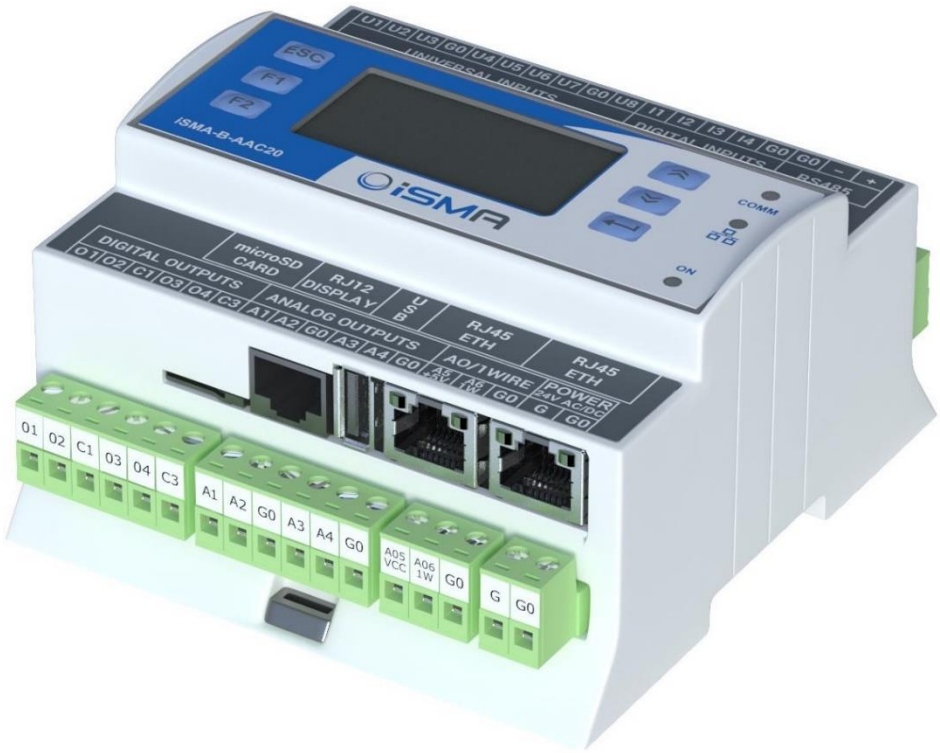
Intelligent Solution Managing Automation

iSMA Control kit

User Manual



Powered by sedona FRAMEWORK™

Global Control 5 Sp. z o.o.
Warsaw, Poland
www.gc5.pl

Table of contents

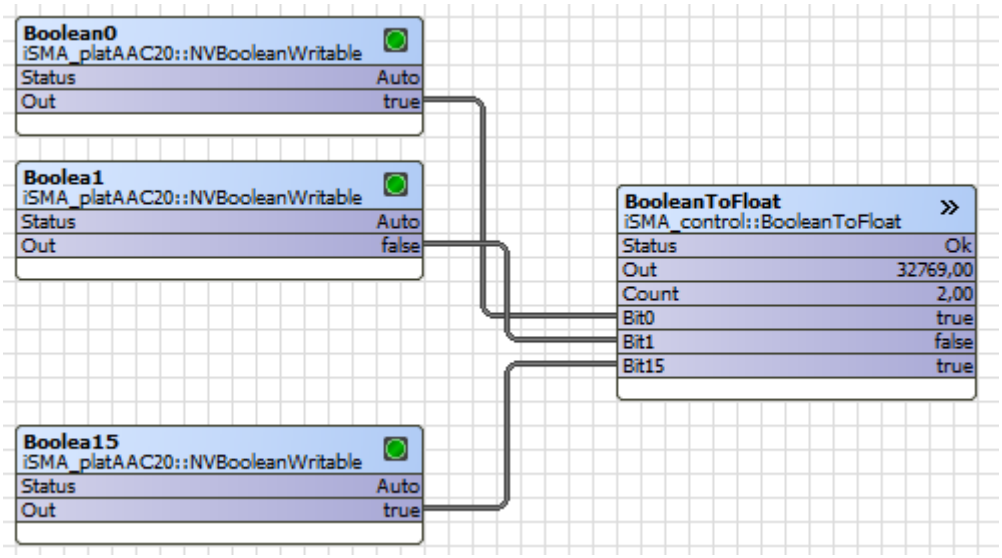
1 iSMA Control kit

This manual contains information about iSMA Control Kit in 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 AAC20 controller and cannot be uninstalled.

2 Conversion Components

2.1.1 BooleanToFloat Component

BooleanToFloat component is a component which converts 16 boolean signals to 1 Float signal.

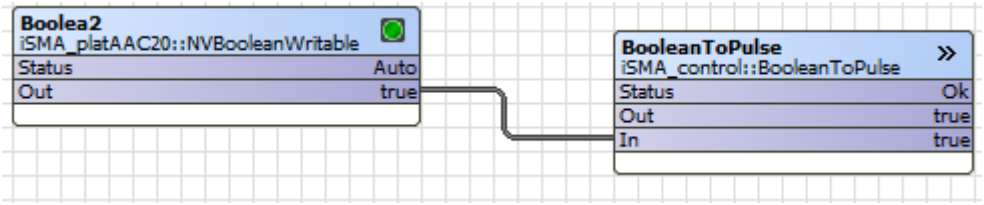


The component has the following slots:

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

2.1.2 BooleanToPulse Component

BooleanToPulse Component is a component which converts 1 Boolean Signal to Pulse. BooleanToPulse is a simple mono-stable oscillator object.

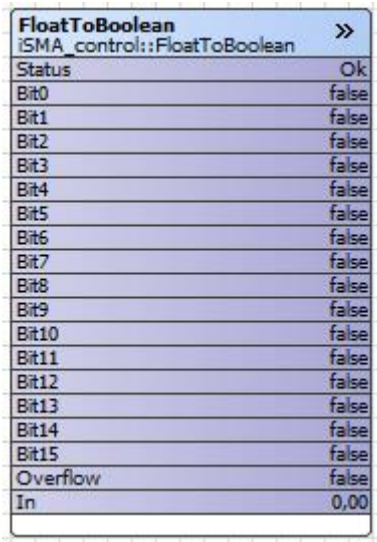


The component has the following slots:

- Out: = in for one scan cycle on the rising edge of in.

2.1.3 FloatToBoolean Component

FloatToBoolean Component is converts a 16-bit Float to Binary Decoder Object.

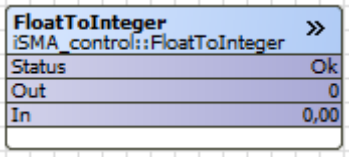


The component has the following slots:

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

2.1.4 FloatToInteger Component

FloatToInteger is a Float to 32-bit integer (Integer) Converter Object.

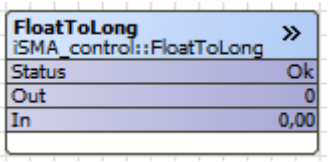


The component has the following slots:

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

2.1.5 FloatToLong Component

FloatToLong is a Float to 64-bit signed integer (Long) Converter Object.

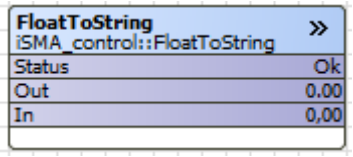


The component has the following slots:

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

2.1.6 FloatToString Component

FloatToStraing Component is a component which converts Float to String Objects.

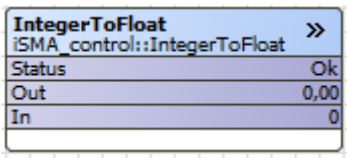


The component has following slots:

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

2.1.7 IntegerToFloat Component

IntegerToFloat Component is component which converts 32 integer bits to float object.

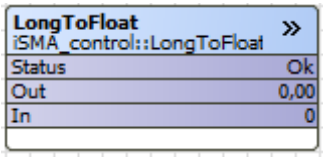


The component has the following slots:

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

2.1.8 LongToFloat Component

LongToFloat Component is component which converts 64-bit signed , Integer (long) to float object.



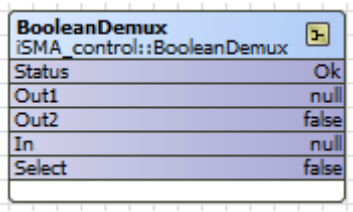
The component has the following slots:

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

3 Demux Components

3.1.1 BooleanDemux Component

BooleanDemux object selects one of two outputs to receive the Input (Boolean) Value, depending on the value of the Boolean select Input. The value of the other Output remains unchanged.

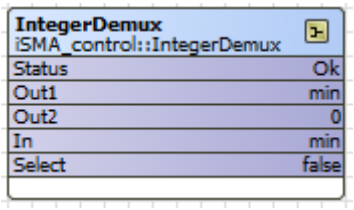


The component has the following slots:

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

3.1.2 IntegerDemux Components

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

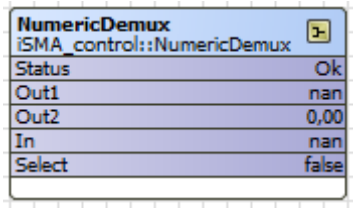


The component has the following slots:

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

3.1.3 NumericDemux Component

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



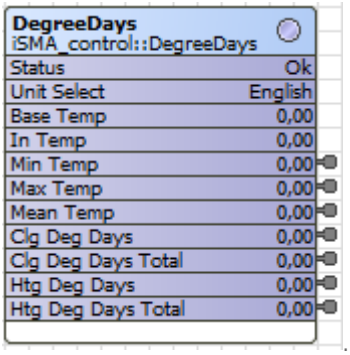
The component has the following slots:

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

4 Energy Components

4.1.1 DegreeDays

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



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

The component has the following slots:

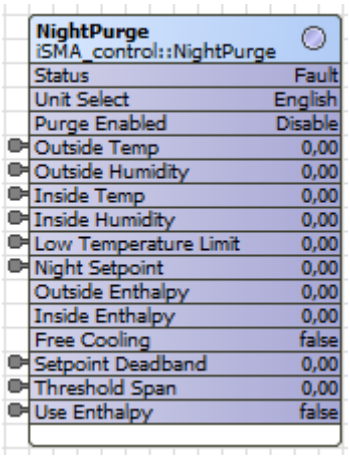
- Unit Select: This is used to set the units of the Temp In, Min Temp, Max Temp, and Mean Temp properties.
- Base Temperature: Specifies the base temperature used in the degree-day calculation.

- Input Temperature: This is 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 last Reset Totals action was invoked. Calculated when Cooling Degree Days changes.
- Heating Degree-day: This is the heating degree-day calculated for the previous day. Calculated when the day changes.

- Totalized Heating Degree-days: This is the totalized heating degree-days since last Reset Totals action was invoked. Calculated when Heating Degree Days changes.

4.1.2 NightPurge Component

It uses the two sets of temperature and humidity inputs to find the air supply with the least amount of heat when the purgeEnabled input is true. The freeCooling output will be set to false if outside >= inside or set to true if outside = nightSetpoint.



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

The component has the following slots:

- Unit Select: Specifies the units of Temperature and Humidity properties.
- Purge Enabled: Boolean, must be true to enable night purge operation. Whenever false, the Free Cooling output is set to the opposite of the Free Cooling Command (or null, if Use Null Output is set to true), and the Current Mode slot value is "Disabled." Often, Purge Enabled is linked to a "Not" object sourced from BooleanSchedule output.
- 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.
- 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 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 Used Null Output is set to true).
- Current Mode: This enumeration indicates which of the following modes this object 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)

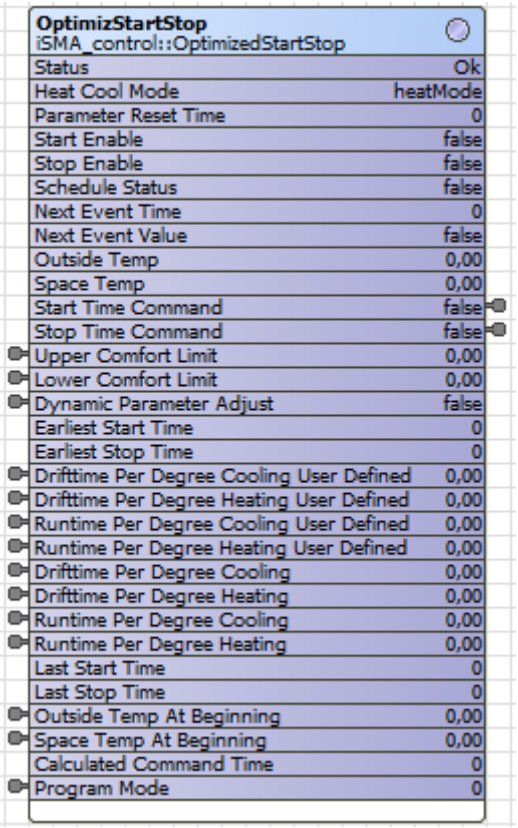
  Input error (A temperature or humidity is invalid (down, fault, etc.), free cooling disabled)

  Satisfied (Inside temperature below Night Setpoint, free cooling disabled)
- Setpoint Deadband: Temperature Setpoint deadband applied when inside temperature falls below Night Setpoint, before free cooling can be enabled. 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. 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.

4.1.3 OptimizedStartStop Component

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.



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

The component has the following slots:

- Heat Cool Mode: This boolean property allows enabling either the heatMode or the coolMode. 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 drifttime properties change to the User Defined values. The OSS component copies the user defined drifttime and runtime property values to the corresponding actual drifttime 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 a schedule for the time of the next scheduled

event.

- Next Event Value: This property is linked to a schedule and reflects the value of the action for 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 you link to a control for invoking an equipment start command. For example, it can be linked to a prioritized input of a boolean writable - or directly to the equipment Start control.
- Stop Time Command : This boolean property is an output that you link to a control for invoking 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 a 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 weight the dynamic parameter adjustment calculation. The value that you specify in this field affects how much weighting you assign to the previous runtime property value when it is used in the dynamic parameter adjustment calculation. A larger value increases the amount of weighting given to the previous runtime and a smaller value decreases the weighting.
- Earliest Start Time: This property allows you 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 this time.
- Earliest Stop Time : This property allows you 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 this time.
- Drifttime Per Degree Cooling User Defined: This property allows you to set a default value for calculating the rate of drift in cooling mode. When you save a value to this field, the value is copied to the Drifttime Per Degree Cooling field.
- Drifttime Per Degree Heating User Defined: This property allows you to set a default value for calculating the rate of drift in heating mode. When you save a value to this field, the value is copied to the Drifttime Per Degree Heating field.
- Runtime Per Degree Cooling User Defined: This property allows you to set a default value for calculating the runtime value in cooling mode. When you save a value to this field, the value is copied to the Runtime Per Degree Cooling field.

- Runtime Per Degree Heating User Defined: This property allows you to set a default value for calculating the runtime value in heating mode. When you save a value to this field, the value 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 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 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 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 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 for 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 for 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 be informative to the system engineer, as well. The Program Mode value displays the current heating or cooling state for 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 valued indicates that the optimized start calculation processes ongoing but that an optimized start or stop is not yet in progress.

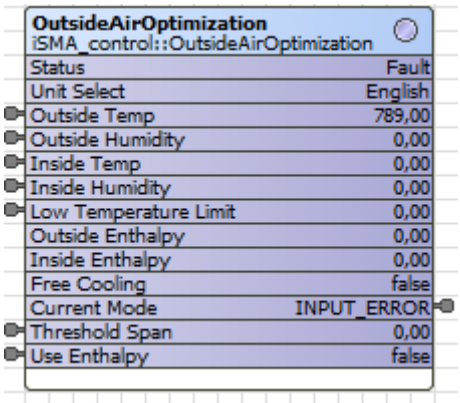
2 ("Start" in Process)
This value indicates that an optimized start has been initiated.

3 ("Stop" Calculation)
This value indicates that an 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 an optimized stop has been initiated.

4.1.4 OutsideAirOptimization Component

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 freeCooling output is set to false if outside >= inside and set to true if outside <= inside - (abs) thresholdSpan. You can select temperature or enthalpy comparisons. There is also a low temperature check to protect against freezing.



*Slots with the dots are hidden by default, when they're connected to, they become uncovered and are being shown in 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 CoolingCommand when it is determined that free cooling should be used. Otherwise, the value is set to null.
- Current Mode: This indicates what mode this object is currently in.

  Input out of range

  Free Cooling

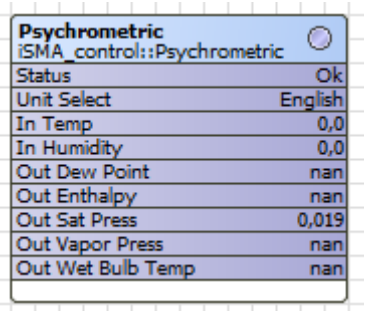
  No Free Cooling

  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.

4.1.5 Psychrometric Component

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



The component has the following slots:

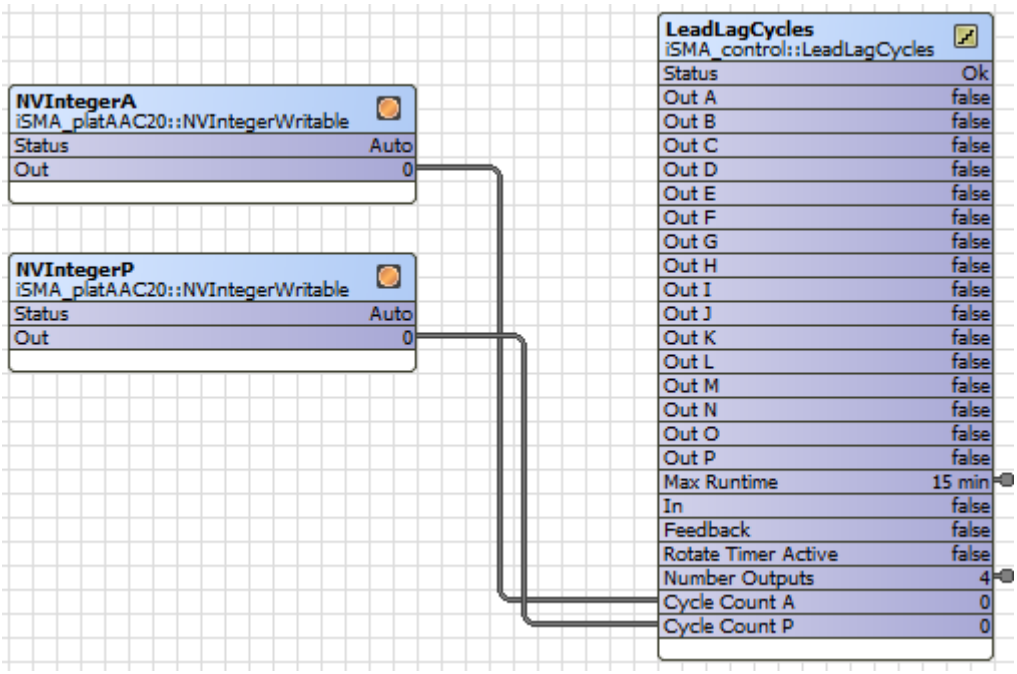
- Unit Select : Used to set the units of the Temperature and Humidity properties.
- Input Temperature: Input temperature
- Input humidity: Input humidity
- Dew Point Temperature: Calculated dew point temperature. Requires valid Input Temperature and Input Humidity to calculate.
- Enthalpy: Calculated enthalpy. Requires valid Input Temperature and Input Humidity to calculate.
- Saturated Pressure: Calculated saturated pressure. Requires valid Input Temp 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 HAVAC Component

5.1.1 LeadLagCycles Component

LeadLagCycles provides lead-lag control of 2 to 16 loads based upon their accumulated COS (change of state) counts. This object 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.

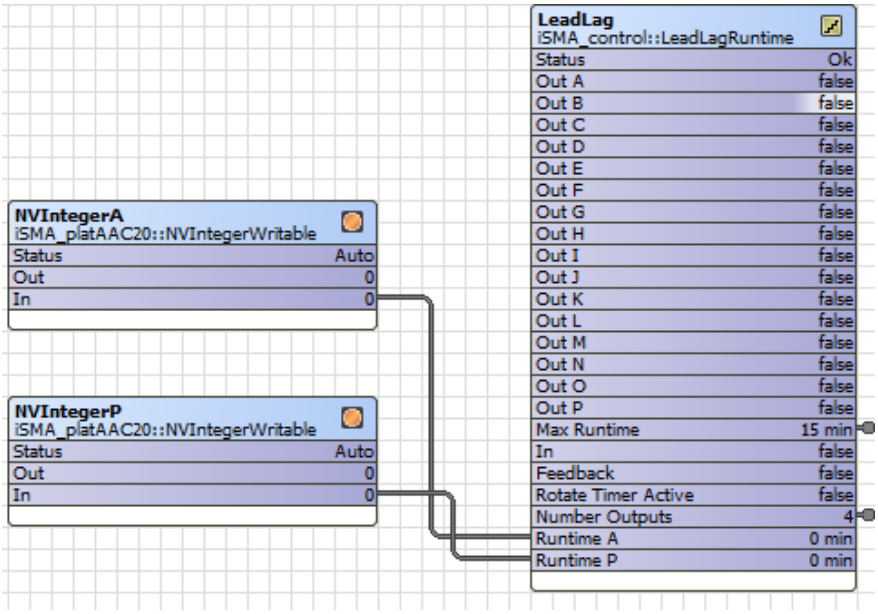


*Slots with the dots are hidden by default, when they're connected to, they become uncovered and are being shown in Wire Sheet The component has the following slots:

- In: A 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: A Boolean input, to provide positive feedback that a controlled device actually started. If the feedback value does not show true within the Feedback Delay time, the current controlled output will show alarm, and the LeadLagCycles switches to the next controlled output. Setting this value to true (and not linking) disables this alarm feature.
- Out A-P: Boolean outputs, each typically linked to a BooleanWritable controlpoint with a DiscreteTotalizerExt. Outputs are typically used to control loads of some type, such as 2 or more pumps.
- Cycle Count A - P: These are 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.

5.1.2 LeadLagRuntime Component

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

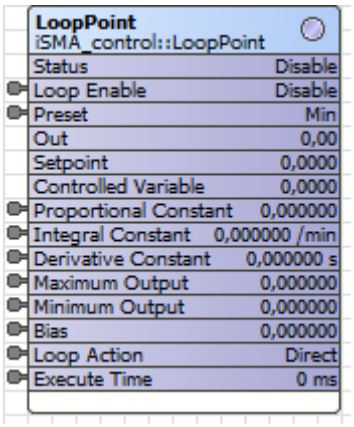


*Slots with the dots are hidden by default, when they're connected to, they become uncovered and are being shown in Wire Sheet The component has the following slots:

- In: A 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: A Boolean input, to provide positive feedback that a controlled device actually started. If the feedback value does not show true within the Feedback Delay time, the current controlled output will show alarm, and the LeadLagRuntime switches to the next controlled output. Setting this value to true (and not linking) disables this alarm feature.
- Out A - P : Boolean outputs, each typically linked to a BooleanWritable control point with a DiscreteTotalizerExt. Outputs are typically used to control loads of some type, such as 2 or more pumps.
- Runtime A - P: These are 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.

5.1.3 LoopPoint Component

LoopPoint 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.



*Slots with the dots are hidden by default, when we they're connected to, they become uncovered and are being shown in 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 increase 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 increase 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.

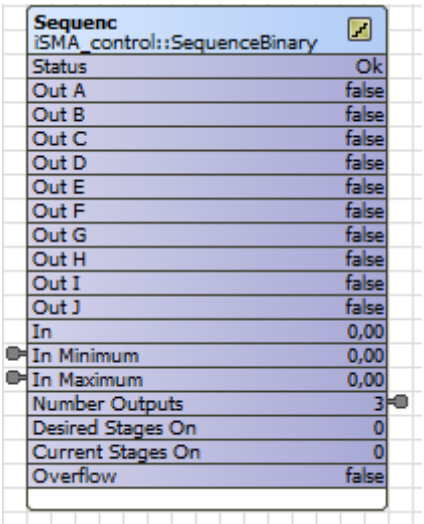
Preset:

- 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 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 output bias added to the output to correct offset error, normally used only used 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.

5.1.4 SequenceBinary Component

SequenceBinary component provides sequenced weighted "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.



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

The component has the following slots:

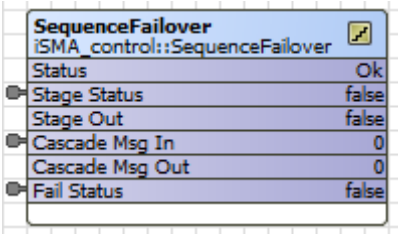
- In: Input property that is used to determine the number of stages that should currently be On.
- In Minimum: Value of the input that produces all outputs off.
- In Maximum: Value of the input that produces all outputs on.
- Number Outputs: This object can be configured to support 2 to 10 outputs or stages.
- OutA - OutJ : These are 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.

5.1.5 SequenceFailover Component

SequenceFailover component is used to cascade and sequence/stage loads based on the stageStatus, cascadeMsgIn and failStatus inputs.

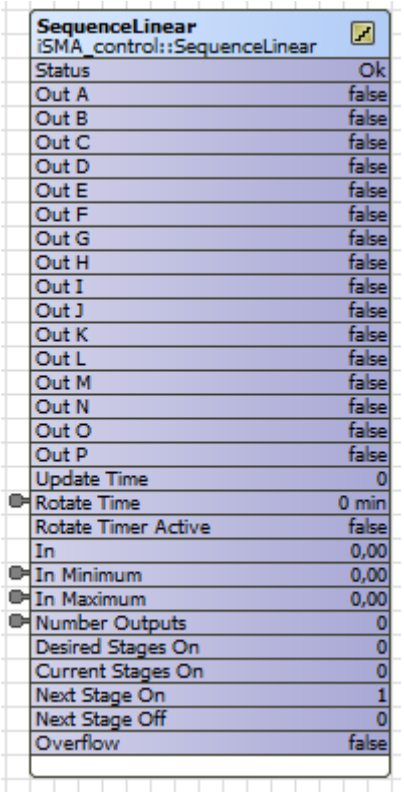


*Slots with the dots are hidden by default, when we they're connected to, they become uncovered and are being shown in Wire Sheet

5.1.6 SequenceLinear Component

SequenceLinear provides 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 weighted method (vs. rotating) for sequencing.

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.



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

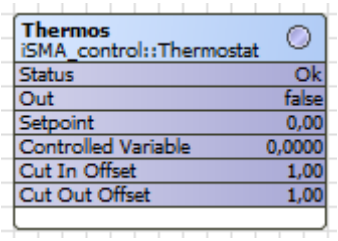
- Next Stage On: Read-only property that indicates the next stage that will be turned on if needed. This is primarily used when the Mode is selected to be Rotating.
- Next Stage Off: Read-only property that indicates the next stage that will be turned off if needed. This is primarily used when the Mode is selected to be Rotating.
- Rotate Time: This configuration property specifies the amount of time that the outputs will remain in a fixed configuration before the outputs 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

5.1.7 Thermostat Component

Thermostat component provides the output control based on the input (process) and the set point value.

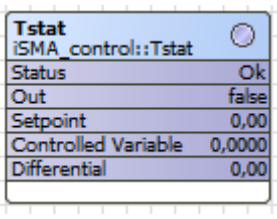


The component has the following slots:

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

5.1.8 Tstat Component

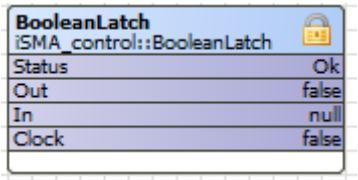
Tstat provides basic thermostatic (On/Off) control with a Boolean Out property and Numeric inputs for controlled variable (Cv), Setpoint (Sp), and differential (Diff).



6 Latch Components

6.1.1 BooleanLatch Component

BooleanLatch 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.

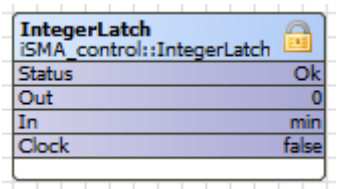


The component has the following slots:

- Clock: This is a 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". This means that a single input property is captured and sent to the output property at the instant that the Clock status changes from a False to a True state and NOT when the property changes from a True to a False state.
- Out : This standard component property provides the actual latched value that is captured from the input property at "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 you link into from a data source. For example, you can link into this property from a control point or a Schedule output.

6.1.2 IntegerLatch Component

IntegerLatch provides a latch for a 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.

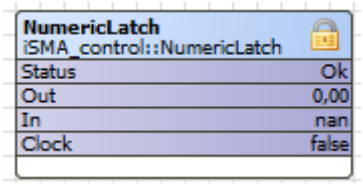


The component has the following slots:

- Clock: This is a 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". This means that a single input property is captured and sent to the output property at the instant that the Clock status changes from a False to a True state and NOT when the property changes from a True to a False state.
- Out : This standard component property provides the actual latched value that is captured from the input property at "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 you link into from a data source. For example, you can link into this property from a control point or a Schedule output.

6.1.3 NumericLatch Component

NumericLatch 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.



The component has the following slots:

- Clock: This is a 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". This means that a single input property is captured and sent to the output property at the instant that the Clock status changes from a False to a True state and NOT when the property changes from a True to a False state.
- Out : This standard component property provides the actual latched value that is captured from the input property at "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 you link into from a data source. For example, you can link into this property from a control point or a Schedule output.

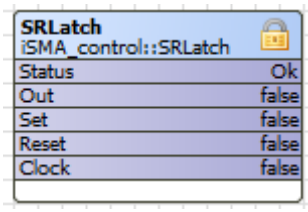
6.1.4 SRLatch Component

Set/Reset Latch — single-bit edge-triggered data storage. The following logic applies on 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.



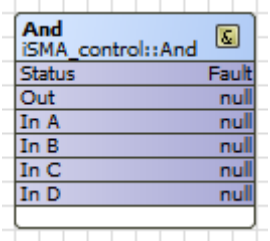
The component has the following slots:

- Out : This standard component property provides the actual latched value that is captured from the input property at "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 you link into from a data source. For example, you can link into this property from a control point or a Schedule output.

7 Logic Components

7.1.1 AND Component

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.



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

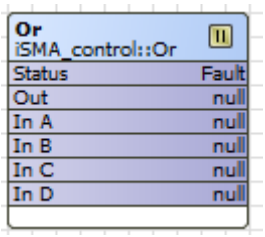
out:= (inA & inB) when operator == 0 (And)
out:= (inA | inB) when operator == 1 (Or)
out := (inA ^ inB) when operator == 2 (Xor)
out := !inA when operator == 3 (Not)

out := !(inA & inB) when operator == 4 (Nand)

out: =! (inA | inB) when operator == 5 (Nor)

7.1.2 OR Component

Or Component performs a logical OR on all valid inputs and writes the boolean result to the out property. The 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.



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

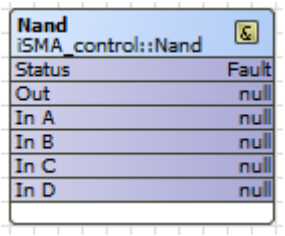
Table 5 Or object truth table (2 inputs)

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)

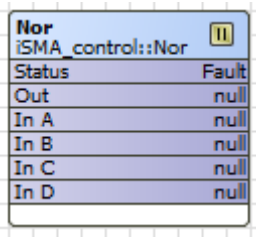
7.1.3 Nand Component

Nand Component performs the operation out is equivalent to false if all inputs are true.



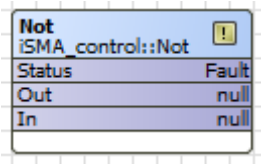
7.1.4 NOR Component

Nor Component performs the operation out is equivalent to true if all inputs are false.



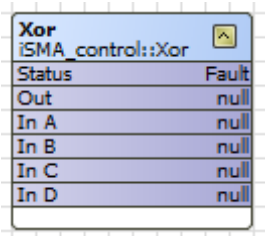
7.1.5 NOT Component

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



7.1.6 XOR Component

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

Comparator

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

equal := (inA == inB)

Note:qual := (inA != inB)

greaterThan	: = (inA > inB)
greaterThanEqual	:= (inA >= inB)
lessThan	: = (inA < inB)
lessThanEqual	:= (inA <= inB)

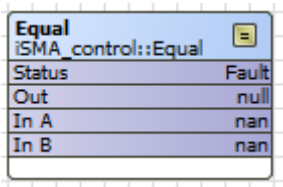
ComparatorExpr

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

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)

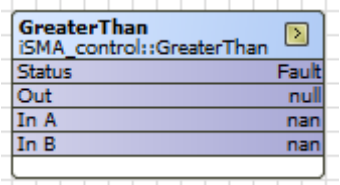
7.1.7 Equal Component

Equal Component performs the operation A == B. Numeric. Nan values are never equal.



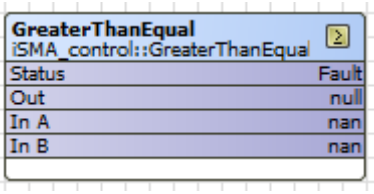
7.1.8 GreaterThan Component

GreaterThan Component performs the operation A > B with a boolean result.



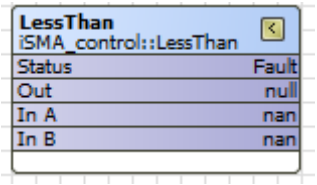
7.1.9 GreaterThanEqual Component

GreaterThanEqual Component performs the operation A >= B with a boolean result.



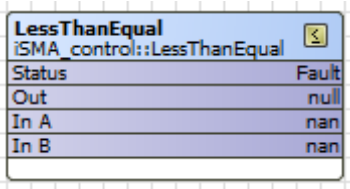
7.1.10 LessThan Component

LessThan Component performs the operation In A < In B with a boolean result.



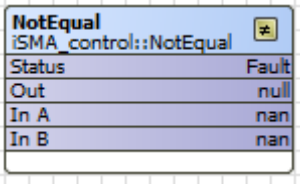
7.1.11 LessThanEqual Component

LessThanEqual Component performs the operation In A <= In B with a boolean result.



7.1.12 NotEqual Component

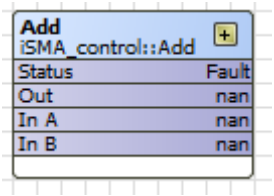
NotEqual Component performs the operation A! = B with a boolean result.



8 Math Components

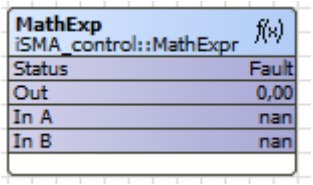
8.1.1 Add Component

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



8.1.2 MathExpr Component

MathExpr Component stands for the Object in which various Mathematical & Trigonometric Operations can be performed on one/two Numeric inputs based on the operator.

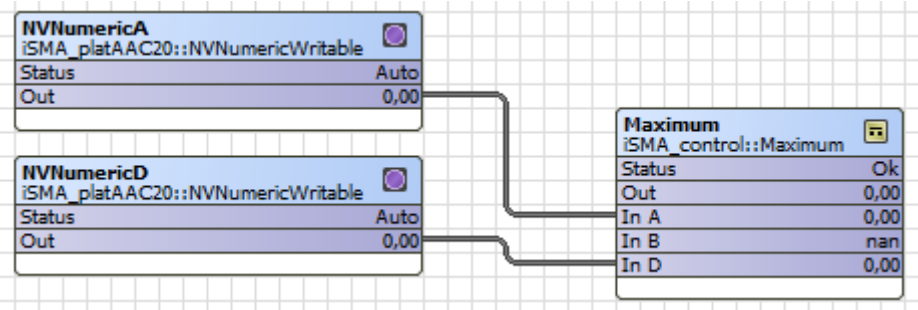


out := fabs (in)	when operator = 0 (AbsValue)
out := inA + inB	when operator = 1 (Add)
out := acos (in)	when operator = 2 (ArcCosine)
out := asin (in)	when operator = 3 (ArcSine)
out := atan (in)	when operator = 4 (ArcTangent)
out := cos (in)	when operator = 5 (Cosine)
out := inA / inB	when operator = 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)

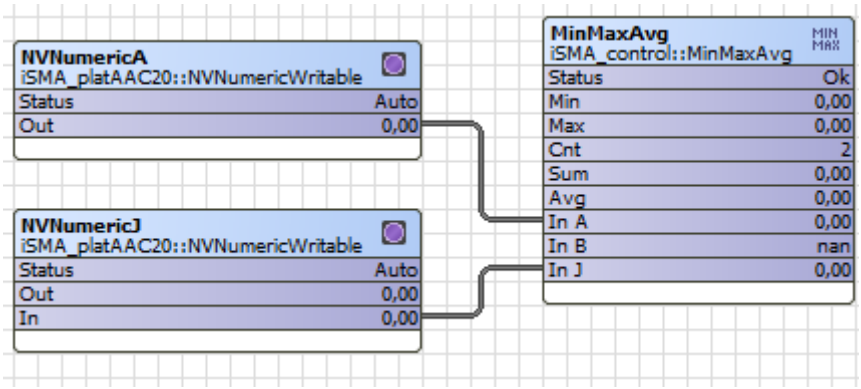
8.1.3 Maximum Component

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



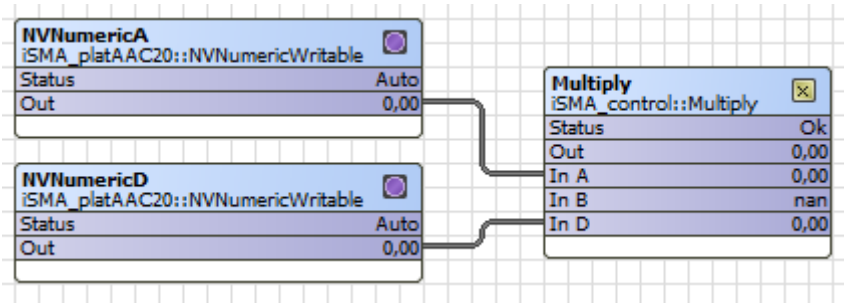
8.1.4 MinMaxAvg Component

MinMaxAvg has 5 Numeric output slots that provide the current minimum, maximum, count, sum and average values of 2 to 10 linked Numeric inputs



8.1.5 Multiply Component

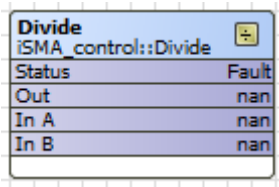
Multiply performs the calculation Out := InA * InB * InC * InD.



The following Math types perform an operation using two inputs:

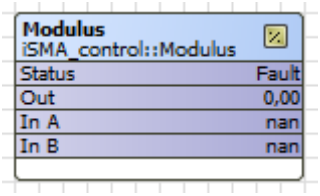
8.1.6 Divide Component

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



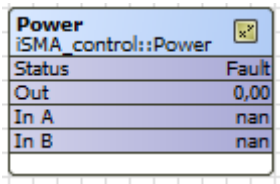
8.1.7 Modulus Component

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



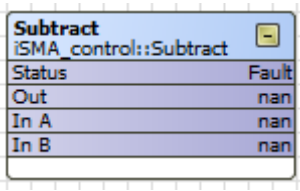
8.1.8 Power Component

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



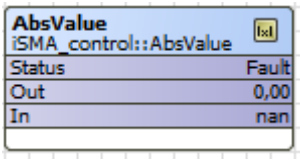
8.1.9 Subtract Component

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



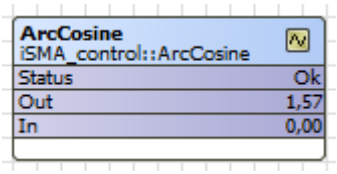
8.1.10 AbsValue Component

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



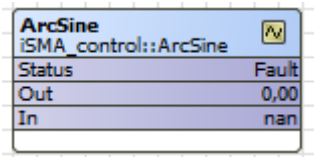
8.1.11 ArcCosine Component

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



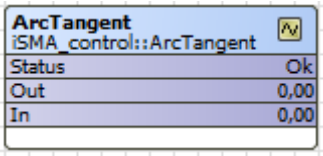
8.1.12 ArcSine Component

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



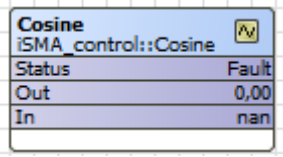
8.1.13 ArcTangent Component

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



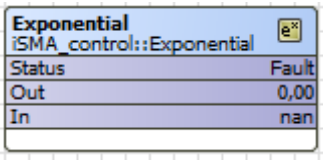
8.1.14 Cosine Component

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



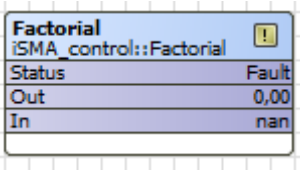
8.1.15 Exponential Component

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



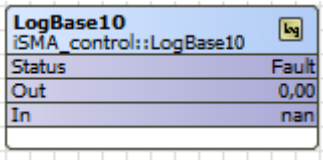
8.1.16 Factorial Component

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.



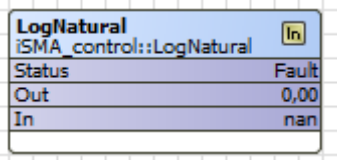
8.1.17 LogBase 10 Component

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



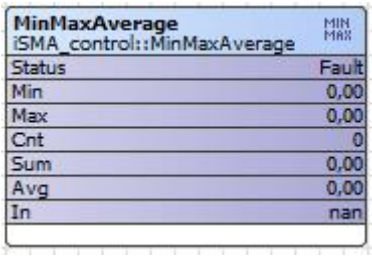
8.1.18 LogNatural Component

LogNatural Component performs the operation out := ln (inA) (log base e of inA)



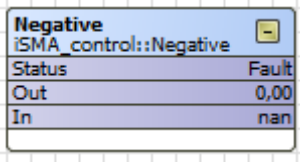
8.1.19 MinMaxAverage Component

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



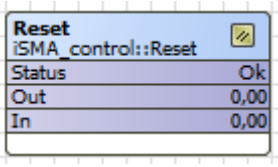
8.1.20 Negative Component

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



8.1.21 Reset Component

This component performs a linear "reset" on the inA value.



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 object 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

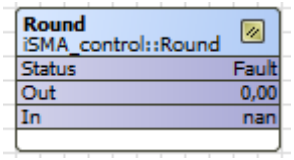
is 200˚F. When the outside air temperature is 75˚F, the hot water Setpoint is 100˚F. The Reset object is configured as:

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.

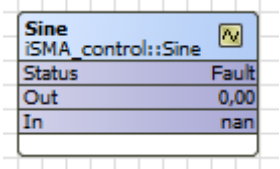
8.1.22 Round Component

Round Component performs the Mathematical operation of returning the nearest Integer, rounding away from zero in the halfway cases.
out := round (in)



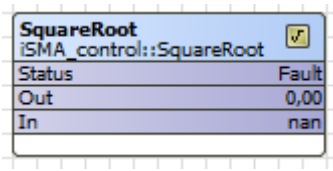
8.1.23 Sine Component

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



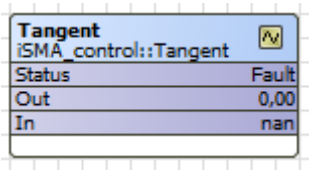
8.1.24 SquareRoot Component

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



8.1.25 Tangent Component

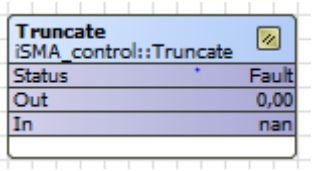
Tangent Component performs the operation out := tan(InA).



8.1.26 Truncate Component

Truncate Component performs the Mathematical operation of returning the nearest Integer, not greater in magnitude than the Input Float.

out := trunc (in)

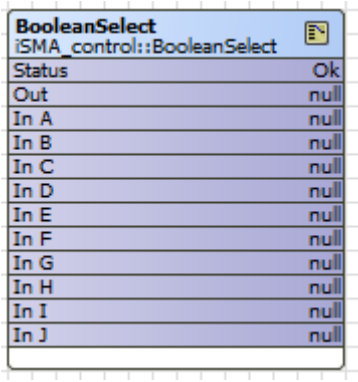


9 Select Components

9.1.1 BooleanSelect Component

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

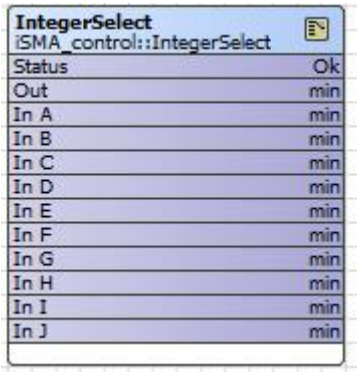
Note: that all select objects require an integer input to perform the selection by the type of input data selected and passed to the "Out" slot.



9.1.2 IntegerSelect Component

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

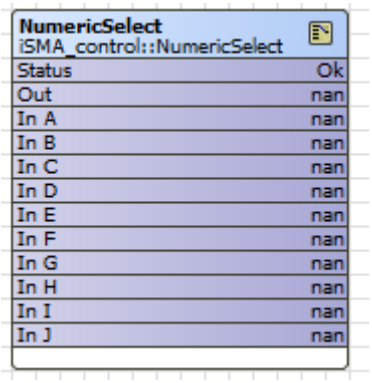
Note: that all select objects require an integer input to perform the selection by the type of input data selected and passed to the "Out" slot.



9.1.3 NumericSelect Component

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

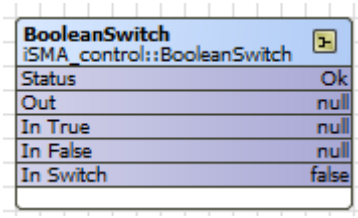
Note: that all select objects require an integer input to perform the selection by the type of input data selected and passed to the "Out" slot.



10 Switch Components

10.1.1 BooleanSwitch Component

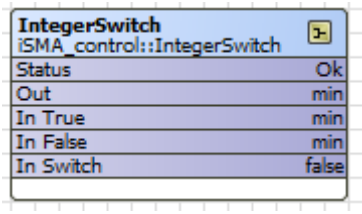
BooleanSwitch Component selects between two Boolean inputs based upon the boolean value at the Boolean input 'In Switch'.



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

10.1.2 IntegerSwitch Component

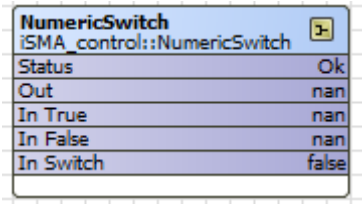
IntegerSwitch Component selects between two Integer inputs based upon the boolean value at the Boolean input 'In Switch'.



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

10.1.3 NumericSwitch Component

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



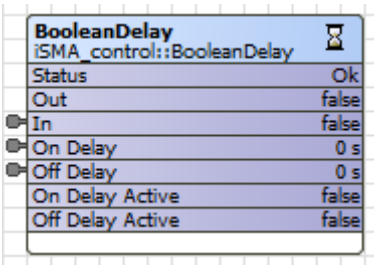
Note: that all select objects require an boolean input to perform the selection by the type of input data selected and passed to the "Out" slot.

11 Timer Components

11.1.1 BooleanDelay Component

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 labeled "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.



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

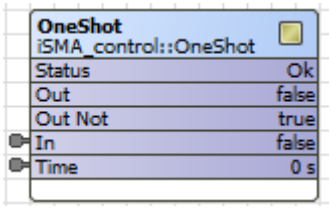
The component has the following slots:

- In : Typically, you set this property by linking a boolean out value into it. You can manually configure the default state to be true or false, so that when no 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 you to set the amount of time (in hours, minutes, and seconds) that you want to 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 is proceeded without delay.
- Off Delay : This property allows you to set the amount of time (in hours, minutes, and seconds) that you want to 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 is proceeded 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, false options available. These values are set at the end of any On Delay or Off Delay to reflect the In property value.

11.1.2 OneShot Component

OneShot Component provides a single, temporary, boolean output for a specified duration (as set in the Time property). A 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.



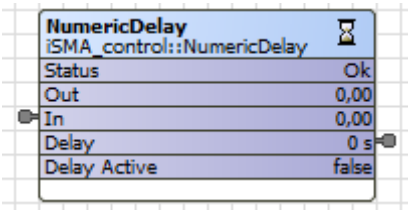
***Slots with the dots are hidden by default, when we they're connected to, they become uncovered and are being shown in Wire Sheet

The component has the following slots:

- In: Typically, you set this property by linking a boolean Out value into it. You can manually configure the default state to a boolean value, so that when no 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 value displays the current value that changes with a False to True transition at the In property value or a "Fire" action. After a 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 a OneShot is triggered and the Time value period expires, this value returns to the default (True) value.

11.1.3 NumericDelay Component

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.



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

The component has the following slots:

- In: Typically, you set this property by linking a numeric out value into it. You can manually configure the default state to be true or false, so that when no 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 you to set the amount of time (in hours, minutes, and seconds) that you want to expire before sending the in value to the Out property. 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 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.

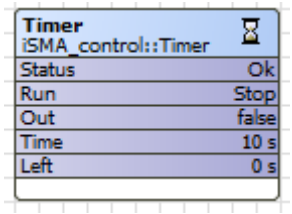
11.1.4 Timer Component

Timer outputs a pulse for the configured amount of time "in" is used to fire the timer:

if low, out is forced to false

if high, out = 1 until timer reaches "time" seconds

Alternatively, the pulse can be fired from the "Start Timer" action if in is not linked



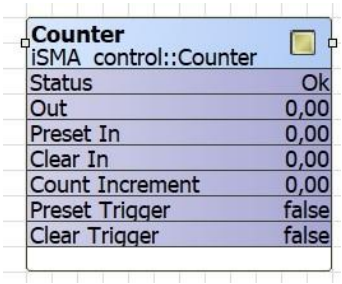
The component has the following slots:

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

12 Util Components

12.1.1 Counter Component

Counter Component will count boolean inactive to active transitions. It supports counting up, counting down, presenting, and clearing.



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

The component has the following slots:

- Count Up: This is a Boolean input. When this input makes 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 inactive to active transition the value of the Out property will be decremented by the Count Increment.
- Preset In: This is a Numeric input which will be set in the Out property when the Preset action is invoked.

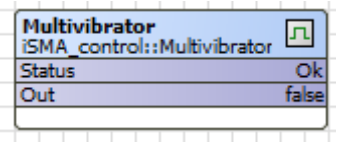
- Clear In: This is a Numeric input which will be set in the Out property when the Clear action is invoked.
- Count Increment: This is the value that the Out property will change for a single count up or count down active transition.
- Preset Trigger: This is a Boolean input. When this input makes inactive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

The Counter component includes the following actions:

- Preset: Preset action when invoked, the value of the Out property will be set to preset in value.
- Clear: Clear action when invoked, the value of the Out property will be set to clear in value.

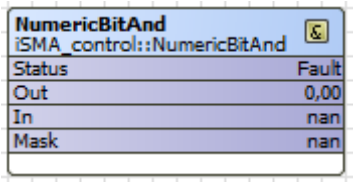
12.1.2 MultiVibrator Component

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



12.1.3 NumericBitAnd Component

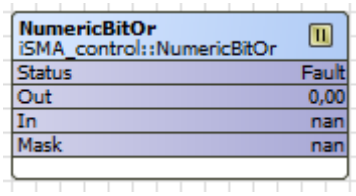
NumericBitAnd Component performs a logical AND on the bit equivalent of the Numeric "In" value against the bit equivalent of its Numeric "Mask" slot value. It may be useful in cases where boolean information is mapped into integer values.



12.1.4 NumericBitOr Component

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 boolean information is mapped into integer values.

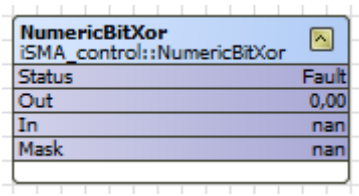
As an example, some manufacturers multiplex binary data into a single numerical point by converting the bits from hexadecimal to decimal format. To obtain the status of the individual binary data, the number must be converted back from decimal to hex format. Each digit of the hex number represents a particular binary parameters state (0 = false, 1 = true). The NumericBitOr object 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.



12.1.5 NumericBitXor Component

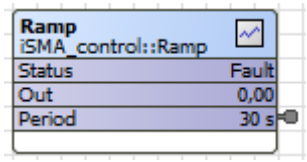
NumericBitXor Component performs a logical XOR on the bit equivalent of the Numeric "In" value against the bit equivalent of its Numeric "Mask" slot value. It may be useful in cases where boolean information is mapped into integer values.

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 object converts a Numeric input to hex value and compares it against the mask value. Each digit is analyzed using exclusive OR (XOR) logic, setting the corresponding digit value to either a 1 or 0.



12.1.6 Ramp Component

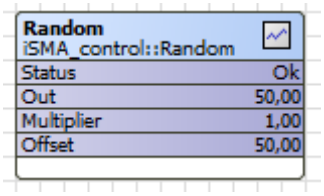
Ramp Component provides a Numeric Out with a linear ramping output. Slots define the Period, Amplitude and Offset.



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

12.1.7 Random Component

This 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 an offset.

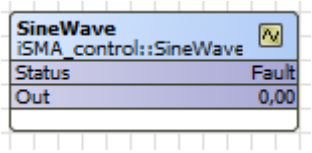


The component has the following slots:

- 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.

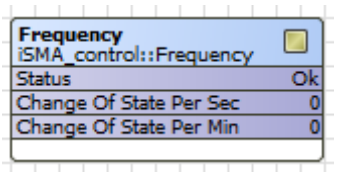
12.1.8 SineWave Component

SineWave Component generates a sine wave as a Numeric out.



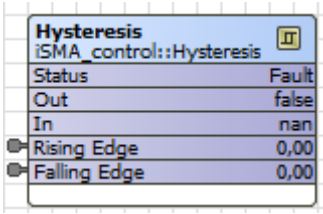
12.1.9 Frequency Component

Frequency Component object calculates a pulse input frequency.



12.1.10 Hysteresis Component

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



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

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

If risingEdge > fallingEdge, then out behaves "normally", ie

out := true when in rises above risingEdge

out := false when in falls below fallingEdge

If risingEdge < fallingEdge, then out behaves "inverted", ie

out := false when in rises above fallingEdge

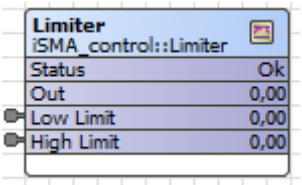
out := true when in falls below risingEdge

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

out := true when in > Rising Edge

12.1.11 Limiter Component

Limiter Component object restricts the output based on the input between lowLimit and highLimit.



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

HighLimit and LowLimit are configurable floats :

out := highLimit when in > highLimit

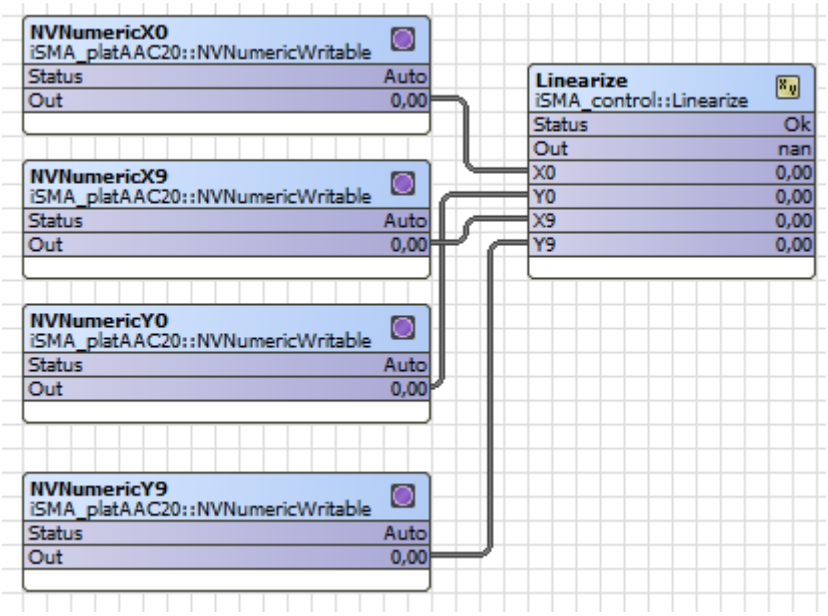
out := lowLimit when in < lowLimit

out := in when lowLimit < in < highLimit

12.1.12 Linearize Component

Linearize — piecewise linearization of a float.

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.



For input values between these points, the component will estimate the output based upon the linear equation:

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 an x value of input there should be a corresponding y value of 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: that slope may be positive or negative, and is indicated by comparison of x1 and x0.

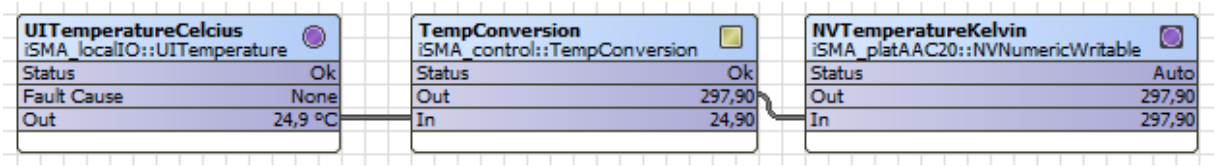
Positive if x1 > x0

Negative if x1 < x0

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

12.1.13 TempConversion Component

TempConversion is a Converter object to convert Temperature from one unit to another.



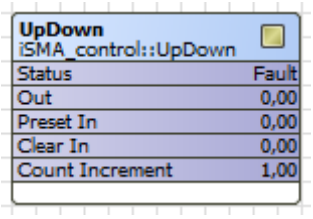

```
out := in                                  when in = Celsius & out = celsius
out := (in - 32.0) * (5.0/9.0)             when in = celsius & out = fahrenheit
out := in + 273.0                          when in = celsius & out = kelvin

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
```

12.1.14 UpDown Component

UpDown component will count based on the countIncrement property. It supports counting up, counting down, presetting, and clearing.



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