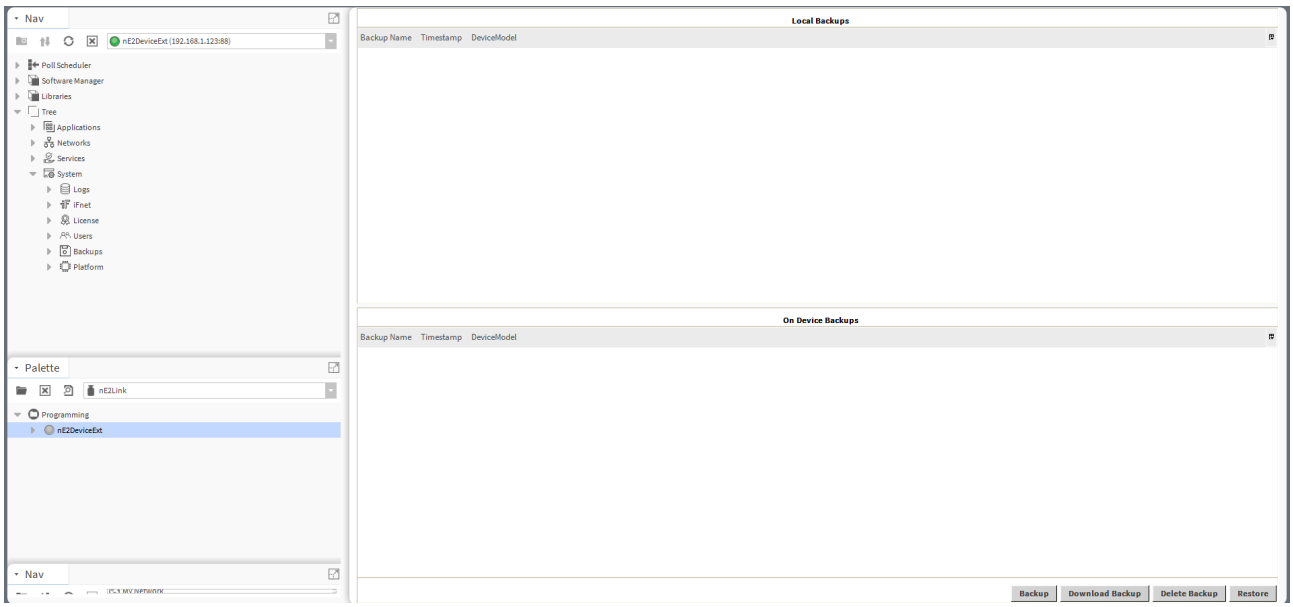


Figure 151. Backup Manager view

To perform a backup, go to the Backup Manager of the device, in the Backups component in the System container. The backup can be restored to the device.

> **Note:** nano EDGE ENGINE controllers allow to store one backup directly in the local device memory. Backups can be downloaded and stored in the Station Files folder.

All local backups are stored in the local Niagara station. To access backups go to Station → Files → nE2Link → nanoEdgeEngine ->backup → *BackupName.* Backups can be imported or exported from this location manually and will become visible in the Backup Manager view.

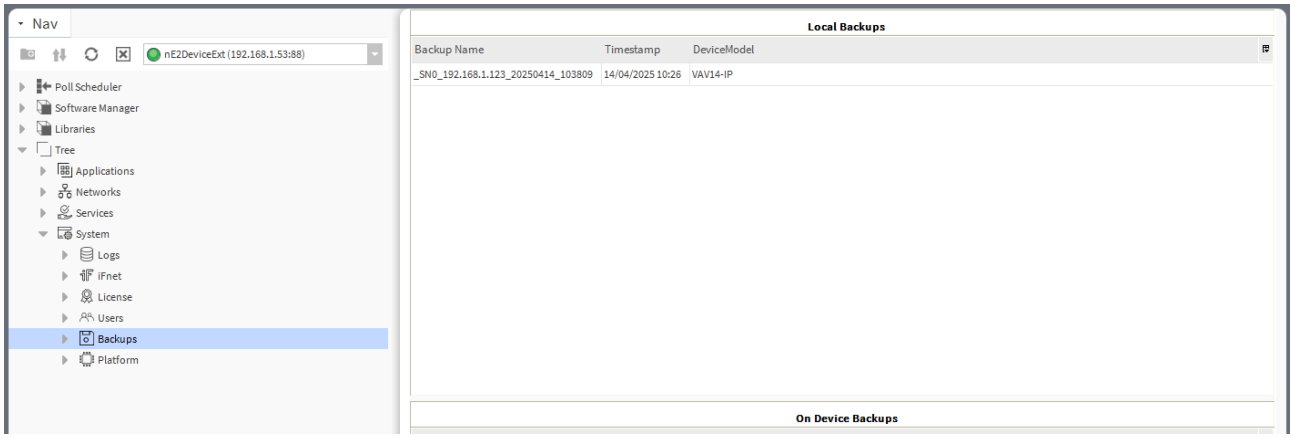


Figure 152. Local station backup stored in the station's Files

To learn more about the Backups, refer to the nano EDGE ENGINE Programming user manual.

Performing Backup

- Click the Backup button to invoke creating a backup.

> **Warning!**
> If there is any existing backup on the device, performing the backup action will overwrite it.

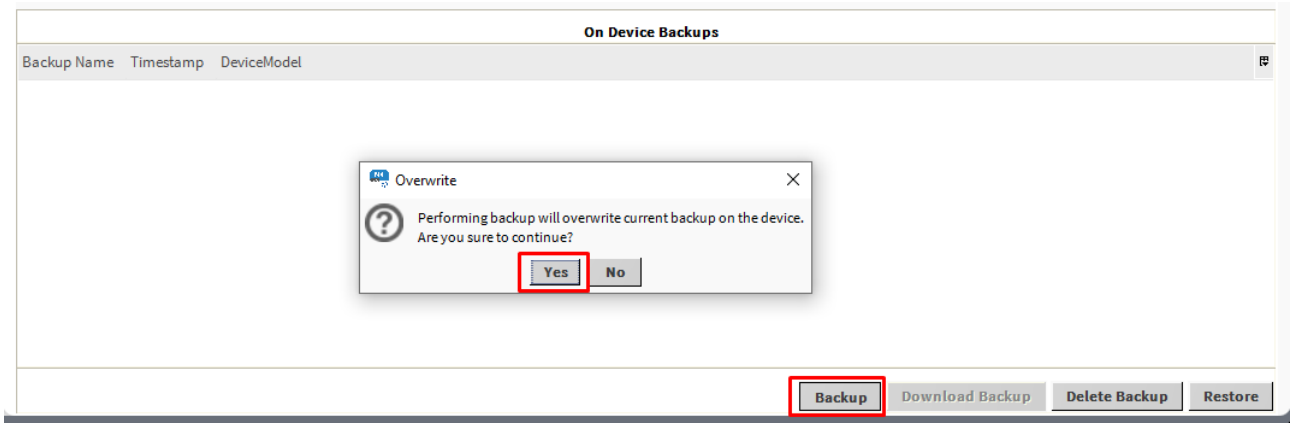


Figure 153. Pop-up informing about the risk of overwriting the existing backup on the device

Once the backup action is confirmed, the device will perform the backup. This process can take up to a few minutes. Wait for the process to finish.

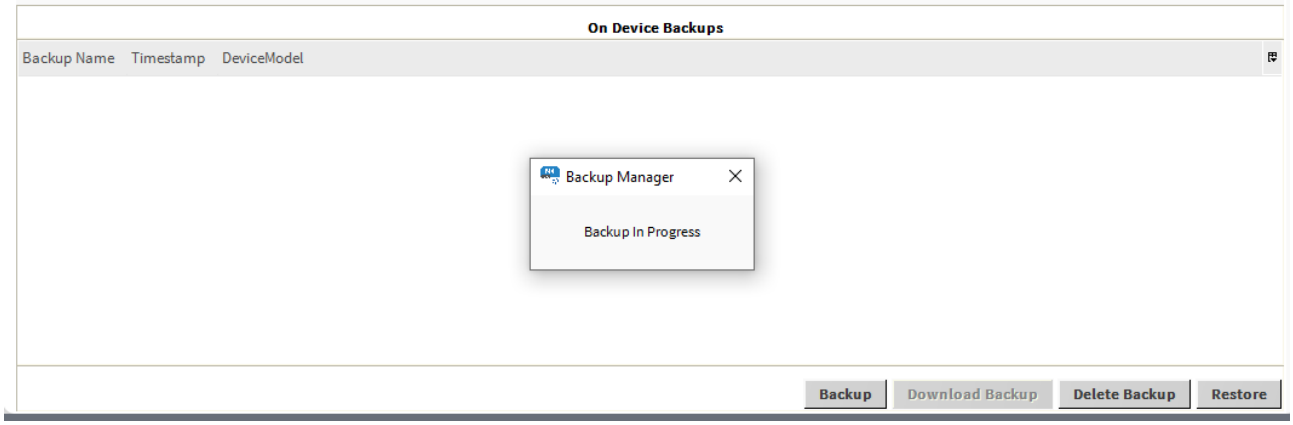


Figure 154. Pop-up informing about the backup progress

Once the process is completed, the backup will be visible in the On Device Backups table in the Backup Manager view.

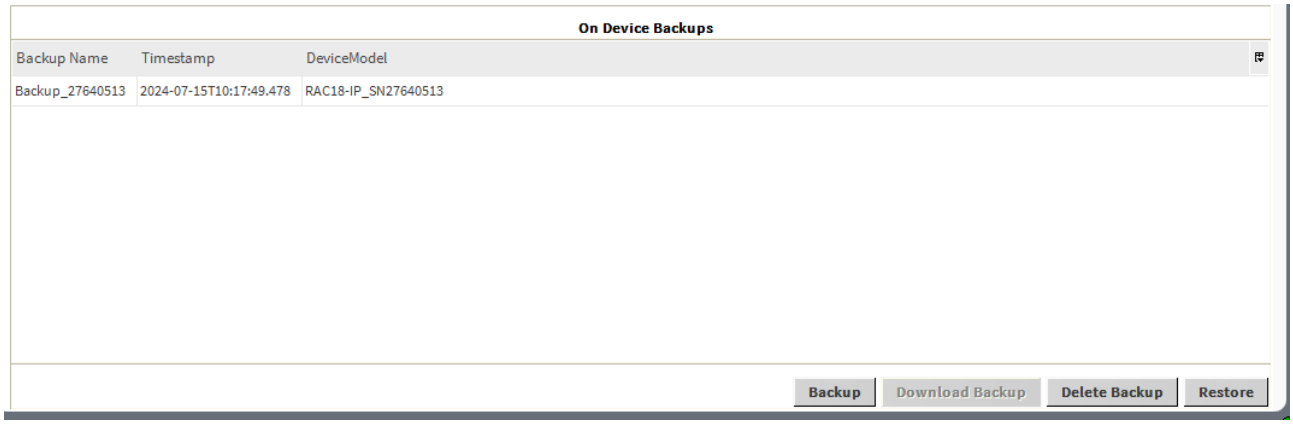


Figure 155. On Device Backups

Downloading Backup

- Click the Download Backup button to download the backup from the device to the local station.

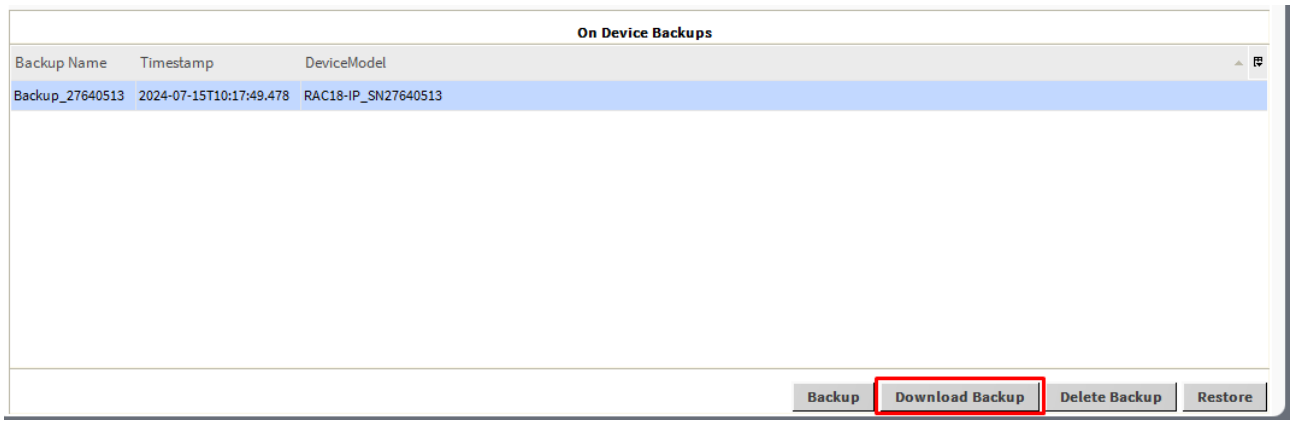


Figure 156. Download Backup button

A pop-up will appear. Set the Backup name or keep the default name. Click Ok to Confirm once the backup name is set. The backup will be downloaded.

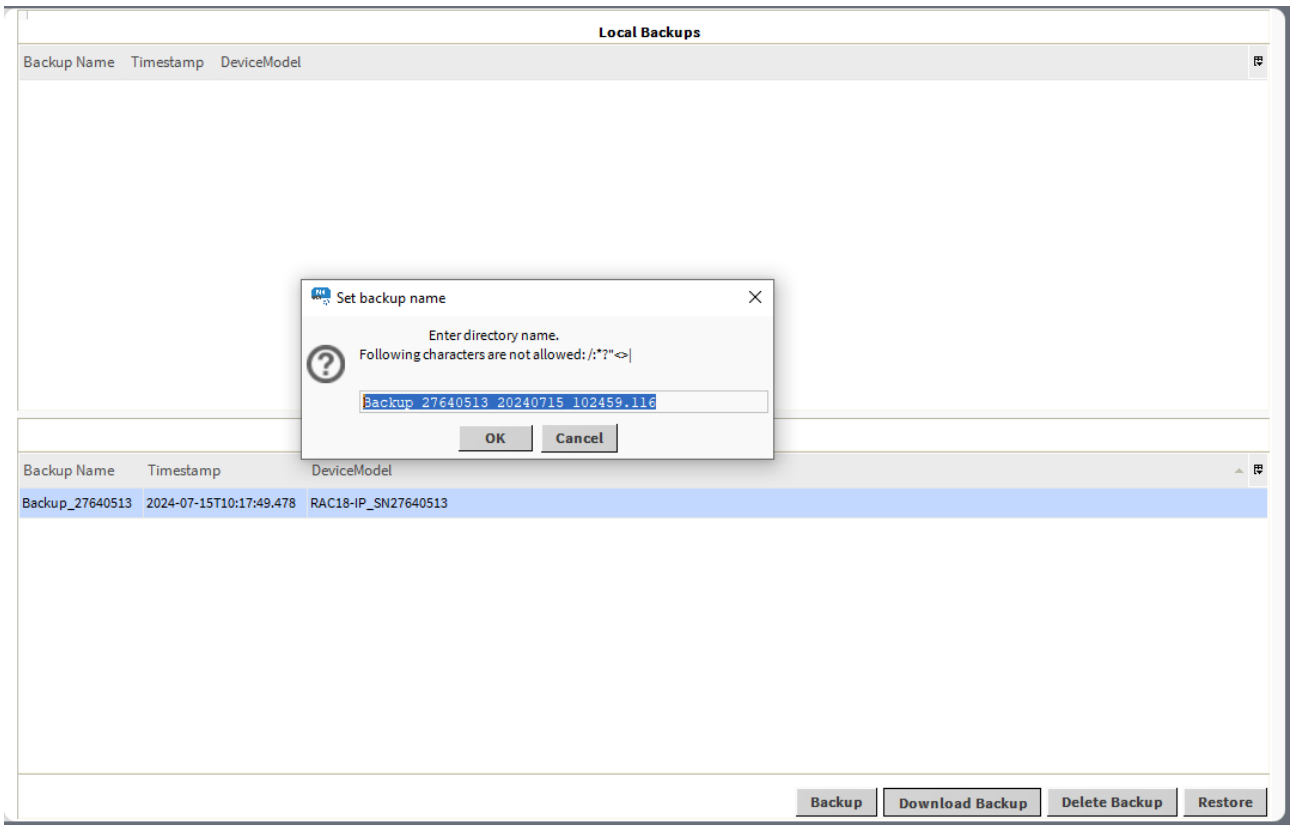


Figure 157. Changing backup name

Note

The Set backup name pop-up informs about characters that are not allowed to be included in a backup's name. If one of them is, the action will be aborted and the following prompt will be displayed:

Figure 158. Invalid characters included in the backup's name

A pop-up will appear informing about the completed download process.

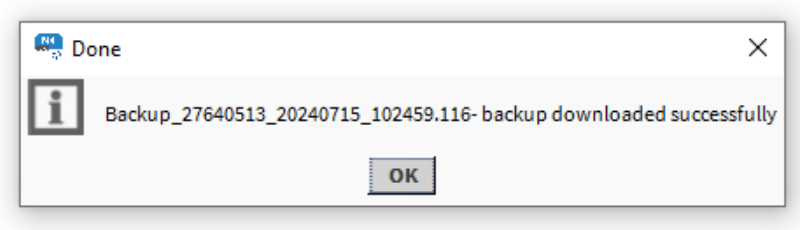


Figure 159. Successful backup confirmation

Once the process is completed, the backup will be visible in the Local Backups table in the Backup Manager view.

All local backups are stored in the local Niagara station. To access backups go to Station → Files → nE2Link → nanoEdgeEngine ->backup → *BackupName.* Backups can be imported or exported from this location manually and will become visible in the Backup Manager view.

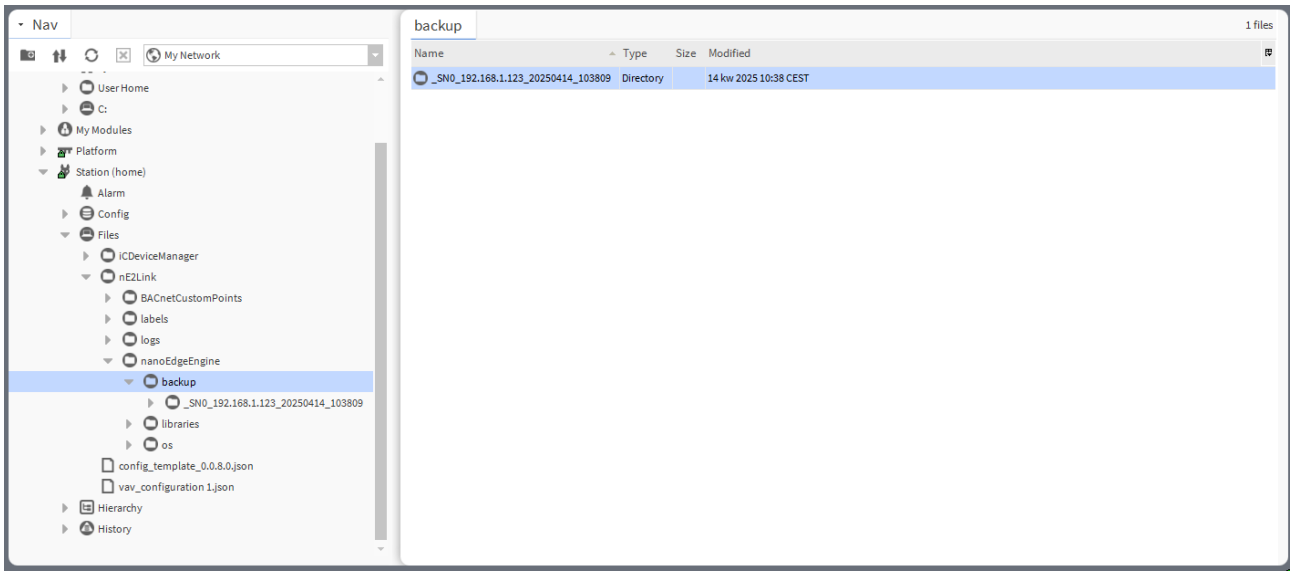


Figure 160. Backup stored in the station files

Restoring Backup

The backup can be restored to the device. To perform the restore function, select the backup to be restored to the device and click the Restore button.

A pop-up will appear with available containers that should be restored to the device. Select the proper configuration and confirm with the OK button.

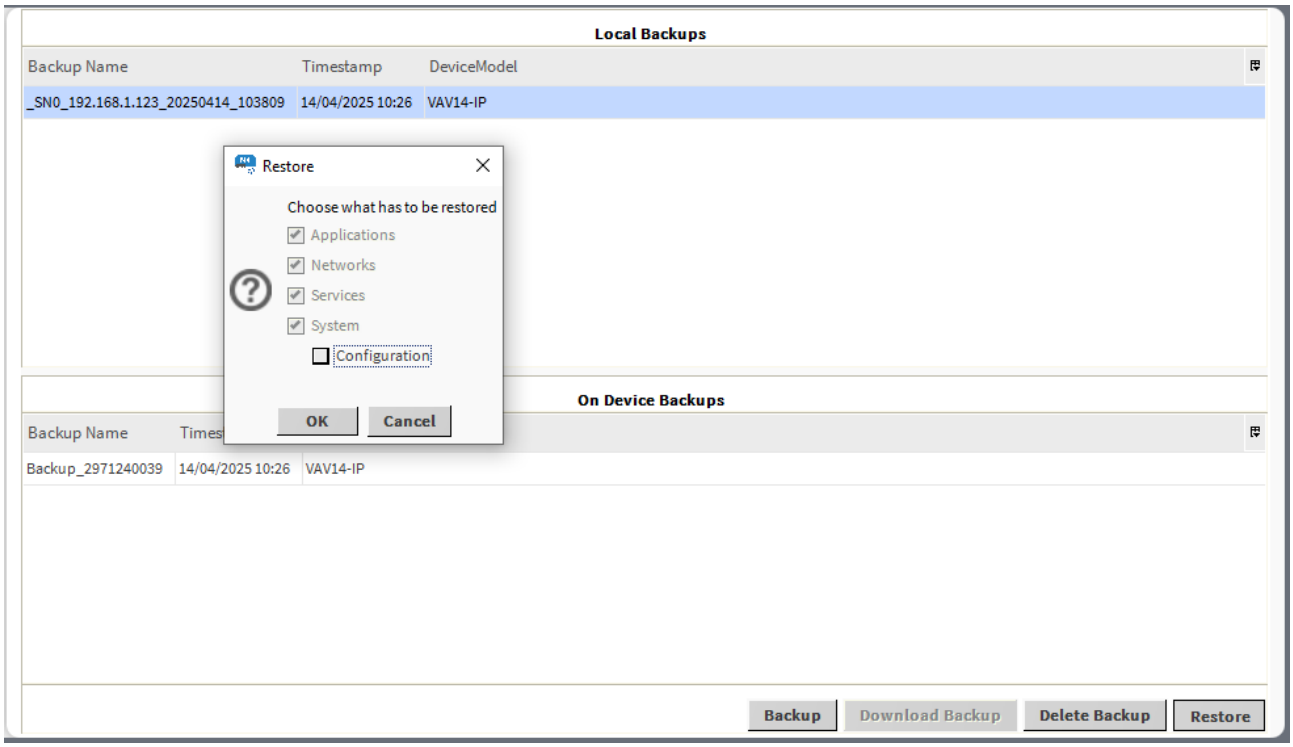


Figure 161. Available restore options will appear after clicking on the Restore button

Warning!
Restoring backup will overwrite the existing application loaded on the device.

A pop-up window will appear to confirm restoring of the backup. Click Yes to start the restoring process.

Please wait until the end of the process a pop-up will inform the user about the process in progress.

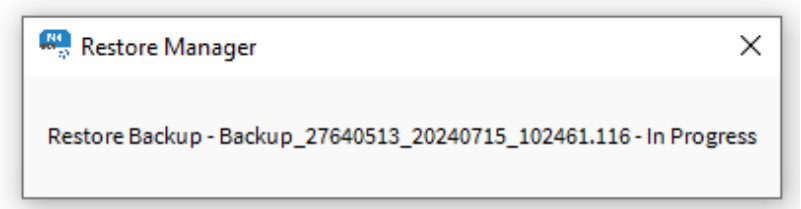


Figure 162. Restoring in progress

Once the process is finished, a pop-up will appear informing that the device has been disconnected. Right-click on the nE2DeviceExt, go to Actions → Connect to reconnect with the device.

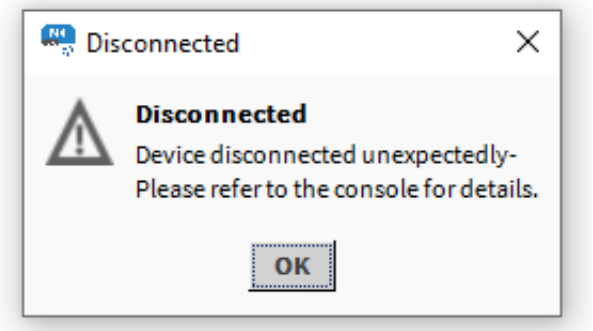


Figure 163. Device disconnected after restoring backup

Step 3: Programming

nano EDGE ENGINE Libraries

nano EDGE ENGINE enables real-time device programming using components from libraries installed on the device. Application programming on the controller is done in the Applications container. The nano EDGE ENGINE controller can run multiple applications in different time cycles, running simultaneously.

To learn more about the Applications, please refer to the nano EDGE ENGINE Programming user manual.

To start programming, make sure that the required libraries are installed on the device.

To learn more about the nano EDGE ENGINE libraries and components, please refer to the nano EDGE ENGINE Programming user manual.

The user can program the nano EDGE ENGINE device using installed libraries and components found in the Libraries folder. In the Application container, basic components can be added by right-clicking on the Application/Equipment.

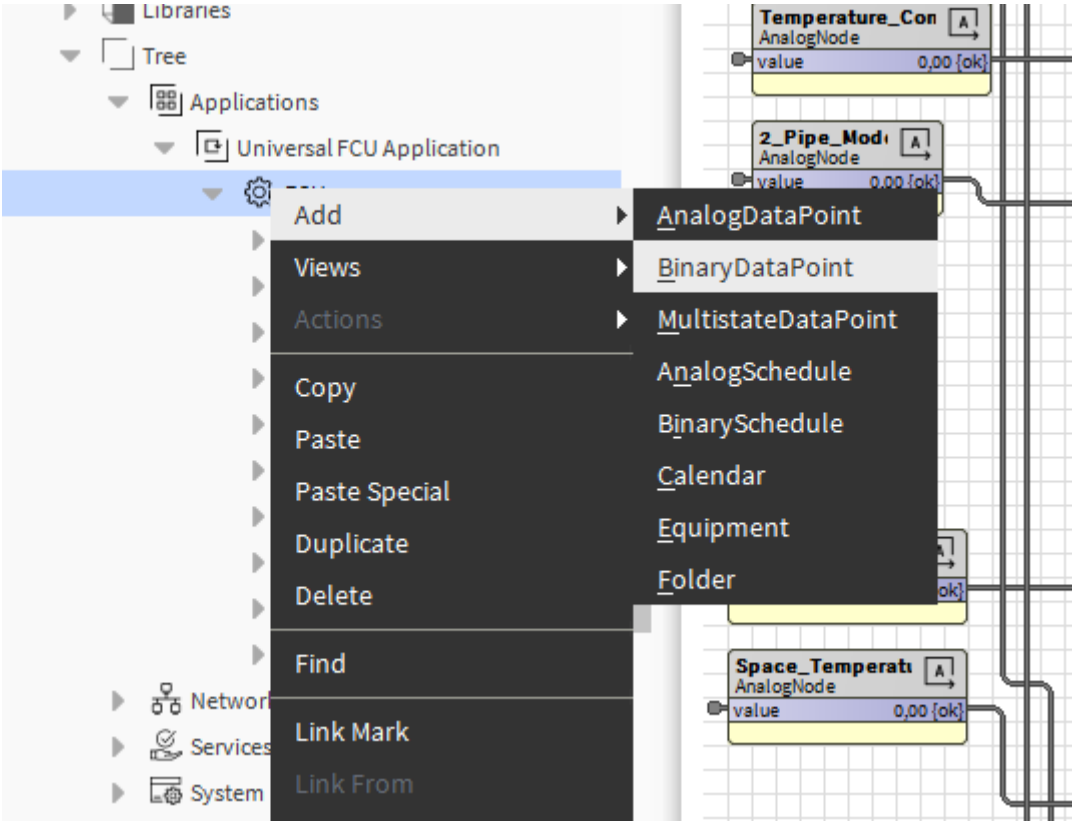


Figure 164. Context menu for adding basic components

The libraries on the device act as a palette of components that can be dropped into the device logic. If the user does not have a dedicated module with nano EDGE ENGINE libraries, it is possible to use those installed on the device. To enhance the user experience, it is recommended to open an additional nav view for the Libraries view:

- in Workbench, go to Window → Sidebar → Nav. Select Nav.

A new Nav view will appear on the left bottom side of the Workbench view. Navigate to the nE2DeviceExt, right-click on the Libraries folder, and select Go Into.

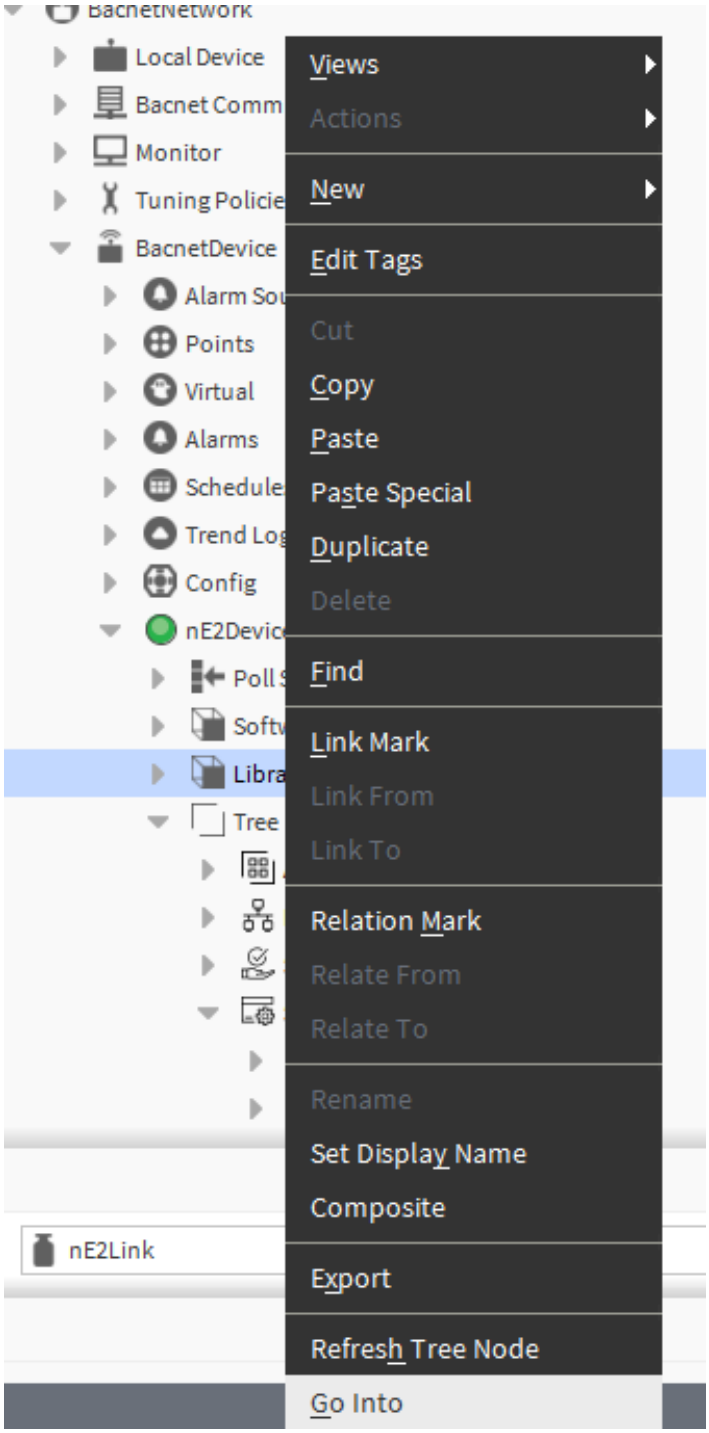


Figure 165. Go Into action in Libraries

An on-device libraries list will appear.

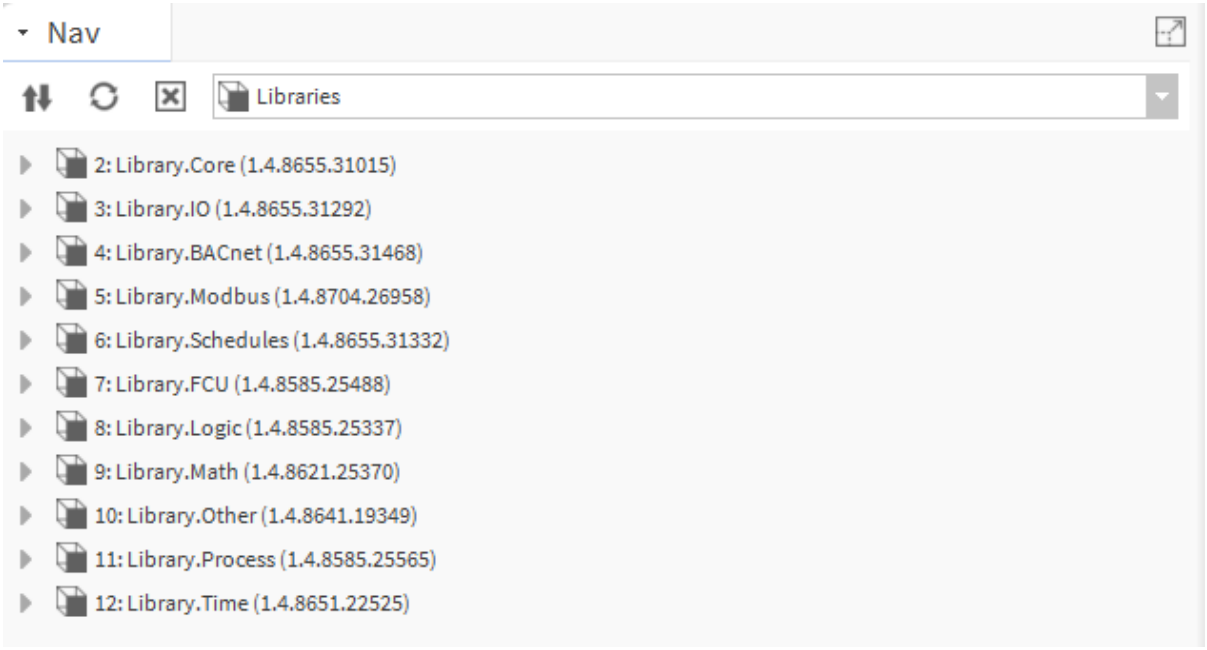


Figure 166. On-device libraries in the Nav view

- Application
- Equipment
- Data Points
- Services
- Local IO
- Linking

Application

The Applications container allows to add multiple Application components for building independent user applications, which are cycle-driven and may work simultaneously.

The user may define the application purpose (heating, lighting, etc.) and a cycle time of algorithms operation (cycles may differ between applications).

To create a first application, drag and drop the Application component from Library.Core to the Applications container and name it as appropriate.

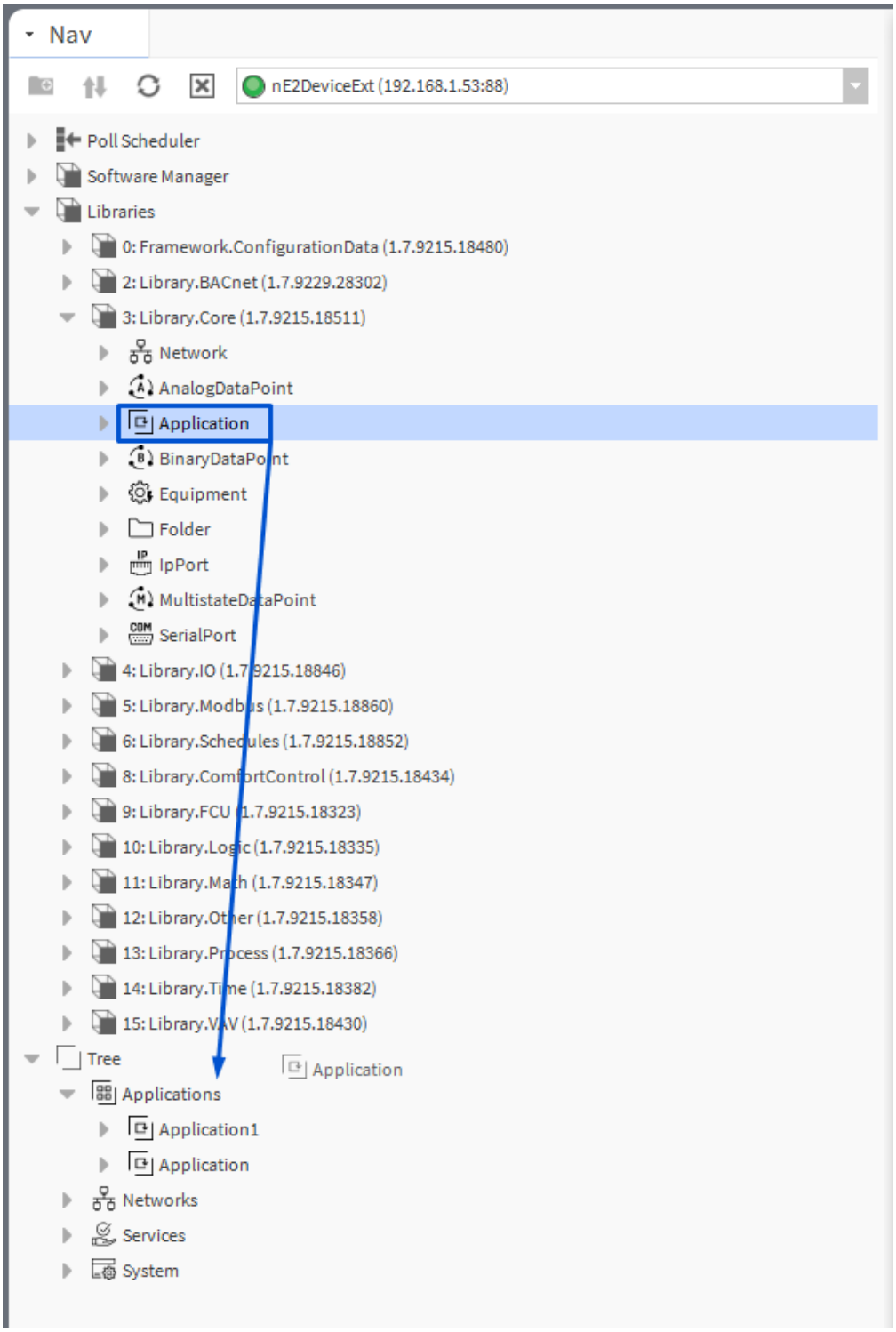


Figure 167. Drag and drop the Application component to Applications

Go to the Application AX Property Sheet. Configure the scanPeriod slot and click Save.

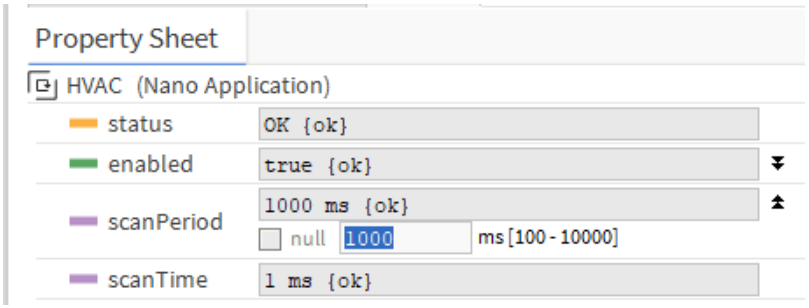


Figure 168. Setting the application's scan period

To learn more about the Application component, please refer to the nano EDGE ENGINE Programming user manual.

Folder

The Folder component is a grouping component, which can be used to gather other components to help organize the tree. The Folder component can be used both in the Applications and Networks containers, however, it cannot be added directly to the container. The Folder component can be freely renamed to facilitate categorization of components included within.

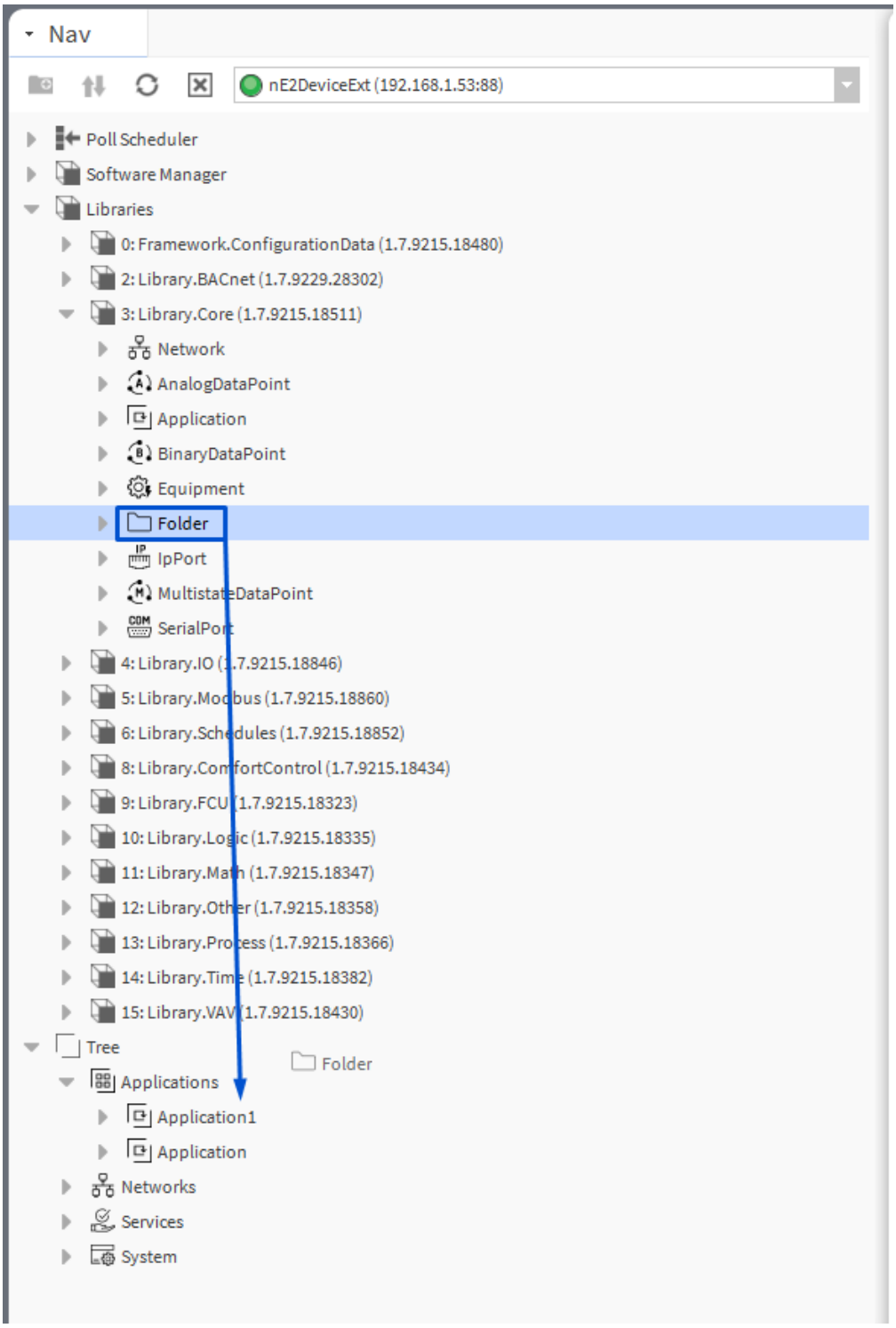


Figure 169. Drag and drop Folder to the Application

It is also possible to add the Folder component directly from the context menu:

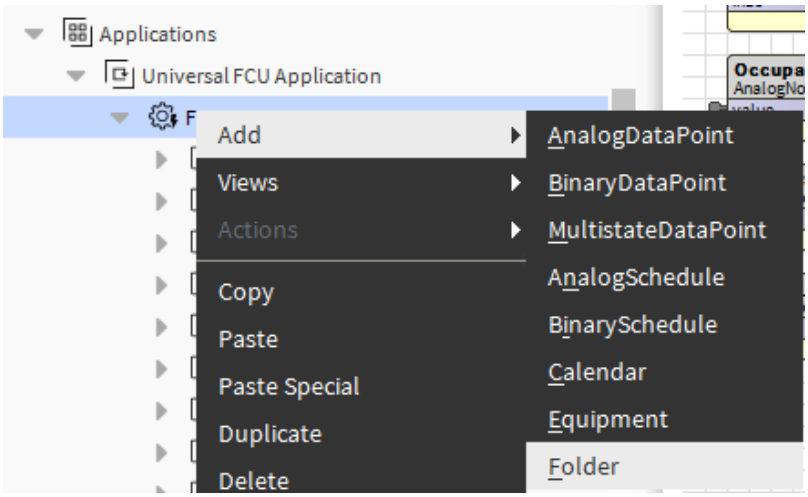


Figure 170. Adding the Folder component from the context menu

To learn more about the Folder component, please refer to the nano EDGE ENGINE Programming user manual.

Equipment

Double-click the Application component (below, renamed as HVAC, as a reference to the purpose of the application), the wire sheet opens. Drag and drop the Equipment component to the wire sheet, and name the component as appropriate (here, FCU).

The Equipment component is a grouping folder-type component, which can be used to gather other components, regarding specific equipment included in the Application, to help organize the Tree. It can be freely renamed to facilitate categorization of components included within.

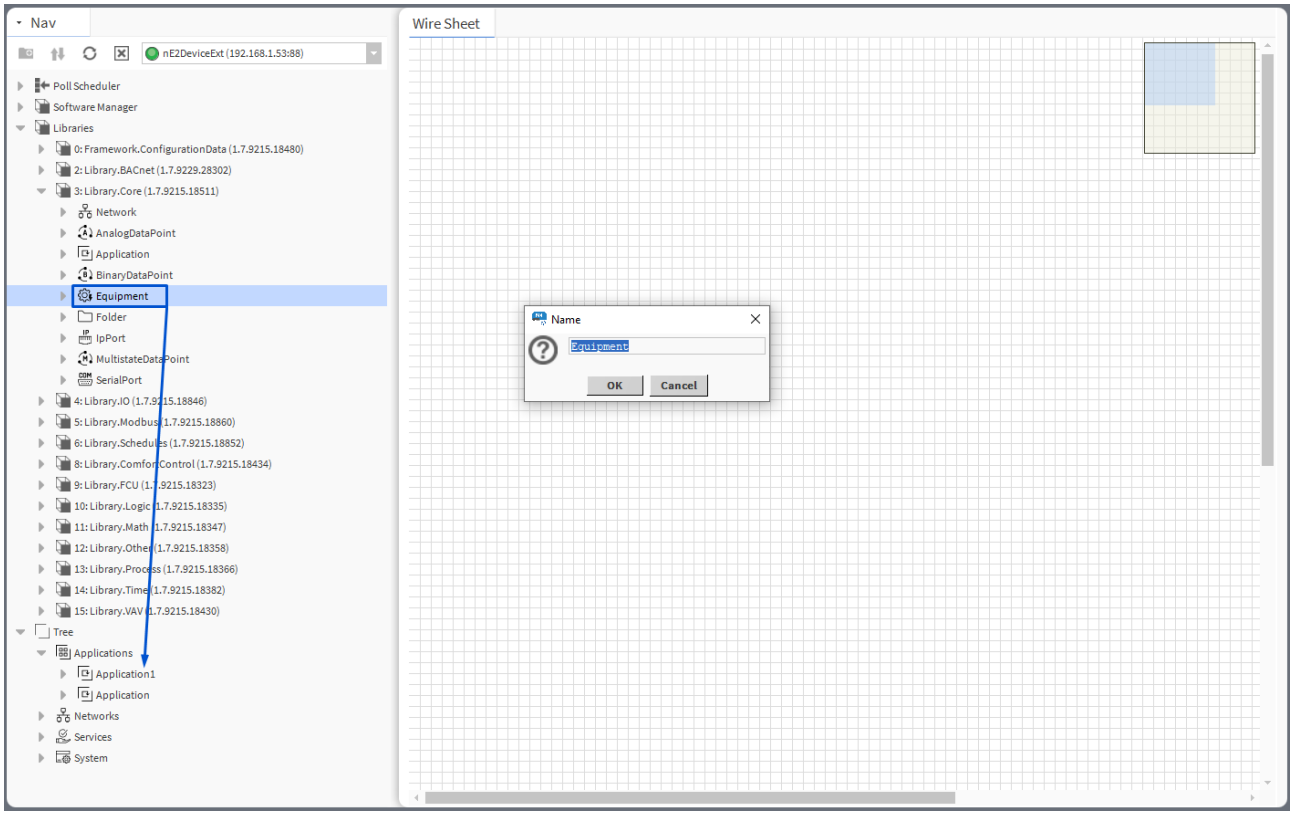


Figure 171. Add (drag and drop) and rename the Equipment component

It is also possible to add the Equipment component directly from the context menu of the Application component:

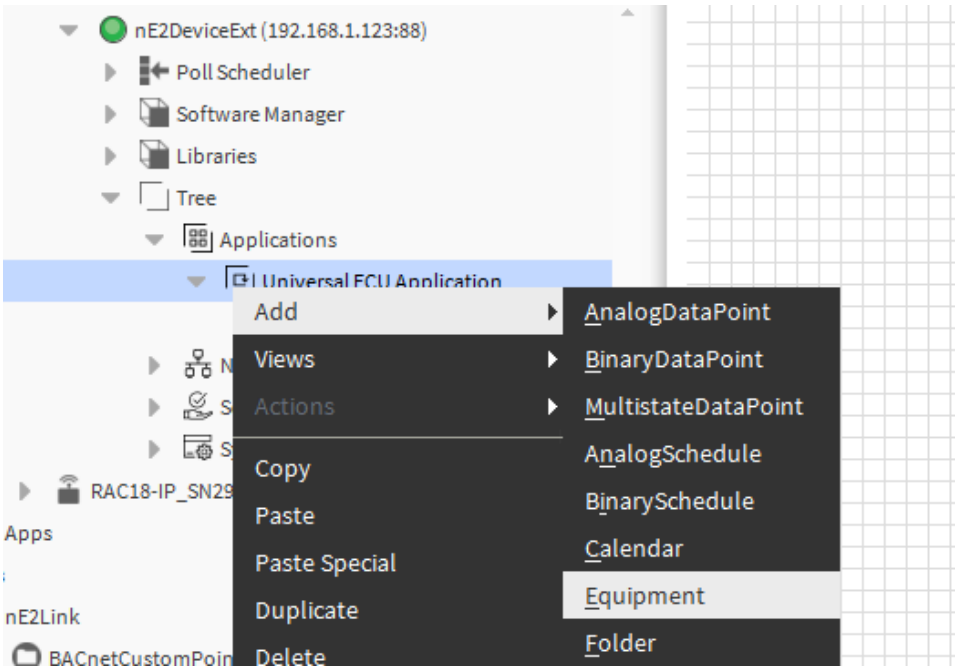


Figure 172. Adding the Equipment component from the context menu

To learn more about the Equipment component, please refer to the nano EDGE ENGINE Programming user manual.

Data Points

Data Points are universal components that represent a value in the application logic. The available Data Points:

- AnalogDataPoint with native BACnetAnalogPoint and ModbusAnalogPoint extensions;
- BinaryDataPoint with native BACnetBinaryPoint and ModbusBinaryPoint extensions;
- MultistateDataPoint with native BACnetMultistatePoint and ModbusMultistatePoint extensions.

To learn more about Data Points, please refer to the nano EDGE ENGINE Programming user manual.

Adding Data Points

To add a Data Point to the application, drag and drop the relevant component (AnalogDataPoint, BinaryDataPoint, or MultistateDataPoint) from the Library.Core to the application.

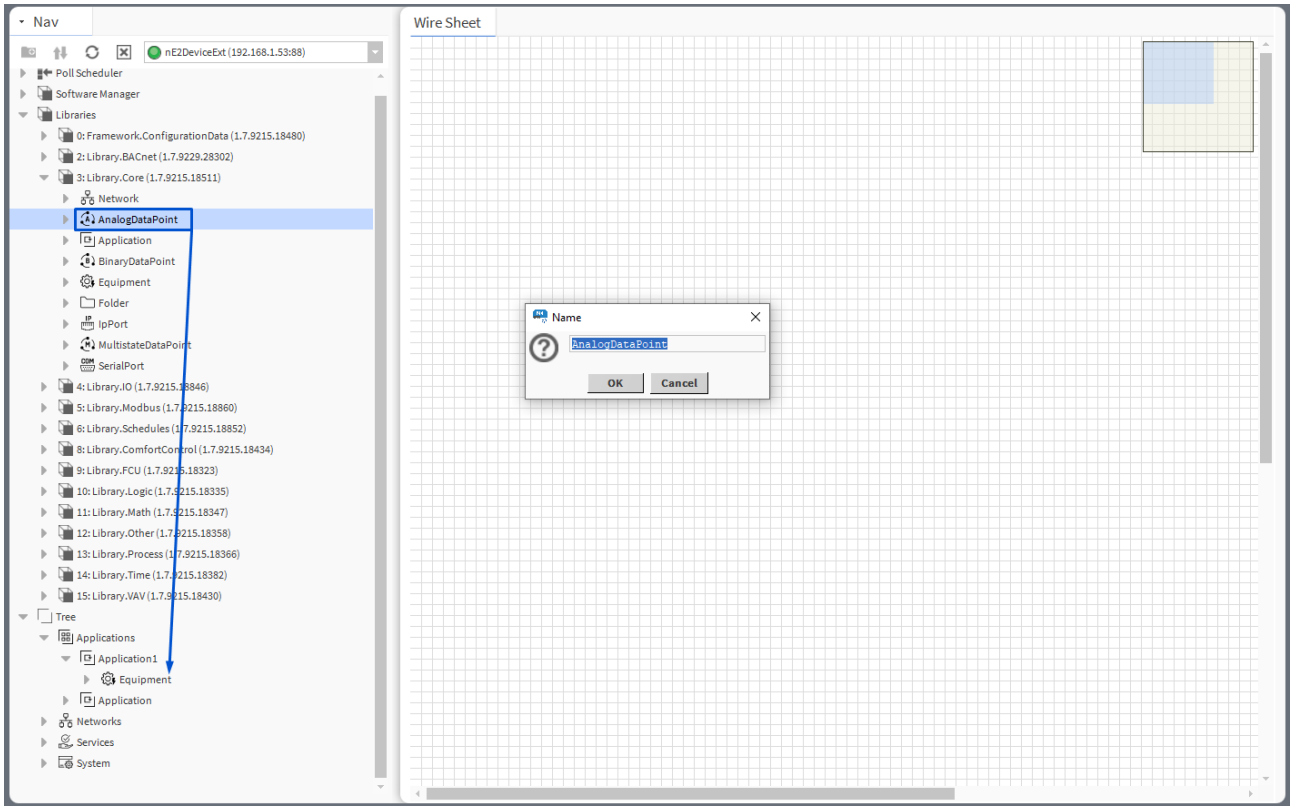


Figure 173. Add (drag and drop) and rename Data Points

It is also possible to add Data Points directly from the context menu of the Application component:

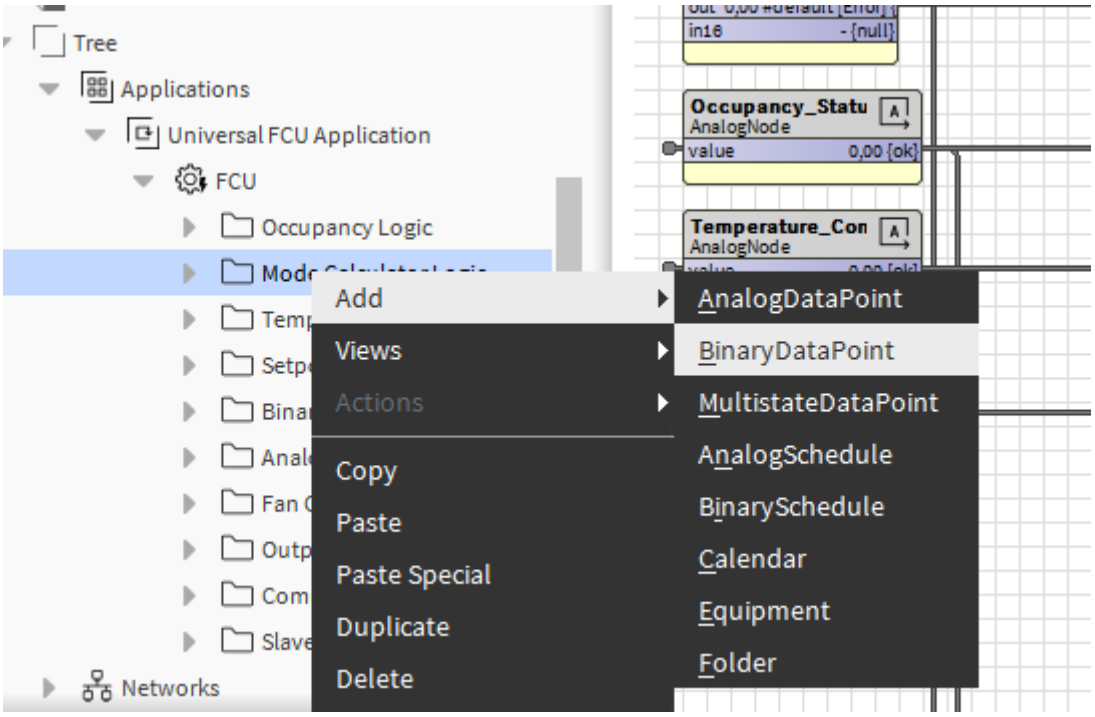


Figure 174. Adding Data Points directly from the context menu

Each Data Point has 3 actions available from the Actions menu. Actions are related with the type of the Data Point and its extensions.

- **Set:** allows entering a value to set the In16 slot;
- **SetId:** sets a BACnet object Id of the Data Point (exposed in the BACnetPoint extension);
- **SetAddress:** sets a Modbus address of the Data Point (exposed in the ModbusPoint extension).

All slots and options are described in the nano EDGE ENGINE Programming user manual: AnalogDataPoint, BinaryDataPoint, MultistateDataPoint.

Data Points Configuration

To configure the added Data Point, go to its Property Sheet. Standard Data Point slots will be visible. The Property Sheet view allows to configure the following parameters:

- mode;
- units;
- extensions;
- other points available for the selected component.

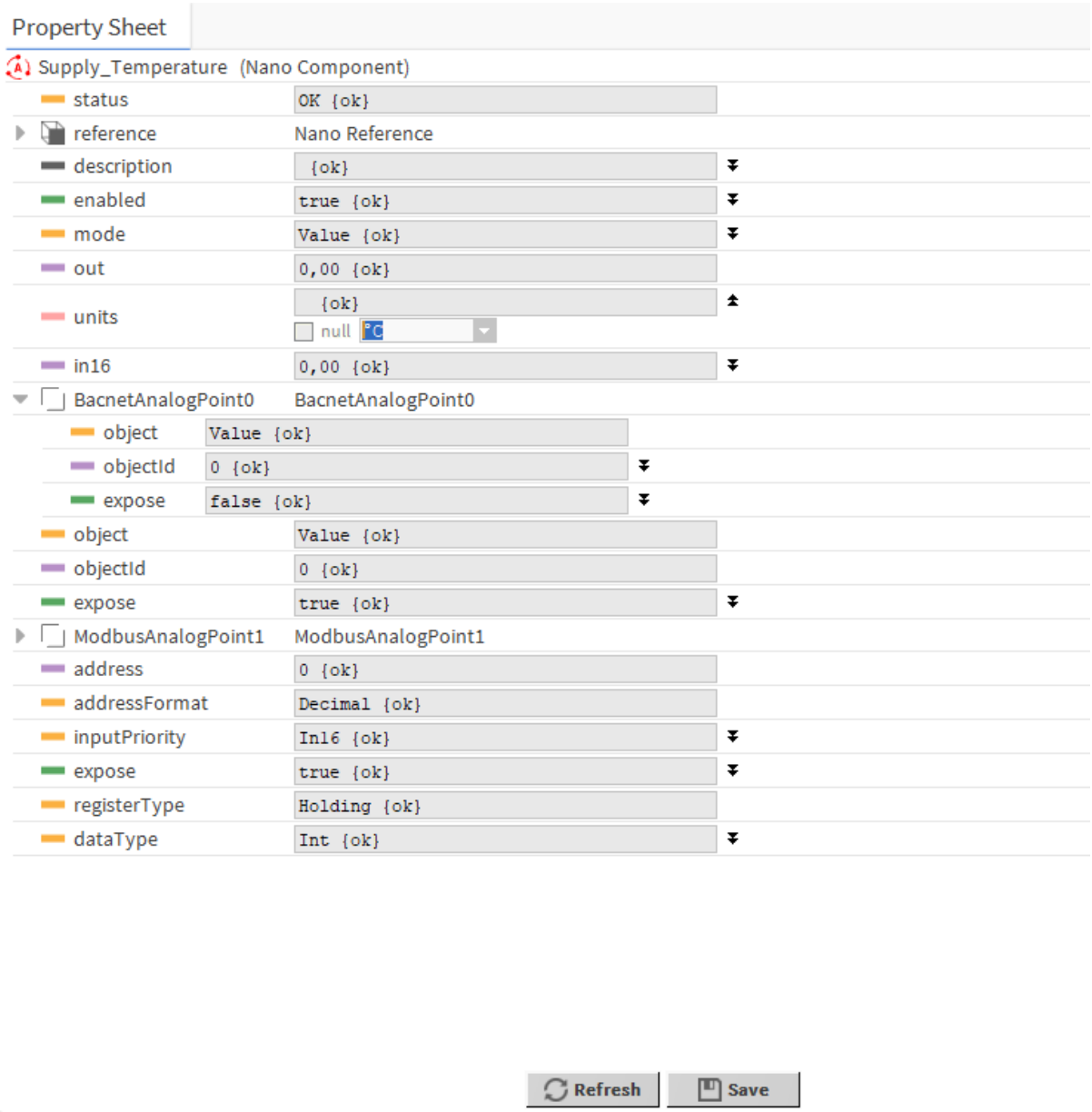


Figure 175. Data Point's Property Sheet view

Adding Extensions

Data Points can have their functionality enhanced by extensions. For example, the AnalogDataPoint is originally equipped with the BACnetAnalogPoint and ModbusAnalogPoint extensions (these cannot be added or removed), but other extensions, which offer different functionalities, can be added or removed as necessary.

Extensions are added from the context menu, select the Add Extension option; add the extension from the list of available options.

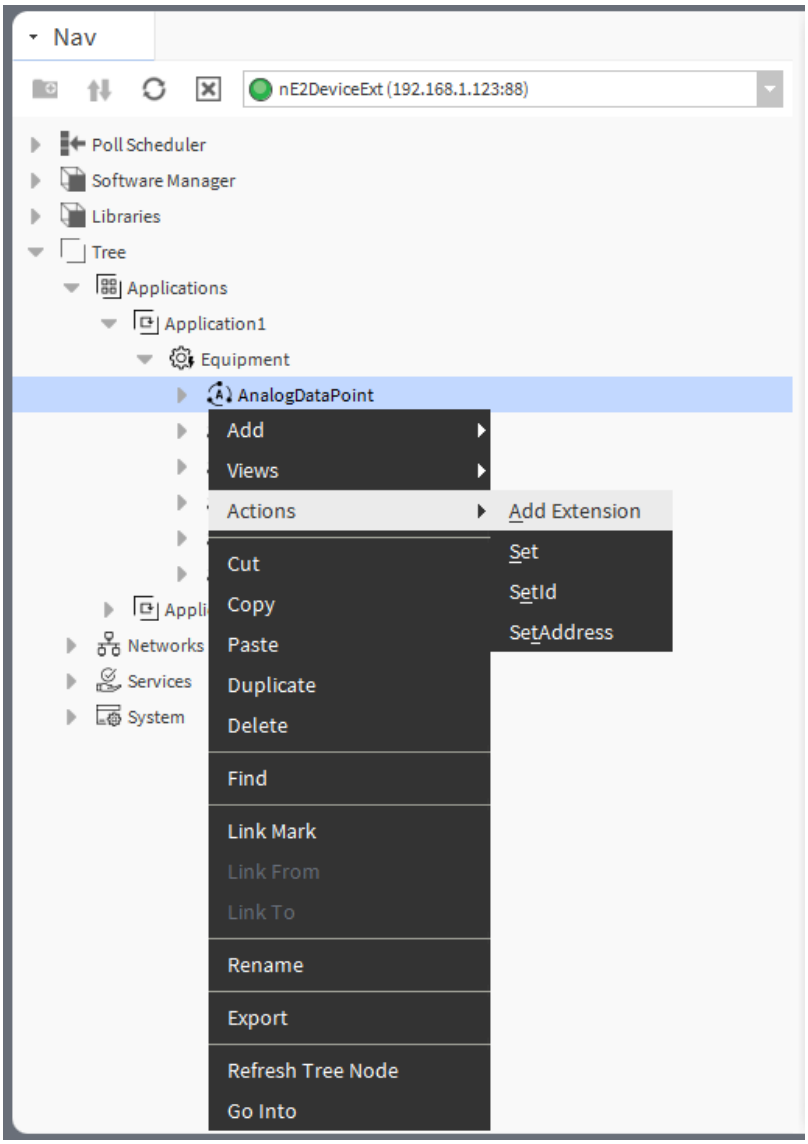


Figure 176. Add Extension action

Available Extensions

BACnetPoint

The BACnetAnalogPoint/BACnetBinaryPoint/BACnetMultistatePoint extension expands the Data Point's functionality giving it an option to expose it to the BACnet IP network as an Analog/Binary/Multistate Value object, and, otherwise, it allows to manually hide it from the network yet preserving its function in the application. It also transfers information to the BACnet IP network about the Data Point's status. The extension is native (cannot be removed), and is visible along with the regular slots and actions of the Data Point as a separate, integral part in the property sheet view.

The extension has the following slots:

- **Object:** a read-only slot showing a type of BACnet object attributed to the extension;
- **ObjectID:** a BACnet object ID, which is automatically numbered from 0 up;
- **Expose:** enables the Data Point to be recognized within the BACnet IP network;
  - Available settings: true (exposed), false (hidden).

ModbusPoint

The ModbusAnalogPoint/ModbusBinaryPoint/ModbusMultistatePoint extension expands the Data Point's functionality giving it an option to expose it to the Modbus TCP/IP network as a Modbus point, and,

otherwise, it allows to manually hide it from the network yet preserving its function in the application. It also transfers information to the Modbus TCP/IP network about the Data Point's status. The extension is native (cannot be removed), and is visible along with the regular slots and actions of the Data Point as a separate, integral part in the property sheet view.

The extension has the following slots:

- Address: a read-only slot showing a Modbus register, which the Data Point is exposed on;
- Address Format: a read-only slot showing a register address format;
  - Available information: decimal, Modbus, HEX;
- Input Priority: allows to select the input number in the Data Point, which the value from the register is synchronized on;
- Expose: enables the Data Point to be recognized within the Modbus TCP/IP network;
  - Available settings: true (exposed), false (hidden);
- Register: a read-only slot showing the type of the register used;
  - Available information: holding register;
- Data Type: allows to select a value data type;
  - Available settings: integer (default), signed integer, long, signed long, float, double.

Configuration Data

The Configuration Data extension has no slots. Its functionality is fully achieved by adding it to the Data Point. It is automatically enabled and allows the Configuration Data service to save and upload slots values of the Data Point.

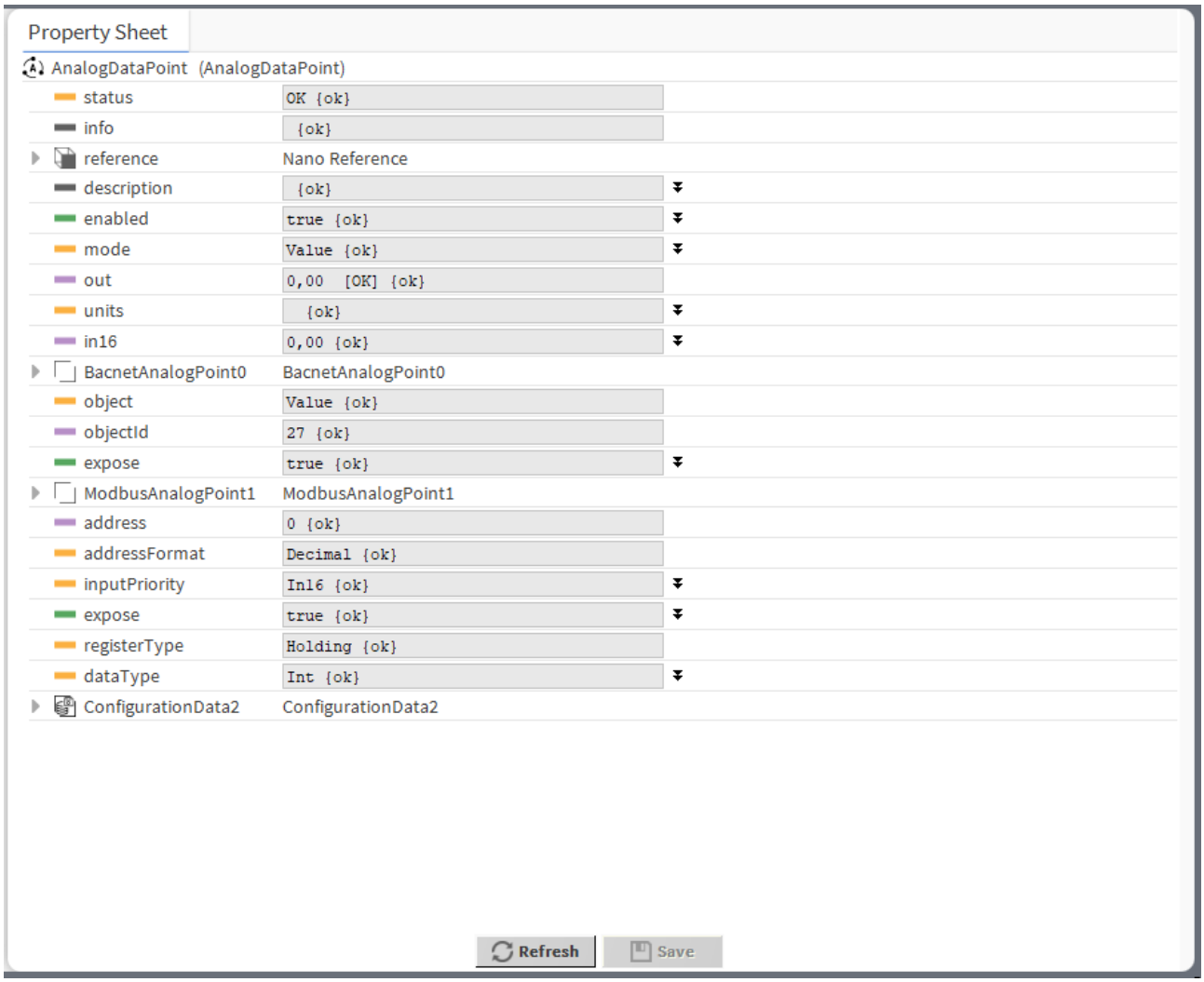


Figure 177. Configuration Data service extension added to the AnalogDataPoint

Priorities

The AnalogPriorities/BinaryPriorities/MultistatePriorities extension adds fifteen writable input slots and the default (lowest) priority slot to the Data Point. The extension includes the Priority slot indicating, which slot is transferring value to the Out slot. The Priorites extension adds In1–In15 slots and the Default slot, which is the lowest, 17th priority. The extension also introduces new actions to the Data Point: EmergencyOverride, EmergencyAuto, Override, and OverrideAuto.

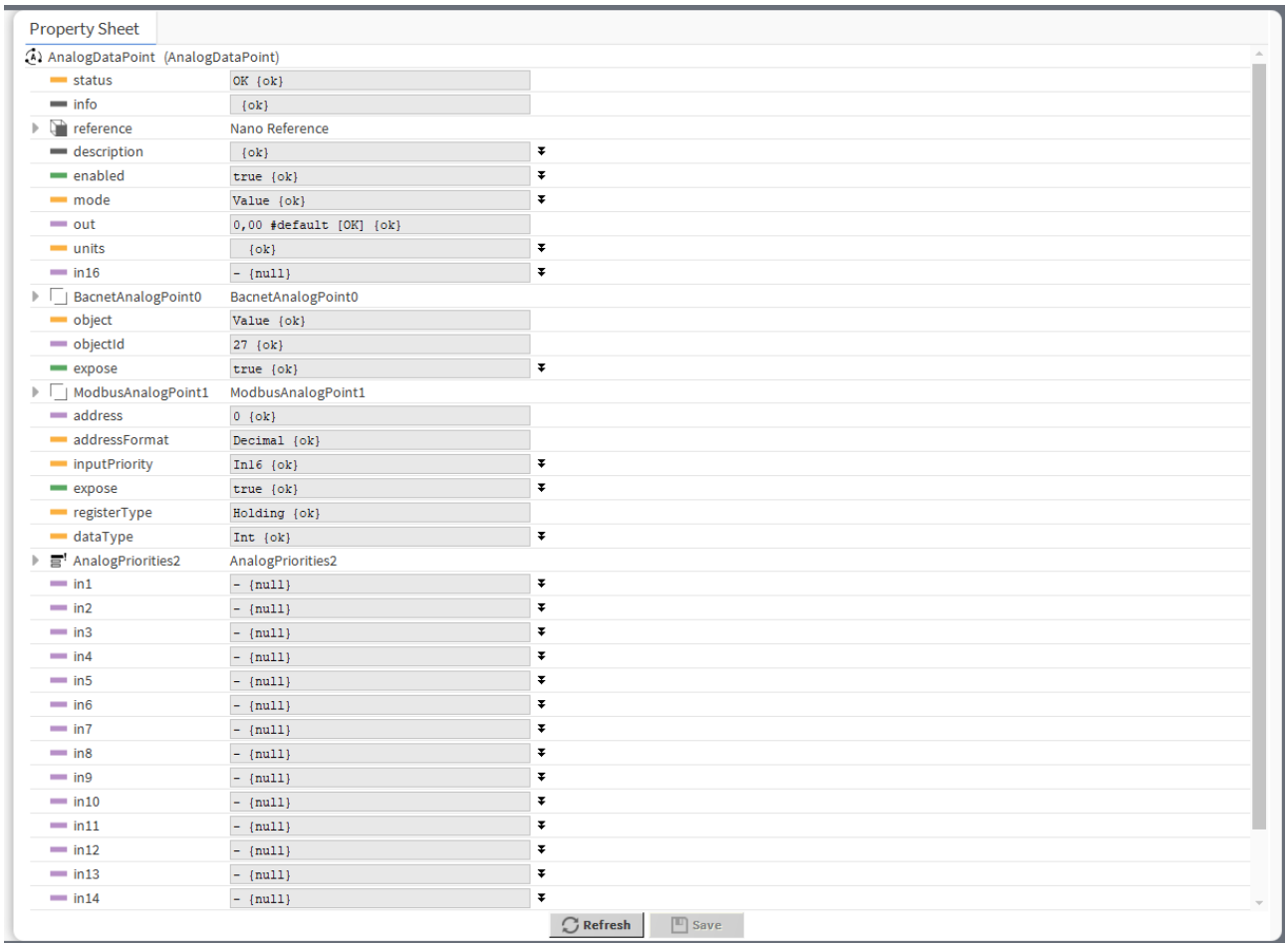


Figure 178. Priorities extension added to the AnalogDataPoint

The extension has the following slots:

- **In1-In15:** input slots providing values to the Out slot (from 1 to 16, the highest priority is In1); only the highest priority value is provided to the Out slot, the rest is dismissed. All input slots are linkable. In the extended mode, the In1 and In8 slots have actions available for overriding their values.

Note: By default, only the In16 is displayed in the Wire Sheet. In case any other input slot receives a value via link, is it displayed in the Wire Sheet along with the In16. Only the null input, which is a lack of value, allows the higher priority input to be dismissed–zero (0) is still a value that will be provided to the Out slot.

- **Default:** the 17th, lowest priority input slot; allows to introduce a default value to the Data Point in case there are no links providing values from other components. If the value to the Data Point is provided by the Reference link, then the Default value is automatically dismissed (the Reference link cannot be directed to the 17th priority, only from the 16th up).

Note: According to BACnet requirements, the Default slot value can never be null; if no other value is set on the slot, it is zero (0).

- **Priority:** shows, which slot is currently providing the value to the Out slot.

The Data Point has the following actions available in the Priorities extension:

- **EmergencyOverride:** enables entering an analog value to the In1 slot;
- **EmergencyAuto:** sets the null value to the In1 slot (cancels the EmergencyOverride action);
- **Override:** enables entering an analog value to the In8 slot;
- **OverrideAuto:** sets the null value to the In8 slot (cancels the Override action).

Note: If the link is connected to the slot that may be affected by an action, the value coming from the link connection has priority over the manually evoked action.

ActionTrigger

The ActionTrigger extension is designed to invoke any action that is available for the component. The extension triggers an action selected in the Action Name on the rising edge of the Action Trigger slot. If the action has parameters to set, the parameter is taken from a relevant slot automatically added to the extension (Analog Value/Binary Value/String Value).

It is possible to add more than one ActionTrigger extension to the component (for example, one for each action in the component).

The extension is added from the context menu of the component.

The ActionTrigger extension has the following slots:

- **Action Name:** allows to select an action to invoke;
- **Action Trigger:** triggers an action selected in the Action Name slot;
- **Action Analog Value/Action Binary Value/Action String Value:** a slot added automatically to the extension if an action selected in the Action Name slot has any specific parameters to set (depending on the type of action and its parameters, the relevant type of value is matched).

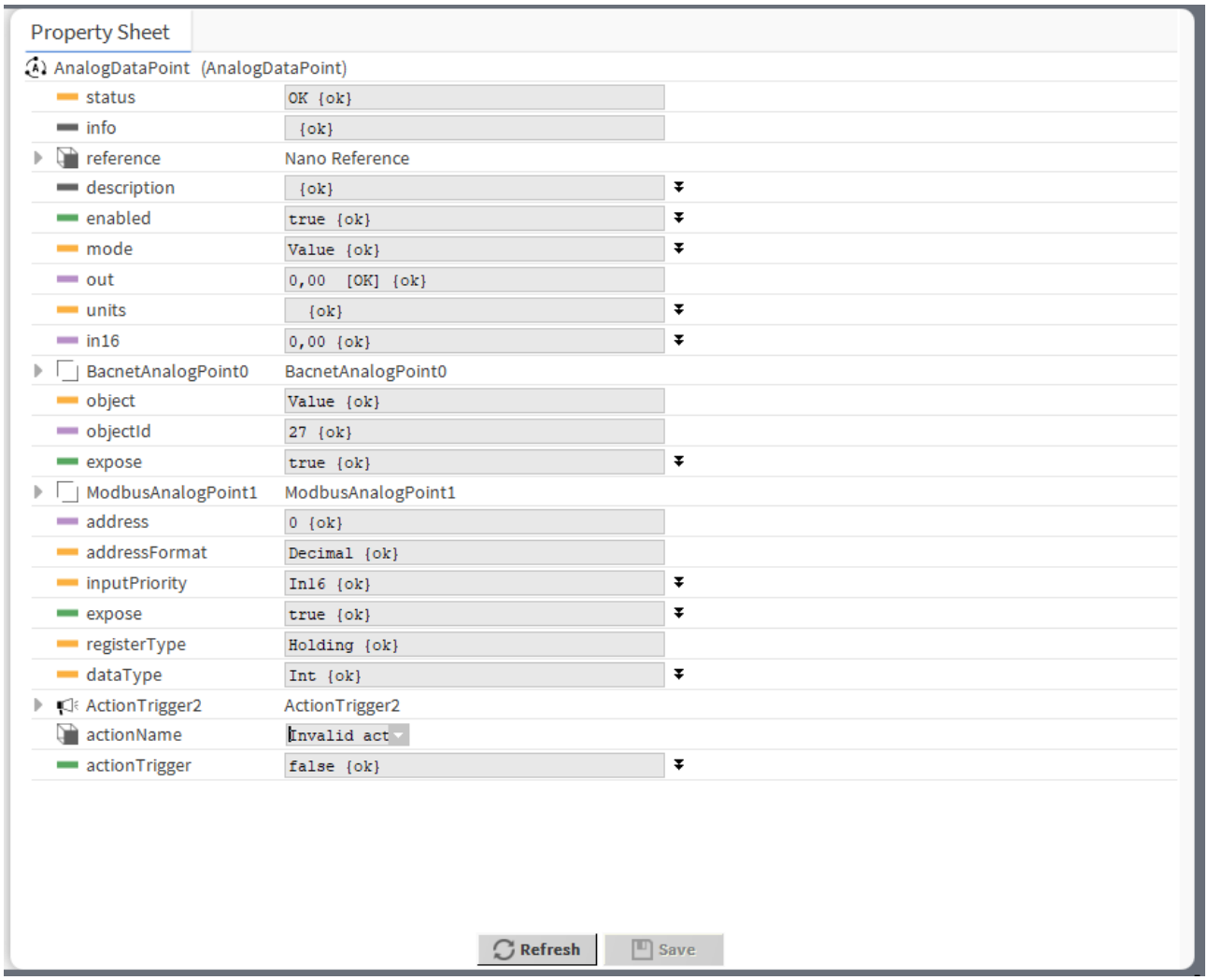


Figure 179. ActionTrigger extension added to the AnalogDataPoint

Services

The Services provide a space for additional services developed to enhance the device's functionalities. Services may be added to the device and then used within applications. They are designed to provide additional functionalities to the basic algorithms included in applications, allowing the device to communicate with systems superior to building automation systems.

- Configuration Data Service

Configuration Data Service

The Configuration Data is a service designed to save configuration settings of Data Points for the purpose of restoring them if changed or lost. The service is executed by adding the Configuration Data extension to Data Points:

- AnalogDataPoint;
- BinaryDataPoint;
- MultistateDataPoint.

The service functions as a backup mechanism for device-specific configurations – it can bring back saved values of Data Points within a single device. To transfer applications between devices, use Backups.

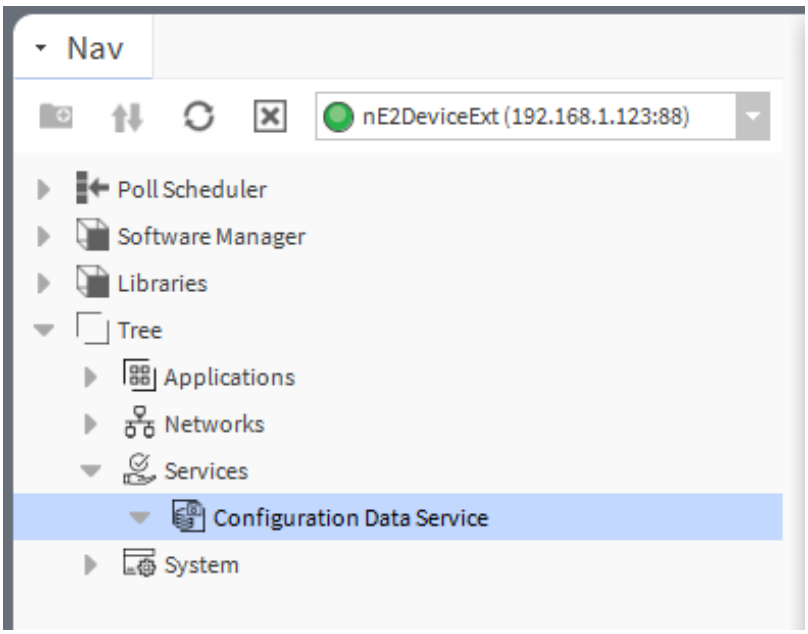


Figure 180. Configuration Data service in the nav tree

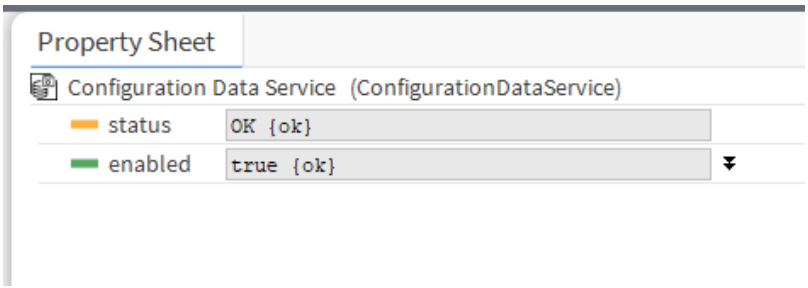
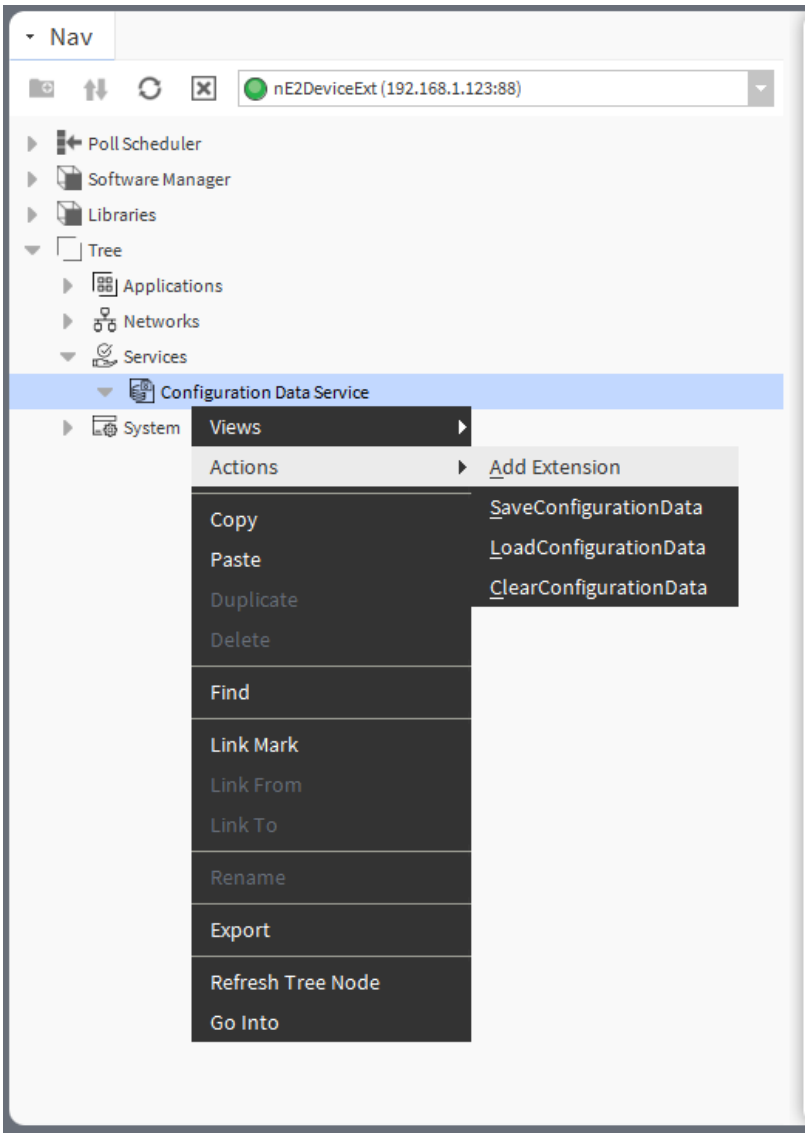


Figure 181. Configuration data service slots

The Configuration Data service has the following slots:

- **Status:** indicates the current status of the component. If the component works properly, its status is OK; however, it changes accordingly when values in other slots are adjusted.
  - Available information: disabled (the Enabled slot is set to false), OK;
- **Enabled:** change of the slot's value enables or disables the component.



Configuration Data Service Manager

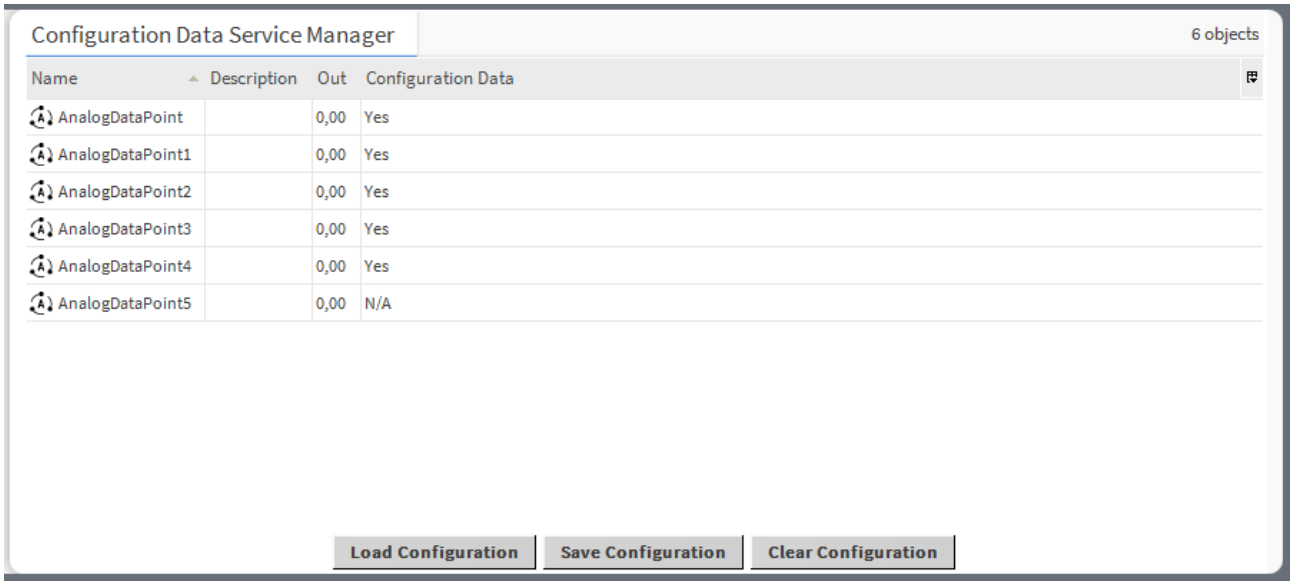


Figure 182. Configuration Data service manager

The Configuration Data service view is a simple table view showing which Data Points have the Configuration Data extension added, along with their Description and Out slots.

The Data Points marked N/A in the view are Data Points added to applications but without the Configuration Data extensions, as the service collects data only from Data Points with added extension. The view shows Data Points from all applications executed in the Applications container (if gathered under the Equipment component, double-click it to show Data Points). Additionally, it allows to export the gathered data with the Export option in the context menu.

The Configuration Data service has the following actions:

> **Note**
> Actions are executed for all applicable Data Points at once.

- **Save Configuration Data:** saves the slots values of Data Points with added Configuration Data extension to the controller's memory;
- **Load Configuration Data:** uploads the saved slots values to Data Points with added Configuration Data extension;

Note: The data can be loaded only to Data Points, which had the extension added at the point of saving the values.

- **Clear Configuration Data:** erases the saved slots values of Data Points with added Configuration Data extension.

> **Warning!**
> Remember that restoring default settings on the controller **by the 6th DIP switch** clears the values saved in the Configuration Data service too.

Configuration Data Extension

The Configuration Data extension has no slots. Its functionality is fully achieved by adding it to the Data Point. It is automatically enabled and allows the Configuration Data service to save and upload slots values of the Data Point.

Local IO

Configuration

To setup local inputs and outputs on the nano EDGE ENGINE device, expand the Library.IO. The IO components must be placed in the Networks container, under the LocalIO component.

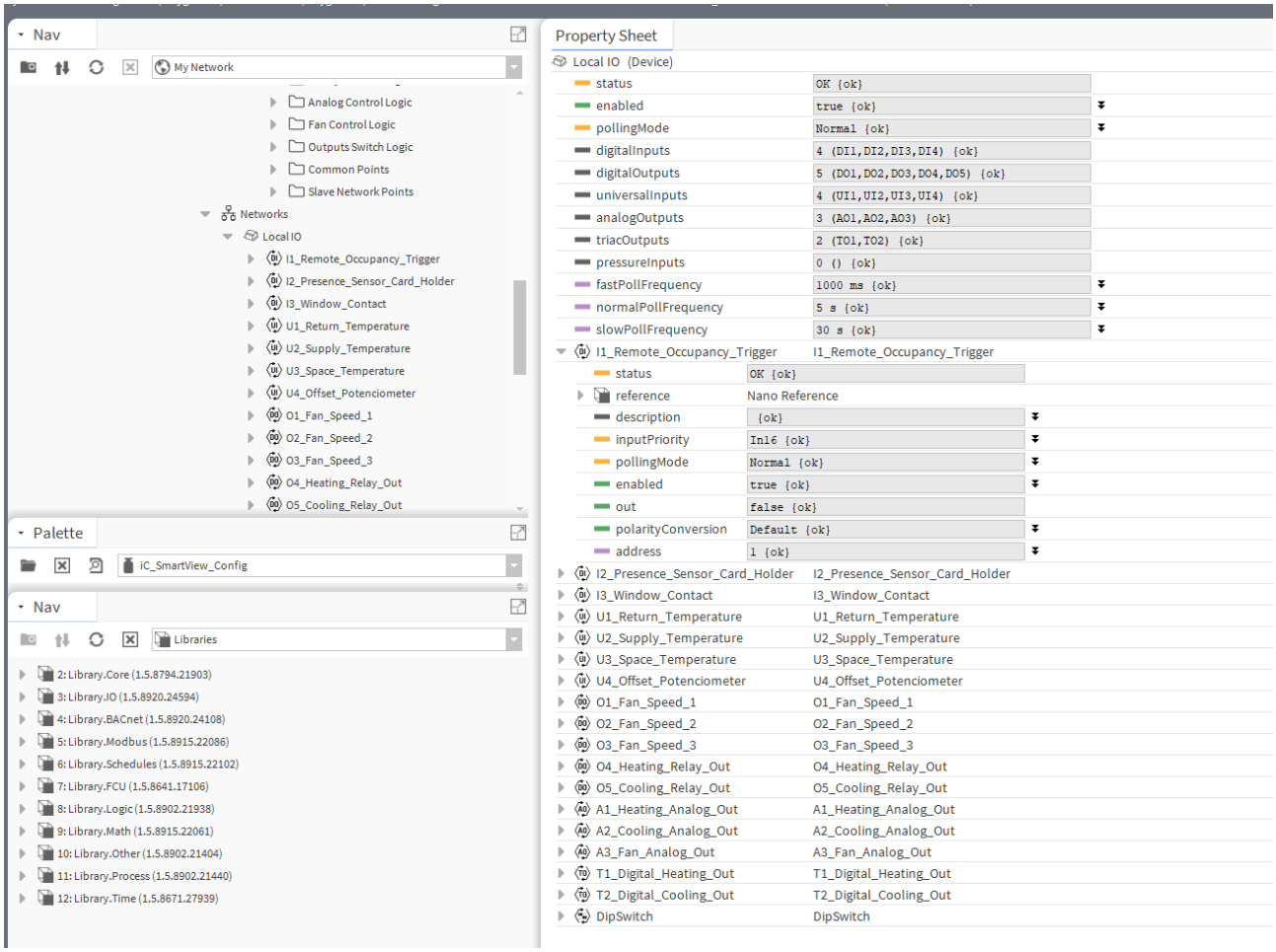


Figure 183. LocalIO components added to the Networks container

To learn more about the local IO components, please refer to the nano EDGE ENGINE Programming user manual.

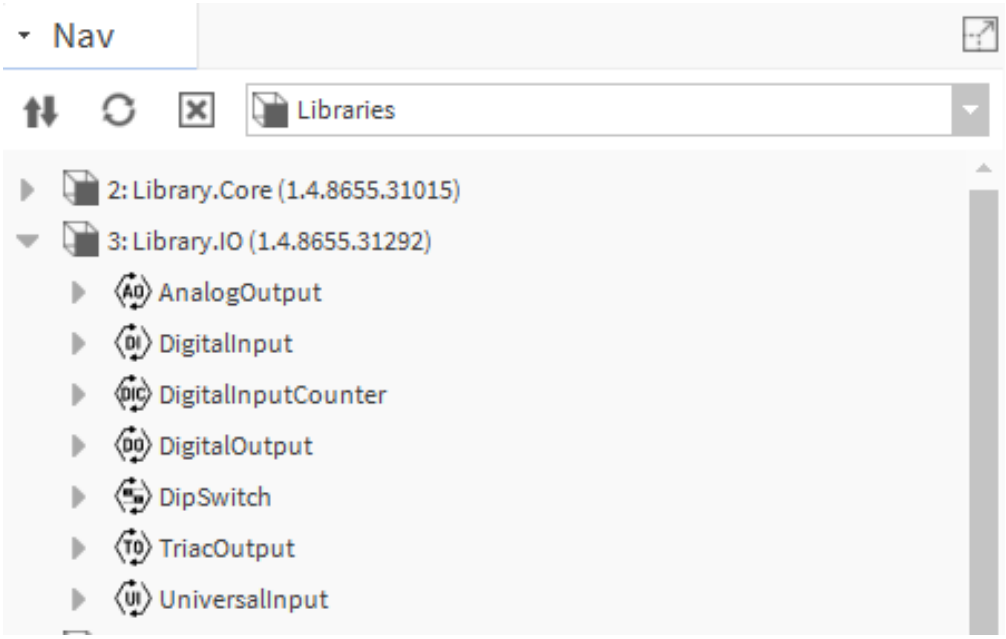


Figure 184. Library.IO

Drag and drop the desired IO component under the Local IO network.

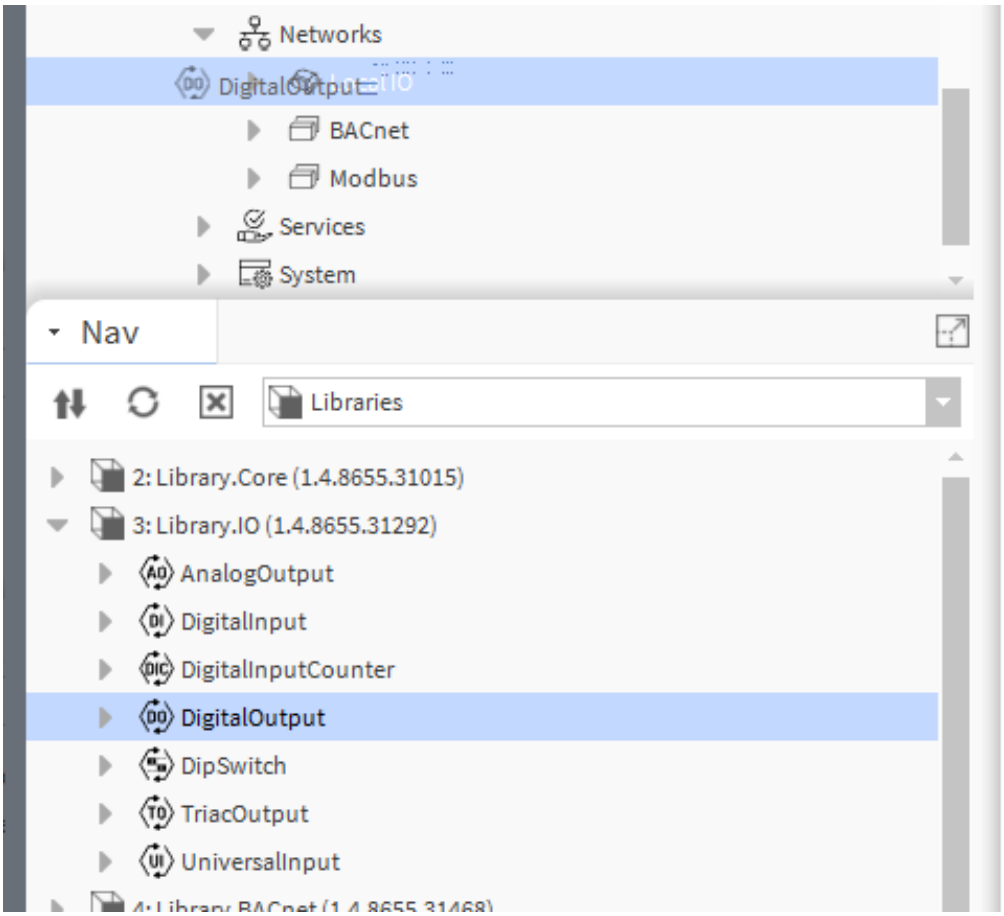


Figure 185. Adding DigitalOutput to the Local IO network

Double-click the IO point to open its property sheet. By default, the status of the point is Fault because the address must be set. Configure the point's address in the Address slot and click Save.

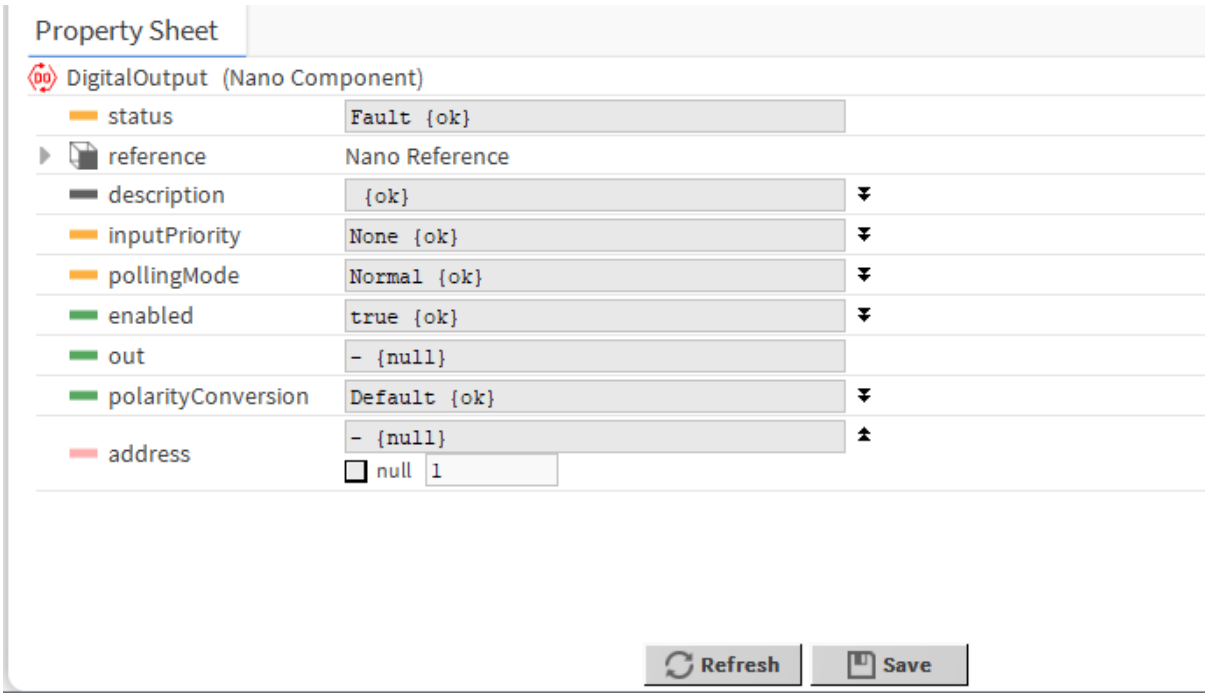


Figure 186. Setting the IO point address

Linking

In nE2DeviceExt, it is possible to link nano EDGE ENGINE components by dragging a wire from one component to another.

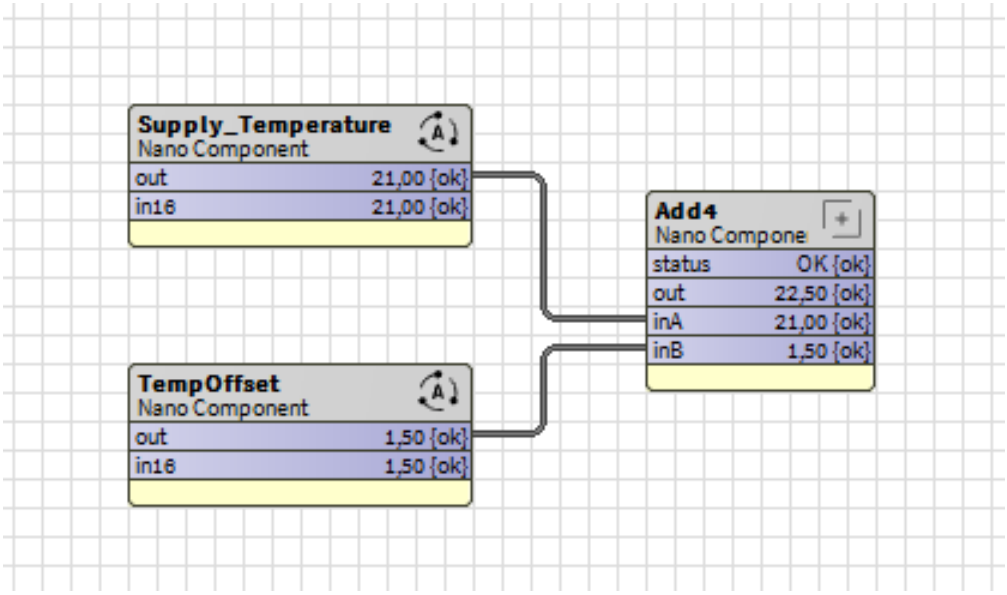


Figure 187. Linked components

Warning!

Linking in nE2DeviceExt works only between nano EDGE ENGINE components. Do not link Niagara components from other modules/drivers with nano EDGE ENGINE components.

Reference Linking

Reference link is designed specifically to connect Data Point class components (in the Applications container) with network point class components (in the Networks container). A reference link transfers values along with the component's status.

To learn more about the reference linking, please refer to the nano EDGE ENGINE Programming user manual.

As network points are situated in the Networks container and Data Points are situated in the Applications container, Reference links are created using the Link Mark and Link From options from the context menu.

To create a reference link, right-click a network point and select the Link Mark option.

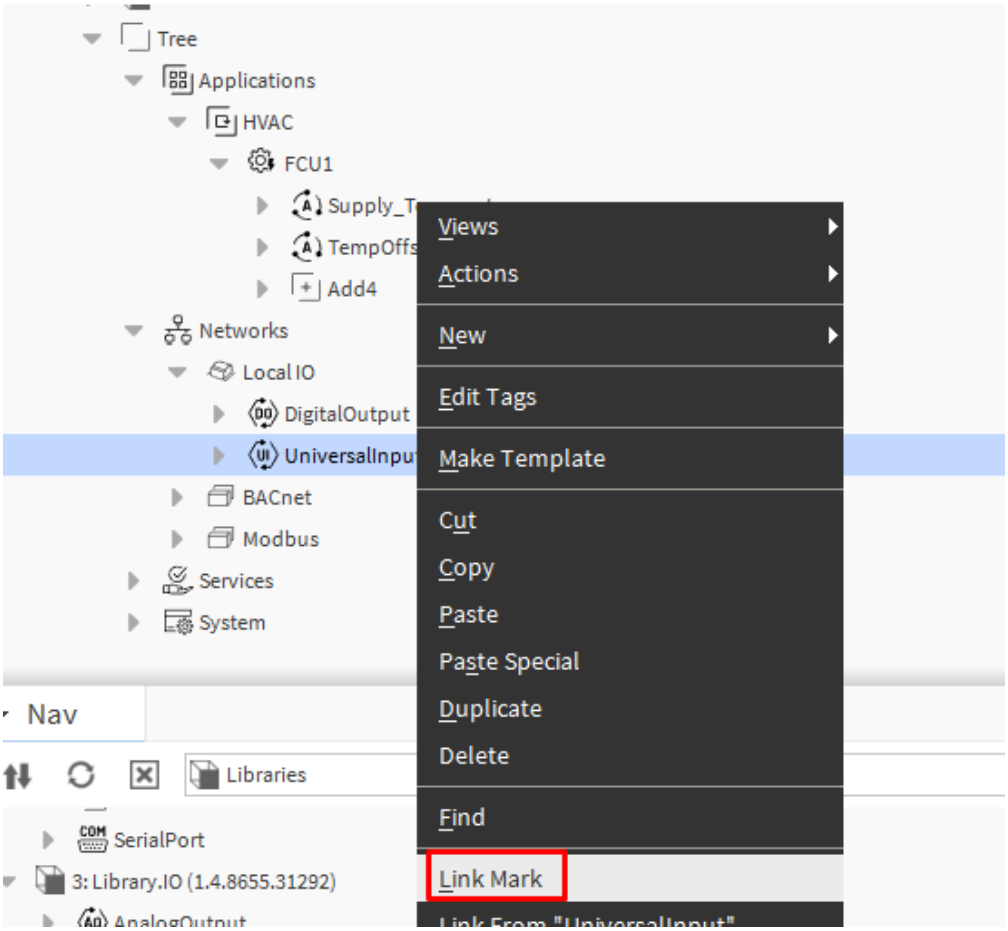


Figure 188. Link Mark option in the network point - UniversalInput

Once the link is marked, right-click on the desired Data Point and click the Link From option.

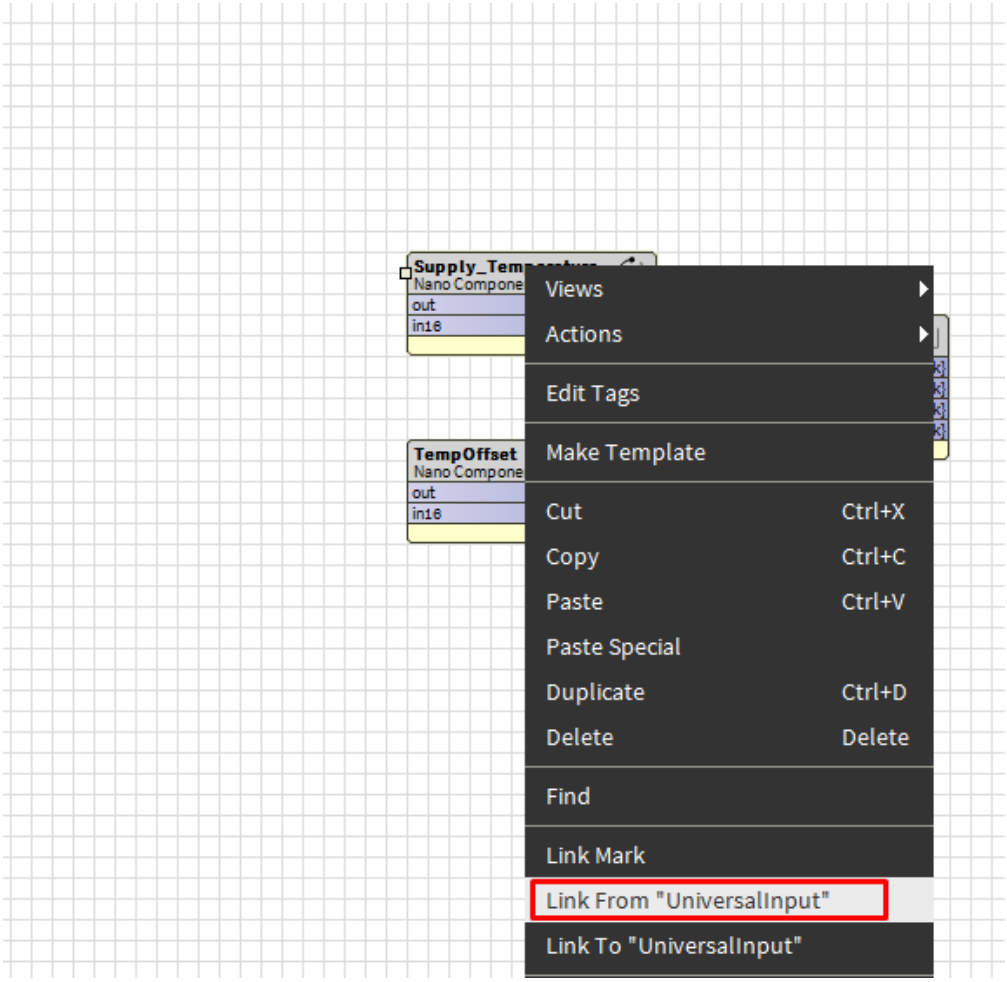


Figure 189. Link From option

In the dialog window that pops up, select “reference” slots on both Source and Target points.

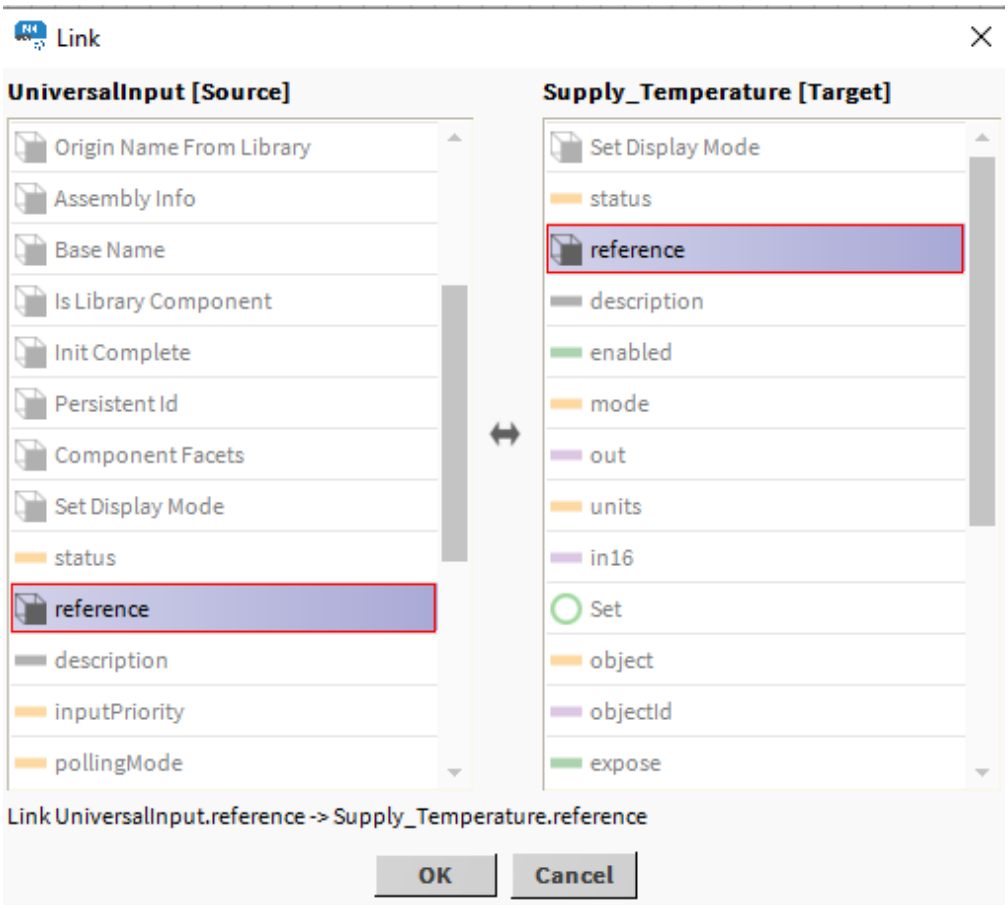


Figure 190. Creating reference between points

Once the reference is created, a new 'Nano Reference' slot appears in the component.

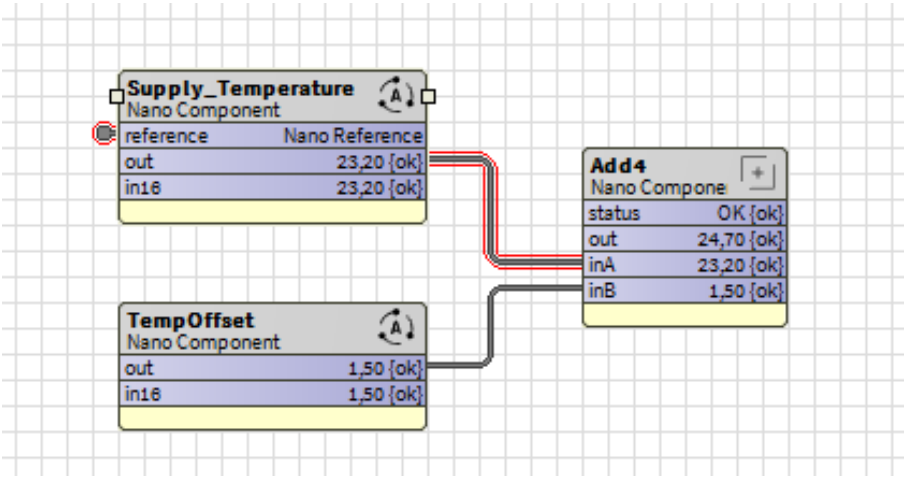


Figure 191. Data Point with a reference

Step 4: Integration to Niagara

Warning!

To integrate data in the Niagara Framework, user must use standard Niagara networks. nano EDGE ENGINE components must not be linked to Niagara components.

To integrate with Niagara, it is important to note that only Data Points can be exposed over networks. Each nano EDGE ENGINE device has a limit on the number of Data Points that can be exposed. The available number of Data Points can be found in the License component in the System container.

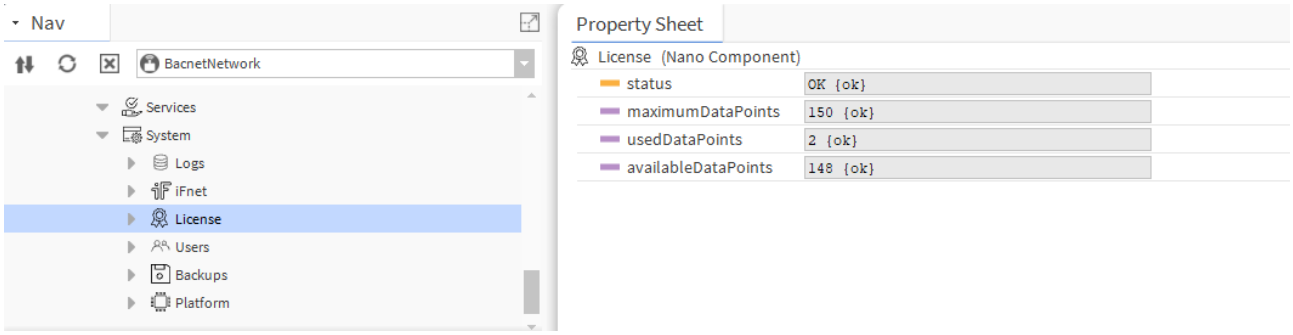


Figure 192. Number of available Data Points visible in the License component

Exposing Data Points

To integrate Data Points with Niagara, the points must be exposed over networks. By default, nano EDGE ENGINE exposes points over BACnet and Modbus.

In BACnet, all Data Points are exposed as BACnet objects by default. Individual Data Points can be hidden by manually changing the Expose slot value in their BACnet extension (e.g., BACnetAnalogPoint, BACnetBinaryPoint, BACnetMultistatePoint). The BACnet object type and object Id is visible in the Data Point BACnet extension. To change the Data Point's BACnetID, right-click on the Data Point and select the SetId action.

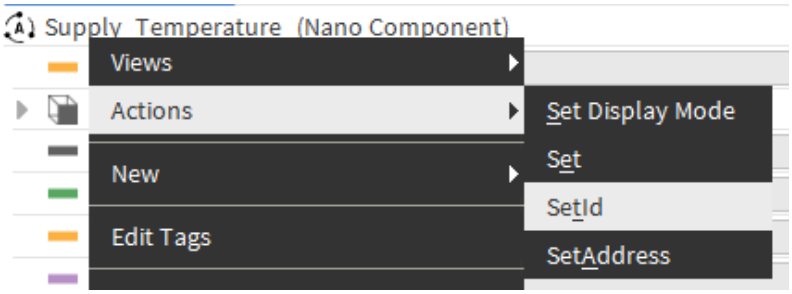


Figure 193. Setting ObjectID to the Data Point

In Modbus, all Data Points in the device are automatically exposed as the Modbus server device. In order to disable the Data Point in the Modbus server network, either set the Autoexposition slot in the Modbus component to false (all Data Points hidden) or go to each Data Point individually and set the Expose slot to false.

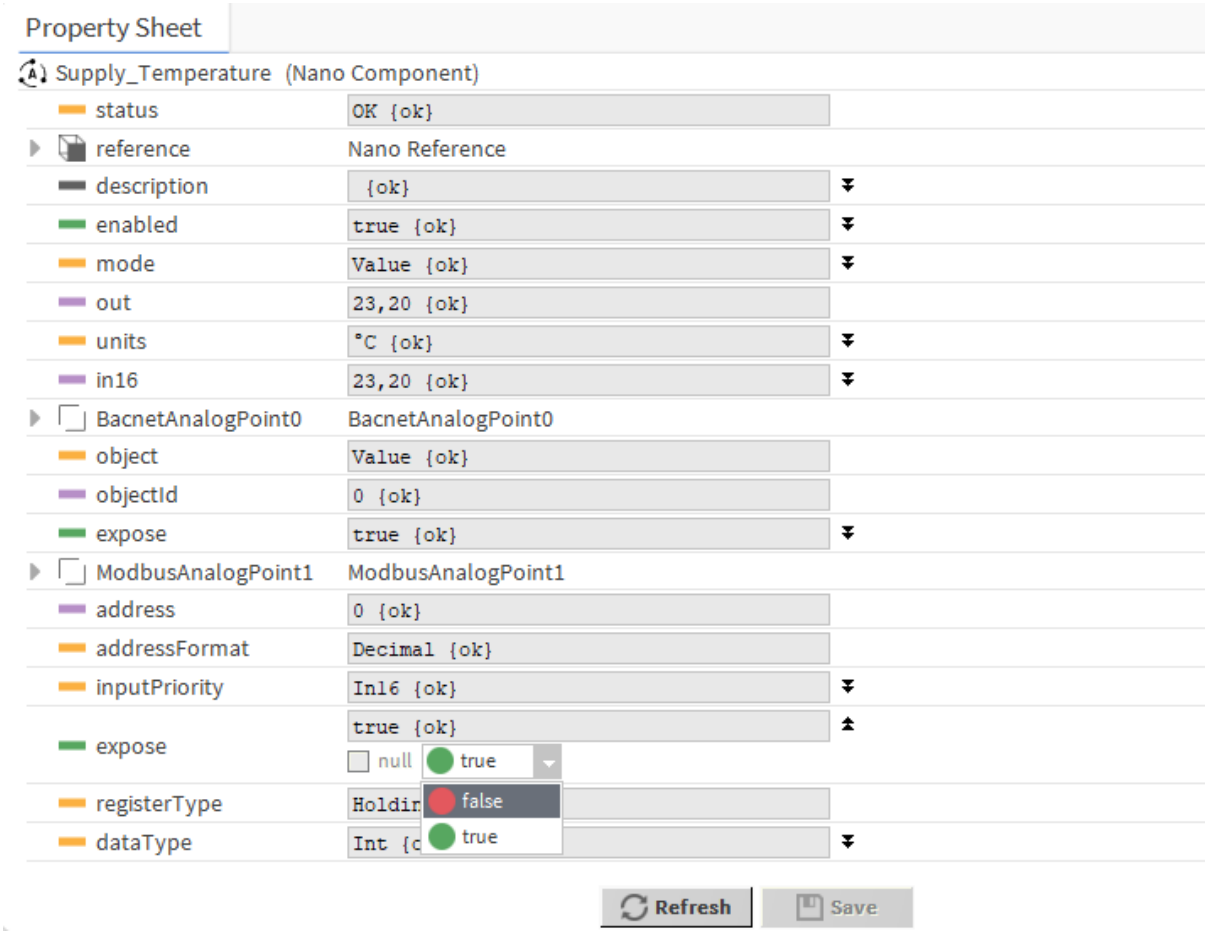


Figure 194. Possibility to disable the exposition on Modbus or BACnet in the Data Point's extension

Modbus address is set automatically. In order to set Modbus address manually, right-click the Data Point and select the SetAddress action.

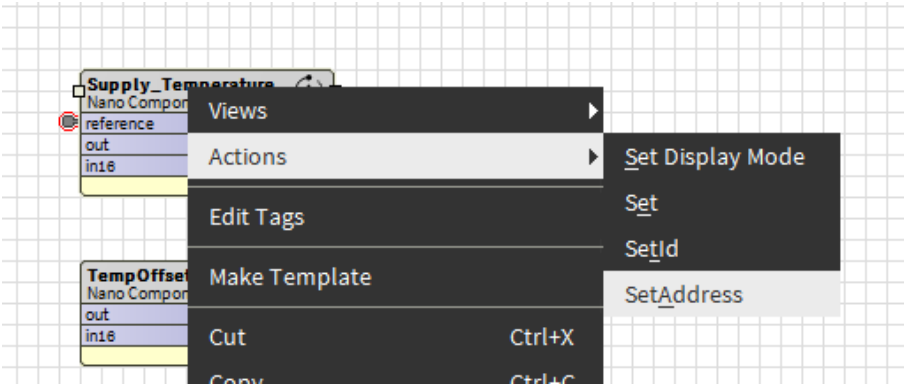


Figure 195. The SetAddress action

BACnetNetwork Niagara Integration

To integrate the nano EDGE ENGINE device and Data Points exposed over BACnet, make sure the LocalDevice component is properly configured. To change the BACnet Device settings, go to Networks → BACnet → LocalDevice, e.g., RAC18-IP.

To learn more about the LocalDevice, please refer to the nano EDGE ENGINE Programming user manual.

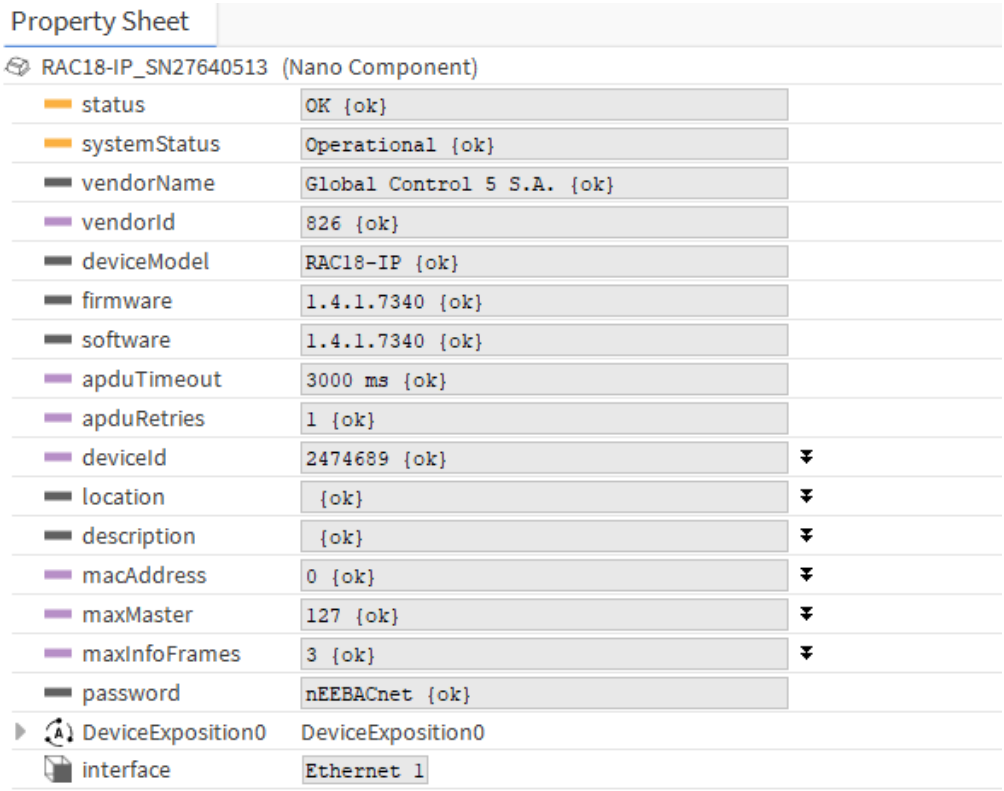


Figure 196. LocalDevice property sheet

Once the deviceID and other parameters are properly configured, go to the BACnetNetwork device in the Niagara station and make sure the configuration of the device is correct.

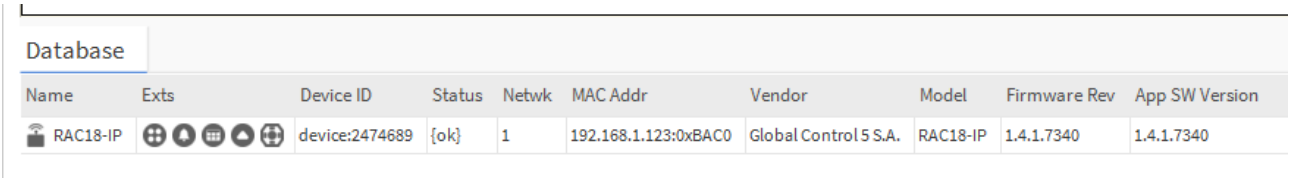


Figure 197. RAC18-IP integrated to Niagara over the BACnet network

To integrate points, go to the device → Points and click Discover. Add required points to Niagara database.

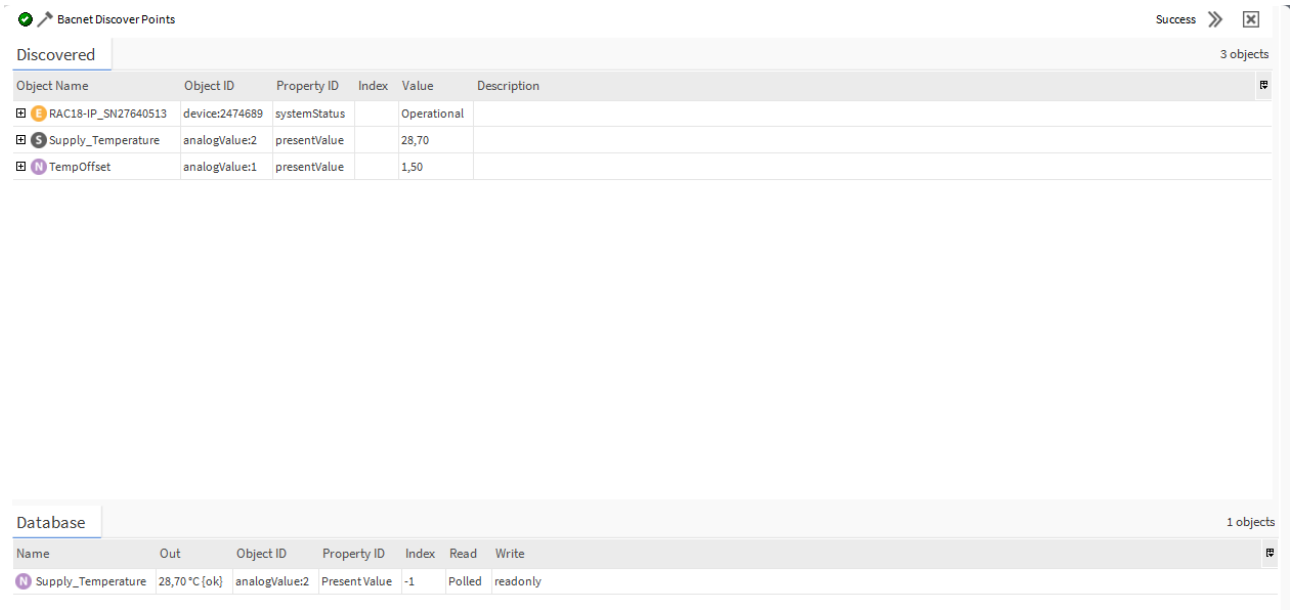


Figure 198. Points added to the Niagara BACnet database

The points have been successfully integrated into the Niagara BACnet network.

6 Download the VAV14-IP Software PDF User Manual