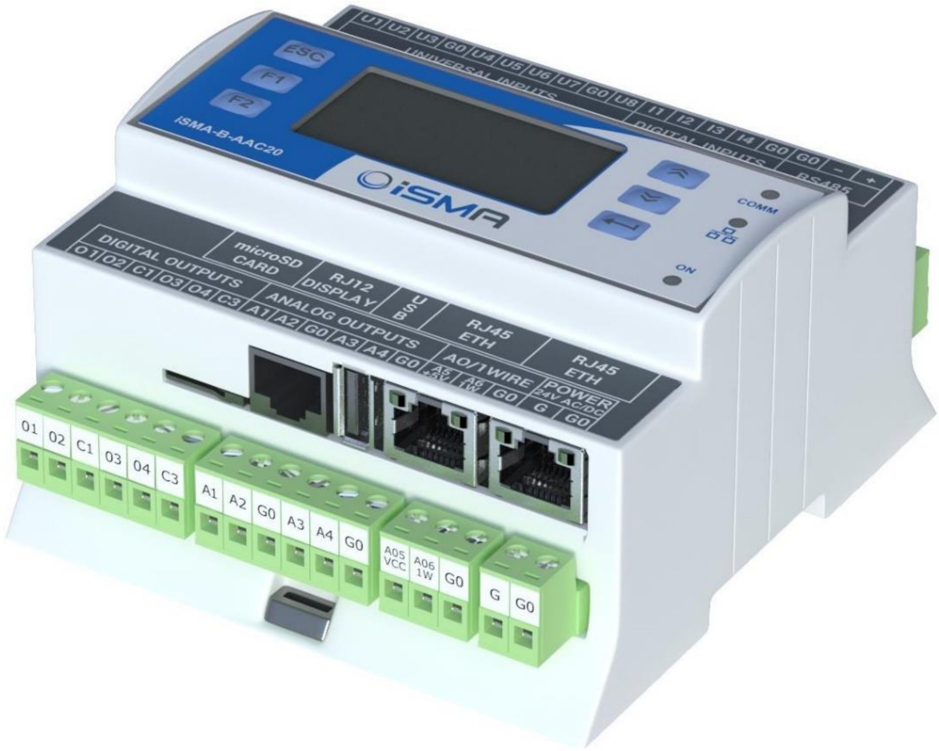
Intelligent Solution Managing Automation

iSMA Control kit

User Manual



Powered by sedona FRAMEWORK™

Global Control 5 S.A.
Warsaw, Poland
www.globalcontrol5.com

Table of Contents

1 Introduction

This manual contains information about iSMA Control kit in an AAC20 controller. The Control kit can be used in all AAC20 hardware versions with firmware 3.4 version or higher. The Control kit is installed by default in the AAC20 controller and cannot be uninstalled.

1.1 Revision History

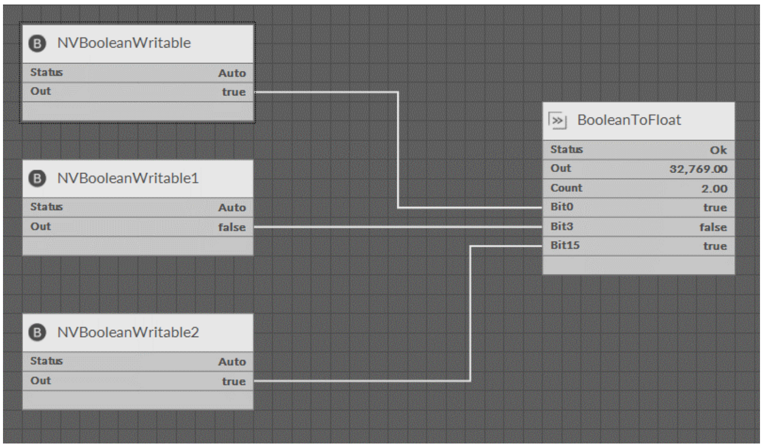
Rev	Date	Description
1.0	30.01.2018	First edition
1.1	21.01.2020	Replaced environment of programming from Workplace to iSMA Tool

Table 1 Revision history

2 Conversion Components

2.1 BooleanToFloat Component

The BooleanToFloat component converts 16 Boolean signals to 1 Float signal.



NVBooleanWritable
Status Auto
Out true

NVBooleanWritable1
Status Auto
Out false

NVBooleanWritable2
Status Auto
Out true

BooleanToFloat
Status Ok
Out 32,769.00
Count 2.00
Bit0 true
Bit3 false
Bit15 true

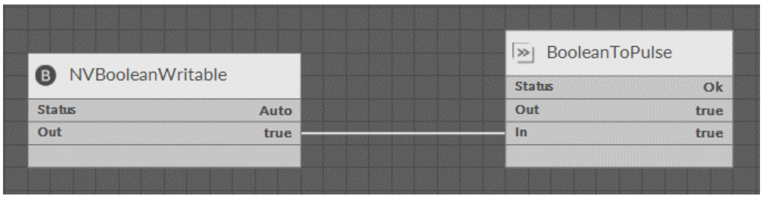
Figure 1 BooleanToFloat component view

The component has the following slots:

- Out: Encoded value of inputs with bit15(MSB) and bit0(LSB);
- Count: Sum of active inputs;
- 16-Bit: Any bit of true (1, +) or false (0, -) value.

2.2 BooleanToPulse Component

The BooleanToPulse component is a component which converts 1 Boolean signal to pulse.BooleanToPulse is a simple monostable oscillator object.



NVBooleanWritable

Status Auto

Out true

BooleanToPulse

Status Ok

Out true

In true

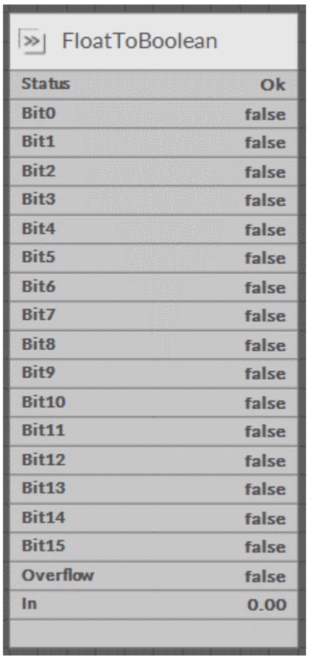
Figure 2 BooleanToPulse component view

The component has the following slots:

- Out: = Generate an impulse during the cycle where the rising edge occurs.

2.3 FloatToBoolean Component

The FloatToBoolean component converts a 16-bit float to binary decoder object.



FloatToBoolean

Status Ok

Bit0 false

Bit1 false

Bit2 false

Bit3 false

Bit4 false

Bit5 false

Bit6 false

Bit7 false

Bit8 false

Bit9 false

Bit10 false

Bit11 false

Bit12 false

Bit13 false

Bit14 false

Bit15 false

Overflow false

In 0.00

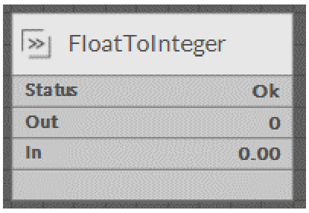
Figure 3 FloatToBoolean component view

The component has the following slots:

- Outputs (bit15-bit0) : Decoded value of inputs with bit15(MSB) and bit0(LSB);
- Overflow: true if inNumeric > 65535.

2.4 FloatToInteger Component

The FloatToInteger is a float to 32-bit integer (Integer) converter object.



FloatToInteger

Status Ok

Out 0

In 0.00

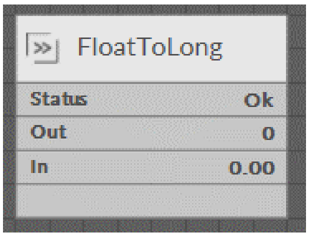
Figure 4 FloatToInteger component view

The component has the following slot:

- Out: = in, except that the outputs are the 32-bit integer with fractional part truncated.

2.5 FloatToLong Component

The FloatToLong is a float to 64-bit signed integer (long) converter object.



FloatToLong

Status Ok

Out 0

In 0.00

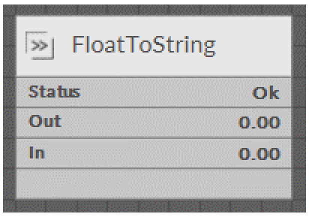
Figure 5 FloatToLong component view

The component has the following slot:

- Out: = in, except that the output is the 64-bit signed integer with fractional part truncated.

2.6 FloatToString Component

The FloatToString component converts float to string objects.



FloatToString

Status Ok

Out 0.00

In 0.00

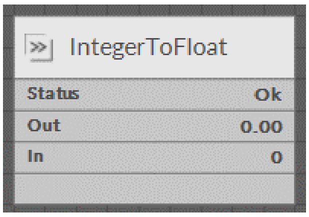
Figure 6 FloatToString component view

The component has the following slot:

- Out: = in, except that the output is the Buff (64-bit) with fractional part truncated.

2.7 IntegerToFloat Component

The IntegerToFloat component converts 32 integer bits to a float object.



IntegerToFloat

Status Ok

Out 0.00

In 0

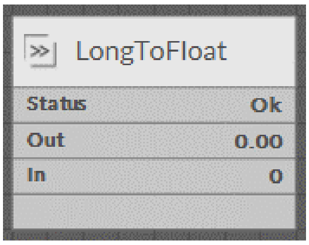
Figure 7 IntegerToFloat component view

The component has the following slot:

- Out: = in, except that the output is the float.

2.8 LongToFloat Component

The LongToFloat component converts a 64-bit signed, integer (long) to a float object.



LongToFloat

Status Ok

Out 0.00

In 0

Figure 8 LongToFloat component view

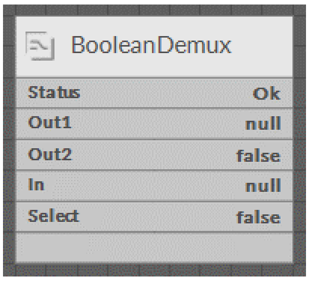
The component has the following slot:

- Out := in, except that the output is the float.

3 Demux Components

3.1 BooleanDemux Component

The BooleanDemux component selects one of two outputs to receive the Input (Boolean) Value, depending on the value of the selected Boolean Input. The value of the other output remains unchanged.



BooleanDemux

Status	Ok
Out1	null
Out2	false
In	null
Select	false

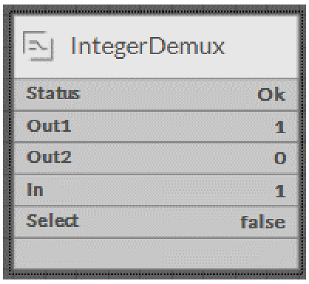
Figure 9 BooleanDemux component view

The component has the following slots:

Select:	Out1	Out2
False:	In	Previous-Value
True:	Previous-Value	In

3.2 IntegerDemux Component

The IntegerDemux component selects one of two outputs to receive the Input (Integer) Value, depending on the value of the selected Boolean Input. The value of the other output remains unchanged.



IntegerDemux

Status	Ok
Out1	1
Out2	0
In	1
Select	false

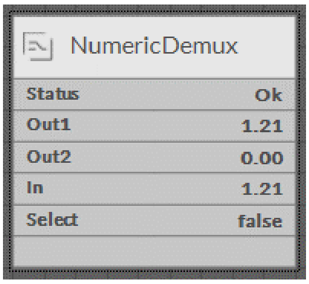
Figure 10 IntegerDemux component view

The component has the following slots:

Select	Out1	Out2
False	In	Previous-Value
True	Previous-Value	In

3.3 NumericDemux Component

The NumericDemux component selects one of two outputs to receive the Input (Numeric) Value, depending on the value of the selected Boolean Input. The value of the other output remains unchanged.



NumericDemux

Status	Ok
Out1	1.21
Out2	0.00
In	1.21
Select	false

Figure 11 NumericDemux component view

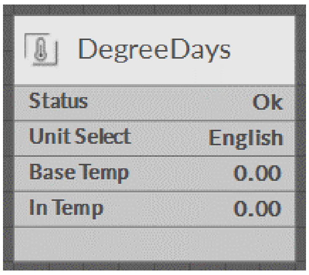
The component has the following slots:

Select	Out1	Out2
False	In	Previous-Value
True	Previous-Value	In

4 Energy Components

4.1 DegreeDays Component

The DegreeDays component provides degree-day calculations, based upon temperature received at the input Temperature slot and values of various other properties.



DegreeDays

Status	Ok
Unit Select	English
Base Temp	0.00
In Temp	0.00

Figure 12 DegreeDays component view

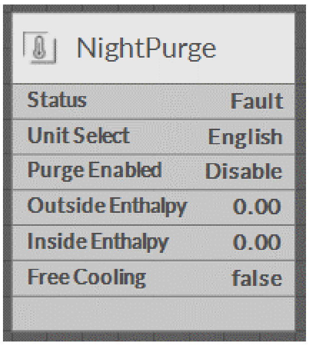
***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- Unit Select: Used to set the units of the Temp In, Min Temp, Max Temp, and Mean Temp properties.
- Base Temperature: Specifies the base temperature used in the degree-day calculation.
- Input Temperature: The input for the outside air temperature used in the degree-day calculation. Note: If this input is not valid then no calculations will be done.
- Minimum Temperature: The minimum temperature recorded for the current day. Tested and set on each calculation.
- Maximum Temperature: The maximum temperature recorded for the current day. Tested and set on each calculation.
- Mean Temperature: The mean temperature recorded for the previous day. Calculated when the day changes. Mean Temp = (Max Temp + Min Temp) / 2.0.
- Cooling Degree-day: This is the cooling degree-day calculated for the previous day. Calculated when the day changes.
- Totalized Cooling Degree-days: This is the totalized cooling degree-days since the last Reset Totals action was invoked. Calculated when the Cooling Degree-day changes.
- Heating Degree-day: The heating degree-day calculated for the previous day. Calculated when the day changes.
- Totalized Heating Degree-days: The totalized heating degree-days since the last Reset Totals action was invoked. Calculated when the Heating Degree-day changes.

4.2 NightPurge Component

The component uses the two sets of temperature and humidity inputs to find the air supply with the least amount of heat when the Purge Enabled input is true. The Free Cooling output will be set to false if outside >= inside or set to true if outside = the Night Setpoint.



NightPurge

Status	Fault
Unit Select	English
Purge Enabled	Disable
Outside Enthalpy	0.00
Inside Enthalpy	0.00
Free Cooling	false

Figure 13 NightPurge component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- Unit Select: Specifies the units of the Temperature and Humidity properties.
- Purge Enabled: Boolean data type, must be true to enable the night purge operation. Whenever false, the Free Cooling output is set to the opposite of the Free Cooling Command (or null, if the Use Null Output is set to true), and the Current Mode slot value is "Disabled." The Purge Enabled is often linked to a "Not" object sourced from the Boolean Schedule output.
- Outside Temperature: Input for the current outside air temperature. The input must be valid for this object to function.
- Outside Humidity: Input for the current outside air humidity. The input must be valid for this object to function.
- Inside Temperature: Input for the current inside air temperature. The input must be valid for this object to function.
- Inside Humidity: Input for the current inside air humidity. The input must be valid for this object to function.
- Low Temperature Limit: This property is used to provide freeze protection.
- Night Setpoint: Inside night temperature Setpoint, at or below which Free Cooling is not applied. Instead, the Current Mode is set to "Satisfied."
- Outside Enthalpy: This is the calculated outside air enthalpy.
- Inside Enthalpy: This is the calculated inside air enthalpy.
- Free Cooling: A Boolean output set to the value of the Free Cooling command when it is determined that free cooling should be used. Otherwise, the value is set to the opposite state or null (if the Used Null Output is set to true).
- Current Mode: This slot indicates which of the following modes the component is currently in:

  Disabled (Purge Enabled is false)

  Free Cooling

No Free Cooling (free cooling not available)

Low temperature (Outside Temp Below Low Temperature Limit, free cooling disabled)

Input error (A temperature or humidity is invalid (down, fault, etc.), free cooling disabled)

Satisfied (Inside temperature below the Night Setpoint, free cooling disabled)

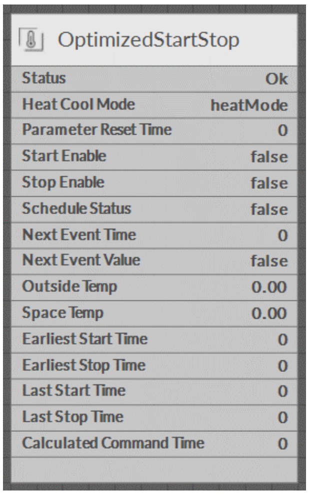
- Setpoint Deadband: The Temperature Setpoint Deadband is applied when the inside temperature falls below the Night Setpoint before free cooling can be enabled. The default value is 1.0.

Threshold Span: The difference between the inside enthalpy and the outside enthalpy must be greater than this value before free cooling will be enabled. The default value is 1.0.

- Use Enthalpy: Setting this property to true will enable the use of enthalpy for determining if free cooling is available. Otherwise, it will just use outside and inside temperature to decide.

4.3 OptimizedStartStop Component

The OptimizedStartStop component allows using Start Time Optimization and Stop Time Optimization for energy saving. This component uses a space temperature input and area characteristics to calculate an optimal amount of lead-time before a scheduled event. It can analyse area temperature changes and adjust the optimization parameters based on the actual temperature change rates after an optimized start or stop.



OptimizedStartStop

Status	Ok
Heat Cool Mode	heatMode
Parameter Reset Time	0
Start Enable	false
Stop Enable	false
Schedule Status	false
Next Event Time	0
Next Event Value	false
Outside Temp	0.00
Space Temp	0.00
Earliest Start Time	0
Earliest Stop Time	0
Last Start Time	0
Last Stop Time	0
Calculated Command Time	0

Figure 14 OptimizedStartStop component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- Heat Cool Mode: This Boolean property allows enabling either the Heat mode or the

Cool mode. The selected option applies only to optimized stop calculations, which means that optimized stop calculations are performed only for the selected mode. Optimized start calculations are performed for both heat and cool modes, regardless of this property value.

- Parameter Reset Time: This property displays the time when any of the four runtime or drift time properties change to the User Defined values. The OSS component copies the user-defined drift time and runtime property values to the corresponding actual drift time and runtime property values.
- Start Enable: This property allows you to manually or automatically enable or disable the optimized start function.
- Stop Enable: This property allows you to manually or automatically enable or disable the optimized stop function.
- Schedule Status: This Boolean property monitors and displays the status of the schedule that is linked to it.
- Next Event Time: This property is linked to the schedule for the time of the next scheduled event.
- Next Event Value: This property is linked to the schedule and reflects the value of the action for the next scheduled event.
- Outside Temp: This property is linked to outside temperature and displays the value for information only.
- Space Temp: This property is linked to a space temperature output and displays the temperature of the area affected by equipment associated with the OSS component.
- Start Time Command: This Boolean property is an output that if linked to the control algorithm invokes an equipment start command. For example, it can be linked to a prioritized input of a Boolean writable point - or directly to the equipment start control.
- Stop Time Command: This Boolean property is an output that if linked to the control invokes an equipment stop command. For example, it can be linked to a prioritized input of a Boolean writable - or directly to the equipment stop control.
- Upper Comfort Limit: This property value is the Cooling mode target temperature.
- Lower Comfort Limit: This property value is the Heating mode target temperature.
- Dynamic Parameter Adjust: This controls whether or not calculation parameters are programmatically adjusted after an execution. After the OSS component completes the start or stop control, if this property value is set to true, the component evaluates the actual recovery rate (degrees/hour) and automatically adjusts the Runtime and Drifttime properties values so that they are influenced by actual drift time and run time.
- Old Parameter Multiplier : This property is used to weigh the dynamic parameter adjustment calculation. The value that is specified in this field affects how much weight is assigned to the previous runtime property value when it is used in the dynamic parameter adjustment calculation. A larger value increases the weight given to the previous runtime and a smaller value decreases the weight.
- Earliest Start Time: This property allows to specify a time before which no optimized

start command may be issued. If this value is set earlier than the Calculated Command Time, the Calculated Command Time is adjusted to equal to this time.

- Earliest Stop Time: This property allows to specify a time before which no stop command may be issued. If this value is set earlier than the Calculated Command Time, the Calculated Command Time is adjusted to equal to this time.
- Drifttime Per Degree Cooling User Defined: This property allows to set a default value for calculating the rate of drift in a cooling mode. A value saved in this field is copied to the Drifttime Per Degree Cooling field.
- Drifttime Per Degree Heating User Defined: This property allows to set a default value for calculating the rate of drift in a heating mode. A value saved in this field is copied to the Drifttime Per Degree Heating field.
- Runtime Per Degree Cooling User Defined: This property allows to set a default value for calculating the runtime value in a cooling mode. A value saved in this field is copied to the Runtime Per Degree Cooling field.
- Runtime Per Degree Heating User Defined: This property allows to set a default value for calculating the runtime value in a heating mode. A value saved in this field is copied to the Runtime Per Degree Heating field.
- Drifttime Per Degree Cooling: This property displays the actual value that is used for calculating an optimized stop time when the equipment is in a cooling mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- Drifttime Per Degree Heating: This property displays the actual value that is used for calculating an optimized stop time when the equipment is in a heating mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- Runtime Per Degree Cooling: This property displays the actual value that is used for calculating an optimized start time when the equipment is in a cooling mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- Runtime Per Degree Heating: This property displays the actual value that is used for calculating an optimized start time when the equipment is in a heating mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- Last Start Time: This is a record of the last Start Time that was used for calculating an optimized start time. Since only one optimized start per day is allowed, this value does not display Start Times (restarts) that are subsequent to the initial Start Time for a day.

- Last Stop Time: This is a record of the last Stop Time that was used for calculating an optimized stop time. Since multiple Optimized Stops are allowed in a day, this value changes to reflect the latest Optimized Stop time.
- Outside Temp at Beginning: This is a record of what the outside air temperature was at the time of the last start or stop command. This is the temperature that was used in calculations of dynamic parameter adjustment.
- Space Temp at Beginning: This is a record of what the space temperature was at the time of the last start or stop command. This is the temperature that was used in

calculations of dynamic parameter adjustment.

- Calculated Command Time: This field shows the calculated time for the next command. This could be a start or a stop command.
- Program Mode: As a part of the logic that the OSS component uses, there are five program mode states. These states serve primarily in logic control; however, they may as well be informative to the system engineer. The Program Mode value displays the current heating or cooling state for the optimized start or stop. The following list describes the possible display values and meanings:

0 ("No" Calculation)

This value indicates that no calculation is being made.

1 ("Start" Calculation)

This value indicates that the optimized start calculation process is ongoing, but that an optimized start or stop is not yet in progress.

2 ("Start" in Process)

This value indicates that the optimized start has been initiated.

3 ("Stop" Calculation)

This value indicates that the optimized stop calculation process is ongoing. but that an optimized start or stop is not yet in progress.

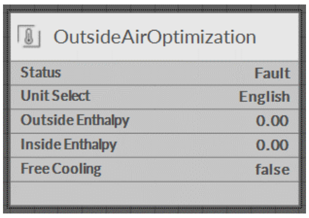
4 ("Stop" in Process)

This value indicates that the optimized stop has been initiated.

4.4 OutsideAirOptimization Component

The OutsideAirOptimization component is used to support applications that need to allow for enthalpy based free cooling. This object is typically used during occupancy periods.

The Free Cooling output is set to false if outside >= inside and set to true if outside <= inside - (abs) Threshold Span. You can select temperature or enthalpy comparisons. There is also a check of the low temperature to protect against freezing.



OutsideAirOptimization

Status	Fault
Unit Select	English
Outside Enthalpy	0.00
Inside Enthalpy	0.00
Free Cooling	false

Figure 15 OutsideAirOptimization component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- Unit Select: This is used to set the units of the Temperature and Humidity properties.
- Outside Temperature: Input for the current outside air temperature. This input must be valid for this object to function.
- Outside Humidity: Input for the current outside air humidity. This input must be valid for this object to function.
- Inside Temperature: Input for the current inside air temperature. This input must be valid for this object to function.
- Inside Humidity: Input for the current inside air humidity. This input must be valid for this object to function.
- Low Temperature Limit: This property is used to provide freeze protection.
- Outside Enthalpy: This is the calculated outside air enthalpy.
- Inside Enthalpy: This is the calculated inside air enthalpy.
- Free Cooling: This Boolean output value is set to the value of the Free Cooling command when it is determined that free cooling should be used. Otherwise, the value is set to null.
- Current Mode: This indicates what mode this component is currently in:

  Input out of range

  Free Cooling

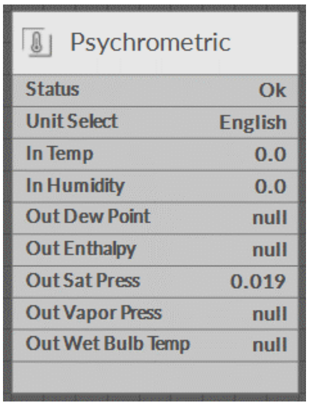
  No Free Cooling

  Low temperature

  Input error
- Threshold Span: The difference between the inside enthalpy and the outside enthalpy must be greater than this value before free cooling will be enabled.
- Use Enthalpy: Setting this property to true will enable the use of enthalpy for determining if free cooling is available. Otherwise, it will just use outside and inside temperature to decide.

4.5 Psychrometric Component

The component is used to support applications that need to calculate the properties of moist air using given temperature and humidity inputs.



Psychrometric

Status	Ok
Unit Select	English
In Temp	0.0
In Humidity	0.0
Out Dew Point	null
Out Enthalpy	null
Out Sat Press	0.019
Out Vapor Press	null
Out Wet Bulb Temp	null

Figure 16 Psychometric component view

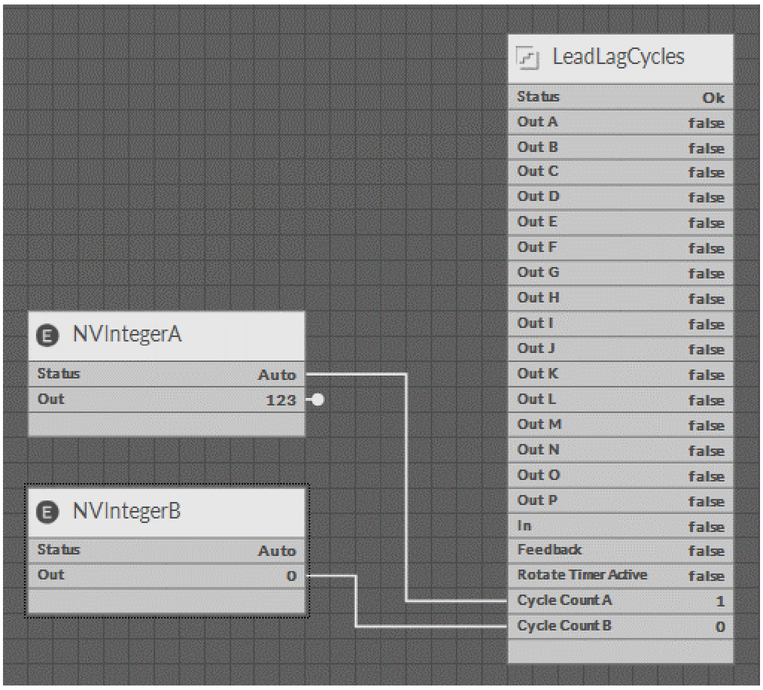
The component has the following slots:

- Unit Select: Used to set the units of the Temperature and Humidity properties.
- Input Temperature: Input temperature.
- Input Humidity: Input humidity.
- Dew Point Temperature: Calculated dew point temperature. Requires valid Input Temperature and Input Humidity to calculate.
- Enthalpy: Calculated enthalpy. Requires valid Input Temperature and Input Humidity to calculate.
- Saturated Pressure: Calculated saturated pressure. Requires valid Input Temperature to calculate.
- Vapor Pressure : Calculated vapor pressure. Requires valid Input Temperature and Input Humidity to calculate.
- Wet Bulb Temperature: Calculated wet-bulb temperature. Requires valid Input Temperature and Input Humidity to calculate.

5 HVAC Components

5.1 LeadLagCycles Component

The LeadLagCycles provides a lead-lag control of 2 to 16 loads based upon their accumulated COS (change of state) counts. This component balances the number of change of states cycles of each of the devices. Only one of the controlled devices will be active at a time based on cycle count.



NVIntegerA
Status Auto
Out 123

NVIntegerB
Status Auto
Out 0

LeadLagCycles
Status Ok
Out A false
Out B false
Out C false
Out D false
Out E false
Out F false
Out G false
Out H false
Out I false
Out J false
Out K false
Out L false
Out M false
Out N false
Out O false
Out P false
In false
Feedback false
Rotate Timer Active false
Cycle Count A 1
Cycle Count B 0

Figure 17 LeadLagCycles component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

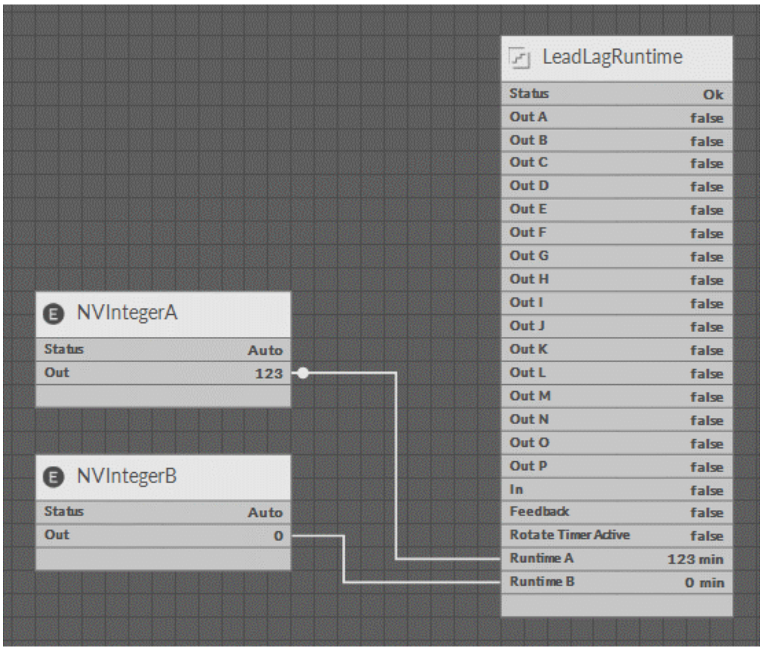
The component has the following slots:

- In: Boolean input that controls whether any control device should be on. If this input is true, one of the outputs will be active based on the cycle count of each controlled device.
- Number Outputs: Specifies the number of devices (outputs) that are controlled.
- Max Runtime: Specifies the maximum amount a given output will be true before switching to another output.
- Feedback: Boolean input which provides positive feedback that a controlled device actually has started. If the feedback value does not show true within the Feedback Delay time, the current controlled output will show an alarm, and the LeadLagCycles switches to the next controlled output. Setting this value to true (and not linking) disables the alarm feature.
- Out A-P: Boolean outputs, each typically linked to the BooleanWritable control point with the DiscreteTotalizerExt. Outputs are typically used to control loads of some type, such as 2 or more pumps.
- Cycle Count A-P: Integer inputs that are used for cycle count feedback for the corresponding Out A-P. These inputs will typically be linked to the ChangeOfStateCount

property of the DiscreteTotalizerExt that is measuring the cycles of the corresponding Out A-P.

5.2 LeadLagRuntime Component

The LeadLagRuntime provides a lead-lag control of from 2 to 16 loads based upon their accumulated runtimes (elapsed active time). This component balances the active runtime of each of the devices. Only one of the controlled devices will be active at a time based on runtime.



LeadLagRuntime
Status Ok
Out A false
Out B false
Out C false
Out D false
Out E false
Out F false
Out G false
Out H false
Out I false
Out J false
Out K false
Out L false
Out M false
Out N false
Out O false
Out P false
In false
Feedback false
Rotate Timer Active false
Runtime A 123 min
Runtime B 0 min

NVIntegerA
Status Auto
Out 123

NVIntegerB
Status Auto
Out 0

Figure 18 LeadLagRuntime component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

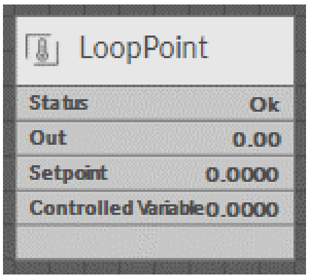
The component has the following slots:

- In: Boolean input that controls whether any control device should be on. If this input is true, one of the outputs will be active based on runtime.
- Number Outputs: Specifies the number of devices (outputs) that are controlled.
- Max Runtime: Specifies the maximum amount a given output will be true before switching to another output.
- Feedback: Boolean input which provides positive feedback that a controlled device actually has started. If the feedback value does not show true within the Feedback Delay time, the current controlled output will show an alarm and the LeadLagRuntime switches to the next controlled output. Setting this value to true (and not linking) disables the alarm feature.

- Out A-P: Boolean outputs, each typically linked to the BooleanWritable control point with the DiscreteTotalizerExt. Outputs are typically used to control loads of some type, such as 2 or more pumps.
- Runtime A-P: Inputs that are used for runtime feedback for the corresponding Out A-P. These inputs will typically be linked to the ElapsedActiveTime property of the DiscreteTotalizerExt that is measuring the runtime of the corresponding Out A-P.

5.3 LoopPoint Component

The LoopPoint component implements a simple PID control loop. Loop objects provide closed-loop PID control (proportional, integral, derivative) at the controller level. Independent gain constants allow the loop to be configured as P-only, PI, or PID.



LoopPoint

Status Ok
Out 0.00
Setpoint 0.0000
Controlled Variable 0.0000

Figure 19 LoopPoint component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- Loop Enable: Setting this input to true will enable the PID loop algorithm to execute at the rate selected by the Execute Time property. Setting this input to false will force the PID loop output to a value dependent on the selection in the Preset property.
- Controlled Variable: Input for the controlled parameter (for example, space temperature). This input must be valid for this object to function.
- Setpoint: Input for the Setpoint value (for example, space temperature Setpoint). This input must be valid for this object to function.
- Execute Time: Controls the execution frequency for the PID algorithm, where the default value is 1 second.
- Loop Action: Determines whether the control algorithm is direct or reverse acting. Loops setup for direct acting mode increases the loop output as the value of the controlled variable becomes greater than the Setpoint value. In a temperature loop, this is typically considered to be a cooling application. Loops setup for reverse acting mode increases the loop output as the value of the controlled variable becomes less than the

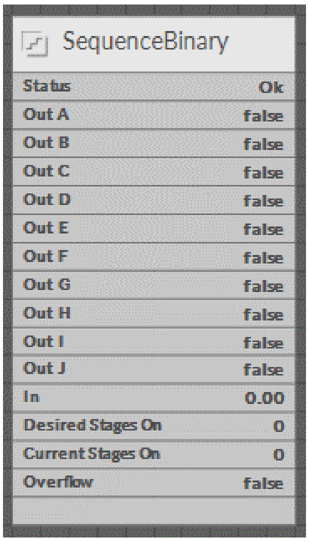
Setpoint value. In a temperature loop, this is typically considered to be a heating application.

Preset:

- Max Value: Sets the loop output value to the Maximum Output property value.
- Min Value: Sets the loop output value to the Minimum Output property value.
- Zero: Sets the loop output value to a zero (0.0) value.
- Proportional Constant: Defines the value of the proportional gain parameter used by the loop algorithm. Used to set the overall gain for the loop. A starting point for this value is found by output range/throttling range.
- Integral Constant: Defines the integral gain parameter, in repeats per minute, used by the loop algorithm. Also, called reset rate. Acts on the magnitude of the Setpoint error. A typical starting point is 0.5.
- Derivative Constant: Defines the derivative gain parameter, in seconds, used by the loop algorithm. Acts on the rate of change of the Setpoint error.
- Bias: Defines the amount of the output bias added to the output to correct offset error, normally used only with proportional control.
- Maximum Output: Defines the maximum output value that the loop algorithm can produce.
- Minimum Output: Defines the minimum output value that the loop algorithm can produce.

5.4 SequenceBinary Component

The SequenceBinary component provides the sequenced weighed "staging" control of 2 to 10 loads based upon the numeric Input value (0-100). It can be used to support applications that need to sequence 2 to 10 loads or stages in a binary sequence. Binary sequencing provides an analog to binary converter function that selects the outputs whose total load rating relates directly to the control need. For each successive output, the output rating is twice the previous output.



SequenceBinary

Status	Ok
Out A	false
Out B	false
Out C	false
Out D	false
Out E	false
Out F	false
Out G	false
Out H	false
Out I	false
Out J	false
In	0.00
Desired Stages On	0
Current Stages On	0
Overflow	false

Figure 20 SequenceBinary component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

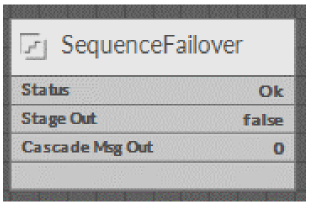
- In: Input property that is used to determine the number of stages that should currently be on.
- In Minimum: Value of the input that produces all outputs off.
- In Maximum: Value of the input that produces all outputs on.
- Number Outputs: This object can be configured to support 2 to 10 outputs or stages.
- OutA - OutJ: Boolean values that can be used to control 2 to 10 loads. The number of outputs used is defined by the Number Outputs property.
- Desired Stages On: Read-only property that indicates the calculated number of stages that should be on based on the In property.
- Current Stages On: Read-only property that indicates the current number of stages that are currently on. Normally the Current Stages On and the Desired Stages On will be the same. They will be different when going through a transition.

Control Signal (In) %	OutC (4kw load size)	OutB (2kw load size)	OutA (1kw load size)	Stage Hysteresis
100	On	On	On	14.3
85,7	On	On	Off	14.3
71,4	On	Off	On	14.3
57,1	On	Off	Off	14.3
42,9	Off	On	On	14.3
28,6	Off	On	Off	14.3
14,3	Off	Off	On	14.3
0	Off	Off	Off	14.3

Table 1 Table illustrates how, by controlling 3 loads, eight unique levels of control can be achieved.

5.5 SequenceFailover Component

The SequenceFailover component is used to cascade and sequence/stage loads based on the Stage status, Cascade Msg In and Fail status inputs.



SequenceFailover

Status	Ok
Stage Out	false
Cascade Msg Out	0

Figure 21 SequenceFailover component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- Stage Status: This configuration property defines where the cascade calculation would be performed.

  If true Msg Out = Msg In.

  If false Msg Out = Msg In – 1.
- Stage Out: Read-only property that indicates Stage Out status.
- Cascade Msg In: Cascade input value slot.
- Cascade Msg Out: Out value slot which indicates cascade value after component calculations.
- Fail Status: This configuration corrects the Cascade output value.

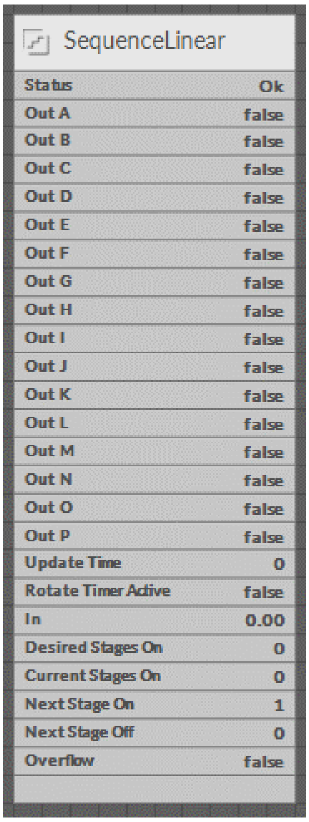
  If true Msg Out = Msg In + 1; Stage Out = false.

  If false Msg Out = Msg In. Stage Out = Stage Status.

5.6 SequenceLinear Component

The SequenceLinear component provides the sequenced rotating "staging" control of 2 to 10 loads based upon the numeric Input value (0-100). A similar object is the SequenceBinary, which uses a weighed method (vs. rotating) for sequencing.

The SequenceLinear component can be used to support applications that need to sequence 2 to 10 loads or stages in a linear or rotating sequence. With linear sequencing, the first stage on will be the last stage off. With rotating sequencing the first stage on will be the first stage off. The In property, which is a Numeric, is used to control the number of stages that should be on. The input range is defined by the InMinimum and InMaximum properties.



SequenceLinear

Status	Ok
Out A	false
Out B	false
Out C	false
Out D	false
Out E	false
Out F	false
Out G	false
Out H	false
Out I	false
Out J	false
Out K	false
Out L	false
Out M	false
Out N	false
Out O	false
Out P	false
Update Time	0
Rotate Timer Active	false
In	0.00
Desired Stages On	0
Current Stages On	0
Next Stage On	1
Next Stage Off	0
Overflow	false

Figure 22 SequenceLinear component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- In: Input property that is used to determine the number of stages that should currently be on.
- In Maximum: Value of the input that produces all outputs on.
- In Minimum: Value of the input that produces all outputs off.

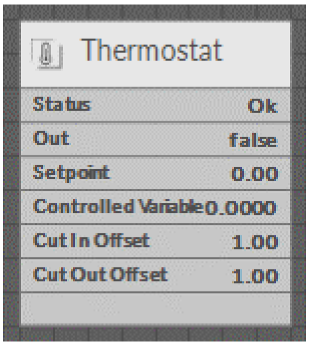
- Number Outputs: This object can be configured to support 2 to 16 outputs or stages.
- OutA - OutP: These are Boolean values that can be used to control 2 to 16 loads. The number of outputs used is defined by the Number Outputs property.
- Desired Stages On: Read-only property that indicates the calculated number of stages that should be on based on the In property.
- Current Stages On: Read-only property that indicates the current number of stages that are currently on. Normally the Current Stages on and the Desired Stages On will be the same. They will be different when going through a transition.
- Next Stage On: Read-only property that indicates the next stage that will be turned on, if needed. This is primarily used when the Mode is selected to be Rotating.
- Next Stage Off: Read-only property that indicates the next stage that will be turned off, if needed. This is primarily used when the Mode is selected to be Rotating.
- Rotate Time: This configuration property specifies the amount of time that the outputs will remain in fixed configuration before they are shifted to the next configuration.
- Rotate Timer Active: Read-only property that indicates that the rotate timer is active.

	Linear	Rotating
Range = InMaximum – InMinimum	100 = 100 – 0	100 = 100 – 0
Delta = range/ NumberOutputs	20 = 100 / 5	20 = 100 / 5
OnSetpointA = 1 * delta	20	20
OnSetpointB = 2 * delta	40	40
OnSetpointC = 3 * delta	60	60
OnSetpointD = 4 * delta	80	80
OnSetpointE = 5 * delta	100	100
OffSetpointA = 0 * delta , 4 * delta	0	80
OffSetpointB = 1 * delta, 3 * delta	20	60
OffSetpointC = 2 * delta, 2 * delta	40	40
OffSetpointD = 3 * delta, 1 * delta	60	20
OffSetpointE = 4 * delta, 0 * delta	80	0

Table 2 SequenceLinear On / Off calculation formulas

5.7 Thermostat Component

The Thermostat component provides the output control based on the input (process) and the setpoint value.



Thermostat

Status Ok
Out false
Setpoint 0.00
Controlled Variable 0.0000
Cut In Offset 1.00
Cut Out Offset 1.00

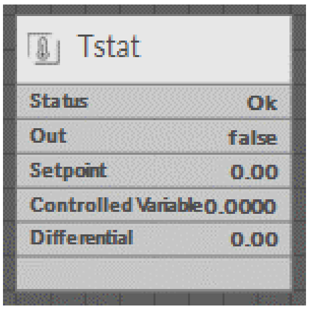
Figure 23 Thermostat component view

The component has the following slots:

- Set Point: Desired/target value.
- Cut in Offset: Defines the differential value between the Controlled Variable and Setpoint to determine the Thermostat output on state. A positive Cut In Offset value means the value is greater than Setpoint, and a negative CutInOffset value means the value is lower than Setpoint during the comparison. For cooling control, use positive value and negative value for heating control.
- Cut Out Offset: Defines the differential value between the Controlled Variable and SetPoint to determine the Thermostat output off state. A positive Cut Out Offset value means greater than SetPoint, and a negative CutOutOffset value means lower than SetPoint during the comparison. For cooling control, use a negative value and positive value for heating control.

5.8 Tstat Component

The Tstat component provides a basic thermostatic (On/Off) control with the Boolean Out property and Numeric inputs for controlled variable (Cv), Setpoint (Sp), and differential (Diff).



Tstat

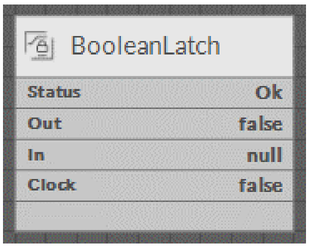
Status Ok
Out false
Setpoint 0.00
Controlled Variable 0.0000
Differential 0.00

Figure 24 Tstat component view

6 Latch Components

6.1 BooleanLatch Component

The BooleanLatch component provides a latch for a Boolean input. Any latch that is invoked using the Clock property must include a method for setting the Clock property status back to False before the Clock is available for latching again.



BooleanLatch

Status	Ok
Out	false
In	null
Clock	false

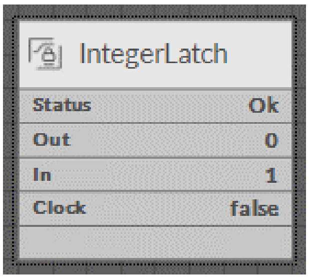
Figure 25 BooleanLatch component view

The component has the following slots:

- Clock: The BooleanBoolean property that has either a True or False state for all latch components. This property latches the input property to the output property on the rising edge of input. It means that a single input property is captured and sent to the output property at the instant that the Clock status changes from False to True state, and NOT when the property changes from True to False state.
- Out: This standard component property provides the actual latched value that is captured from the input property at the latch time. Link to this property to display the value on a graphic or to process the value with another component.
- In: This is the standard component input property that may be linked into from a data source. For example, there may be a link into this property from a control point or the Schedule output.

6.2 IntegerLatch Component

The IntegerLatch component provides a latch for an integer input. Any latch that is invoked using the Clock property must include a method for setting the Clock property status back to False before the Clock is available for latching again.



IntegerLatch

Status Ok

Out 0

In 1

Clock false

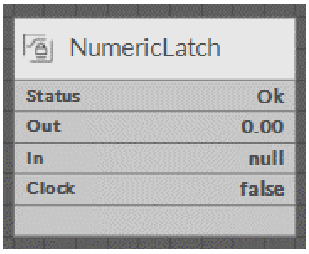
Figure 26 IntegerLatch component view

The component has the following slots:

- Clock: The Boolean property that has either a True or False state for all latch components. This property latches the input property to the output property on the rising edge of input. This means that a single input property is captured and sent to the output property at the instant that the Clock status changes from False to True state, and NOT when the property changes from True to False state.
- Out: This standard component property provides the actual latched value that is captured from the input property at the latch time. Link to this property to display the value on a graphic or to process the value with another component.
- In: The standard component input property that may be linked into from a data source. For example, there may be a link to this property from a control point or a Schedule output.

6.3 NumericLatch Component

The NumericLatch component provides a latch for a Boolean input. Any latch that is invoked using the Clock property must include a method for setting the Clock property status back to False before the Clock is available for latching again.



NumericLatch

Status Ok

Out 0.00

In null

Clock false

Figure 27 NumericLatch component view

The component has the following slots:

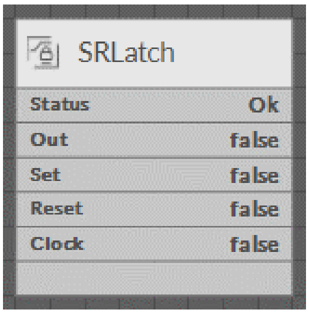
- Clock: The BBoolean property that has either a True or False state for all latch components. This property latches the input property to the output property on the rising edge of input. This means that a single input property is captured and sent to the output property at the instant that the Clock status changes from False to True state, and NOT when the property changes from True to False state.

- Out: This standard component property provides the actual latched value that is captured from the input property at the latch time. Link to this property to display the value on a graphic or to process the value with another component.
- In: This is the standard component input property that may be linked into from a data source. For example, there may be a link into this property from a control point or a Schedule output.

6.4 SRLatch Component

The Set/Reset Latch component — a single-bit edge-triggered data storage. The following logic applies to the false-to-true transition of S or R:

If S goes true and R does not change, then Out = true and remains true.
If R goes true and S does not change, then Out = false and remains false.
If both S and R go true on the same scan, then Out = false and remains false.



SRLatch

Status	Ok
Out	false
Set	false
Reset	false
Clock	false

Figure 28 SRLatch component view

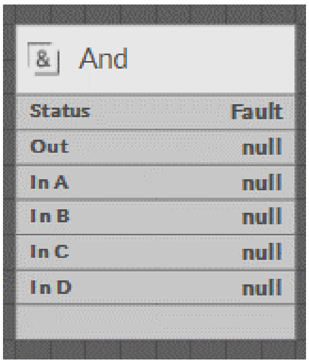
The component has the following slots:

- Out: This standard component property provides the actual latched value that is captured from the input property at the latch time. Link to this property to display the value on a graphic or to process the value with another component.
- In: This is the standard component input property that may be linked into from a data source. For example, there may be a link into this property from a control point or a Schedule output.

7 Logic Components

7.1 AND Component

The AND Component performs a logical AND on all inputs and writes the result to the out property. Table 3 shows the AND object truth table when using two inputs. Table 4 shows the AND object truth table if using all four inputs.



And

Status Fault

Out null

In A null

In B null

In C null

In D null

Figure 29 AND component view

In A	In B	Out
False	False	False
False	True	False
True	False	False
True	True	True

Table 3 AND object truth table (2 inputs)

In A	In B	In C	In D	Out
False	False	False	False	False
False	False	False	True	False
False	False	True	False	False
False	False	True	True	False
False	True	False	False	False
False	True	False	True	False
False	True	True	False	False
False	True	True	True	False

True	False	False	False	False
True	False	False	True	False
True	False	True	False	False
True	False	True	True	False
True	True	False	False	False
True	True	False	True	False
True	True	True	False	False
True	True	True	True	True

Table 4 AND object truth table (4 inputs)

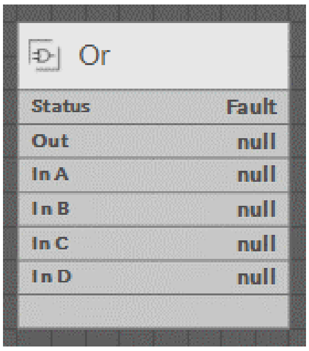
LogicExpr:

LogicExpr is a Binary Logic Object where various Logic Operations are being performed on one/two Boolean inputs based on the operator.

out:= (inA & inB)	when operator == 0 (And)
out:= (inA | inB)	when operator == 1 (Or)
out:= (inA ^ inB)	when operator == 2 (Xor)
out:= !inA	when operator == 3 (Not)
out:= !(inA & inB)	when operator == 4 (Nand)
out: =! (inA | inB)	when operator == 5 (Nor)

7.2 OR Component

The Or component performs a logical OR on all valid inputs and writes the Boolean result to the out property. Table 5 shows the OR object truth table when using two inputs. Table 6 shows the OR object truth table when using all four inputs. NOR gate logic is accomplished by linking to a Not object.



Or

Status	Fault
Out	null
In A	null
In B	null
In C	null
In D	null

Figure 30 OR component view

In A	In B	Out
False	False	False
False	True	True
True	False	True
True	True	True

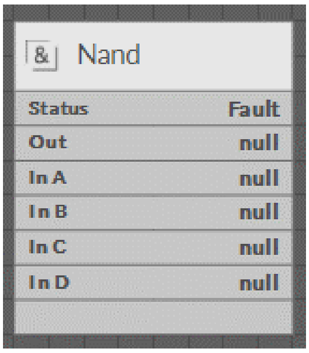
Table 5 OR object truth table (2 inputs)

In A	In B	In C	In D	Out
False	False	False	False	False
False	False	False	True	True
False	False	True	False	True
False	False	True	True	True
False	True	False	False	True
False	True	False	True	True
False	True	True	False	True
False	True	True	True	True
True	False	False	False	True
True	False	False	True	True
True	False	True	False	True
True	False	True	True	True
True	True	False	False	True
True	True	False	True	True
True	True	True	False	True
True	True	True	True	True

Table 6 OR object truth table (4 inputs)

7.3 Nand Component

The Nand component set the out value to false if all inputs are true.



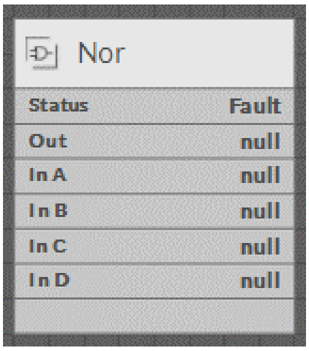
Nand

Status	Fault
Out	null
In A	null
In B	null
In C	null
In D	null

Figure 31 Nand component view

7.4 NOR Component

The Nor Component set the out value to true if all inputs are false.



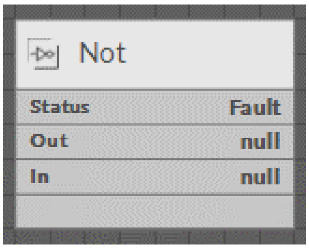
Nor

Status	Fault
Out	null
In A	null
In B	null
In C	null
In D	null

Figure 32 Nor component view

7.5 NOT Component

The Not Component simply inverts the Boolean logic value currently at the (single) object input.



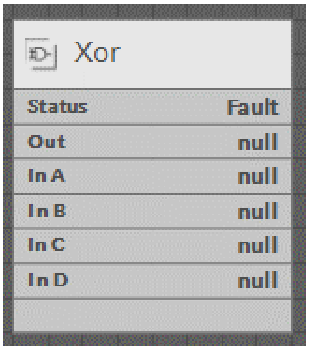
Not

Status	Fault
Out	null
In	null

Figure 33 NOT component view

7.6 XOR Component

The XOR Component performs a logical XOR on all valid inputs and writes the result to the out property. Table 7 shows the XOR component truth table if using all four inputs. Table 8 shows the XOR component truth table when using two inputs (typical).



Xor

Status	Fault
Out	null
In A	null
In B	null
In C	null
In D	null

Figure 34 XOR component view

In A	In B	In C	In D	Out
False	False	False	False	False
False	False	False	True	True
False	False	True	False	True
False	False	True	True	False
False	True	False	False	True
False	True	False	True	False
False	True	True	False	False
False	True	True	True	True
True	False	False	False	True
True	False	False	True	False
True	False	True	False	False
True	False	True	True	True
True	True	False	False	False
True	True	False	True	True
True	True	True	False	True
True	True	True	True	False

Table 6 Xor object truth table (4 inputs)

In A	In B	Out
False	False	False

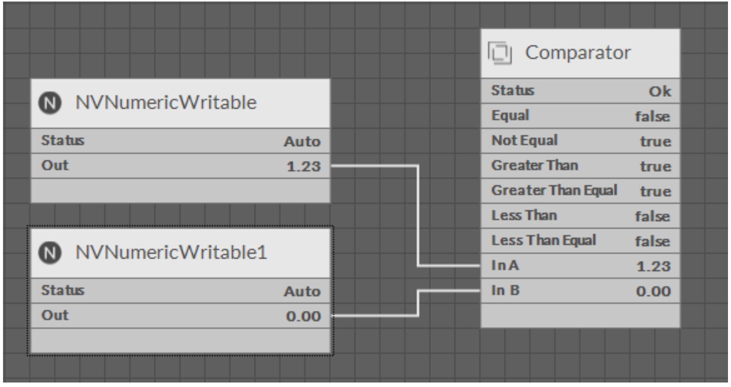
False	True	True
True	False	True
True	True	False

Table 8 Xor object truth table (2 inputs)

Eight types of components have Numeric inputs:

7.7 Comparator Component

The Comparator component performs a Numeric Comparison of two numeric inputs and raises the respective Flags.



NVNumericWritable
Status Auto
Out 1.23

NVNumericWritable1
Status Auto
Out 0.00

Comparator
Status Ok
Equal false
Not Equal true
Greater Than true
Greater Than Equal true
Less Than false
Less Than Equal false
In A 1.23
In B 0.00

Figure 35 Comparator component view

equal := (inA == inB)

Note:qual := (inA != inB)

greaterThan := (inA > inB)

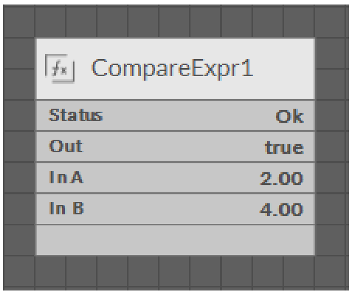
greaterThanEqual := (inA >= inB)

lessThan := (inA < inB)

lessThanEqual := (inA <= inB)

7.8 ComparatorExpr Component

The ComparatorExpr Component is a Comparator Object where various Comparator Operations are being performed on two Float inputs based on the operator.



CompareExpr1

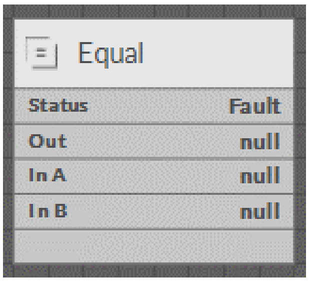
Status	Ok
Out	true
In A	2.00
In B	4.00

Figure 36 XOR component view

out:= (inA == inB)	when operator == 0 (Equal)
out:= (inA != inB)	when operator == 1 (Note:qual)
out:= (inA > inB)	when operator == 2 (GreaterThan)
out:= (inA >= inB)	when operator == 3 (GreaterThanEqual)
out:= (inA < inB)	when operator == 4 (LessThan)
out:= (inA <= inB)	when operator == 5 (LessThanEqual)

7.9 Equal Component

The Equal component performs the operation InA == InB for Numeric values. Nan values are never equal.



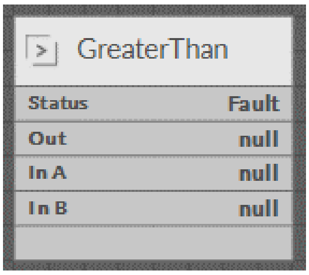
Equal

Status	Fault
Out	null
In A	null
In B	null

Figure 37 Equal component view

7.10GreaterThan Component

The GreaterThan component performs the operation InA > InB with a Boolean result.



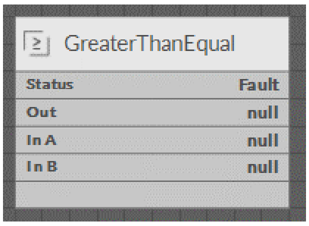
GreaterThan

Status	Fault
Out	null
In A	null
In B	null

Figure 38 GreaterThan component view

7.11 GreaterThanEqual Component

The GreaterThanEqual component performs the operation InA >= InB with a Boolean result.



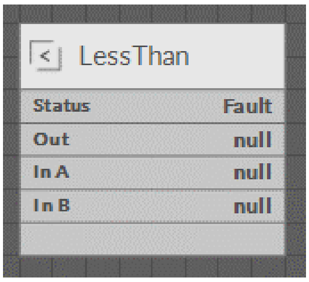
GreaterThanEqual

Status	Fault
Out	null
In A	null
In B	null

Figure 39 GreaterThanEqual component view

7.12 LessThan Component

The LessThan component performs the operation InA < InB with a Boolean result.



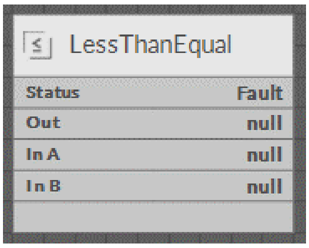
LessThan

Status	Fault
Out	null
In A	null
In B	null

Figure 40 LessThan component view

7.13 LessThanEqual Component

The LessThanEqual component performs the operation InA <= InB with a Boolean result.



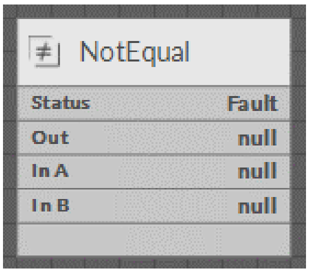
LessThanEqual

Status	Fault
Out	null
In A	null
In B	null

Figure 41 LessThanEqual component view

7.14NotEqual Component

The NotEqual component performs the operation InA! = InB with a Boolean result.



NotEqual

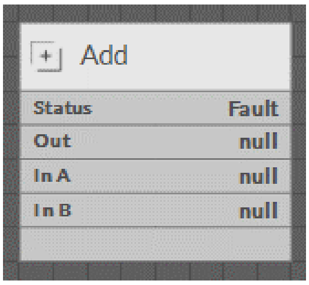
Status	Fault
Out	null
In A	null
In B	null

Figure 42 NotEqual component view

8 Math Components

8.1 Add Component

The Add component performs the operation out:= (InA + InB + InC + InD).



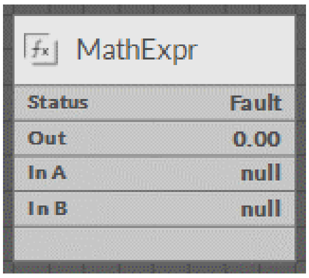
Add

Status	Fault
Out	null
In A	null
In B	null

Figure 43 Add component view

8.2 MathExpr Component

The MathExpr is a component in which various Mathematical & Trigonometric Operations can be performed on one/two Numeric inputs based on the operator.



MathExpr

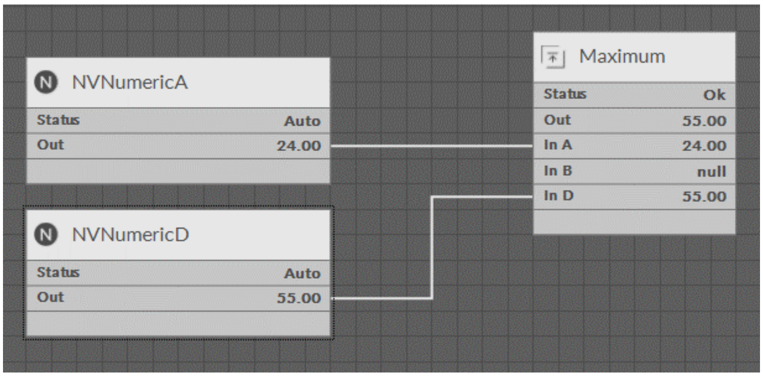
Status	Fault
Out	0.00
In A	null
In B	null

Figure 44 MathExpr component view

out:= fabs (in)	when operator = 0 (AbsValue)
out := inA + inB	when operator = 1 (Add)
out := acos (in)	when operator = 2 (ArcCosine)
out := asin (in)	when operator = 3 (ArcSine)
out := atan (in)	when operator = 4 (ArcTangent)
out := cos (in)	when operator = 5 (Cosine)
out := inA / inB	when operator = 6 (Divide)
out := e ^ in	when operator = 7 (Exponential)
out := inA!	when operator = 8 (Factorial)
out := log10 (in)	when operator = 9 (LogBase10)
out := ln (in)	when operator = 10 (LogNatural)
out := inA % inB	when operator = 11 (Modulus)
out := inA * inB	when operator = 12 (Multiply)
out := -in	when operator = 13 (Negative)
out := inA ^ inB	when operator = 14 (Power)
out := round (in)	when operator = 15 (Round)
out := sin (in)	when operator = 16 (Sine)
out := sqrt (in)	when operator = 17 (SquareRoot)
out := inA - inB	when operator = 18 (Subtract)
out := tan (in)	when operator = 19 (Tangent)
out := trunc (in)	when operator = 20 (Truncate)

8.3 Maximum Component

The Maximum component determines the maximum value of valid inputs and writes that value to out. Out:= max (InA, InB, InC, InD)



NVNumericA
Status Auto
Out 24.00

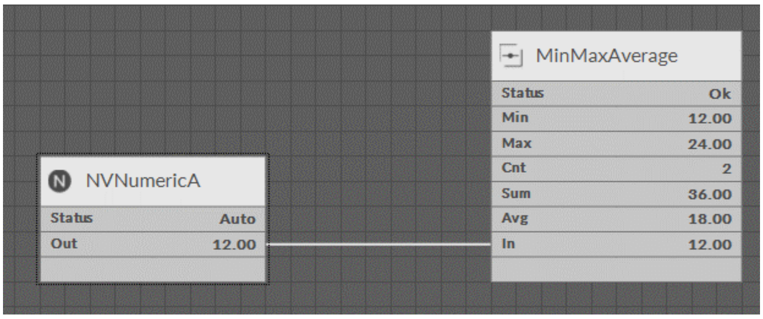
NVNumericD
Status Auto
Out 55.00

Maximum
Status Ok
Out 55.00
In A 24.00
In B null
In D 55.00

Figure 45 Maximum component view

8.4 MinMaxAvg Component

The MinMaxAvg component has 5 Numeric output slots that provide the current minimum, maximum, count, sum, and average values of linked Numeric input.



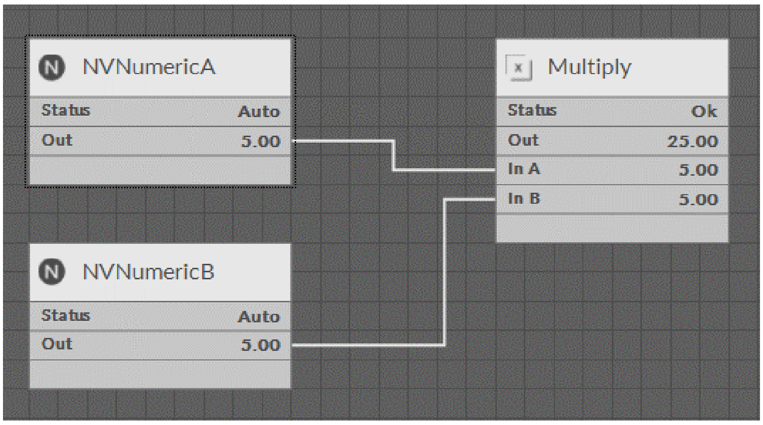
MinMaxAverage
Status Ok
Min 12.00
Max 24.00
Cnt 2
Sum 36.00
Avg 18.00
In 12.00

NVNumericA
Status Auto
Out 12.00

Figure 46 MinMaxAvg component view

8.5 Multiply Component

The Multiply component performs the calculation Out := InA * InB * InC * InD.



NVNumericA
Status Auto
Out 5.00

NVNumericB
Status Auto
Out 5.00

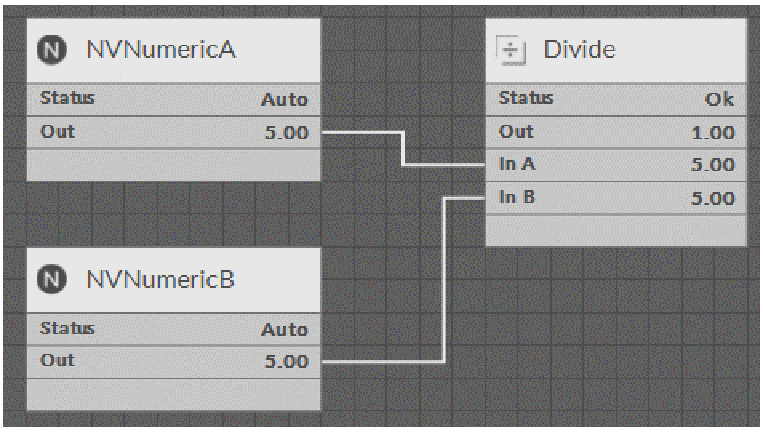
Multiply
Status Ok
Out 25.00
In A 5.00
In B 5.00

Figure 47 Multiply component view

The following Math types perform an operation using two inputs:

8.6 Divide Component

The Divide component performs the operation out := (in A / in B). If either input is Numeric.Null, the output will be Numeric.Null.



NVNumericA
Status Auto
Out 5.00

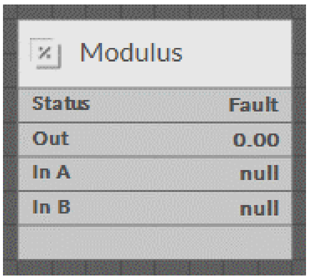
NVNumericB
Status Auto
Out 5.00

Divide
Status Ok
Out 1.00
In A 5.00
In B 5.00

Figure 48 Divide component view

8.7 Modulus Component

The Modulus component provides a modulus operation based on values at its two Numeric inputs. The output is the remainder of dividing the InA value by the InB value. If the InB value is 0, the output is NaN (not a number).



Modulus

Status Fault

Out 0.00

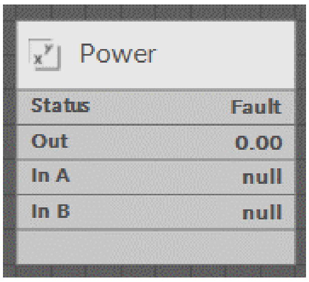
In A null

In B null

Figure 49 Modulus component view

8.8 Power Component

The Power component performs the operation out := (InA ^ InB) or InA raised to the InB power.



Power

Status Fault

Out 0.00

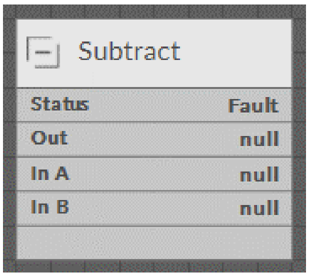
In A null

In B null

Figure 50 Power component view

8.9 Subtract Component

The Subtract component performs the operation out := (InA - InB). If either input is Numeric.NaN, the output will be Numeric.NaN.



Subtract

Status Fault

Out null

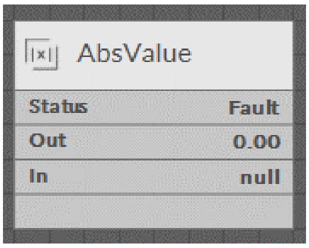
In A null

In B null

Figure 51 Substract component view

8.10AbsValue Component

The AbsValue component performs the operation out := abs (In) (absolute value of In).



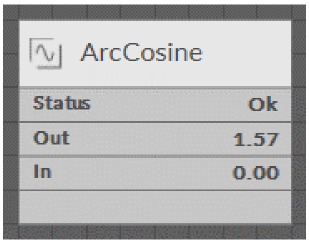
AbsValue

Status	Fault
Out	0.00
In	null

Figure 52 AbsValue component view

8.11 ArcCosine Component

The ArcCosine component performs the operation out := acos (inA).



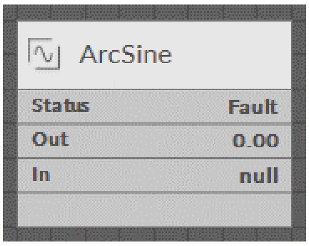
ArcCosine

Status	Ok
Out	1.57
In	0.00

Figure 53 ArcCosine component view

8.12 ArcSine Component

The ArcSine component performs the operation out := asin (inA).



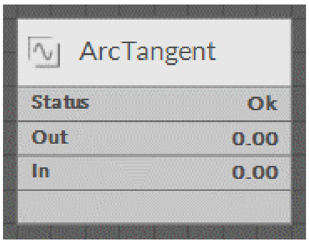
ArcSine

Status	Fault
Out	0.00
In	null

Figure 54 ArcSine component view

8.13 ArcTangent Component

The ArcTangent component performs the operation out := atan (inA).



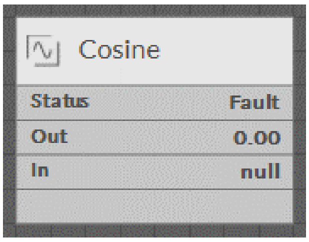
ArcTangent

Status	Ok
Out	0.00
In	0.00

Figure 55 ArcTangent component view

8.14Cosine Component

The Cosine component performs the operation out := cos (in A).



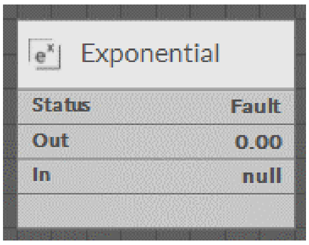
Cosine

Status Fault
Out 0.00
In null

Figure 56 Cosine component view

8.15Exponential Component

The Exponential component performs the operation out := e ^ inA (e raised to the inA power).



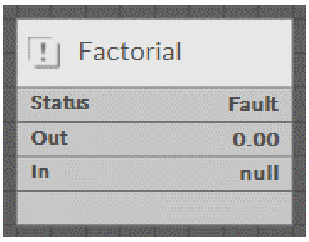
Exponential

Status Fault
Out 0.00
In null

Figure 57 Exponential component view

8.16Factorial Component

The Factorial component provides a factorial math output, based upon the value present at its Numeric input. Only the integer portion of the input value is evaluated, for example, either value of 1.03 or 1.9999 is evaluated as 1.



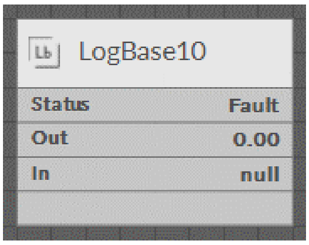
Factorial

Status Fault
Out 0.00
In null

Figure 58 Factorial component view

8.17LogBase10 Component

The LogBase10 component performs the operation out := log10 (inA) (log base 10 of inA).



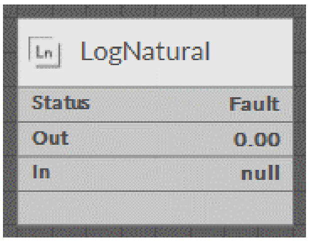
LogBase10

Status	Fault
Out	0.00
In	null

Figure 59 LogBase10 component view

8.18LogNatural Component

The LogNatural component performs the operation out := ln (inA) (log base e of inA).



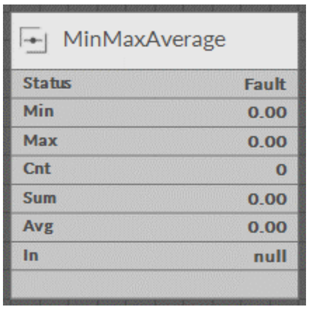
LogNatural

Status	Fault
Out	0.00
In	null

Figure 60 LogNatural component view

8.19MinMaxAverage Component

The MinMaxAverage component has 5 numeric output slots that provide the current minimum, maximum, count, sum, and average values of a linked Numeric input.



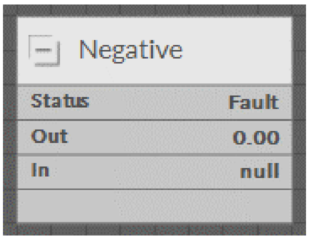
MinMaxAverage

Status	Fault
Min	0.00
Max	0.00
Cnt	0
Sum	0.00
Avg	0.00
In	null

Figure 61 MinMaxAverage component view

8.20Negative Component

The Negative component simply converts any numeric input to a negative output value.



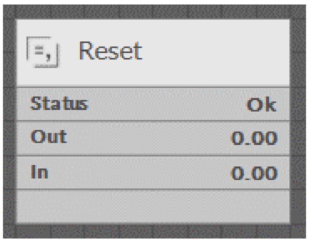
Negative

Status	Fault
Out	0.00
In	null

Figure 62 Negative component view

8.21 Reset Component

The Reset component performs a linear "reset" on the inA value.



Reset

Status	Ok
Out	0.00
In	0.00

Figure 63 Reset component view

The Reset operation is defined by the following four slots:

Input Low Limit: must be less than the Input High Limit.
Input High Limit: must be greater than the Input Low Limit.
Output Low Limit: may (or may not) be greater than the Output High Limit.
Output High Limit: may (or may not) be greater than the Output Low Limit.

For example, a Reset component is used to establish a hot water control Setpoint, based on the outside air temperature at inA. When the outside air temperature is 0˚F, the hot water Setpoint is 200˚F. When the outside air temperature is 75˚F, the hot water Setpoint is 100˚F. The Reset component is configured as:

Input Low Limit = 0.0

Input High Limit = 75.0

Output Low Limit = 200.0

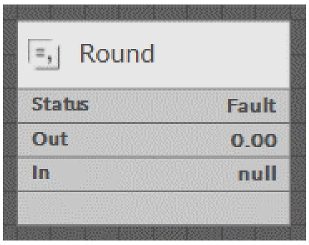
Output High Limit = 100.0

Whenever the inA value is beyond the input limits, the output is limited by the corresponding output limit (in this case, 200 at 0˚F or below, 100 at 75˚F, or above). When the input is at an intermediate value, the output scales linearly. For example, when the outside air temperature is at 38.2˚F, the Reset output is 149.1˚F.

8.22Round Component

The Round component performs the mathematical operation of returning the nearest integer, rounding away from zero in halfway cases.

Out := round (in)



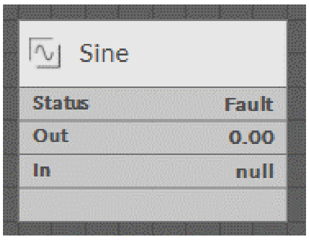
Round

Status	Fault
Out	0.00
In	null

Figure 64 Round component view

8.23Sine Component

The Sine component performs the operation out := sin (InA).



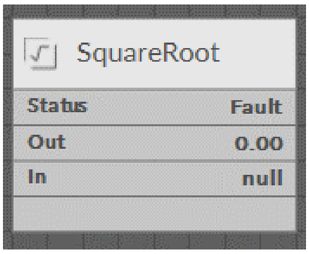
Sine

Status	Fault
Out	0.00
In	null

Figure 65 Sine component view

8.24SquareRoot Component

The SquareRoot component performs the operation out := sqrt (InA) (square root of InA).



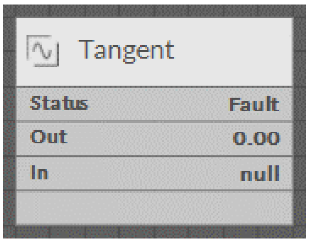
SquareRoot

Status	Fault
Out	0.00
In	null

Figure 66 SquareRoot component view

8.25Tangent Component

The Tangent component performs the operation out := tan(InA).



Tangent

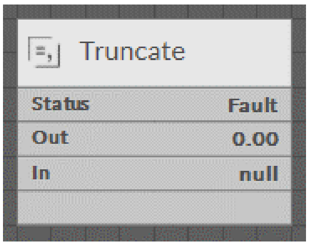
Status	Fault
Out	0.00
In	null

Figure 67 Tangent component view

8.26Truncate Component

The Truncate component performs the mathematical operation of returning the nearest integer, not greater in magnitude than the Input Float.

Out := trunc (in)



Truncate

Status	Fault
Out	0.00
In	null

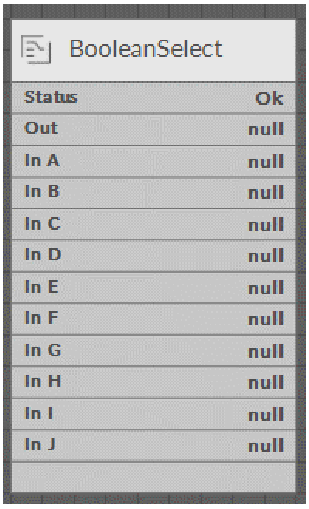
Figure 68 Truncate component view

9 Select Components

9.1 BooleanSelect Component

The BooleanSelect component allows one of the multiple Boolean inputs to be selected (passed to the output) upon selection by the value at its "Select" (Integer) input. Inputs from 3 to 10 can be specified.

Note: All select objects require an integer input to perform the selection by the type of an input data selected and passed to the "Out" slot.



BooleanSelect

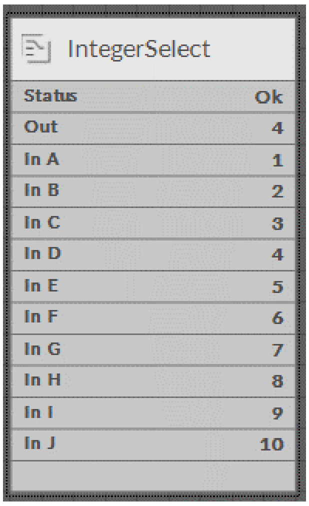
Status	Ok
Out	null
In A	null
In B	null
In C	null
In D	null
In E	null
In F	null
In G	null
In H	null
In I	null
In J	null

Figure 69 BooleanSelect component view

9.2 IntegerSelect Component

The IntegerSelect component allows one of the multiple Integer inputs to be selected (passed to the output) upon selection by the value at its "Select" (Integer) input. Inputs from 3 to 10 can be specified.

Note: All select objects require an integer input to perform the selection by the type of an input data selected and passed to the "Out" slot.



IntegerSelect

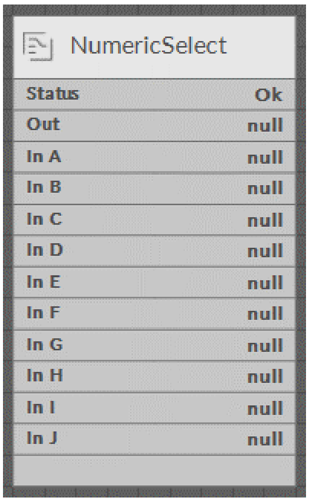
Status	Ok
Out	4
In A	1
In B	2
In C	3
In D	4
In E	5
In F	6
In G	7
In H	8
In I	9
In J	10

Figure 70 IntegerSelect component view

9.3 NumericSelect Component

The NumericSelect component allows one of the multiple Numeric inputs to be selected (passed to the output) upon selection by the value at its "Select" (Integer) input. Inputs from 3 to 10 can be specified.

Note: All select objects require an integer input to perform the selection by the type of an input data selected and passed to the "Out" slot.



NumericSelect

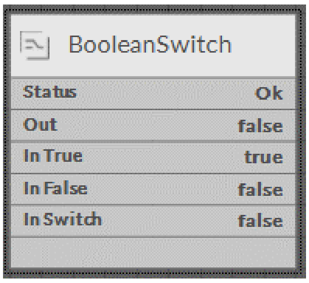
Status	Ok
Out	null
In A	null
In B	null
In C	null
In D	null
In E	null
In F	null
In G	null
In H	null
In I	null
In J	null

Figure 71 NumericSelect component view

10 Switch Components

10.1 BooleanSwitch Component

The BooleanSwitch component selects between two Boolean inputs based upon the Boolean value at the Boolean input 'In Switch'.



BooleanSwitch

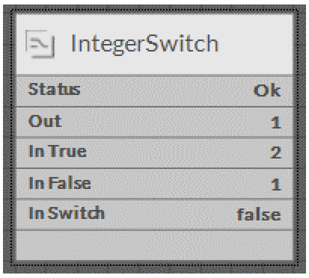
Status	Ok
Out	false
In True	true
In False	false
In Switch	false

Figure 72 BooleanSwitch component view

Note: All select objects require a Boolean input to perform the selection by the type of an input data selected and passed to the "Out" slot.

10.2IntegerSwitch Component

The IntegerSwitch component selects between two Integer inputs based upon the Boolean value at the Boolean input 'In Switch'.



IntegerSwitch

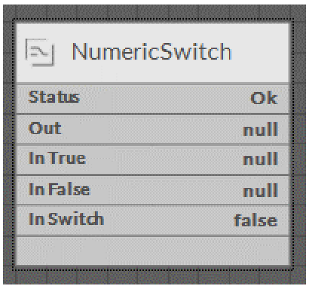
Status	Ok
Out	1
In True	2
In False	1
In Switch	false

Figure 73 IntegerSwitch component view

Note: All select objects require a Boolean input to perform the selection by the type of an input data selected and passed to the "Out" slot.

10.3NumericSwitch Component

The NumericSwitch component selects between two Numeric inputs based upon the Boolean value at the Boolean input 'In Switch'.



NumericSwitch

Status	Ok
Out	null
In True	null
In False	null
In Switch	false

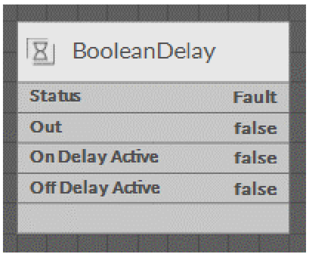
Figure 74 NumericSwitch component view

Note: All select objects require a Boolean input to perform the selection by the type of an input data selected and passed to the "Out" slot.

11 Timer Components

11.1 BooleanDelay Component

The BooleanDelay component provides the way to delay the change of a Boolean out property value by configuring an associated "Delay" property. Delay properties are provided for on (true) and off (false) statuses and are labelled "On Delay" and "Off Delay", respectively. The delay applies to any transition (status change from on to off or off to on) at the component's Boolean input. Both delay times are configurable in terms of hours, minutes, and seconds.



BooleanDelay

Status	Fault
Out	false
On Delay Active	false
Off Delay Active	false

Figure 75 BooleanDelay component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

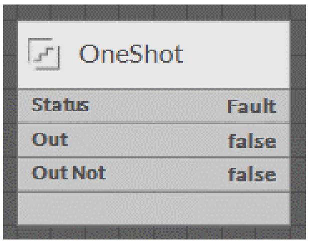
- In: Typically, this property is set by linking a Boolean out value into it. The default state can be manually configured to a true or false value so that when no other value is linked into this property, the default value is used. This property value is passed to the Out (after any On Delay or Off Delay) whenever there is a change in this property.
- On Delay: This property allows to set the amount of time (in hours, minutes, and seconds) that will expire before sending a true (On) value to the Out property. Time begins to expire at the moment that a change in the In property occurs (a transition from false or null to true). If the On Delay value is 0, the change of the state proceeds without delay.
- Off Delay: This property allows to set the amount of time (in hours, minutes, and seconds) that will expire before sending a false (Off) value to the Out property. The time begins at the moment that a change in the In property occurs (a transition from true to false or false to true). If the Off Delay value is 0, the change of the state proceeds without delay.
- On Delay Active: This read-only property shows whether or not the On Delay time is actively counting down to expiration. This (normally false) value changes to true anytime that a transition from false to true occurs at the In property and stays at true until any Off Delay time is expired. If the On Delay value is set to "0", then this value does

not change to true.

- Off Delay Active: This read-only property shows whether or not the Off Delay time is actively counting down to expiration. This (normally false) value changes to true anytime that a transition from true to false occurs at the In property and stays at true until any Off Delay time is expired. If the On Delay value is set to "0", then this value does not change to true.
- Out: This property has true and false options available. These values are set at the end of any On Delay or Off Delay to reflect the In property value.

11.2OneShot Component

The OneShot component provides a single, temporary, Boolean output for a specified duration (as set in the Time property). The OneShot action occurs with a False-to-True value transition at the In property or with an invoked Fire action. When either of these conditions occurs, the Out property value is set to True, and the Out Not property value is set to False for a time that is equal to the value of the Time property. When the time expires, these values revert to the previous (default) values.



OneShot

Status	Fault
Out	false
Out Not	false

Figure 76 OneShot component view

***Slots with the dots are hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

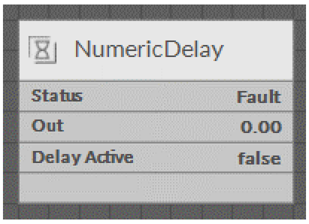
The component has the following slots:

- In: Typically, this property is set by linking a Boolean Out value into it. The default state can be manually configured to a Boolean value so that when no other value is linked into this property, the default value is used. This property value is passed to the component's Out property for the amount of time set in the Time property.
- Time: The value of this property determines how long the Out and Out Not properties hold their "one-shot" values.
- Out: This property displays the current value that changes with a False to True transition at the In property value. After the OneShot is triggered and the Time value period expires, this value returns to the default (False) value.

- Out Not: This property has true or false options available. The Out value change with a False to True transition at the In property value or a "Fire" action. After the OneShot is triggered and the Time value period expires, this value returns to the default (True) value.

11.3NumericDelay Component

The NumericDelay component provides a way to delay the change of a numeric Out property value by configuring an associated Delay property. The delay applies to any change at the component's numeric input. The delay time is configurable in terms of hours, minutes and seconds.



NumericDelay

Status	Fault
Out	0.00
Delay Active	false

Figure 77 NumericDelay component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

- In: Typically, this property is set by linking a numeric out value into it. The default state can be manually configured to a true or false value so that when no other value is linked into this property, the default value is used. This property value is passed to the Out (after Delay) whenever there is a change in this property.
- Delay: This property allows to set the amount of time (in hours, minutes, and seconds) that will expire before sending the In value to the Out property. The time begins to expire at the moment that a change in the In property occurs.
- Delay Active: This read-only property shows whether or not the Delay time is actively counting down to expiration. This (normally false) value changes to true anytime that a change in the In property occurs and stays at true until any Delay time is expired. If the Delay value is set to "0", then this value does not change to true.
- Out: This property is a numeric output. These values are set at the end of any Delay to reflect the In-property value.

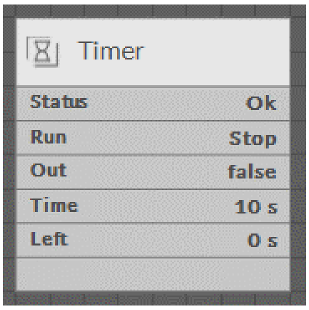
11.4Timer Component

The Timer component outputs a pulse for the configured amount of time; the In slot is used to fire the timer:

if low, out is forced to false,

if high, out = 1 until the timer reaches Time value in seconds.

Alternatively, the pulse can be fired from the Start Timer action if In is not linked.



Timer	
Status	Ok
Run	Stop
Out	false
Time	10 s
Left	0 s

Figure 78 Timer component view

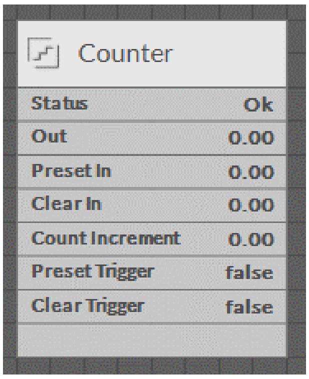
The component has the following slots:

- Out: A timed pulse output.
- Run: Used to fire the timer on transition from false -> true
- Time: Desired duration of the output pulse.
- Left: Remaining time before the output transition from true -> false.

12 Util Components

12.1 Counter Component

The Counter component will count the Boolean inactive to active transitions. It supports counting up, counting down, presenting, and clearing.



Counter	
Status	Ok
Out	0.00
Preset In	0.00
Clear In	0.00
Count Increment	0.00
Preset Trigger	false
Clear Trigger	false

Figure 79 Counter component view

***Slots with the dots are hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

The component has the following slots:

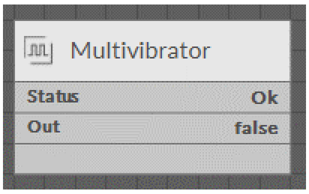
- Count Up: This is a Boolean input. When this input makes an inactive to active transition the value of the Out property increments by the Count Increment value.
- Count Down: This is a Boolean input. When this input makes an inactive to active transition the value of the Out property will be decremented by the Count Increment value.
- Preset In: This is a Numeric input which will be set in the Out property when the Preset action is invoked.
- Clear In: This is a Numeric input which will be set in the Out property when the Clear action is invoked.
- Count Increment: This is the value that the Out property will change for a single count up or count down in active transition.
- Preset Trigger: This is a Boolean input. When this input makes an inactive to active transition it invokes the Preset action.
- Clear Trigger: This is a Boolean input. When this input makes an inactive to active transition it clears the Preset value.

The Counter component includes the following actions:

- Preset: The Preset action when invoked, the value of the Out property will be set to the Preset In value.
- Clear: The Clear action when invoked, the value of the Out property will be set to the Clear In value.

12.2MultiVibrator Component

The MultiVibrator component provides an oscillating binary pulse output (Boolean) with a period configurable from 1s and a duty cycle configurable from 0 to 100.



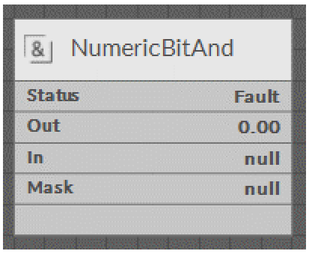
Multivibrator

Status	Ok
Out	false

Figure 80 MultiVibrator component view

12.3NumericBitAnd Component

The NumericBitAnd component performs a logical AND on the bit equivalent of the Numeric In value against the bit equivalent of its Numeric Mask slot value. It may be useful in cases where the Boolean information is mapped into integer values.



NumericBitAnd

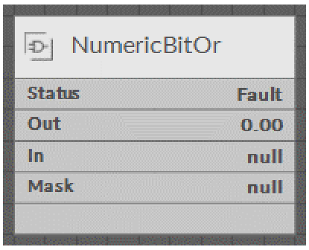
Status	Fault
Out	0.00
In	null
Mask	null

Figure 81 NumericBitAnd component view

12.4NumericBitOr Component

The NumericBitOr component performs a logical OR on the bit equivalent of the Numeric In value against the bit equivalent of its Numeric Mask slot value. It may be useful in cases where the Boolean information is mapped into integer values.

As an example, some manufacturers multiplex binary data into a single numerical point by converting the bits from hexadecimal to decimal format. To obtain the status of the individual binary data, the number must be converted back from decimal to hex format. Each digit of the hex number represents a particular binary parameters state (0 = false, 1 = true). The NumericBitOr component converts a Numeric input to a hex value and compares it against the Mask value. Any digits with a value of 1 in the Mask or the input will result in a corresponding value of 1 in the same digit of the output. Any value on the output slot greater than 1 indicates that at least one of the binary parameters is true.



NumericBitOr

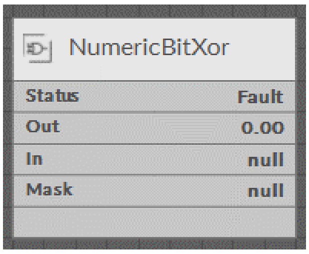
Status	Fault
Out	0.00
In	null
Mask	null

Figure 82 NumericBitOr component view

12.5NumericBitXor Component

The NumericBitXor component performs a logical XOR on the bit equivalent of the Numeric In value against the bit equivalent of its Numeric Mask slot value. It may be useful in cases where the Boolean information is mapped into integer values.

As an example, some manufacturers multiplex binary data into a single numerical point by converting the bits from hexadecimal to decimal format. To obtain the status of the individual binary data, the number must be converted back from decimal to hex format. Each digit of the hex number represents a particular binary parameters state (0 = false, 1 = true). The NumericBitXor component converts a Numeric input to a hex value and compares it against the Mask value. Each digit is analysed using exclusive OR (XOR) logic, setting the corresponding digit value to either 1 or 0.



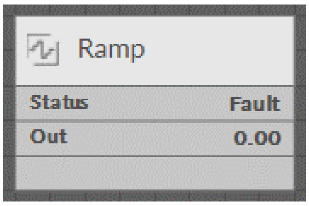
NumericBitXor

Status	Fault
Out	0.00
In	null
Mask	null

Figure 83 NumericBitXor component view

12.6Ramp Component

The Ramp component provides a Numeric Out with a linear ramping output. Slots define the Period, Amplitude, and Offset.



Ramp

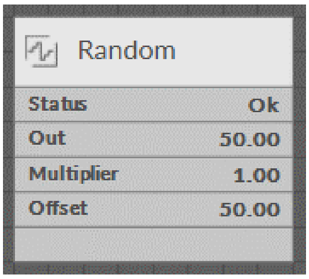
Status	Fault
Out	0.00

Figure 84 Ramp component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

12.7Random Component

The Random component can be used to generate random numbers. The output is derived by multiplying a random number (that is greater than 0 but less than 1) times a variable Multiplier plus Offset value.



Random

Status	Ok
Out	50.00
Multiplier	1.00
Offset	50.00

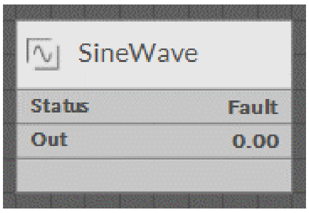
Figure 85 Random component view

The component has the following slots:

- Multiplier: This is a double value that is used to multiply by the random number (the random number is >=0.0 but <1.0). The multiplier is set to 1.0 by default.
- Offset: This is the positive or negative distance from zero that the wave's amplitude is centered on. The default offset value is 50.

12.8SineWave Component

The SineWave component generates a sine wave as a Numeric out.



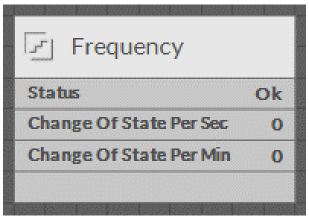
SineWave

Status	Fault
Out	0.00

Figure 86 Random component view

12.9Frequency Component

The Frequency component calculates a pulse input frequency.



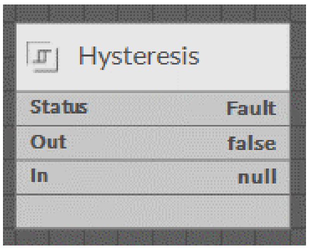
Frequency

Status	Ok
Change Of State Per Sec	0
Change Of State Per Min	0

Figure 87 Frequency component view

12.10 Hysteresis Component

The Hysteresis component sets on/off trip points to an input variable.



Hysteresis

Status	Fault
Out	false
In	null

Figure 88 Hysteresis component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

There are two internal floats called Rising Edge and Falling Edge which are configurable.

If risingEdge > fallingEdge, then out behaves "normally", i.e.

out := true when in rises above risingEdge

out := false when in falls below fallingEdge

If risingEdge < fallingEdge, then out behaves "inverted", i.e.

out := false when in rises above fallingEdge

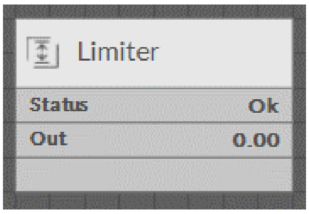
out := true when in falls below risingEdge

If risingEdge == fallingEdge, this object behaves as a simple comparator,

out := true when in > Rising Edge

12.11 Limiter Component

The Limiter component object restricts the output based on the input between lowLimit and highLimit values.



Limiter

Status	Ok
Out	0.00

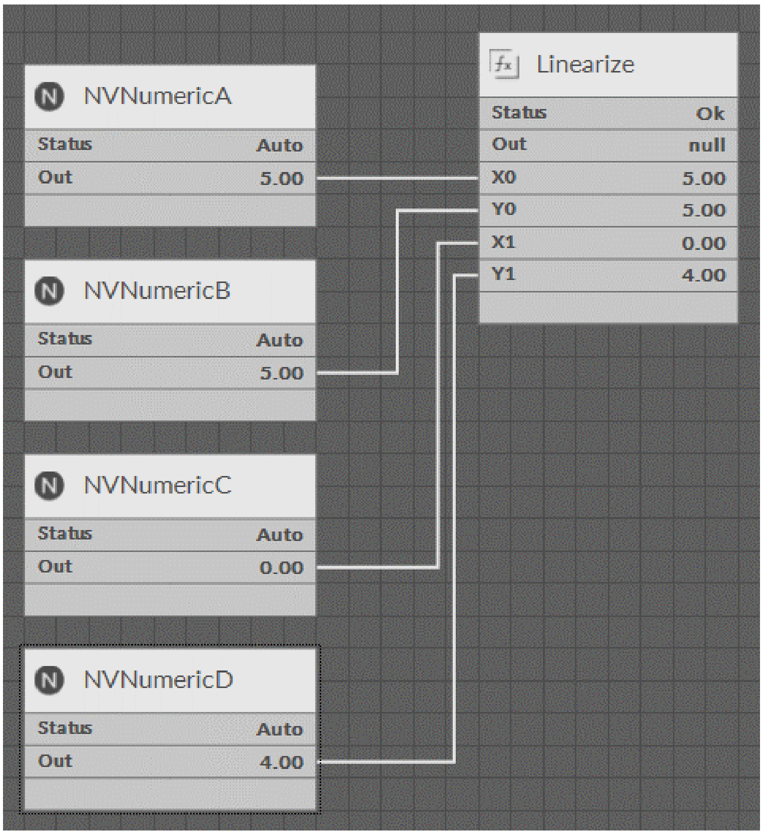
Figure 89 Limiter component view

***Some slots may be hidden by default, when they are connected, they become uncovered and are being shown in the Wire Sheet.

HighLimit and LowLimit are configurable floats:
out := highLimit when in > highLimit
out := lowLimit when in < lowLimit
out := in when lowLimit < in < highLimit.

12.12 Linearize Component

The Linearize component performs a piecewise linearization of afloat.
For piecewise linearization of a nonlinear input, there are ten pairs of x, y parameters that must be configured into this component. The x, y pairs indicate points along the input curve. For an x value of the input, there should be a corresponding y value of the output.



NVNumericA
Status Auto
Out 5.00

NVNumericB
Status Auto
Out 5.00

NVNumericC
Status Auto
Out 0.00

NVNumericD
Status Auto
Out 4.00

Linearize
Status Ok
Out null
X0 5.00
Y0 5.00
X1 0.00
Y1 4.00

Figure 90 Linearize component view

For input values between these points, the component will estimate the output based upon the linear equation, therefore it converts a table of values into a curve using linear interpolation between the values.

The x,y pairs indicate points along the input curve, for each x value of the input there should be the corresponding y value of the output.

Individual slope/intercept constants are computed between the x's and y's using the formula y = mx + b, where m = ym - yn/xm - xn.

If In is not in the range of x0 to x9, then output is set to "nan".

Note: The slope may be positive or negative, and it is indicated by a comparison of x1 and x0.

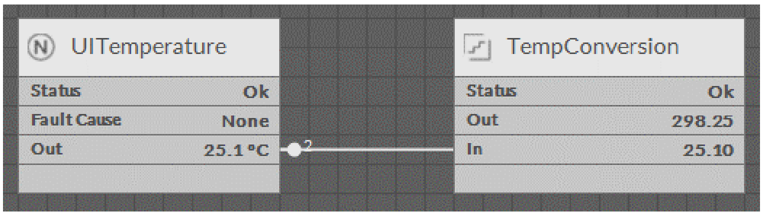
Positive if x1 > x0

Negative if x1 < x0

Out:= (m * in) + b, where m is the slope between the adjacent points and b is the Y Intercept

12.13 TempConversion Component

The TempConversion component converts the Temperature from one unit to another.



UITemperature	
Status	Ok
Fault Cause	None
Out	25.1 °C

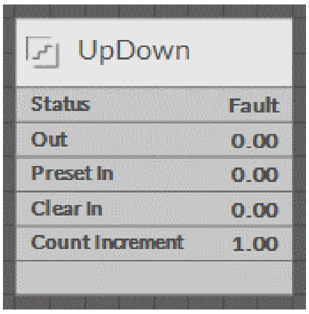
TempConversion	
Status	Ok
Out	298.25
In	25.10

Figure 91 TempConversion component view

out := in	when in = Celsius & out = Celsius
out := (in - 32.0) * (5.0/9.0)	when in = Celsius & out = fahrenheit
out := in + 273.0	when in = Celsius & out = kelvin
out := (in * 1.8) + 32.0	when in = Fahrenheit & out = Celsius
out := in	when in = Fahrenheit & out = Fahrenheit
out := (in * 1.8) + 32.0 + 273.0	when in = Fahrenheit & out = kelvin
out := in - 273.0	when in = kelvin & out = Celsius
out := ((in - 273.0) - 32.0) * (5.0/9.0)	when in = kelvin & out = Fahrenheit
out := in	when in = kelvin & out = kelvin

12.14 UpDown Component

The UpDown component will count basing on the Count Increment property. It supports counting up, counting down, presetting, and clearing.



UpDown	
Status	Fault
Out	0.00
Preset In	0.00
Clear In	0.00
Count Increment	1.00

Figure 92 UpDown component view

out := out + countIncrement	when mode = true (Up Mode)
out := out - countIncrement	when mode = false(Down Mode)
out := No Change	when mode = null (Disable)
out := presetValue	when preset action is fired
out := 0.0	when clear action is fired