ID UM.L82341.2012

# User Manual

## LX-M10 MultiController

### SW type A


LONIX LTD
www.lonix.com

Teollisuuskatu 33
FI-00510 Helsinki, Finland

Tel. +358 9 349 9853
Fax +358 9 349 9863

VAT 1043054-9
Trade Reg.No. 678.356

# Index

# Introduction

This document describes technical details and functionality of the Lonix M10 MultiController.

Lonix M10 MultiController includes the needed functionality for HVACE processes, such as controls of heating, cooling, ventilation, air-conditioning and lighting. The M10 MultiController includes the freely configurable software application, which is configured to the needed functionality.

Each M10 includes 10 flexible I/O points:

- 5 UI
- 5 DI / UO
- 2 Controllers (PID)

This document consists of two main parts: LX-M10 Hardware and LX-M10 Software. The first part describes the physical features of the Controller and characteristics of the Input / Output points. The second part gives detailed view on Configuration Parameters of the I/O Objects and the Node Object.

Reader should have basic level of understanding about Lonix PCT (Project Creation Tool) software, which is used as a Configuration Tool for LX-M10 MultiControllers and other Lonix devices.

This document is meant primarily for installers, system integrators and other automation system professionals.

# LX-M10 Hardware

## Technical Data of LX-M10 Hardware

- General Data
- Inputs / Outputs
- Connections
- Front Panel

LONIX LTD
www.lonix.com
Teollisuuskatu 33
FI-00510 Helsinki, Finland
Tel. +358 9 349 9853
Fax +358 9 349 9863
VAT 1043054-9
Trade Reg.No. 678.356

# 1. Technical Data

## 1.1 General

General technical data about M10 is described in the following table.

| Parameter | Description |
|---|---|
| **Operating Voltage** | 24V AC/DC (±10%) |
| **Operating Temperature** | 10-50°C |
| **Power Required** | 0.9 W (with no I/O connected) |
| **Overload Protection** | Automatic PSU safety shutdown |
| **Main Processor** | ARM Cortex™ M4 (Kinetis K10) |
| **Network Processor** | Echelon FT-5000 |
| **Network Interface** | TP/FT-10 channel |
| **Memory** | 256 kb Flash, 64kb SRAM |
| **Clock Frequency** | 50 MHz |
| **IP Class** | IP20 |
| **Size** | 110 x 71.30 x 62 mm (4M width) |
| **Mounting** | 35 mm DIN rail |
| **Connection Strips** | Detachable, wire max 2,5 mm2, in blocks of two terminals, except the Bus connection in block of three terminals |
| **EMC Compatibility** | Compliance according to EN 55022, EN 61000 |
| **Production Standards** | ISO-9001, ISO-14001 |

## 1.2 Inputs / Outputs

The Lonix M10 MultiController contains 10 I/O points in total:

- 5 DI/UO (Digital Input / Universal Output)
- 5 UI (Universal Input)

Points 1-5 are Digital Input / Universal Output points, so they can be configured as Digital Input, Digital Output or Analog Output. Points 6-10 are Universal Input points, so they can be configured as Digital Input or Analog Input points.

**Digital Input:**

Potential free contact input. Points 1-5 are using +24V and points 6-10 +5V excitation voltage to detect the state of the circuit to be monitored. Higher voltage of points 1-5 can be utilized e.g. when signal cables are located in environment with high interference, thus higher excitation voltage might reduce the impact of the interference.

Plus terminals of Digital Input points 1-5 are internally connected together (common plus) and minus (ground) terminals of Digital Input points 6-10 are internally connected together (common minus).

**Digital Output:**

Open collector type, max 750 mA per controller. Switching mode of the output can be selected per point from the following modes: *Low(-)*, *High(+)* or *Low(-) and High(+)*. This feature can be used when common plus or common minus configuration needs to be utilized.

*See also: Digital Output (DO) configuration parameter 'Switching Mode'.*

**Analog Input:**

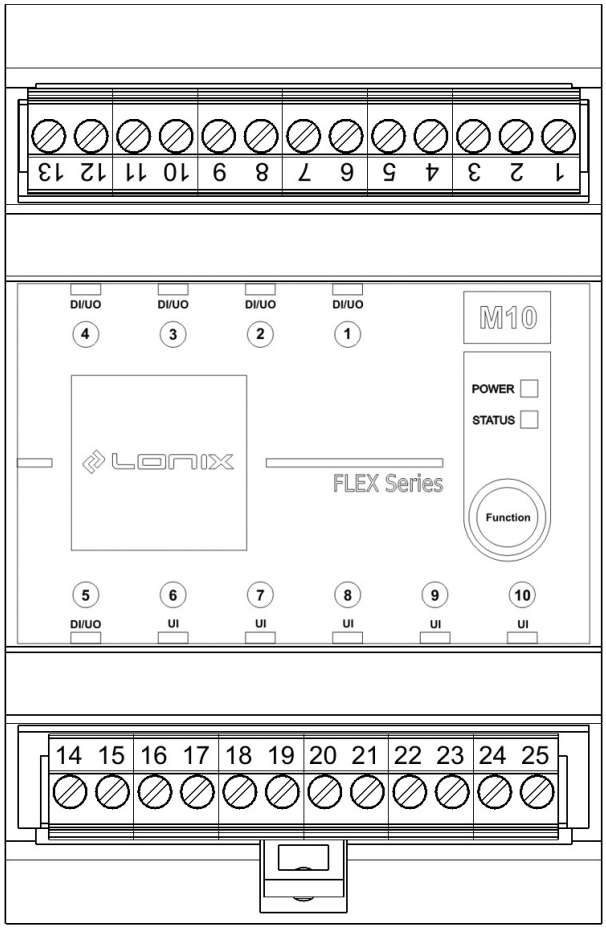
0-10 VDC input, passive sensors: Pt1000, Ni1000-DIN, Ni1000-LG and raw resistance measure (0…2100 Ohm). All Analog Input points share the same ground potential through minus terminals of the points.

**Analog Output:**

0-10 VDC output, max 20 mA per point. All Analog Output points share the same ground potential through minus terminals of the points.

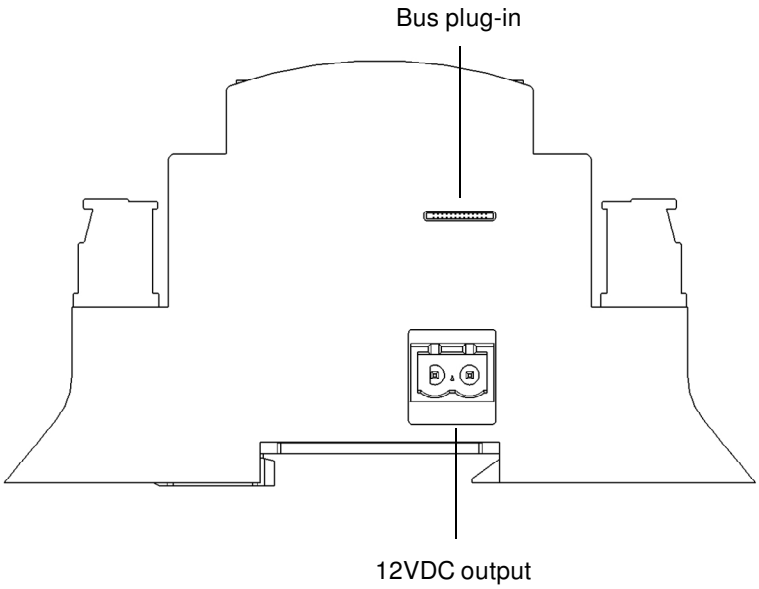
## 1.3 Connections

The layout of M10 is as in the following figure and the connections are explained in the following table.




| Nr | Connection |
|---|---|
| **1** | (Bus connection Shield) |
| **2** | Bus connection A |
| **3** | Bus connection B |
| **4** | G / + (operating voltage 24VAC/DC) |
| **5** | G0 / - (operating voltage 24VAC/DC) |
| **6** | Point 1 UO/DI+ |
| **7** | Point 1 UO/DI - |
| **8** | Point 2 UO/DI + |
| **9** | Point 2 UO/DI - |
| **10** | Point 3 UO/DI + |
| **11** | Point 3 UO/DI - |
| **12** | Point 4 UO/DI + |
| **13** | Point 4 UO/DI - |
| **14** | Point 5 UO/DI - |
| **15** | Point 5 UO/DI + |
| **16** | Point 6 UI - |
| **17** | Point 6 UI + |
| **18** | Point 7 UI - |
| **19** | Point 7 UI + |
| **20** | Point 8 UI - |
| **21** | Point 8 UI + |
| **22** | Point 9 UI - |
| **23** | Point 9 UI + |
| **24** | Point 10 UI - |
| **25** | Point 10 UI + |

The M10 includes additional connections in the side for bus plug-in and 12 VDC output.




**Bus plug-in:**

Connector for bus and power chaining between adjacent controllers. Located on both sides of the controller (external connector LX-BUS required).

**12 VDC output:**

Power output for external devices e.g. occupancy detector. Max. current 80mA. Located on the right side of controller under field removable cover.

## 1.4 Front Panel

The front panel of M10 includes 10 I/O LEDs, Power LED, Status LED and a Function button.

| LED / Button | Description |
|---|---|
| **LED 1…5** | Input status (GRN), Output status (YEL) |
| **LED 6…10** | Input status (GRN) |
| **Power LED** | Power indicator (GRN) |
| **Status LED** | Controller functional status (YEL) |
| **Function button** | Service Pin |

LED functions indicate the status of the I/O:

| I/O status | | LED function |
|---|---|---|
| **DI - Circuit closed** |  | LED on |
| **DI - Circuit open** |  | LED off |
| **AI - Measure OK** |  1x/s | Short LED blink once in a second.<br>(Active (0-10V) and Resistance: LED always on) |
| **AI - Not connected** |  4x/s | LED blinking slowly, 4x /second<br>(Active (0-10V) and Resistance: LED always on) |
| **AI - Short-circuited** |  10x/s | LED blinking fast, 10x /second<br>(Active (0-10V) and Resistance: LED always on) |
| **DO – Active (on)** |  | LED on |
| **DO – Inactive (off)** |  | LED off |
| **AO - Output value** |  | LED with 0-100% blinking frequency:<br>0V = off, increasing from >0V = slow to 10V = fast |

**NOTE:**
LEDs always indicate the physical state of the I/O. For example, programmatic inverting does not change LED functionality.

# LX-M10 Software

## Configuration of M10 Application

- General
- Digital Input (DI)
- Digital Output (DO)
- Analog Input (AI)
- Analog Output (AO)
- Controller (PID)
- Node Object
- Module Parameters

# 2. General

The M10 MultiController includes the embedded software application, which is configured for the needed functionality. These chapters describe the functionality of the M10 application.

## 2.1 Generic Parameters

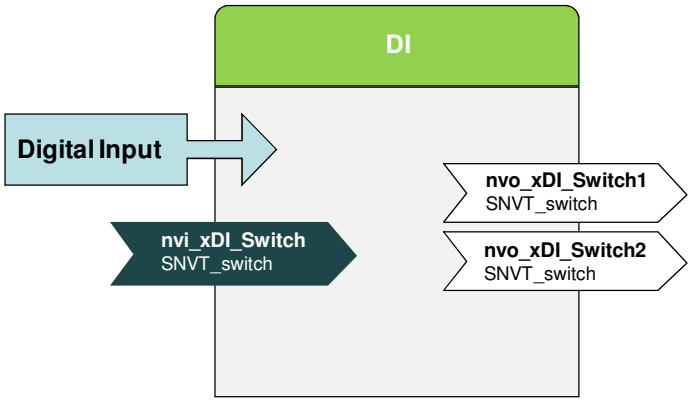
The following table includes the Generic Parameters which are applicable for all I/O objects:

| Parameter | Description | Range |
|---|---|---|
| **Minimum Send Time** | The minimum send time of output network variables. The object does not send the values of its output network variables to LON-network until this time has elapsed, even if the object state changes during that time period. When the value equals zero (0 s), sending is event based i.e. the values of the output network variables are sent to the LON-network whenever their value changes. NOTE! Sending of values of output network variables to LON-network is event based, if either the Minimum Send Time or Maximum Send Time parameter is zero (0). | 0 … 65535s |
| **Maximum Send Time** | The maximum send time of output network variables. Setting this parameter to non-zero causes the object to send values of its nvo's to LON-network at frequent intervals even if their state has not changed. | 0 … 65535s |

## 2.2 Network Variable Aliases

The M10 MultiController supports ten (10) network variable alias table entries. The aliases are used to solve network variable selector conflicts during binding of network variables. Network Management Tool will automatically take care of usage of aliases when they are needed.

# 3. Digital Input (DI)




**The Digital Input (DI) includes the following functions:**

- Indication
- Pulse Switch
- Scene (Normal)
- Scene (State Operated)
- Monitored Alarm
- Monitored Alarm Arming
- Slide (ON/OFF - SUB/ADD)
- Slide (ADD)
- Slide (SUB)

## 3.1 Indication

Indication is used for monitoring the state information of e.g. pumps, fans, filters, smoke detectors, etc. It can also monitor conflicts between the object's physical state and the state of the network input variable. This is often used e.g. for confirming that a device with a run status signal actually turned on by binding the output of an DO object (relay) to the network variable input of this DI object.

### Functionality of Network Variables:

**nvi_xDI_Switch:** If the control state information of another object is bound to the nvi_xDI_Switch and the Alarming State parameter is set appropriately, the object monitors for conflicts. Comparison is done between the state-field of this input network variable and the selected output network variable (see Alarming State parameter).

**nvo_xDI_Switch1:** The value and state of the object.

**nvo_xDI_Switch2:** The value and state of the object.

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. If the object is configured as an alarm point, the alarm takes place only after the On Delay time. In case the object compares conflicts (for alarming) the On Delay time is used for conflict information as well. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. In case the object compares conflicts (for alarming) the Off Delay time is used for conflict information as well. | 0 … 65535s |
| **Invert** | This parameter inverts the values of the output network variables i.e. when the object is changed to ON-state, OFF-state values of the output network variables are updated and vice versa. | Yes<br>No |
| **Alarm Priority** | This parameter defines the alarm priority. The higher the value, the more important the alarm is. When set to zero (0), the object does not send any alarms. When a Lonix STAM controller is involved, an alarm of priority 4 or greater will trigger the alarm siren. An alarm with a priority of 3 will trigger the alarm buzzer. | 0 … 7 |
| **On Value.State of nvo_xDI_Switch1** | This parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |
| **Off Value.State of nvo_xDI_Switch1** | The parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |
| **On Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |
| **Off Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |
| **Operational Hours Init Value** | Initiation value for the usage hour information stored in nvoTrend variable of the controller. | 0 … 65535s |

| | | |
|---|---|---|
| **Alarming State** | The object triggers an alarm when the value of the state-field in the selected nvo_xDI_Switch variable equals the value of this parameter. The object's input network variable nvi_xDI_Switch should not be bound to anything. If the object's nvi_xDI_Switch is bound and this parameter is set to "nvo_xDI_Switch conflict", the object will compare the state informations of the input network variable and the selected output network variable. If the state informations differ from each other, an alarm is generated. | nvoDIswitch1 conflict, nvoDIswitch2 conflict, nvoDIswitch1.state = 0/1, nvoDIswitch2.state = 0/1 |

***Note 1:*** *Notation: 200.01 means that value is 200 and state information 1.*

## 3.2 Pulse Switch

The most common application is push buttons of lighting systems. The state of this object changes every time the circuit is closed and opened again. For an object of this type to function properly it must have feedback information from the object controlling the physical device.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xDI_Switch:** | The object uses the state field (feedback of device status) of this variable for selecting its next state (ON/OFF). For example, if the feedback information is OFF, the next push of the button sends values configured for ON-state to output variables. |
| **nvo_xDI_Switch1:** | Shows the ON/OFF value configured for this variable according to the object's state. |
| **nvo_xDI_Switch2:** | Shows the ON/OFF value configured for this variable according to the object's state. |

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. If the object is configured as an alarm point, the alarm takes place only after the On Delay time. In case the object compares conflicts (for alarming) the On Delay time is used for conflict information as well. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. In case the object compares conflicts (for alarming) the Off Delay time is used for conflict information as well. | 0 … 65535s |
| **Invert** | This parameter inverts the values of the output network variables i.e. when the object is changed to ON-state, OFF-state values of the output network variables are updated and vice versa. | Yes<br>No |
| **On Value.State of nvo_xDI_Switch1** | This parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |

| | | |
|---|---|---|
| **Off Value.State of nvo_xDI_Switch1** | The parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |
| **On Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |
| **Off Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>*See Note 1.* |

***Note 1:*** *Notation: 200.01 means that value is 200 and state information 1.*

## 3.3 Scene (Normal & State Operated)

This function is used e.g. in push buttons of lighting systems, extension time buttons of air-conditioning units and running info of pumps and fans. In *Normal* mode the state of this object changes every time the circuit is first closed and then opened again (pulse). In *State Operated* mode the state of the object is changed according to circuit state: closed → ON, open → OFF similar to "Indication" function. In both modes, the value of nvi_xDI_Switch is always passed on to both nvo_xDI_Switches to allow chaining of the scene objects.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xDI_Switch:** | The value of nvi_xDI_Switch is passed directly to both nvo_xDI_Switch1 and nvo_xDI_Switch2 so the object can be used for chaining of scene objects. |
| **nvo_xDI_Switch1:** | Shows the ON/OFF value configured for this variable according to the object's state. |
| **nvo_xDI_Switch2:** | Shows the ON/OFF value configured for this variable according to the object's state. |

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. If the object is configured as an alarm point, the alarm takes place only after the On Delay time. In case the object compares conflicts (for alarming) the On Delay time is used for conflict information as well. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. In case the object compares conflicts (for alarming) the Off Delay time is used for conflict information as well. | 0 … 65535s |

| | | |
|---|---|---|
| **Invert** | This parameter inverts the values of the output network variables i.e. when the object is changed to ON-state, OFF-state values of the output network variables are updated and vice versa. | Yes<br>No |
| **Netvar Invert** | (Only in State Operated function). When this parameter has value Yes, the value and state of the input network variable nvi_xDI_Switch is inverted (with respect to 100) when passed on to the output network variables.<br>Example: nvi_xDI_Switch = 20.0 0 => nvo_xDI_Switch1&2 = 80.0 1 | Yes<br>No |
| **On Value.State of nvo_xDI_Switch1** | This parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Off Value.State of nvo_xDI_Switch1** | The parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **On Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Off Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Dead Time for Feedback** | This parameter defines the time (beginning from the closing of the object's dry contacts i.e. the push of a button) during which received input signals are not relayed to the object's output variable. With this feature the risk of a signal traversing a loop more than one time is eliminated. | |

***Note 1:*** *Notation: 200.01 means that value is 200 and state information 1.*

## 3.4 Monitored Alarm

This function is used mostly in burglar and fire alarm applications when the system needs to be monitored for tampering. The loop is formed by one or many objects of this function and by a single DI-object configured as "Monitored Alarm Arming".

***See also:*** *Monitored Alarm Arming*

**NOTE:**
In smart home systems utilizing Lonix STAM module, when a single burglar alarm loop for occupancy sensors and another one for magnetic switches is sufficient, it is recommended to use the ShieldLoop and AreaLoop objects of the STAM module.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xDI_Switch:** | In burglar alarm applications this variable is bound with the nvo_xDI_Switch1 of another DI-object in the same burglar alarm loop. This enables transmission of the monitoring token of the alarm loop for tampering. |
| **nvo_xDI_Switch1:** | When the burglar alarm loop is unarmed, the value of this variable shows either the value of nvi_xDI_Switch variable or the physical state of the object, whichever has changed most recently. When the object is armed i.e. the loop this object |

| | |
|---|---|
| | belongs to is armed, this variable passes the monitoring token to the next object in the loop. |
| **nvo_xDI_Switch2:** | Follows the object's physical state. When the circuit is closed, the variable shows the configured ON-value of the object. When the circuit is open, the variable shows the configured OFF-value. When the object is alarming, the value of this variable shows either burglar alarm code or fire alarm code, depending on the configuration. |

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. If the object is configured as an alarm point, the alarm takes place only after the On Delay time. In case the object compares conflicts (for alarming) the On Delay time is used for conflict information as well. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. In case the object compares conflicts (for alarming) the Off Delay time is used for conflict information as well. | 0 … 65535s |
| **Invert** | This parameter inverts the values of the output network variables i.e. when the object is changed to ON-state, OFF-state values of the output network variables are updated and vice versa. | Yes<br>No |
| **Alarm Delay** | Alarm and monitoring delay i.e. safe time. When the object switches to armed mode, it will not cause any alarms during the Alarm Delay period. The monitored space can be safely left during this delay without causing alarms. This parameter also defines the sending time of the alarm loop monitoring token. | 0 … 320s |
| **Alarm Priority** | This parameter defines the alarm priority. The higher the value, the more important the alarm is. When set to zero (0), the object does not send any alarms. When a Lonix STAM controller is involved, an alarm of priority 4 or greater will trigger the alarm siren. An alarm with a priority of 3 will trigger the alarm buzzer. | 0 … 7 |
| **Alarm Type** | The parameter defines whether a burglar alarm or a fire alarm is caused when the object is alarming. This influences the alarm code shown in variable nvo_xDI_Switch2. | Burglar alarm<br>Fire alarm |
| **On Value.State of nvo_xDI_Switch1** | This parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Off Value.State of nvo_xDI_Switch1** | The parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **On Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Off Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |

| | | |
|---|---|---|
| **Operational Hours Init Value** | Initiation value for the usage hour information stored in nvoTrend variable of the controller. | 0 … 65535s |

***Note 1:*** *Notation: 200.01 means that value is 200 and state information 1.*

## 3.5 Monitored Alarm Arming

This function is used in alarm applications for switching the monitored alarm loop status into armed/disarmed (circuit closed = armed). Switching to disarmed-state also functions as an acknowledgement and resets the alarms. Field devices (occupancy sensors, magnetic switches, etc), which are connected to other DI-points and configured as type ”Monitored Alarm”, are bound into loop with this object to form a monitored alarm loop. The object monitors the alarm loop by sending a periodical monitoring token to the member objects of the loop. If the token does not pass the whole loop, a sabotage alarm is triggered after a set delay. A loop must not include more than one object of this type.

***See also:*** *Monitored Alarm*

### Functionality of Network Variables:

| | |
|---|---|
| **nvi_xDI_Switch:** | Used for the feedback of the alarm loop. It shall be connected with the nvo_xDI_Switch1 of the last object in the alarm loop. |
| **nvo_xDI_Switch1:** | Shall be connected with the nvi_xDI_Switch of the first object in the alarm loop (see above). It can also be bound to a DO-object that functions as an alarm buzzer or indication. |
| **nvo_xDI_Switch2:** | Indicates the state of nvi_xDI_Switch (loop armed/disarmed) and can be used e.g. as a control of an alarm arming indication lamp. |

### Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object’s physical state. When the object’s state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object’s physical state. When the object’s state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Invert** | This parameter inverts the values of the output network variables i.e. when the object is changed to ON-state, OFF-state values of the output network variables are updated and vice versa. | Yes<br>No |
| **Alarm Delay** | Alarm and monitoring delay i.e. safe time. When the object switches to armed mode, it will not cause any alarms during the Alarm Delay period. The monitored space can be safely left during this delay without causing alarms. This parameter also defines the sending time of the alarm loop monitoring token. | 0 … 320s |

| | | |
|---|---|---|
| **Alarm Priority** | This parameter defines the alarm priority. The higher the value, the more important the alarm is. When set to zero (0), the object does not send any alarms. When a Lonix STAM controller is involved, an alarm of priority 4 or greater will trigger the alarm siren. An alarm with a priority of 3 will trigger the alarm buzzer. | 0 … 7 |
| **On Value.State of nvo_xDI_Switch1** | This parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Off Value.State of nvo_xDI_Switch1** | The parameter defines the value and state of the output network variable nvo_xDI_Switch1 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **On Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in ON-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Off Value.State of nvo_xDI_Switch2** | The parameter defines the value and state of the output network variable nvo_xDI_Switch2 when the object is in OFF-state. | Value: 0 ... 200<br>State: 0 or 1<br>See Note 1. |
| **Alarm Type** | The parameter defines whether this object arms a burglar alarm or a fire alarm loop. This influences the alarm code shown in variable nvo_xDI_Switch1 and in alarm object_id during alarm event. | Burglar alarm<br>Fire alarm |

***Note 1:*** *Notation: 200.01 means that value is 200 and state information 1.*

## 3.6 Slide (all operation types)

The most common application of this type is increase/decrease buttons of dimmable light groups. It can be used e.g. for continuous dimming of a light group or linear control of a fan. A long control pulse increases (ADD type) or decreases (SUB type) the control value and once it reaches the maximum/minimum a new control pulse will not have any effect on the control value until a feedback signal with a new control value is provided. Hence when using the ADD or SUB type, two DI objects (one ADD, one SUB) should be used.

In (ON/OFF-SUB/ADD) mode every second long control pulse increases and every second decreases the control level. Short control pulses switch the object fully OFF, or - in case it already is fully OFF - to fully ON.

**NOTE:**
Short control pulse: <10ms
Long control pulse: =>10ms

**Functionality of Network Variables:**

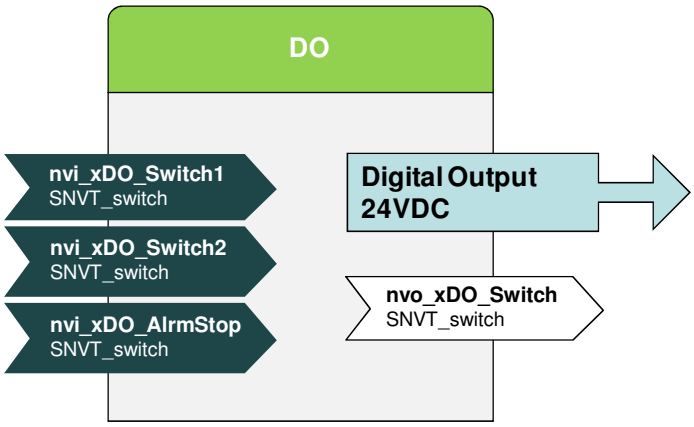
| | |
|---|---|
| **nvi_xDI_Switch:** | The initial value from which the object begins to change the value of its output, often a feedback from the controlled object. |
| **nvo_xDI_Switch1:** | The controlled output value of the object. |
| **nvo_xDI_Switch2:** | The controlled output value of the object. |

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Invert** | This parameter inverts the values of the output network variables i.e. when the object is changed to ON-state, OFF-state values of the output network variables are updated and vice versa. | Yes<br>No |
| **Button Press Delay** | Delay for the button press. The values of the output variables are changed only after the object's physical input has remained in ON-state (closed circuit) for the time defined with this parameter. | 10 … 100ms |
| **Steps per Button Press** | Step size of the control value. The output variable values change the amount defined in this parameter with every long control pulse. | 0 … 200 |
| **Min Level** | Defines the minimum value of the control variables. (Not used in ADD mode). | 0 … 200 |
| **Max Level** | Defines the maximum value of the control variables. | 0 … 200 |

***Note 1:*** *Notation: 200.01 means that value is 200 and state information 1.*

# 4. Digital Output (DO)




**The Digital Output (DO) includes the following functions:**

- On/Off
- Scene
- Trigger
- PWM-Control Normal / Constant time
- Alarm Priority Control PRI == DO PRI / PRI => DO PRI
- Binary Step
- Linear Step
- Logical AND / AND-AND-OR / AND-OR-AND / AND-OR-OR / OR / OR-AND-OR

## 4.1 On/Off

This function is commonly used for turning devices such as pumps, fans, and lighting ON/OFF.

**Functionality of Network Variables:**

**nvi_xDO_Switch1:** nvi_xDO_switch1 and nvi_xDO_switch2 are bound to incoming control commands. If both variables are connected, the latest control command is effective. The object's physical digital output is controlled with the state of the incoming control command (1 =

| | |
|---|---|
| | ON, 0 = OFF). The value of the control command can also be used to control the physical digital output in conjunction with the Trigger Level parameter. |
| **nvi_xDO_Switch2:** | See nvi_xDO_Switch1. |
| **nvi_xDO_AlrmStop:** | This variable is used to lock the object to OFF-state. This happens when the state of this variable is one (1). Locking and unlocking of the object is achieved also by configuring the Locking Level and Locking Release Level, and using the value of nvi_xDO_AlrmStop variable. |
| **nvo_xDO_Switch:** | This variable shows the value of the latest control command received through nvi_xDO_Switch1 or nvi_xDO_Switch2. The state corresponds with the current state of the physical digital output. |

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | 0...200 |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **Trigger Level (Value.State)** | The value field of nvi_xDO_Switch1/2 variable can be used to control the physical digital output by configuring this parameter with a desired trigger value, higher values changing the physical digital output into ON-state and lower into OFF-state. If the state field of this parameter is set to zero, then the object will be ON when the state of an nvi_xDO_Switch1/2 variable is 0 and OFF when the state is 1, and vice versa.<br><br>Example: If the Trigger Level (value.state) parameter is set to 11.00, the DO | Value 0…255<br>State 0…1<br><br>E.g. 200.01 =<br>Value: 200,<br>State: 1 |

| | | |
|---|---|---|
| | will be in ON state whenever the value of an nvi_xDO_Switch1/2 is greater than 5.5, or the state of an nvi_xDO_Switch1/2 is 0. The object will be in OFF state when the value is less or equal to 5.5 and the state is 1. | |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |

## 4.2 Scene

This function is commonly used in lighting and ventilation applications. Six (6) ON scenes and six (6) OFF scenes can be configured for this object. In addition, the object can be forced to ON state by inputting a control command of value 100 and OFF with a value of zero (0). The object does not react to control commands that are not specified in the object's scene parameters but passes them on with nvo_xDO_Switch.

**Functionality of Network Variables:**

**nvi_xDO_Switch1:** Variables nvi_xDO_Switch1 and nvi_xDO_Switch2 are used for incoming control commands. The latest command is effective. The value of the control command is regarded as a scene number, the state has no significance.

**nvi_xDO_Switch2:** See nvi_xDO_Switch1.

**nvi_xDO_AlrmStop:** This variable is used to lock the object to OFF-state. This happens when the state of this variable is one (1). Locking and unlocking of the object is achieved also by configuring the Locking Level and Locking Release Level, and using the value of nvi_xDO_AlrmStop variable.

**nvo_xDO_Switch:** This variable shows the value of the latest control command received via nvi_xDO_Switch1 or nvi_xDO_Switch2. The state corresponds with the current state of the object.

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network | 0 … 65535s |

| | | |
|---|---|---|
| | variables are updated only after this time has elapsed. | |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | 0...200 |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |
| **(On) Scene1.Scene 2** | The object is transferred to ON state when the value of the control command received by nvi_xDO_Switch1 or nvi_xDO_Switch2 corresponds to either of the scene numbers configured in this parameter. Scene number 1 is the first part of this parameter and scene number 2 the second part (separated with a dot). For example, scene numbers 30 and 4 need to switch the object to ON state. The value of this parameter is configured as either 60.08 or 8.60. The object is switched to ON state when a control command of value 30 or 4 is received by nvi_xDO_Switch1 or nvi_xDO_Switch2.<br><br>NOTE! If a scene number greater than 49 is used, it needs to be configured to the first part of the parameter e.g. scene numbers 40 and 60 should be stated as 120.80. Also, scene numbers less than 5 in the second part of the parameter need to have a zero in front of them e.g. 3.09 for scene numbers 1.5 and 4.5. | (See description) |
| **(On) Scene 3.Scene 4** | See (On) Scene1.Scene 2 | (See description) |
| **(On) Scene 5.Scene 6** | See (On) Scene1.Scene2 | (See description) |
| **(Off) Scene 1.Scene 2** | The object is switched to OFF state when the value of the control command received by nvi_xDO_Switch1 or nvi_xDO_Switch2 corresponds to either of the scene numbers configured in this parameter. Scene number 1 is the first part of this parameter and scene number 2 the second part (separated with a dot). For example, scene numbers 20 and 5 need to switch the object to OFF state. The value of this parameter should be configured to 40.10. The object is switched to OFF state when a control command of value 20 or 5 is received by nvi_xDO_Switch1 or nvi_xDO_Switch2.<br><br>NOTE! If a scene number greater than 49 is used, it needs to be configured to the first part of the parameter e.g. scene numbers 40 and 60 should be stated as 120.80. Also, scene numbers less than 5 in the second part of the parameter need to have a zero in front of them e.g. 3.09 for scene numbers 1.5 and 4.5. | (See description) |

| | | |
|---|---|---|
| **(Off) Scene 3.Scene 4** | See (Off) Scene1.Scene2 | (See description) |
| **(Off) Scene 5.Scene 6** | See (Off) Scene1.Scene2 | (See description) |

## 4.3 Trigger

This function is mostly used in access control functions when e.g. a lock should stay open only for a configured period of time, for example 5sec. The received control input is passed on to the output network variable but reset to zero after a configured delay (Off Delay parameter).

**Functionality of Network Variables:**

**nvi_xDO_Switch1:** Variables nvi_xDO_Switch1 and nvi_xDO_Switch2 are bound to incoming control commands. If both variables are connected, the latest control command is effective. The object's physical digital output is controlled with the state of the incoming control command (1 = ON, 0 = OFF). The value of the control command can also be used to control the physical digital output in conjunction with the Trigger Level parameter.

**nvi_xDO_Switch2:** See nvi_xDO_Switch1.

**nvi_xDO_AlrmStop:** This variable is used to lock the object to OFF-state. This happens when the state of this variable is one (1). Locking and unlocking of the object is achieved also by configuring the Locking Level and Locking Release Level, and using the value of nvi_xDO_AlrmStop variable.

**nvo_xDO_Switch:** This variable shows the value of the latest control command received via nvi_xDO_Switch1 or nvi_xDO_Switch2. The state corresponds with the current state of the physical digital output.

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the time after which the object resets itself back to zero after a control command is received. | 0 … 65535s |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of | 0...200 |

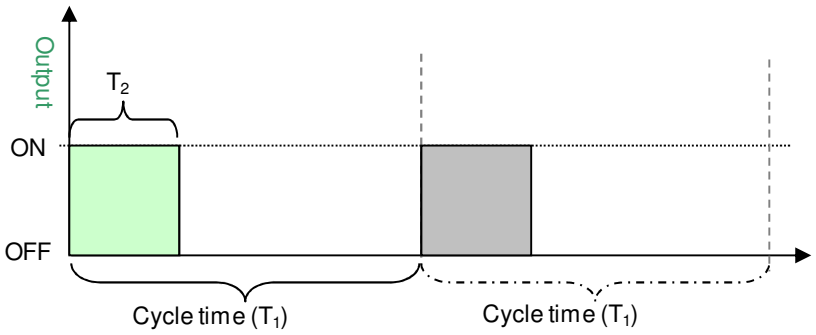
| | | |
|---|---|---|
| | nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **Trigger Level (Value.State)** | The value field of nvi_xDO_Switch1/2 variable can be used to control the physical digital output by configuring this parameter with a desired trigger value, higher values changing the physical digital output into ON-state and lower into OFF-state. If the state field of this parameter is set to zero, then the object will be ON when the state of an nvi_xDO_Switch1/2 variable is 0 and OFF when the state is 1, and vice versa.<br><br>Example: If the Trigger Level (value.state) parameter is set to 11.00, the DO will be in ON state whenever the value of an nvi_xDO_Switch1/2 is greater than 5.5, or the state of an nvi_xDO_Switch1/2 is 0. The object will be in OFF state when the value is less or equal to 5.5 and the state is 1. | Value 0…255<br>State 0…1<br><br>E.g. 200.01 =<br>Value: 200,<br>State: 1 |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |

## 4.4 PWM-Control (Normal)

This function is mostly used in PWM-based heating and cooling control such as floor heating systems and cooling systems. The output capacity of the system is adjusted by altering the time the object is in ON-state during a configured duty cycle time.

For example, a 40% output capacity is achieved with a 100 second duty cycle by keeping the object 40 seconds in ON-state and 60 seconds in OFF-state.

**PWM Normal**




$T_1$ = PWM Cycle Time x PWM Cycle Multiplier

$T_2$ = nvi_DO_Switch1 x 0.01 x $T_1$ (or nvi_DO_Switch2 x 0.01 x $T_1$)

## Functionality of Network Variables:

**nvi_xDO_Switch1:** Variables nvi_xDO_Switch1 and nvi_xDO_Switch2 are used for incoming control commands. If both variables are connected, the latest control command is effective. The state of the control command has no significance. The value of the control command is the percentage of the total cycle time which the object stays in ON-state i.e. the power output of the connected device in percentage.

**nvi_xDO_Switch2:** See nvi_xDO_Switch1.

**nvi_xDO_AlrmStop:** This variable is used to lock the object to OFF-state. This happens when the state of this variable is one (1). Locking and unlocking of the object is achieved also by configuring the Locking Level and Locking Release Level, and using the value of nvi_xDO_AlrmStop variable.

**nvo_xDO_Switch:** This variable shows the value of the latest control command received via nvi_xDO_Switch1 or nvi_xDO_Switch2. The state corresponds with the current state of the object.

## Configuration Parameters:

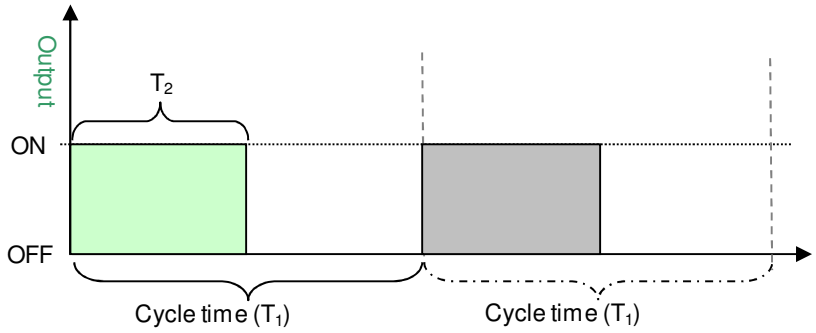
| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |

| | | |
|---|---|---|
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | 0...200 |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |
| **PWM Cycle Time** | This parameter defines the base duty cycle for the pulse width modulation. | Fast (20 s)<br>Slow (200 s) |
| **PWM Cycle Time Multiplier** | Multiplier for the base duty cycle time. With a 20 second base duty cycle time and a multiplier of 4 a cycle time of 80 seconds is obtained. | 0 … 65535 |

## 4.5 PWM-Control (Constant Time)

This function is suitable for the control of signal lights etc. which need to blink constantly. This mode is not suitable for power output control as it is fixed to 50% ouput i.e. the object is in ON-state 50% of the duty cycle.

**PWM Constant Time**




$T_1 = 2 \times T_2$
$T_2$ = nvi_DO_Switch1 x 0.01 x PWM Cycle Time x PWM Cycle Multiplier
(or nvi_DO_Switch2 x 0.01 x PWM Cycle Time x PWM Cycle Multiplier)

## Functionality of Network Variables:

**nvi_xDO_Switch1:** Variables nvi_xDO_Switch1 and nvi_xDO_Switch2 are used for incoming control commands. If both variables are connected, the latest control command is effective. The state of the control command has no significance. The value states the percentage from the cycle time which the object is in ON- and OFF-states. For example with a 20 second cycle time and a 5% (5.0 0) control command the object switches from ON-state to OFF-state every 1 second. With a 100% control command the object is in ON-state for 10 seconds and then switches back to OFF state for another 10 seconds.

**nvi_xDO_Switch2:** See nvi_xDO_Switch1.

**nvi_xDO_AlrmStop:** This variable is used to lock the object to OFF-state. This happens when the state of this variable is one (1). Locking and unlocking of the object is achieved also by configuring the Locking Level and Locking Release Level, and using the value of nvi_xDO_AlrmStop variable.

**nvo_xDO_Switch:** This variable shows the value of the latest control command received via nvi_xDO_Switch1 or nvi_xDO_Switch2. The state corresponds with the current state of the object.

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |

| | | |
|---|---|---|
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | 0...200 |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **(AlarmStop ON) nvoDOswitch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |
| **PWM Cycle Time** | This parameter defines the base duty cycle for the pulse width modulation. | Fast (20 s)<br>Slow (200 s) |
| **PWM Cycle Time Multiplier** | Multiplier for the base duty cycle time. With a 20 second base duty cycle time and a multiplier of 4 a cycle time of 80 seconds is obtained. | 0 … 65535 |

## 4.6 Alarm Priority Control

This function is used in alarm functions to indicate alarms of different alarm classes, e.g. fire alarms, burglar alarms, and HVAC-alarms. The object's physical output can be connected to e.g. alarm signal lights, buzzers, alarm forwarding devices etc. The object monitors all the alarms that are input to the module's nviAlarm network variable. If an alarm with a certain priority is detected this object will switch to ON state. The object will switch back to OFF state only when either of its nvi_xDO_Switch variables receives an input with a value of 0. A module with a DO object of this function type should not have any alarm generating objects as this may lead to unwanted behavior of the Alarm Priority Control object.

**Alarm Priority Control (PRI = DO PRI):**

Alarm information and priority is monitored from the nviAlarm variable of the module the object resides in. If the alarm priority equals the value of "Alarm Priority to Operate" parameter, the object is switched

to ON-state. Only acknowledgement using value 0.0 in variable nvi_xDO_Switch1 or nvi_xDO_Switch2 returns the object to OFF-state.

**Alarm Priority Control (PRI => DO PRI):**

Alarm information and priority is monitored from the nviAlarm variable of the module the object resides in. If the alarm priority equals or is greater than the value of "Alarm Priority to Operate" parameter, the object is switched to ON-state. Only acknowledgement using value 0.0 in variable nvi_xDO_Switch1 or nvi_xDO_Switch2 returns the object to OFF-state.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xDO_Switch1:** | This variable is used for acknowledgement information after an alarm is detected. Acknowledgement is done by entering the value 0.0 in the value field of either of nvi_xDO_Switch variables. The state has no significance. The object's physical state and output variable are switched to OFF-state after an acknowledgement. |
| **nvi_xDO_Switch2:** | See nvi_xDO_Switch1. |
| **nvi_xDO_AlrmStop:** | This variable locks the state of nvo_xDO_Switch and the physical digital output into standstill. Locking takes place when the state is one (1). Also the value of the variable can be used for locking, see configuration parameters of Locking Level and Locking Release Level. |
| **nvo_xDO_Switch:** | If the object is not locked, this variable has always a value of 0. The state of this variable is 1 if the object is in ON state, otherwise the state is 0. |

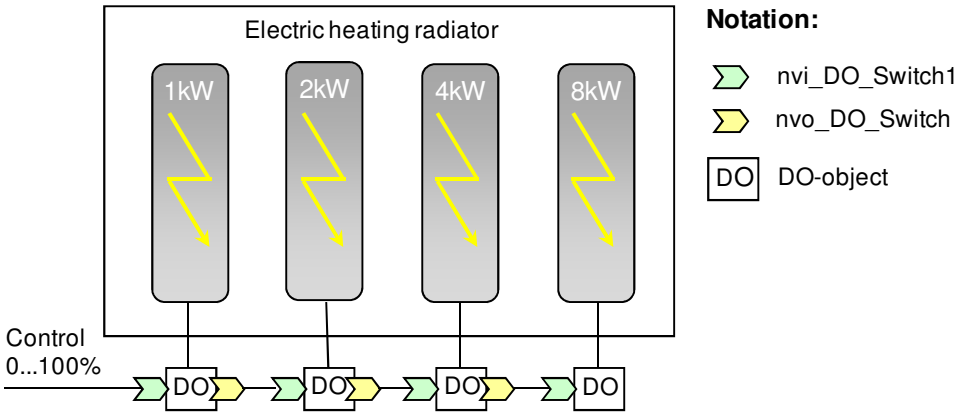
**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value | 0...200 |

| | | |
|---|---|---|
| | of nvi_xDO_AlrmStop. | |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **Alarm Priority to Operate** | This parameter defines the alarm priority that object should operate. The object only responds to alarms of certain priority (Alarm Priority Control (PRI = DO PRI)) or all alarms with a priority equal or greater than the value defined with this parameter (Alarm Priority Control (PRI => DO PRI)). | 0 … 7 |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |

## 4.7 Binary Step

This function is used in electrical heating applications and compressor power adjustment applications when the system is controlled through several devices with different power properties (in most cases the next device has a double power property compared to the previous device to achieve linear power output steps for the whole system). Each of the devices is controlled with a single DO object attached in series with the DO objects of the other devices.



For example, an electrical heating radiator might consist of 4 separate heating resistors with different power outputs. As the control value increases the first resistor is switched on. Further increasing the control value the first resistor is switched off and the second on. If the control value still increases also the first resistor is switched on to achieve a small additional increase in heating power. Next power increase step would be switching the first and the second resistors off and turning on the third resistor. Thus with a binary sequence (each bit corresponding to the state information (ON/OFF) of each device) multiple levels of total power output of the system is achieved (in this case a total of 24=16 power levels).

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xDO_Switch1:** | The control command from another object or from a controller object is bound to nvi_xDO_Switch1 or to nvi_xDO_Switch2. Latest command is effective. |
| **nvi_xDO_Switch2:** | See nvi_xDO_Switch1. |
| **nvi_xDO_AlrmStop:** | This variable locks the state of nvo_xDO_Switch and the physical digital output into standstill. Locking takes place when the state is one (1). Also the value of the variable can be used for locking, see configuration parameters of Locking Level and Locking Release Level. |
| **nvo_xDO_Switch:** | This variable passes on the control command received with the difference that the state of the output variable is adjusted according to the object's physical state (ON = 1, OFF = 0) calculated from the control command. This variable is bound to the nvi_xDO_Switch1/2 of the next Binary Step object of the binary sequence. |

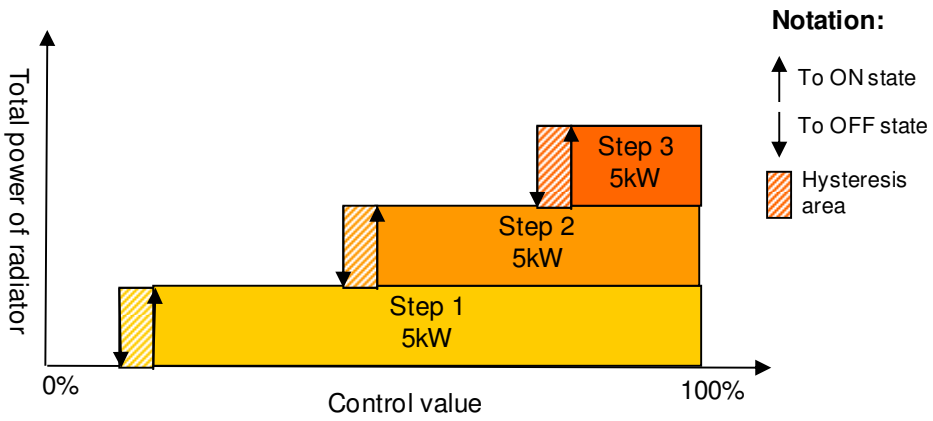
## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | 0...200 |

| | | |
|---|---|---|
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **Binary Step Selection** | The binary position of the object in the sequence of binary step DO objects. | 1..7 |
| **Amount of Binary Steps** | The number of DO objects in this binary step sequence. | 2..7 |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |

## 4.8 Linear Step

This function is used in electrical heating applications and compressor power adjustment applications, when the system is controlled through several devices with similar power properties. The system consists of DO objects of Linear Step type connected in series. Control of the devices with similar power properties are connected to physical outputs of these DO-objects.




For example, the system includes an electrical heating radiator with three similar resistors (5kW, 5kW and 5kW). As control value increases, the first step is turned on. As the value further increases, the

second one turned on. When the value still increases, the third step is also turned on. Combined output power is then 15kW (5kw+5kW+5kW=15kW).

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xDO_Switch1:** | The control command from another object or from a controller object is bound to nvi_xDO_Switch1 or to nvi_xDO_Switch2. Latest command is effective. |
| **nvi_xDO_Switch2:** | See nvi_xDO_Switch1. |
| **nvi_xDO_AlrmStop:** | This variable locks the state of nvo_xDO_Switch and the physical digital output into standstill. Locking takes place when the state is one (1). Also the value of the variable can be used for locking, see configuration parameters of Locking Level and Locking Release Level. |
| **nvo_xDO_Switch:** | This variable passes on the control command received with the difference that the state of the output variable is adjusted according to the object's physical state (ON = 1, OFF = 0). This variable is bound to the nvi_xDO_Switch1/2 of the next Linear Step object of the linear step sequence. |

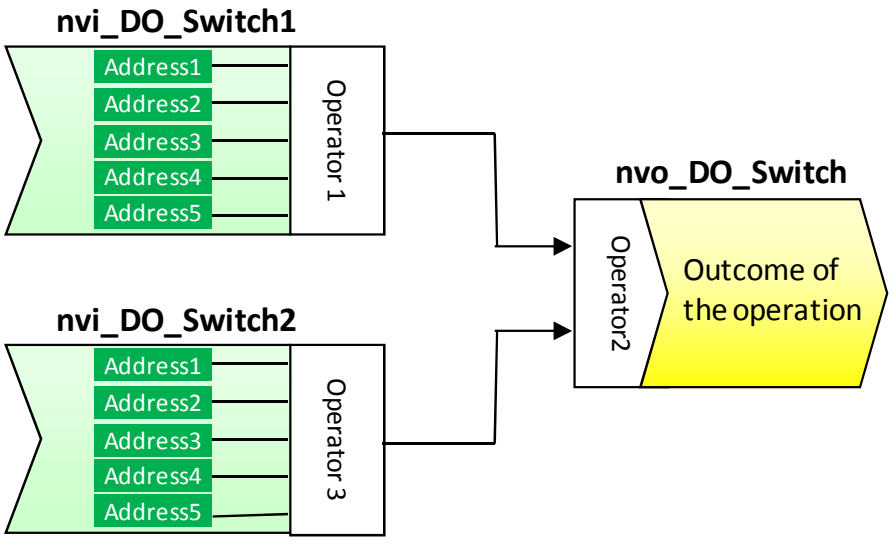
## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | 0...200 |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br><br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when | Yes<br>No |

| | | |
|---|---|---|
| | the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br><br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked).<br><br>Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |
| **Off Level** | A control value less than this parameter will change the objects to OFF state. | 0...200 |
| **On Level** | A control value greater than this parameter will change the object to ON state. | 0...200 |

## 4.9 Logical

This function is used for logical operations. It consists of three (3) logical operators and a maximum of ten (10) signals bound to two input network variables. First the logical operation is carried out individually for both input network variables. After that the truth value of the total system is determined with the truth values of the two input network variables, and the physical output of the object is functioning according to the truth value (TRUE=on, FALSE=off). This can be visualized in the following way:




The logical operators for the different operation types of this function are listed in the following table.

| Operation type | Operator 1 (nvi_DO_Switch1) | Operator 2 (nvo_DO_Switch) | Operator 3 (nvi_DO_Switch2) |
|---|---|---|---|
| **OR** | OR | OR | OR |
| **AND** | AND | AND | AND |
| **AND-AND-OR** | AND | AND | OR |
| **AND-OR-AND** | AND | OR | AND |
| **AND-OR-OR** | AND | OR | OR |
| **OR-AND-OR** | OR | AND | OR |

An incoming signal is considered to be TRUE if the value is greater than zero (0) or its state is one (1). Otherwise it is FALSE.

**NOTE!**

The object needs to be taught the state information of the input network variables in order to initialize the state information in the module's address table. This is done by changing the physical state of each bound object once from OFF state to ON state and back to OFF state (e.g. short circuiting the connection strips of the DI objects' physical input or controlling them in Manual mode).

When the function of a DO object is changed to this type, the state of its nvo_xDO_Switch will be -1. This indicates that the object has not been initialized yet. After each object bound to the object's nvi_xDO_Switch1 and nvi_xDO_Switch2 has been changed to ON state (and back to OFF state) the state will be zero and the object is ready for use.

Do not initialize the state information by inputting values directly to this DO object's nvi_xDO_Switch1 and nvi_xDO_Switch2 variables manually from PCT. It will result in object's address table to be filled with your LON network interface address instead of the real LON addresses of the objects involved in the logical operation!

**Functionality of Network Variables:**

**nvi_xDO_Switch1:** The input for the first logical operator is bound to nvi_xDO_Switch1. Maximum of five (5) bindings can be made.

**nvi_xDO_Switch2:** The input for the third logical operator is bound to nvi_xDO_Switch2. A maximum of five (5) bindings can be made.

**nvi_xDO_AlrmStop:** This variable is used to lock the object to OFF-state. This happens when the state of this variable is one (1). Locking and unlocking of the object is achieved also by configuring the Locking Level and Locking Release Level, and using the value of nvi_xDO_AlrmStop variable.

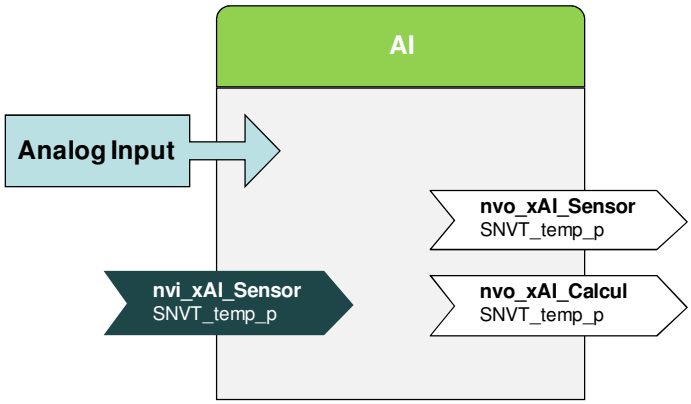
**nvo_xDO_Switch:** This variable shows the outcome of the logical operations: FALSE = 0.0 0, TRUE = 0.5 1.

If the object's Operation type is "OR", then the value of the outcome equals the number of TRUE signals bound to nvi_xDO_Switch1 and nvi_xDO_Switch2 (e.g. 3.5 1 indicates that 7 of the incoming signals are TRUE and hence also the physical state of the object is TRUE).

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **On Delay** | This parameter defines the ON delay of the object's physical state. When the object's state changes from OFF to ON, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Off Delay** | This parameter defines the OFF delay of the object's physical state. When the object's state changes from ON to OFF, the values of the output network variables are updated only after this time has elapsed. | 0 … 65535s |
| **Switching Mode** | When value Low(-) is selected, the object disconnects in OFF state only minus terminal from the power supply. When High(+) is used, only plus terminal is disconnected from power supply. Value High(+) and Low(-) uses both terminals to switch the output ON and OFF. | Low(-)<br>High(+)<br>High(+) and Low(-) |
| **Locking Level** | Locking level of the object. The value of this parameter defines the limit level for the value field of nvi_xDO_AlrmStop variable, higher values causing the object to lock into standstill. To enable locking with only the value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). If the state of nvi_xDO_AlrmStop is 1, then the DO object is locked regardless of the value of nvi_xDO_AlrmStop. | 0...200 |
| **Locking Release Level** | Object locking release level. The value of this parameter defines the limit level for value field of nvi_xDO_AlrmStop variable, lower values releasing the object from standstill. To enable locking with value field, the state field of nvi_xDO_AlrmStop variable needs to be zero (0). | 0...200 |
| **Invert** | This parameter inverts the state of the output network variable and the physical state of the object.<br>NOTE! Object locking with nvi_xDO_AlrmStop variable functions also when the object is inverted. The state of the nvi_xDO_AlrmStop variable still needs to be zero for the object locking to work. | Yes<br>No |
| **(AlarmStop ON) nvo_xDO_Switch value.active** | The influence of the object's locking functionality on the output network variable. The value of this parameter is the value shown in nvo_xDO_Switch (state is 0) variable when the object is locked and this parameter is active (1).<br>When this parameter is not active (0), the value of nvo_xDO_Switch follows that of the nvi_xDO_Switch1/2 variable (with a state of 0 if the object is locked). Example: If this parameter is set to 32.01 and the object is locked, then nvo_xDO_Switch will be 16.0 0. | Value 0…200<br>Active 0…1 |

# 5. Analog Input (AI)




**The Analog Input (AI) includes the following functions:**

- Pt-1000 Normal / Maximum / Minimum / Sub / Average
- Ni-1000-LG Normal / Maximum / Minimum / Sub / Average
- Ni-1000-DIN Normal / Maximum / Minimum / Sub / Average
- Active (0-10V) Normal / Maximum / Minimum / Sub / Average
- Resistance Normal / Maximum / Minimum / Sub / Average
- Condensation point (dew point)

***Operation types for measurement functions:***

| | |
|---|---|
| Normal: | The object measures the value of the connected sensor and outputs it to nvo_xAI_Sensor. There is no function for nvi_xAI_Sensor in this case. |
| Maximum: | The object measures the value of the connected sensor and outputs the result to nvo_xAI_Sensor. The calculated maximum of the measured value and the value of nvi_xAI_Sensor is output to nvo_xAI_Calcul. |
| Minimum: | The object measures the value of the connected sensor and outputs the result to nvo_xAI_Sensor. The calculated minimum of the measured value and the value of nvi_xAI_Sensor is output to nvo_xAI_Calcul. |
| Sub: | The object measures the value of the connected sensor and outputs the result to nvo_xAI_Sensor. The calculated difference of the measured value and the value of nvi_xAI_Sensor is output to nvo_xAI_Calcul. |
| Average: | The object measures the value of the connected sensor and outputs the result to nvo_xAI_Sensor. The calculated average of the measured value and the value of nvi_xAI_Sensor is output to nvo_xAI_Calcul. |

## 5.1 Passive Sensors: Pt-1000 / Ni-1000-LG / Ni-1000-DIN / Resistance

These functions are used when connecting passive sensor elements to AI object. Function type has to be selected according to the sensor element type (Pt-1000, Ni-1000-LG or Ni-1000-DIN). Resistance function can be used to show a raw resistance value of AI input without any sensor element type based scaling. It is capable of measuring resistance value from 0 to 2100 Ohm. All passive sensor functions include Normal, Maximum, Minimum, Sub and Average operation types.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xAI_Sensor:** | Input values for the desired operation (Maximum, Minimum, Sub or Average). |
| **nvo_xAI_Sensor:** | Shows the measurement value of the sensor attached to the object. |
| **nvo_xAI_Calcul:** | Shows the outcome of the operation defined by the operation type (Maximum, Minimum, Sub, Average). In the *Normal* operation type, nvo_xAI_Calcul shows the measurement value of the attached sensor. |

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **Trending NV** | This parameter selects which of the two output variables is used for trending. | nvo_xAI_Sensor<br>nvo_xAI_Calcul |
| **Alarming NV** | This parameter selects which of the two output variables is used for alarm triggering. | nvo_xAI_Sensor<br>nvo_xAI_Calcul |
| **Alarm Priority** | The priority of the alarms generated by this object. If the parameter is zero (0) then no alarms are generated. The higher the value, the more important the alarm is. | 0 … 7 |
| **Alarm On Delay** | This parameter defines the time that the value of the appropriate output variable (configured via Alarming NV) needs to exceed/fall below the alarm limit, before an alarm is generated.<br>NOTE! An AI object that needs to generate alarms must have an Alarm On Delay greater than zero (0). Otherwise no alarms are generated. | 0 … 65535s |
| **Alarm Off Delay** | This parameter defines the time that the value of the appropriate output variable (configured via Alarming NV) needs to fall below/exceed the alarm limit, before the alarm is inactivated. | 0 … 65535s |
| **HiHi-Level Alarm** | High high level limit for the alarm. If the measurement exceeds this value, then an alarm with a priority of +1 to the configured alarm priority is generated. | -250 … +300 |
| **Hi-Level Alarm** | High level limit for the alarm. If the measurement exceeds this value, then an alarm with the configured alarm priority is generated. | -250 … +300 |
| **Lo-Level Alarm** | Low level limit for the alarm. If the measurement falls below this value, then an alarm with the configured alarm priority is generated. | -250 … +300 |

| | | |
|---|---|---|
| **LoLo-Level Alarm** | Low low level limit for the alarm. If the measurement falls below this value, then an alarm with a priority of +1 to the configured alarm priority is generated. | -250 … +300 |
| **Offset** | The offset for the measurement reading can be set with this parameter. It can be used to shift the zero point of the measurement e.g. in field calibration. | -250 … +300 |
| **Sensitivity** | The sensitivity for the measurement output. The value of the output variables is not updated unless the measurement changes more than the value of this parameter. | 0 … 250 |
| **Factor** | Only used in Resistance mode. Used for scaling the output value.<br>Output value = (Factor x resistance value) x 0,001 | 0 … 327 |

## 5.2 Active (0-10V)

Active (0-10V) function is used when connecting transmitters or other 0-10V voltage transmitting devices to AI object. It is also possible to connect 4-20mA transmitters by adding external load resistor (500 Ohm) parallel to AI object's terminals. This function includes Normal, Maximum, Minimum, Sub and Average operation types.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xAI_Sensor:** | Input values for the desired operation. |
| **nvo_xAI_Sensor:** | Shows the measurement value of the sensor attached to the object. |
| **nvo_xAI_Calcul:** | Shows the outcome of the operation defined by the operation type (Maximum, Minimum, Sub, Average). In the *Normal* operation type, nvo_xAI_Calcul shows the measurement value of the attached sensor. |

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **Trending NV** | This parameter selects which of the two output variables is used for trending. | nvo_xAI_Sensor<br>nvo_xAI_Calcul |
| **Alarming NV** | This parameter selects which of the two output variables is used for alarm triggering. | nvo_xAI_Sensor<br>nvo_xAI_Calcul |
| **Alarm Priority** | The priority of the alarms generated by this object. If the parameter is zero (0) then no alarms are generated. The higher the value, the more important the alarm is. | 0 … 7 |
| **Alarm On Delay** | This parameter defines the time that the value of the appropriate output variable (configured via Alarming NV) needs to exceed/fall below the alarm limit, before an alarm is generated.<br>NOTE! An AI object that needs to generate alarms must have an Alarm On | 0 … 65535s |

| | | |
|---|---|---|
| | Delay greater than zero (0). Otherwise no alarms are generated. | |
| **Alarm Off Delay** | This parameter defines the time that the value of the appropriate output variable (configured via Alarming NV) needs to fall below/exceed the alarm limit, before the alarm is inactivated. | 0 … 65535s |
| **HiHi-Level Alarm** | High high level limit for the alarm. If the measurement exceeds this value, then an alarm with a priority of +1 to the configured alarm priority is generated. | -250 … +300 |
| **Hi-Level Alarm** | High level limit for the alarm. If the measurement exceeds this value, then an alarm with the configured alarm priority is generated. | -250 … +300 |
| **Lo-Level Alarm** | Low level limit for the alarm. If the measurement falls below this value, then an alarm with the configured alarm priority is generated. | -250 … +300 |
| **LoLo-Level Alarm** | Low low level limit for the alarm. If the measurement falls below this value, then an alarm with a priority of +1 to the configured alarm priority is generated. | -250 … +300 |
| **Offset** | The offset for the measurement reading can be set with this parameter. It can be used to shift the zero point of the measurement e.g. in field calibration. | -250 … +300 |
| **Sensitivity** | The sensitivity for the measurement output. The value of the output variables is not updated unless the measurement changes more than the value of this parameter. | 0 … 250 |
| **Range** | This parameter defines the total operating range of the sensor attached to the object. For example if the operating range of the sensor is from -50 to +50, then this parameter needs to be set at 100. The true lower limit of the sensor's operating range can then be set with the Offset parameter. In this case, however, as the object is connected to a relative humidity measurement, the range should be 100 (and the offset 0). | 0 … 327 |

## 5.3 Condensation Point (Dew Point)

This function calculates the dew point in Celsius degrees. The calculation is based on a temperature measurement bound to nvi_xAI_Sensor and a relative humidity measurement from a humidity sensor physically attached to the object. This function is used e.g. for adjustment of the minimum temperature in a cooling water network to prevent condensation on the surfaces of pipes and cooling elements.

**NOTE!**

For the dew point calculation to be accurate the temperature measurement should be between 0ºC and 30ºC. The relative humidity measurement should be between 20% and 100%.

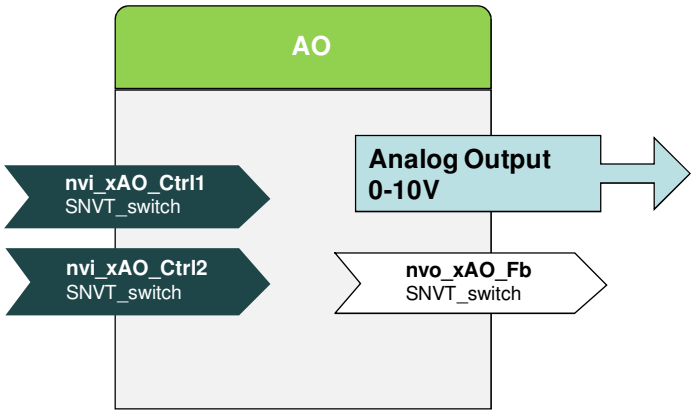
**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xAI_Sensor:** | Used to input the temperature measurement for the Dew Point calculation. |
| **nvo_xAI_Sensor:** | Shows the measurement value of the humidity sensor attached to the object. |
| **nvo_xAI_Calcul:** | Shows the calculated Dew Point value. |

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Trending NV** | This parameter selects which of the two output variables is used for trending. | nvo_xAI_Sensor<br>nvo_xAI_Calcul |
| **Alarming NV** | This parameter selects which of the two output variables is used for alarm triggering. | nvo_xAI_Sensor<br>nvo_xAI_Calcul |
| **Alarm Priority** | The priority of the alarms generated by this object. If the parameter is zero (0) then no alarms are generated. The higher the value, the more important the alarm is. | 0 … 7 |
| **Alarm On Delay** | This parameter defines the time that the value of the appropriate output variable (configured via Alarming NV) needs to exceed/fall below the alarm limit, before an alarm is generated.<br>NOTE! An AI object that needs to generate alarms must have an Alarm On Delay greater than zero (0). Otherwise no alarms are generated. | 0 … 65535s |
| **Alarm Off Delay** | This parameter defines the time that the value of the appropriate output variable (configured via Alarming NV) needs to fall below/exceed the alarm limit, before the alarm is inactivated. | 0 … 65535s |
| **HiHi-Level Alarm** | High high level limit for the alarm. If the measurement exceeds this value, then an alarm with a priority of +1 to the configured alarm priority is generated. | -250 … +300 |
| **Hi-Level Alarm** | High level limit for the alarm. If the measurement exceeds this value, then an alarm with the configured alarm priority is generated. | -250 … +300 |
| **Lo-Level Alarm** | Low level limit for the alarm. If the measurement falls below this value, then an alarm with the configured alarm priority is generated. | -250 … +300 |
| **LoLo-Level Alarm** | Low low level limit for the alarm. If the measurement falls below this value, then an alarm with a priority of +1 to the configured alarm priority is generated. | -250 … +300 |
| **Offset** | The offset for the measurement reading can be set with this parameter. It can be used to shift the zero point of the measurement e.g. in field calibration. | -250 … +300 |
| **Sensitivity** | The sensitivity for the measurement output. The value of the output variables is not updated unless the measurement changes more than the value of this parameter. | 0 … 250 |
| **Range** | This parameter defines the total operating range of the sensor attached to the object. For example if the operating range of the sensor is from -50 to +50, then this parameter needs to be set at 100. The true lower limit of the sensor's operating range can then be set with the Offset parameter. In this case, however, as the object is connected to a relative humidity measurement, the range should be 100 (and the offset 0). | 0 … 327 |

# 6. Analog Output (AO)




**The Analog Output (AO) includes the following functions:**

- 0-100% Control
- Maximum Control
- Minimum Control
- Scene Control

## 6.1 0-100% Control

This function is used in continuous control (0-10V) of valve actuators, dimmers, or speed settings of Variable Frequency Drives, for example.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xAO_Ctrl1:** | Used for controlling the physical output of the object. The latest value update to either nvi_xAO_Ctrl1 or nvi_xAO_Ctrl2 is effective for the control. |
| **nvi_xAO_Ctrl2:** | Equal to nvi_xAO_Ctrl1, see above. |
| **nvo_xAO_Fb:** | The output value of the object. This variable shows a percentage value of the output range (configured with Min Operation Value and Max Operation Value). |

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **Invert** | This parameter inverts the output value. It defines the influence of the input network variables on the output network variable and the output voltage of the object.<br>Range definitions:<br>Output invert: The output voltage of the object is inverted.<br>Range invert: The value of nvo_xAI_Fb is inverted.<br>Range+Output invert: Both of above. | No<br>Output invert<br>Range invert<br>Range+Output invert |
| **Min Operation Value** | This parameter defines the minimum value of the object's range of operation with only larger input values having an influence on the object. For example a parameter value of 50 means that the value of nvo_xAO_Fb remains zero (0) until the value of nvi_xAO_Ctrl exceeds 25%. | 0..200 |
| **Max Operation Value** | This parameter defines the maximum value of the object's range of operation with larger input values having no influence. For example, a parameter value of 150 means that the value of nvo_xAO_Fb will be 100% with all inputs equal to or greater than 75%. | 0..200 |
| **Min Output Voltage** | This parameter defines the minimum physical output voltage of the object. For example if this parameter is set at 2.0 then the output voltage of the object is 2V when nvi_xAO_Ctrl is zero (0). | 0..10 Volts |
| **Max Output Voltage** | This parameter defines the maximum physical output voltage of the object. For example if this parameter is set at 8.0 then the output voltage of the object is 8V when nvi_xAO_Ctrl is 100. | 0..10 Volts |
| **Value of State -field** | Selects how the state field of the nvo_xAO_Fb is functioning. | 1 when output > 0%<br>1 when output = 100%<br>Follow Input State |

## 6.2 Maximum / Minimum Control

This function is commonly used e.g. in air-conditioning applications for preventing frost to form on the heat recovery unit.

**Maximum Control:**

The object selects the higher input value of the two input network variables as its control value.

**Minimum Control:**

The object selects the lower input value of the two input network variables as its control value.

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xAO_Ctrl1:** | Used for controlling the physical output of the object. The latest value update to either nvi_xAO_Ctrl1 or nvi_xAO_Ctrl2 is effective for the control. |
| **nvi_xAO_Ctrl2:** | Equal to nvi_xAO_Ctrl1, see above. |

| | |
|---|---|
| **nvo_xAO_Fb:** | The output value of the object. This variable shows a percentage value of the output range (configured with Min Operation Value and Max Operation Value). |

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **Invert** | This parameter inverts the output value. It defines the influence of the input network variables on the output network variable and the output voltage of the object.<br>Range definitions:<br>Output invert: The output voltage of the object is inverted.<br>Range invert: The value of nvo_xAI_Fb is inverted.<br>Range+Output invert: Both of above. | No<br>Output invert<br>Range invert<br>Range+Output invert |
| **Min Operation Value** | This parameter defines the minimum value of the object's range of operation with only larger input values having an influence on the object. For example a parameter value of 50 means that the value of nvo_xAO_Fb remains zero (0) until the value of nvi_xAO_Ctrl exceeds 25%. | 0..200 |
| **Max Operation Value** | This parameter defines the maximum value of the object's range of operation with larger input values having no influence. For example, a parameter value of 150 means that the value of nvo_xAO_Fb will be 100% with all inputs equal to or greater than 75%. | 0..200 |
| **Min Output Voltage** | This parameter defines the minimum physical output voltage of the object. For example if this parameter is set at 2.0 then the output voltage of the object is 2V when nvi_xAO_Ctrl is zero (0). | 0..10 Volts |
| **Max Output Voltage** | This parameter defines the maximum physical output voltage of the object. For example if this parameter is set at 8.0 then the output voltage of the object is 8V when nvi_xAO_Ctrl is 100. | 0..10 Volts |
| **Value of State -field** | Selects how the state field of the nvo_xAO_Fb is functioning. | 1 when output > 0%<br>1 when output = 100%<br>Follow Input State |

## 6.3 Scene Control

This function is mainly used for controlling lighting systems (e.g. dimmers) in different scenes. The scene numbers can be selected with e.g. push buttons connected to DI objects.

**Functionality of Network Variables:**

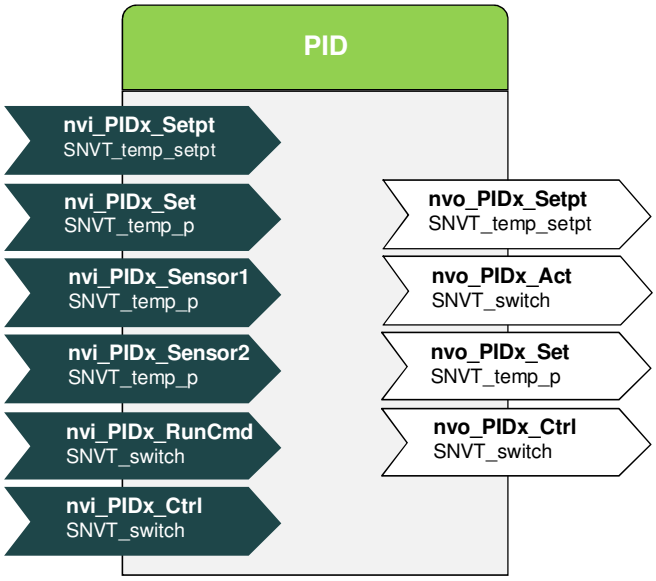
| | |
|---|---|
| **nvi_xAO_Ctrl1:** | Used for continuous control of the physical output. The input value defines the output of the object the same way as in 0-100% Control function. |
| **nvi_xAO_Ctrl2:** | Used to input current scene number. To activate the scene control the value of the input must equal one of the configured scene numbers (see Configuration Parameters). If the value does not equal any of the configured scene numbers the scene input is |

| | |
|---|---|
| | ignored. |
| **nvo_xAO_Fb:** | The output value of the object. This variable shows a percentage value of the output range (configured with Min Operation Level and Max Operation Level parameters) determined with the input value of nvi_xAO_Ctrl1 or a scene triggered with nvi_xAO_Ctrl2. |

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Invert** | This parameter inverts the output value. It defines the influence of the input network variables on the output network variable and the output voltage of the object.<br>Range definitions:<br>Output invert: The output voltage of the object is inverted.<br>Range invert: The value of nvo_xAI_Fb is inverted.<br>Range+Output invert: Both of above. | No<br>Output invert<br>Range invert<br>Range+Output invert |
| **Min Operation Value** | This parameter defines the minimum value of the object's range of operation with only larger input values having an influence on the object. For example a parameter value of 50 means that the value of nvo_xAO_Fb remains zero (0) until the value of nvi_xAO_Ctrl exceeds 25%. | 0..200 |
| **Max Operation Value** | This parameter defines the maximum value of the object's range of operation with larger input values having no influence. For example, a parameter value of 150 means that the value of nvo_xAO_Fb will be 100% with all inputs equal to or greater than 75%. | 0..200 |
| **Min Output Voltage** | This parameter defines the minimum physical output voltage of the object. For example if this parameter is set at 2.0 then the output voltage of the object is 2V when nvi_xAO_Ctrl is zero (0). | 0..10 Volts |
| **Max Output Voltage** | This parameter defines the maximum physical output voltage of the object. For example if this parameter is set at 8.0 then the output voltage of the object is 8V when nvi_xAO_Ctrl is 100. | 0..10 Volts |
| **Value of State -field** | Selects how the state field of the nvo_xAO_Fb is functioning. | 1 when output > 0%<br>1 when output = 100%<br>Follow Input State |
| **Scene Configuration 1** | The physical output of the object with a specific scene number. The desired level of output is entered first, then the specific scene number, separated with a dot. For example 75% output on scene 3 means a parameter value of 150.06. | Output 0..200<br>Scene nr 0..99 |
| **Scene Configuration 2** | See Scene Configuration 1 | Output 0..200<br>Scene nr 0..99 |
| **Scene Configuration 3** | See Scene Configuration 1 | Output 0..200<br>Scene nr 0..99 |
| **Scene Configuration 4** | See Scene Configuration 1 | Output 0..200<br>Scene nr 0..99 |
| **Scene Configuration 5** | See Scene Configuration 1 | Output 0..200<br>Scene nr 0..99 |
| **Scene Configuration 6** | See Scene Configuration 1 | Output 0..200<br>Scene nr 0..99 |

# 7. Controller (PID)




**The Controller (PID) includes the following functions:**

- Constant (Normal)
- Constant (Cascade)
- Curve
- Thermostat functions
- Linear Setpoint (nvi_xPID_Ctrl)
- Linear Setpoint (nvi_xPID_Sensor2)
- Setpoint Calculation

**NOTE:** The PID controller should be tuned separately for each application (not relevant for Thermostat functions and Setpoint Calculation). There are some rules of thumb to help this.

- Large gain shortens the rise time, but on the downside it also increases overshoot.
- Integration corrects steady state errors, but if TI is too short it can cause oscillations and can make the system unstable.
- Derivation makes the controller faster and removes oscillations, but if TD is too short it makes the system very sensitive to high frequency noise and delays.
- Use derivation with caution, in 99% of cases a PI-controller is enough.

## 7.1 Constant (Normal)

This function is used with applications that use constant continuous control. The controller adjusts the attached actuator with respect to the process measurement and the configured setpoint (reference). If the process measurement resides in the same module with the controller the process measurement can be brought to the controller via an internal binding. The control command for the actuator can also be sent via an internal binding.

*See also: Configuration parameters Sensor Selection, Actuator Selection.*

For example, the domestic hot water temperature is fixed at 58ºC (controller setpoint). If the process measurement i.e. the current domestic hot water temperature differs from the setpoint, the controller opens/closes the attached actuator (a valve regulating the flow of heating water that is used to heat the domestic hot water circuit) to maintain the domestic hot water temperature at 58ºC.

The controller can be configured for six separate scenes, which deviate the configured setpoint with the value in nvi_xPID_Sensor2 or the three last fields in nvi_xPID_Setpt.

**Functionality of Network Variables:**

**nvi_xPID_Setpt:** This variable determines the setpoint of the controller. The six fields of the variable in Normal mode are as follows:

| Field no. in nvi_xPID_Setpt | Meaning |
|---|---|
| 1 | Setpoint |
| 2 | Not in use |
| 3 | Not in use |
| 4 | Deviation value for Scene 4 |
| 5 | Deviation value for Scene 5 |
| 6 | Deviation value for Scene 6 |

**nvi_xPID_Set:** This variable deviates the setpoint of the controller with this value. If the controller is acting as a pilot controller in a cascade control loop, this variable is bound to nvo_xPID_Set of the cascade main controller to receive current setpoint information.

**nvi_xPID_Sensor1:** This variable is the input for process measurement.

**nvi_xPID_Sensor2:** This variable the deviation for the setpoint during Scenes 1, 2 and 3.

**nvi_xPID_RunCmd:** Input for run command to activate the controller. The controller is activated when the value or state of the variable is greater than 0. If not activated, the actuator output is the value configured with ”Actuator OFF-state Value” parameter.

**nvi_xPID_Ctrl:** This variable is used for scene number input. The value of nvi_xPID_Ctrl is regarded as the scene number. The state has no significance.

**nvo_xPID_Setpt:** This variable outputs the value of nvi_xPID_Setpt.

| | |
|---|---|
| **nvo_xPID_Set:** | This variable outputs the current setpoint of the controller. |
| **nvo_xPID_Ctrl:** | This variable outputs the value of nvi_xPID_Ctrl. |
| **nvo_xPID_Actuator:** | This variable is the control value for the actuator. The state of the variable is 1 only when the value is 100. Otherwise the state is 0. |

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Start Up Ramping Time** | Linearly adjusts the setpoint of the controller from the initial process measurement value to the configured setpoint during the time set with this parameter (disabled with 0). This feature is intended for smooth start ups of processes. The function also activates after power failures. | 0..32767 |
| **Proportional** | The gain KP of the PID controller. | -100..100 |
| **Integration Time** | Integration time TI of the PID controller. Integration disabled if this is set to zero (0). The longer the time the more effect it has on the control. | 0..9999 s |
| **Derivation Time** | Derivation time TD of the PID controller. Derivation disabled if this is set to zero (0). The longer the time the more effect it has on the control. | 0..9999 s |
| **Sensitivity of PID** | This parameter defines the minimum deviation of the measurement from the reference point (setpoint) before the controller's output is adjusted. | 0..100 |
| **Invert** | The controller normally adjusts the system with respect to the difference between the reference (setpoint) and the process measurement. With this parameter the difference can be inverted i.e. the controller adjusts the system with the difference between the process measurement and the reference. Also the actuator output of the controller can be inverted which is useful for example in cooling systems, when a valve drive actually needs to open when the temperature is rising. | No<br>PID invert<br>Actuator invert<br>Actuator+PID invert |
| **Sensor Selection (nvi_xPID_Sensor1)** | Selection of the process measurement for the controller. The process measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need of using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Sensor Selection (nvi_xPID_Sensor2)** | Selection of the compensation measurement for the controller. The compensation measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need for using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Hi-Level Sliding Alarm** | High level alarm limit for the process measurement. The actual high level alarm limit is determined with this parameter and the setpoint of the process i.e. if the setpoint is 20 and this parameter 5, then the high level alarm threshold is 25. Also when using this parameter the Hi- and HiHi Alarm Limit parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | -255..+255 |
| **Lo-Level Sliding Alarm** | Low level alarm limit for the process measurement. The actual low level alarm limit is determined with this parameter and the setpoint of the process | -255..+255 |

| | | |
|---|---|---|
| | i.e. if the setpoint is 20 and this parameter -5, then the low level alarm threshold is 15. Also when using this parameter the Lo- and LoLo Alarm Limit parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | |
| **(Sensor2)SceneNo1.(Sensor2)SceneNo2** | The controller behaviour i.e. the setpoint can be adjusted with predetermined scenes. The scenes are marked with unique identifiers (numbers) that are sent to the controller via nvi_xPID_Ctrl. When the value of nvi_xPID_Ctrl matches that of a configured scene, the setpoint of the controller is deviated from the normal setpoint by an amount defined with nvi_xPID_Sensor2.<br>For example the temperature of a room is lowered by 3 degrees during night time. The scene number for this is defined as 11. The value of this parameter is then configured as 11.99. When an SNVT_switch of value 5.5 is input to nvi_xPID_Ctrl and nvi_xPID_Sensor2 has the value -3.0, the controller lowers the setpoint by 3 degrees. This scene is active until a new scene number is input to nvi_xPID_Ctrl (if it is not a valid scene number, the controller will return back to its normal state).<br>Note! A scene number greater than 99 must be configured in the first part of the parameter. Scene numbers lower than 10 defined in the second part of the parameter must have a zero prefix e.g. 120.04. | 0.0...200.99 |
| **(Sensor2)SceneNo3.(SPfield4)SceneNo4** | See: (Sensor2)SceneNo1.(Sensor2)SceneNo2.<br>Note! If a scene number defined in the second part of the parameter activates, the deviation for the setpoint is the fourth field of nvi_xPID_Setpt variable, not the value of nvi_xPID_Sensor2. | 0.0...200.99 |
| **(SPfield5)SceneNo5.(SPfield6)SceneNo6** | See: (Sensor2)SceneNo1.(Sensor2)SceneNo2.<br>Note! If a scene number defined in this parameter activates, the deviation for the setpoint is the fifth/sixth field of nvi_xPID_Setpt parameter, not the value of nvi_xPID_Sensor2. | 0.0...200.99 |
| **Actuator Selection** | When an internal binding is used, this parameter defines the controlled AO object's input inside the same module. If an internal binding is not used, this should be set to "nvo_xPID_Act". | nvo_xPID_Act<br>nvi_x1AO_Ctrl1…<br>nvi_x5AO_Ctrl1<br>nvi_x1AO_Ctrl1 +<br>nvi_x2AO_Ctrl1 |
| **Actuator Range(Max.Min)** | The maximum and minimum output values of nvo_xPID_Act can be set with this parameter. | 0.0...200.99 |
| **Actuator OFF-State Value** | OFF-state value of the controller. When the controller is not running (no permission to run), the nvo_xPID_Act is set to this value. | 0..200 |

## 7.2 Constant (Cascade)

In Cascade mode the controller is usually the main controller of compensation in air-conditioning. The compensated setpoint is output via nvo_xPID_Set variable and it is bound to the nvi_xPID_Set of the pilot controller. The range for nvo_xPID_Set can be configured with the nvi_xPID_Setpt variable. When using cascade mode, it is recommended that a P-controller with a gain of 1 is used (i.e. Integration Time and Derivation Time are 0, Proportional = 1).

If the process measurement resides in the same module with the controller the process measurement can be brought to the controller via an internal binding. The control command for the actuator can also be sent via an internal binding. See: Configuration parameters Sensor Selection, Actuator Selection.

The controller can be configured for six separate scenes, which deviate the configured setpoint with the value in nvi_xPID_Sensor2 or the three last fields in nvi_xPID_Setpt.

## Functionality of Network Variables:

**nvi_xPID_Setpt:** The fields of the variable in Cascade mode are as follows:

| Field no. in nvi_xPID_Setpt | Meaning |
|---|---|
| 1 | Setpoint or deviation |
| 2 | Minimum value of nvo_xPID_Set |
| 3 | Maximum value of nvo_xPID_Set |
| 4 | Deviation value for Scene 4 |
| 5 | Deviation value for Scene 5 |
| 6 | Deviation value for Scene 6 |

**nvi_xPID_Set:** The first field of nvi_xPID_Setpt is then used to deviate this value.

**nvi_xPID_Sensor1:** This variable is the input for process measurement.

**nvi_xPID_Sensor2:** This variable the deviation for the setpoint during Scenes 1, 2 and 3.

**nvi_xPID_RunCmd:** Input for run command to activate the controller. The controller is activated when the value or state of the variable is greater than 0. If not activated, the actuator output is the value configured with ”Actuator OFF-state Value” parameter.

**nvi_xPID_Ctrl:** This variable is used for scene number input. The value of nvi_xPID_Ctrl is regarded as the scene number. The state has no significance.

**nvo_xPID_Setpt:** This variable outputs the value of nvi_xPID_Setpt.

**nvo_xPID_Set:** This variable outputs the current setpoint of the controller.

**nvo_xPID_Ctrl:** This variable outputs the value of nvi_xPID_Ctrl.

**nvo_xPID_Actuator:** This variable is the control value for the actuator. The state of the variable is 1 only when the value is 100. Otherwise the state is 0.

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Start Up Ramping Time** | Linearly adjusts the setpoint of the controller from the initial process measurement value to the configured setpoint during the time set with this parameter (disabled with 0). This feature is intended for smooth start ups of processes. The function also activates after power failures. | 0..32767 |
| **Proportional** | The gain KP of the PID controller. | -100..100 |
| **Integration Time** | Integration time TI of the PID controller. Integration disabled if this is set to | 0..9999 s |

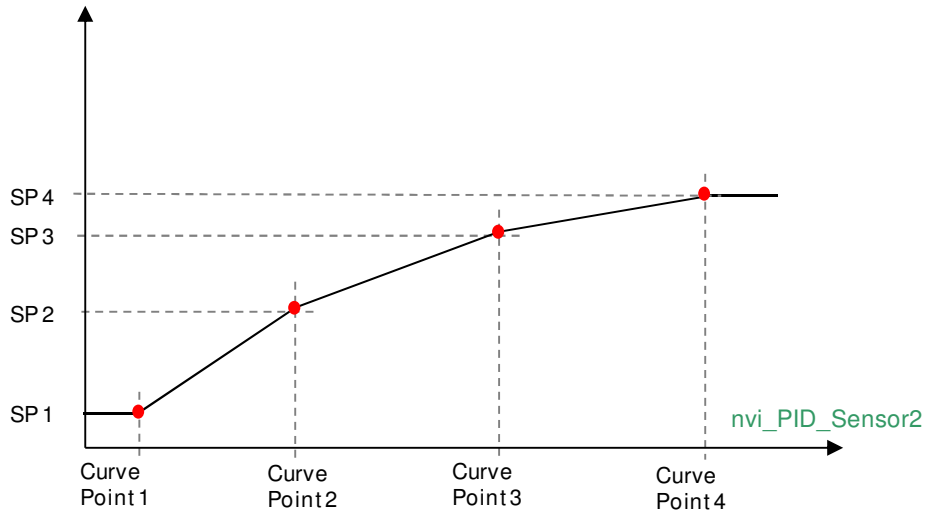
| | | |
|---|---|---|
| | zero (0). The longer the time the more effect it has on the control. | |
| **Derivation Time** | Derivation time TD of the PID controller. Derivation disabled if this is set to zero (0). The longer the time the more effect it has on the control. | 0..9999 s |
| **Sensitivity of PID** | This parameter defines the minimum deviation of the measurement from the reference point (setpoint) before the controller's output is adjusted. | 0..100 |
| **Invert** | The controller normally adjusts the system with respect to the difference between the reference (setpoint) and the process measurement. With this parameter the difference can be inverted i.e. the controller adjusts the system with the difference between the process measurement and the reference. Also the actuator output of the controller can be inverted which is useful for example in cooling systems, when a valve drive actually needs to open when the temperature is rising. | No<br>PID invert<br>Actuator invert<br>Actuator+PID invert |
| **Sensor Selection (nvi_xPID_Sensor1)** | Selection of the process measurement for the controller. The process measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need of using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Sensor Selection (nvi_xPID_Sensor2)** | Selection of the compensation measurement for the controller. The compensation measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need for using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Hi-Level Sliding Alarm** | High level alarm limit for the process measurement. The actual high level alarm limit is determined with this parameter and the setpoint of the process i.e. if the setpoint is 20 and this parameter 5, then the high level alarm threshold is 25. Also when using this parameter the Hi- and HiHi Alarm Limit parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | -255..+255 |
| **Lo-Level Sliding Alarm** | Low level alarm limit for the process measurement. The actual low level alarm limit is determined with this parameter and the setpoint of the process i.e. if the setpoint is 20 and this parameter -5, then the low level alarm threshold is 15. Also when using this parameter the Lo- and LoLo Alarm Limit parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | -255..+255 |
| **(Sensor2)SceneNo1.(Sensor2)SceneNo2** | The controller behaviour i.e. the setpoint can be adjusted with predetermined scenes. The scenes are marked with unique identifiers (numbers) that are sent to the controller via nvi_xPID_Ctrl. When the value of nvi_xPID_Ctrl matches that of a configured scene, the setpoint of the controller is deviated from the normal setpoint by an amount defined with nvi_xPID_Sensor2.<br>For example the temperature of a room is lowered by 3 degrees during night time. The scene number for this is defined as 11. The value of this parameter is then configured as 11.99. When an SNVT_switch of value 5.5 is input to nvi_xPID_Ctrl and nvi_xPID_Sensor2 has the value -3.0, the controller lowers the setpoint by 3 degrees. This scene is active until a new scene number is input to nvi_xPID_Ctrl (if it is not a valid scene number, the controller will return back to its normal state).<br>Note! A scene number greater than 99 must be configured in the first part of the parameter. Scene numbers lower than 10 defined in the second part of the parameter must have a zero prefix e.g. 120.04. | 0.0...200.99 |

| | | |
|---|---|---|
| **(Sensor2)SceneNo3.(SPfield4)SceneNo4** | See: (Sensor2)SceneNo1.(Sensor2)SceneNo2.<br>Note! If a scene number defined in the second part of the parameter activates, the deviation for the setpoint is the fourth field of nvi_xPID_Setpt variable, not the value of nvi_xPID_Sensor2. | 0.0...200.99 |
| **(SPfield5)SceneNo5.(SPfield6)SceneNo6** | See: (Sensor2)SceneNo1.(Sensor2)SceneNo2.<br>Note! If a scene number defined in this parameter activates, the deviation for the setpoint is the fifth/sixth field of nvi_xPID_Setpt parameter, not the value of nvi_xPID_Sensor2. | 0.0...200.99 |
| **Actuator Selection** | When an internal binding is used, this parameter defines the controlled AO object's input inside the same module. If an internal binding is not used, this should be set to "nvo_xPID_Act". | nvo_xPID_Act<br>nvi_x1AO_Ctrl1…<br>nvi_x5AO_Ctrl1<br>nvi_x1AO_Ctrl1 +<br>nvi_x2AO_Ctrl1 |
| **Actuator Range(Max.Min)** | The maximum and minimum output values of nvo_xPID_Act can be set with this parameter. | 0.0...200.99 |
| **Actuator OFF-State Value** | OFF-state value of the controller. When the controller is not running (no permission to run), the nvo_xPID_Act is set to this value. | 0..200 |

## 7.3 Curve

With this type of controller the setpoint is not fixed but follows a piecewise linear curve defined by four points. The current setpoint is defined by the setpoint curve and a compensation measurement. This type of controller is commonly used e.g. in heating networks which need to be outdoor temperature compensated.

The process and compensation measurement can be input via internal bindings, see: configuration parameters Sensor Selection, Sliding Alarms.



## Functionality of Network Variables:

**nvi_xPID_Setpt:** This variable is used for defining ”Y-coordinates” of Curve Points 1-4. Note that ”X-coordinates” of Curve Points are defined with separate configuration parameters Curve Point 1, Curve Point 2, Curve Point 3 and Curve Point 4.

| Field no. in nvi_xPID_Setpt | Meaning |
|---|---|
| 1 | Current setpoint (calculated automatically by the controller) |
| 2 | Vertical shift of the curve |
| 3 | Setpoint value for Curve Point 1 |
| 4 | Setpoint value for Curve Point 2 |
| 5 | Setpoint value for Curve Point 3 |
| 6 | Setpoint value for Curve Point 4 |

**nvi_xPID_Set:** This variable is the deviation of the setpoint.

**nvi_xPID_Sensor1:** This variable is the input for process measurement.

**nvi_xPID_Sensor2:** This variable is the input for compensation measurement. This defines the current setpoint with the curve.

**nvi_xPID_RunCmd:** Input for run command to activate the controller. The controller is activated when the value or state of the variable is greater than 0. If not activated, the actuator output is the value configured with ”Actuator OFF-state Value” parameter.

**nvi_xPID_Ctrl:** This variable is used for scene number input. The value of nvi_xPID_Ctrl is regarded as the scene number. The state has no significance.

**nvo_xPID_Setpt:** This variable outputs the value of nvi_xPID_Setpt.

**nvo_xPID_Set:** This variable outputs the current setpoint of the controller.

**nvo_xPID_Ctrl:** This variable outputs the value of nvi_xPID_Ctrl.

**nvo_xPID_Actuator:** This variable is the control value for the actuator. The state of the variable is 1 only when the value is 100. Otherwise the state is 0.

## Configuration Parameters:

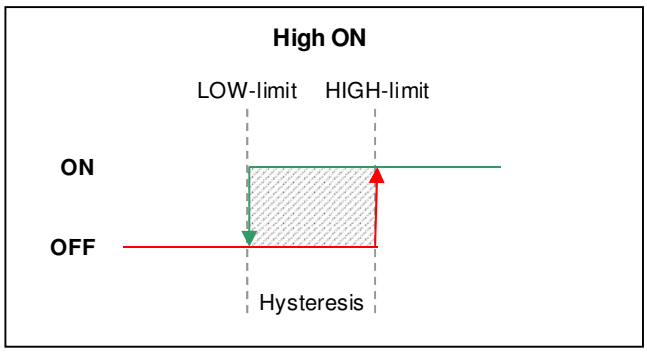
| Parameter | Description | Range |
|---|---|---|
| **Start Up Ramping Time** | Linearly adjusts the setpoint of the controller from the initial process measurement value to the configured setpoint during the time set with this parameter (disabled with 0). This feature is intended for smooth start ups of processes. The function also activates after power failures. | 0..32767 |
| **Proportional** | The gain KP of the PID controller. | -100..100 |

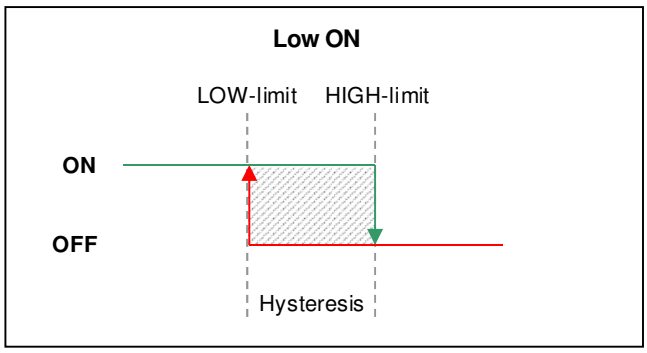
| | | |
|---|---|---|
| **Integration Time** | Integration time TI of the PID controller. Integration disabled if this is set to zero (0). The longer the time the more effect it has on the control. | 0..9999 s |
| **Derivation Time** | Derivation time TD of the PID controller. Derivation disabled if this is set to zero (0). The longer the time the more effect it has on the control. | 0..9999 s |
| **Sensitivity of PID** | This parameter defines the minimum deviation of the measurement from the reference point (setpoint) before the controller's output is adjusted. | 0..100 |
| **Invert** | The controller normally adjusts the system with respect to the difference between the reference (setpoint) and the process measurement. With this parameter the difference can be inverted i.e. the controller adjusts the system with the difference between the process measurement and the reference. Also the actuator output of the controller can be inverted which is useful for example in cooling systems, when a valve drive actually needs to open when the temperature is rising. | No<br>PID invert<br>Actuator invert<br>Actuator+PID invert |
| **Sensor Selection (nvi_xPID_Sensor1)** | Selection of the process measurement for the controller. The process measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need of using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Sensor Selection (nvi_xPID_Sensor2)** | Selection of the compensation measurement for the controller. The compensation measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need for using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Hi-Level Sliding Alarm** | High level alarm limit for the process measurement. The actual high level alarm limit is determined with this parameter and the setpoint of the process i.e. if the setpoint is 20 and this parameter 5, then the high level alarm threshold is 25. Also when using this parameter the Hi- and HiHi Alarm Limit parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | -255..+255 |
| **Lo-Level Sliding Alarm** | Low level alarm limit for the process measurement. The actual low level alarm limit is determined with this parameter and the setpoint of the process i.e. if the setpoint is 20 and this parameter -5, then the low level alarm threshold is 15. Also when using this parameter the Lo- and LoLo Alarm Limit parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | -255..+255 |
| **Actuator Selection** | When an internal binding is used, this parameter defines the controlled AO object's input inside the same module. If an internal binding is not used, this should be set to "nvo_xPID_Act". | nvo_xPID_Act<br>nvi_x1AO_Ctrl1…<br>nvi_x5AO_Ctrl1<br>nvi_x1AO_Ctrl1 +<br>nvi_x2AO_Ctrl1 |
| **Actuator Range(Max.Min)** | The maximum and minimum output values of nvo_xPID_Act can be set with this parameter. | 0.0...200.99 |
| **Actuator OFF-State Value** | OFF-state value of the controller. When the controller is not running (no permission to run), the nvo_xPID_Act is set to this value. | 0..200 |
| **Curve Point 1 (Min)** | Measurement value (X-coordinate) for Curve Point 1 i.e. the 'nvi_xPID_Sensor2 coordinate' of curve point 1. | -250..250 |

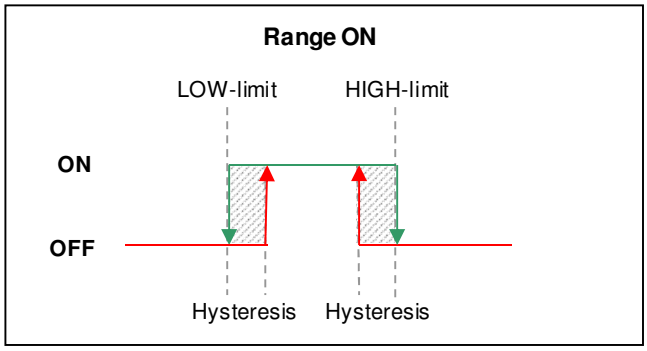
| | | |
|---|---|---|
| **Curve Point 2** | Measurement value (X-coordinate) for the curve point 2. | -250..250 |
| **Curve Point 3** | Measurement value (X-coordinate) for the curve point 3. | -250..250 |
| **Curve Point 4 (Max)** | Measurement value (X-coordinate) for the curve point 4. | -250..250 |

## 7.4 Thermostat Functions

This function controls its ON/OFF output (nvo_xPID_Ctrl) with respect to measurement values of nvi_xPID_Sensor1 and nvi_xPID_Sensor2. Note that nvo_xPID_Actuator is not in use when using this function. The two measurement value inputs are compared to values configured in nvi_xPID_Setpt and assigned a truth value (ON/OFF).



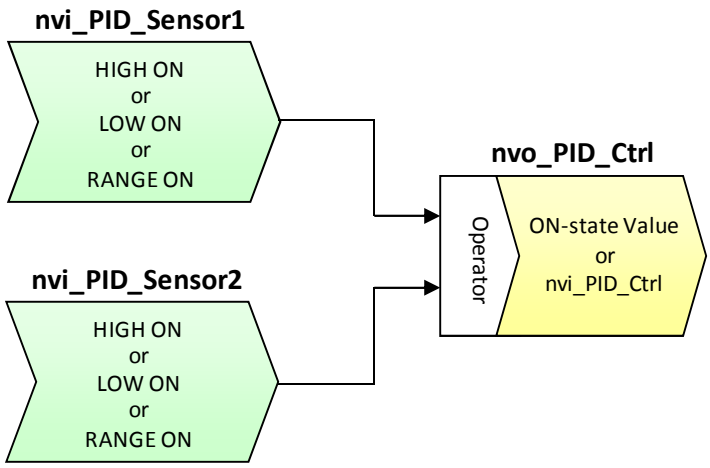



In High ON mode the input is considered to be ON when the measurement exceeds the high limit value configured in nvi_xPID_Setpt. The input is considered to be in ON-state until the measurement falls below the assigned low limit value. Thus, if the low and high limit values are not equal, hysteresis occurs. In Low ON mode controller functions vice versa compared to above description.

In Range ON mode, the input is considered to be ON when the measurement value is between the high and low limit values. Hysteresis in this mode is defined with a separate configuration parameter.

The final state of the controller is then determined with the selected logical operation and the truth values from the two measurement inputs. This means that each controller object configured to any of the Thermostat functions contains always two separate thermostats whose output state (ON/OFF) are combined with logical operator (and/or). When only one thermostat is needed, the selected logical operator has to be OR and the high and low limit values of the second thermostat have to be configured so that the second thermostat is constantly in OFF-state. This can be easily achieved by selecting for example function *(Sensor1) High ON or (Sensor2) High ON* and setting high limit to value 200 and low limit to value 199. In addition, verify that nvi_xPID_Sensor2 does not have any binding so it will continuously stay at value 0.00. Above steps will guarantee that the second thermostat will stay always in OFF-state and only the first thermostat's output will have effect on the output of the controller.




Available Thermostat functions and their logical operators:

- (Sensor1) High ON and (Sensor2) High ON
- (Sensor1) High ON and (Sensor2) Low ON
- (Sensor1) High ON or (Sensor2) High ON
- (Sensor1) High ON or (Sensor2) Low ON
- (Sensor1) Low ON and (Sensor2) Low ON
- (Sensor1) Low ON or (Sensor2) Low ON
- (Sensor1) Range ON and (Sensor2) High ON
- (Sensor1) Range ON and (Sensor2) Low ON
- (Sensor1) Range ON and (Sensor2) Range ON
- (Sensor1) Range ON or (Sensor2) High ON
- (Sensor1) Range ON or (Sensor2) Low ON
- (Sensor1) Range ON or (Sensor2) Range ON

## Functionality of Network Variables:

**nvi_xPID_Setpt:** Defines the high and low limits of the measurement inputs. It also defines the value of nvo_xPID_Set for both controller states.

| Field no. in nvi_xPID_Setpt | Meaning |
|---|---|
| 1 | Low limit value of nvi_xPID_Sensor1 measurement |
| 2 | High limit value of nvi_xPID_Sensor1 measurement |
| 3 | Low limit value of nvi_xPID_Sensor2 measurement |
| 4 | High limit values of nvi_xPID_Sensor2 measurement |
| 5 | Value of nvo_xPID_Set variable in ON-state |
| 6 | Value of nvo_xPID_Set variable in OFF-state |

**nvi_xPID_Set:** This variable deviates the value of nvo_xPID_Set.

**nvi_xPID_Sensor1:** This variable is used for measurement input.
***Note:*** *This value can also be received through internal binding. See configuration parameter Sensor Selection (nvi_xPID_Sensor1)*

**nvi_xPID_Sensor2:** This variable is used for measurement input.
***Note:*** *This value can also be received through internal binding. See configuration parameter Sensor Selection (nvi_xPID_Sensor2)*

**nvi_xPID_RunCmd:** (This variable is not in use in this function.)

**nvi_xPID_Ctrl:** This variable is used for activation number input for the controller. The controller doesn't operate until a suitable activation number (the value of nvi_xPID_Ctrll) is received. The state of the variable has no significance. See also configuration parameters ActivationNo1.ActivationNo2, ActivationNo3.ActivationNo4 and ActivationNo5.ActivationNo6.

**nvo_xPID_Setpt:** This variable outputs the value of nvi_xPID_Setpt.

**nvo_xPID_Set:** This variable outputs the configured ON- or OFF-state value of the controller, deviated with the value of nvi_xPID_Set.

**nvo_xPID_Ctrl:** When the controller is in ON-state, this variable has the value configured with the parameter "nvo_xPID_Ctrl ON-state Value" and the state is 1. When the controller is in OFF-state, this variable equals the value of nvi_xPID_Ctrl.

**nvo_xPID_Actuator:** This variable follows the value of nvo_xPID_Ctrl.

**Configuration Parameters:**

| Parameter | Description | Range |
|---|---|---|
| **Sensor Selection (nvi_xPID_Sensor1)** | Selection of the process measurement for the controller. The process measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need of using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Sensor Selection (nvi_xPID_Sensor2)** | Selection of the compensation measurement for the controller. The compensation measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need for using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **nvo_xPID_Ctrl ON-state Value** | This defines the value of nvo_xPID_Ctrl when the controller is in ON-state. | 0..200 |
| **Hysteresis** | In Range mode only, the hysteresis is defined with this parameter. The controller is considered to be in ON-state only after the measured value exceeds the low limit value + hysteresis (i.e. the value of this parameter) or falls below high limit value - hysteresis. The input is considered to be in ON-state until the measurement exceeds the high limit value or falls below the low limit value. | 0…250 |
| **Activation No1.Activation No2** | The controller is activated when the value of nvi_xPID_Ctrl equals either value of this parameter. The controller compares the measurements and their corresponding limits only if it is activated. Note! Activation numbers greater than 99 need to be entered in the first part of the parameter. Activation numbers smaller than 10 in the second part of the parameter need to have a zero prefix, i.e. 120.03. | 0.0...200.99 |
| **Activation No3.Activation No4** | See: Activation No1.Activation No2 | 0.0...200.99 |
| **Activation No5.Activation No6** | See: Activation No1.Activation No2 | 0.0...200.99 |

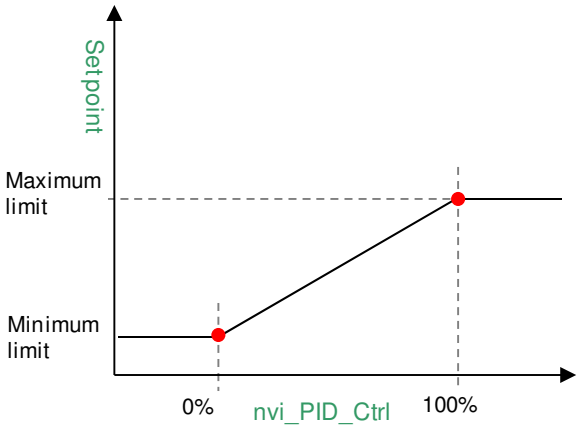
## 7.5 Linear Setpoint (All types)

This function is used to adjust the controller's setpoint linearly between an upper and a lower limit according to a compensation value. In Linear Setpoint (nvi_xPID_Sensor2) function the value of nvi_xPID_Sensor2 is used as a compensation measurement and in Linear Setpoint (nvi_xPID_Ctrl) function setpoint is adjusted according to a 0..100% control value of the nvi_xPID_Ctrl.

### Linear Setpoint (nvi_xPID_Ctrl)

In this type, the controller setpoint is adjusted with a control value bound to nvi_xPID_Ctrl. The upper and lower limits of the setpoint are configured with the second and third field of nvi_xPID_Setpt, see table below. When nvi_xPID_Ctrl has the value 0 (%), the controller's setpoint equals the configured lower limit of its range. As the control value increases the setpoint linearly approaches its upper limit so that with a control value of 100 the setpoint equals the configured upper limit of its range.

**Linear Setpoint (nvi_PID_Ctrl)**

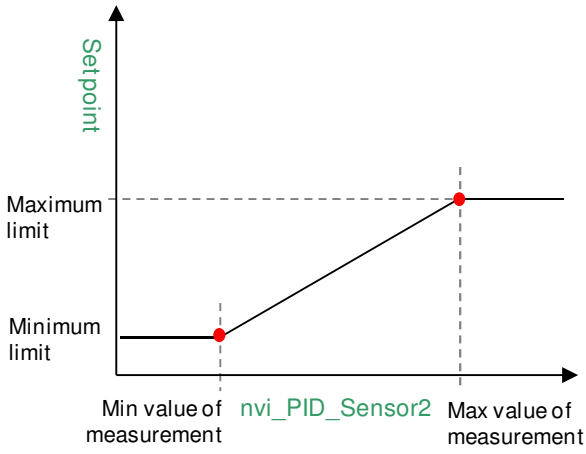



**Linear Setpoint (nvi_xPID_Sensor2)**

In this type, the controller's setpoint is adjusted with a compensation measurement bound to nvi_xPID_Sensor2. The range for the setpoint as well as the range for the compensation measurement are configured with nvi_xPID_Setpt, see table below.

When the compensation measurement is below the lower limit of the measurement range, the setpoint equals the lower limit of the setpoint range. When the compensation measurement exceeds the upper limit of the measurement range, the setpoint equals the upper limit of the setpoint range. When the compensation measurement is within the range, the setpoint is adjusted within its range so that both the compensation measurement and the setpoint cover the same amount of their individual ranges in percentages. For example, the range for the compensation measurement is 20..30 and the range for the setpoint is 18..22. When the compensation measurement is 25 (50 % of its range), the setpoint of the controller is adjusted to 20 (also 50 % of its range).

**Linear Setpoint (nvi_PID_Sensor2)**



## Functionality of Network Variables:

**nvi_xPID_Setpt:** This variable is used for configuring the ranges for the setpoint and the compensation measurement.

***Linear Setpoint (nvi_xPID_Ctrl)***

| Field no. in nvi_xPID_Setpt | Meaning |
|---|---|
| 1 | Deviation for the calculated setpoint, but not recommended for use. Use nvi_xPID_Set instead. |
| 2 | Lower limit for the setpoint range. |
| 3 | Upper limit for the setpoint range. |
| 4 | Lower limit for the control range: must be 0. |
| 5 | Upper limit for the control range: must be 2. |
| 6 | Not in use. |

***Linear Setpoint (nvi_xPID_ Sensor2)***

| Field no. in nvi_xPID_Setpt | Meaning |
|---|---|
| 1 | Deviation for the calculated setpoint, but not recommended for use. Use nvi_xPID_Set instead. |
| 2 | Lower limit for the compensation measurement. |
| 3 | Upper limit for the compensation measurement. |
| 4 | Lower limit for the setpoint range. |
| 5 | Upper limit for the setpoint range. |
| 6 | Not in use. |

**nvi_xPID_Set:** This variable is used to deviate the calculated setpoint.

**nvi_xPID_Sensor1:** This variable is used to input the process measurement.

**nvi_xPID_Sensor2:** This variable used to input the compensation measurement in Linear Setpoint (nvi_xPID_Sensor2) mode only. In Linear Setpoint (nvi_xPID_Ctrl) mode it has no significance.

**nvi_xPID_RunCmd:** (This variable is not in use in this function, the controller is always active)

**nvi_xPID_Ctrl:** This variable is used to input the control value in Linear Setpoint (nvi_xPID_Ctrl) mode only. No significance in Linear Setpoint (nvi_xPID_Sensor2) mode.

**nvo_xPID_Setpt:** This variable outputs the value of nvi_xPID_Setpt.

**nvo_xPID_Set:** This variable outputs the current setpoint of the controller.

| | |
|---|---|
| **nvo_xPID_Ctrl:** | This variable outputs the value of nvi_xPID_Ctrl. |
| **nvo_xPID_Actuator:** | This variable is the control value for the actuator. The state of the variable is 1 only when the value is 100. Otherwise the state is 0. |

## Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Start Up Ramping Time** | Linearly adjusts the setpoint of the controller from the initial process measurement value to the configured setpoint during the time set with this parameter (disabled with 0). This feature is intended for smooth start ups of processes. The function also activates after power failures. | 0..32767 |
| **Proportional** | The gain KP of the PID controller. | -100..100 |
| **Integration Time** | Integration time TI of the PID controller. Integration disabled if this is set to zero (0). The longer the time the more effect it has on the control. | 0..9999 s |
| **Derivation Time** | Derivation time TD of the PID controller. Derivation disabled if this is set to zero (0). The longer the time the more effect it has on the control. | 0..9999 s |
| **Sensitivity of PID** | This parameter defines the minimum deviation of the measurement from the reference point (setpoint) before the controller's output is adjusted. | 0..100 |
| **Invert** | The controller normally adjusts the system with respect to the difference between the reference (setpoint) and the process measurement. With this parameter the difference can be inverted i.e. the controller adjusts the system with the difference between the process measurement and the reference. Also the actuator output of the controller can be inverted which is useful for example in cooling systems, when a valve drive actually needs to open when the temperature is rising. | No<br>PID invert<br>Actuator invert<br>Actuator+PID invert |
| **Sensor Selection (nvi_xPID_Sensor1)** | Selection of the process measurement for the controller. The process measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need of using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Sensor Selection (nvi_xPID_Sensor2)** | Selection of the compensation measurement for the controller. The compensation measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need for using a LON binding tool.<br><br>See also: Hi-Level Sliding Alarm, Lo-Level Sliding Alarm. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Hi-Level Sliding Alarm** | High level alarm limit for the process measurement. The actual high level alarm limit is determined with this parameter and the setpoint of the process i.e. if the setpoint is 20 and this parameter 5, then the high level alarm threshold is 25. Also when using this parameter the Hi- and HiHi Alarm Limit parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | -255..+255 |
| **Lo-Level Sliding Alarm** | Low level alarm limit for the process measurement. The actual low level alarm limit is determined with this parameter and the setpoint of the process i.e. if the setpoint is 20 and this parameter -5, then the low level alarm threshold is 15. Also when using this parameter the Lo- and LoLo Alarm Limit | -255..+255 |

| | | |
|---|---|---|
| | parameters of the AI object have no significance.<br><br>Note! This parameter has any significance only if an internal binding is used, see Sensor Selection (nvi_xPID_Sensor1).<br><br>Note! The alarm priority and the alarm delays are set in the process measurement AI object. | |
| **Actuator Selection** | When an internal binding is used, this parameter defines the controlled AO object's input inside the same module. If an internal binding is not used, this should be set to "nvo_xPID_Act". | nvo_xPID_Act<br>nvi_x1AO_Ctrl1…<br>nvi_x5AO_Ctrl1<br>nvi_x1AO_Ctrl1 +<br>nvi_x2AO_Ctrl1 |
| **Actuator Range(Max.Min)** | The maximum and minimum output values of nvo_xPID_Act can be set with this parameter. | 0.0...200.99 |
| **Actuator OFF-State Value** | In Linear Setpoint (nvi_xPID_Ctrl) mode this is the value that the nvo_xPID_Act shows all the time. In Linear Setpoint (nvi_xPID_Sensor2) mode this value is shown in nvo_xPID_Act if the controller is used as an internal controller i.e. the Use Internal PID parameter is set to "Yes". | 0…200 |
| **Use Internal PID** | (Used in Linear Setpoint (nvi_xPID_Sensor2) mode only!) If this is set to "Yes", the object doesn't control the process (i.e. nvo_xPID_Act shows the controller's OFF state value configured with Actuator OFF-state Value parameter) but simply outputs the calculated setpoint value to nvo_xPID_Set. When this parameter is "No", the control value is output by both nvo_xPID_Set and nvo_xPID_Act variables. | Yes<br>No |

## 7.6 Setpoint Calculation

This function is used for arithmetic calculation of input values. The object takes a maximum of three inputs and performs a different calculation in four predefined scenes.

Scene 1: nvo_xPID_Set = nvi_xPID_Set + nvi_xPID_Sensor1
Scene 2: nvo_xPID_Set = nvi_xPID_Set + nvi_xPID_Sensor2
Scene 3: nvo_xPID_Set = nvi_xPID_Set + nvi_xPID_Sensor1 – nvi_xPID_Sensor2
Scene 4: nvo_xPID_Set = nvi_xPID_Set – nvi_xPID_Sensor1 – nvi_xPID_Sensor2

**Functionality of Network Variables:**

| | |
|---|---|
| **nvi_xPID_Setpt:** | This variable is not in use. |
| **nvi_xPID_Set:** | This variable is the first input value for the calculation. |
| **nvi_xPID_Sensor1:** | This variable is the second input value for the calculation. |
| **nvi_xPID_Sensor2:** | This variable is the third input value for the calculation. |
| **nvi_xPID_RunCmd:** | If the value of this variable is greater than 90 and SceneNo2 is activated, then the object calculates the efficiency ratio of e.g. a heat recovery unit. See example below. |
| **nvi_xPID_Ctrl:** | This variable is used for scene input. If the value of the variable doesn't equal any of the configured scene numbers, nvo_xPID_Set will be 0. If the value of nvi_xPID_Ctrl is zero, then |

| | |
|---|---|
| | the object calculates a weighted average of the three inputs. See example below. |
| **nvo_xPID_Setpt:** | This variable is not in use. |
| **nvo_xPID_Set:** | This variable outputs the result of the calculation. |
| **nvo_xPID_Ctrl:** | This variable is not in use. |
| **nvo_xPID_Actuator:** | This variable is not in use. |

### Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Sensor Selection (nvi_xPID_Sensor1)** | Selection of the process measurement for the controller. The process measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need of using a LON binding tool.<br><br>Note! Sliding alarm limits not in use. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **Sensor Selection (nvi_xPID_Sensor2)** | Selection of the compensation measurement for the controller. The compensation measurement can be brought to the controller from AI objects located in the same module with the controller object via an internal binding using this parameter. This eliminates the need for using a LON binding tool.<br><br>Note! Sliding alarm limits not in use. | nvi_xPID_Sensor1 (default),<br>nvo_x6AI_Sensor…<br>nvo_x10AI_Sensor<br>nvo_x6AI_Calcul…<br>nvo_x10AI_Calcul |
| **SceneNo1 (nvo_xPID_Set= PID_Set+ PID_Sensor1)** | Activation value for scene 1. | 0..200 |
| **SceneNo2 (nvo_xPID_Set=PID_Set+PID_Sensor2)** | Activation value for scene 2. | 0..200 |
| **SceneNo3 (nvo_xPID_Set=PID_Set+PID_Sensor1-PID_Sensor2)** | Activation value for scene 3. | 0..200 |
| **SceneNo4 (nvo_xPID_Set=PID_Set-PID_Sensor1-PID_Sensor2)** | Activation value for scene 4. | 0..200 |

### Example 1: Calculation of weighted average

The weights for each input are configured with the three first fields of nvi_xPID_Setpt variable; first field is the weight for nvi_xPID_Set, second for nvi_xPID_Sensor1 and third for nvi_xPID_Sensor2. The last three fields of nvi_xPID_Setpt have no significance.

nvi_xPID_Setpt: 1.00,2.00,4.00,0.00,0.00,0.00
nvi_xPID_Set: 12.00
nvi_xPID_Sensor1: 15.00

nvi_xPID_Sensor2: 20.00
nvi_xPID_Ctrl: 0.0 0

nvo_xPID_Set: 17.42 = (1*12 + 2*15 + 4*20) / 7

**Example 2: Efficiency ratio calculation**

This is usually used when there is a need to calculate the efficiency ratio of a heat recovery unit. The inputs should be then as follows:

nvi_xPID_Set: outlet temperature before HRU
nvi_xPID_Sensor1: inlet temperature after HRU
nvi_xPID_Sensor2: outdoor temperature
nvi_xPID_RunCmd: e.g. the rotational speed of the heat recovery wheel
nvi_xPID_Ctrl: SceneNo2

The following temperature measurements:

nvi_xPID_Set: 25.00
nvi_xPID_Sensor1: 17.00
nvi_xPID_Sensor2: 2.00

…yield an efficiency ratio of:

nvo_xPID_Set: 66.00 ≈ (17 – 2) / (25 – 2) *100

**NOTE!**
The efficiency ratio is not calculated unless the value of nvi_xPID_RunCmd is greater than 90.

# 8. Node Object

Node Object contains network variables for alarming functions.

**Functionality of Network Variables:**

| | |
|---|---|
| **nviAlarm:** | Used for alarm bindings (nvoAlarm of another object can be bound to this). A module forwards all alarm messages. |
| **nviAlarmPos:** | Used for alarm position bindings (nvoAlarmPos of another object can be bound to this). A module forwards all alarm position messages. |
| **nvoAlarm:** | Shows the latest alarm event coming either from the module's nviAlarm or from an alarming object inside the module. |
| **nvoAlarmPos:** | Shows the position of the latest alarm event coming either from the module's nviAlarmPos or from an alarming object inside the module. |

**Configuration Parameters:**

This object does not contain any Configuration Parameters.

# 9. Module Parameters

Module parameters can be accessed on the “Module” tab of the Lonix PCT.

### Functionality of Network Variables:

**nci_Cmd:**

A command variable for the controller. Commands are carried out by entering a desired value to this variable from the following table:

| Value of nci_Cmd | Function |
|---|---|
| 10000 – 10019 | Browsing of active alarms |
| 65535 | Module RESET |

When browsing the alarms, the latest active alarm is read by entering value 10000 to nci_Cmd, the oldest with value 10019. Corresponding alarm details (object_id, priority, etc.) are shown in nvoAlarm variable of the Node Object and alarming point position string in nvoAlarmPos. Even though the alarms are shown in the alarm output variables no data is propagated to the LON network.

**nvoTrend:**

Shows I/O-object specific data e.g. usage (operational) hours of DI-objects. See also chapter: “Tables on nvoTrend and nvi_x_Override variables”.

### Configuration Parameters:

| Parameter | Description | Range |
|---|---|---|
| **Module Name** | This parameter is for the identification of the module. For example the COBA BOS Server application uses this information to identify alarming module. Six first characters of the parameter are included in the beginning of nvoAlarm variable when an object inside the module triggers an alarm. It is recommended that the value of this parameter is exactly the same as the module’s name in the LON database. | 0…6 characters |

# 10. Tables on nvoTrend and nvi_x_Override variables

The following table shows the meaning of the different fields of **nvoTrend** variable.

| UNVT_Trend_LXM10 | | |
|---|---|---|
| **Variable field** | **Type** | **Description** |
| 1 P01_DATA | Unsigned Long | Point 1 data e.g. operational hours |
| 2 P01_DATA | Unsigned Long | Point 2 data e.g. operational hours |
| 3 P01_DATA | Unsigned Long | Point 3 data e.g. operational hours |
| 4 P01_DATA | Unsigned Long | Point 4 data e.g. operational hours |
| 5 P01_DATA | Unsigned Long | Point 5 data e.g. operational hours |
| 6 P01_DATA | Unsigned Long | Point 6 data e.g. operational hours |
| 7 P01_DATA | Unsigned Long | Point 7 data e.g. operational hours |
| 8 P01_DATA | Unsigned Long | Point 8 data e.g. operational hours |
| 9 P01_DATA | Unsigned Long | Point 9 data e.g. operational hours |
| 10 P01_DATA | Unsigned Long | Point 10 data e.g. operational hours |
| DIa_ALARM_STATUS | Unsigned Short | DI alarm status bits of points 1-8 |
| DIb_ALARM_STATUS | Unsigned Short | DI alarm status bits of points 9-10 |
| AI0_ALARM_STATUS | Enum (AI_ALARM_STATUS) | AI alarm status of point 6 |
| AI1_ALARM_STATUS | Enum (AI_ALARM_STATUS) | AI alarm status of point 7 |
| AI2_ALARM_STATUS | Enum (AI_ALARM_STATUS) | AI alarm status of point 8 |
| AI3_ALARM_STATUS | Enum (AI_ALARM_STATUS) | AI alarm status of point 9 |
| AI4_ALARM_STATUS | Enum (AI_ALARM_STATUS) | AI alarm status of point 10 |

Description of Enum (AI_ALARM_STATUS):

NO_AI_ALARM_CONDITION: 0
LOLO_ALARM: 1
LO_ALARM: 2
HI_ALARM: 4
HIHI_ALARM: 8

The following table shows the meaning of the different fields of **nvi_x_Override** variable (here x is the I/O point number, value between 1 … 10).

| UNVT_Manual_LXM10 | | |
|---|---|---|
| **Variable field** | **Type** | **Description** |
| 1 state | Enum (manual_override) | Defines the state of manual override (on/off) |
| 2 valueSwitch.value<br>valueSwitch.state | Struct (Unsigned short, signed short) | Defines value of override for DI/DO/AO points |
| 3 valueTemp | Signed long | Defines value of override for AI points |