Focus Config Block Reference

INNOTECH®

Proprietary

Trademark

Disclaimer

Version History	Version	Date
First Edition	1.0	14/8/2018
Various updates from feedback and added the Template manger section	1.1	28/8/2018
Added new UIO and internal blocks.	1.2	30/9/2020

Contents

Focus project capabilities

The following items need to be taken into consideration when designing your project.

A Focus Project supports up to 50 Devices in the Network Tree. Support for the following per Device Configuration blocks:

- **1,500 Blocks total per configuration**
- 1,500 BACnet Objects (Native and Protocol Watches) *
- 200 BACnet Comms Inputs *
- 200 BACnet Comms Outputs *
- 200 BACnet COV Registrations *
- 250 Modbus Comms Blocks *
- 100 Innotech Global Outputs
- 100 Calendars (each with up to 100 events, maximum of 2,000 events per Device)
- 256 Alarm Blocks
- 250 Log Blocks
- 100 Universal Curve Blocks
- 50 Recipe Blocks
- 50 Lookup Blocks
- No limit on Innotech Global Inputs (up to 1,500 Total Blocks configuration limit)
- No limit on Logic blocks (up to 1,500 Total Blocks configuration limit)

* Note - Not available when programming with Focus Lite.

General notes

1. Items and descriptions used in this reference have been designed from Focus version 1.14. As block functions change or new blocks are added, this will be updated and reflected in this document as soon as possible.
2. It is recommended to view the in-built help application installed with Focus upon a new release to check for any new blocks or operation. Always check the release notes included with each new software release for the latest added features and known issues.
3. All examples shown within this guide are intended for reference only to assist in explaining the function or operation of the relevant block operation.
4. For all support related questions for products or software, please refer to the Contact section at the end of this manual.

Blocks Overview and links

To learn more about a block, click on the descriptor text to navigate to the relevant page.

Innotech Logic

Added Blocks (new releases)

Input and Output Blocks

UIO Template Manager information


Current Loop Input


Current Loop Output


Current/Voltage Input


Digital Input Contact


Digital Input Voltage


Pressure Input


Digital Output


Duty Cycle Output


Pulse Input Contact


Pulse Input Voltage


Pulse Output


Sensor Input


Voltage Input


Voltage Output


TRIAC Digital Output


TRIAC Pulse Output

Communications Blocks


BACnet Comms Input


BACnet comms Output


Global Input


Global Output


ISS Comms Input


ISS Comms Output


Modbus Comms Input


Modbus Coms Output

BACnet Objects

Note that the BACnet Object blocks are not available when using Focus Lite.

Analog Input

Analog Output

Analog Value

Binary Input

Binary Output

Binary Value

Calendar


Large Analog Value


Loop


Multistate Input


Multistate Output


Multistate Value


Trend


User Variable

Note that the block descriptors on the following pages are in alphabetical order and may not follow the sequence presented in the tables above. Use the links in the document to easily navigate.

ACCUMULATOR

The Accumulator block accumulates either a value over time or accumulates just time, depending on the internal settings of the block.

NODE DESCRIPTORS

Enable / Value - Analog Input: The value used as the input to the block depends on the setting for the internal block property accumulation type.

Reset - Digital Input: Pulse used to reset the value accumulated by the block back to zero.

Accumulated Value - Point which is either the time accumulated since the last reset of the block, the total accumulated value or the total number of pulses from a Pulse Input.

INTERNAL BLOCK PROPERTIES

Accumulation Type: Select the appropriate accumulation type.

- Pulses
- Time
- Value

Input Value Rate: [*Value accumulation type only*] Specify the rate to which your input refers. Choose from one of the following.

- Seconds
- Minutes
- Hours

Time Units: [*Time accumulation type only*] Specify the rate to which your input refers. Choose one of the following.

- Seconds
- Minutes
- Hours

Initial Value [*Value accumulation type only*] This is the starting accumulated time or value which will be used at the output of the block. For example, you may be using the block to log the hours run on a compressor, but when sending the configuration, you may need to specify that the compressor has already accumulated a run time of 150 hours.

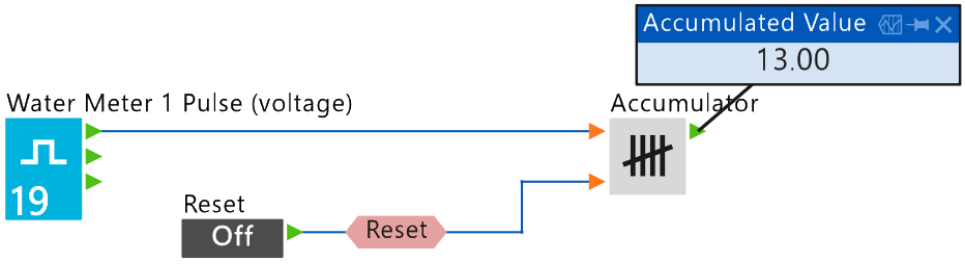
Initial Accumulated Pulses: [*Pulse Accumulation Type Only*] Specify the initial pulses value.

Initial Value: [*Time Accumulation Type Only*] Specify the initial time value.

CONFIG EXAMPLES

Accumulator Pulses

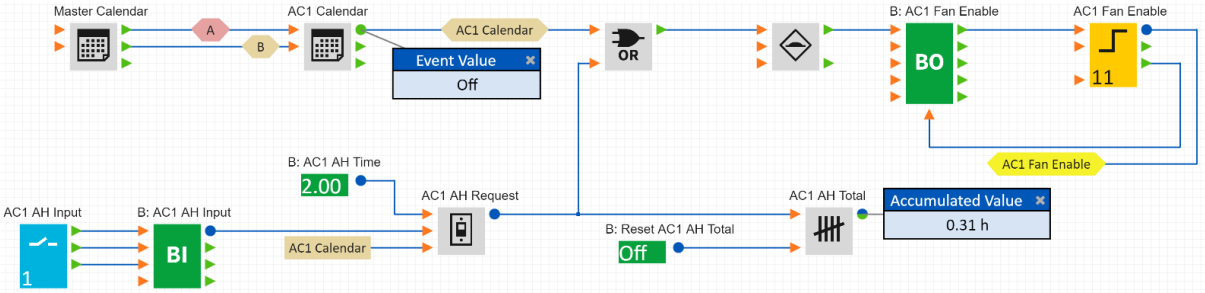
When the accumulation type is set to pulses, the block only accumulates when an input pulse is received via a **Pulse Input Block**. If a reset pulse is received, the count is reset. The block will not accumulate while the reset node is active.



Accumulator Time (seconds, minutes or hours)

The accumulator continually counts up while an input is received. This counts depending if the time is set to seconds, minutes or hours. If a reset pulse is received, the count is reset. If the reset node is active, the accumulation will not count.

This accumulation type can be used to log hours run or after hours' time for equipment such as a AHU after hours run totals.

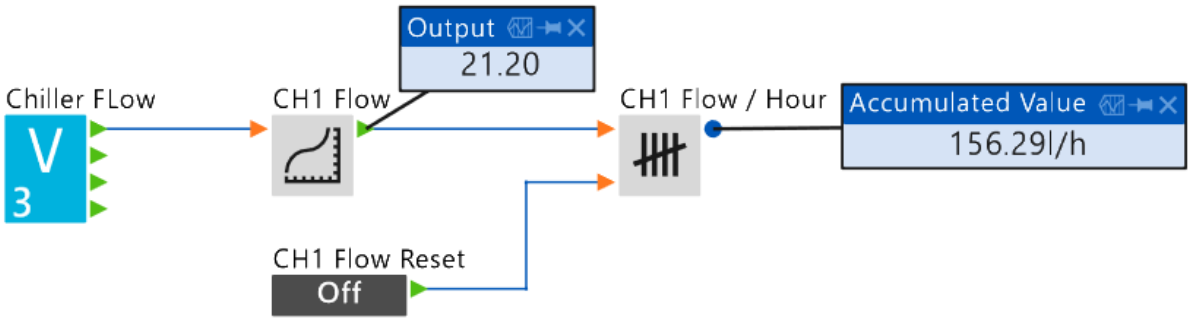


Note that a watch on the Latch output must be added and configured correctly for the accumulator to provide the hours reading in the correct format.

Accumulator Value (per second, minute or hour)

The accumulator can be used to monitor the total flow rate of an input, such as a flow meter. When the input provides a reading of zero, the accumulator will stop accumulating at that point providing the current total value.

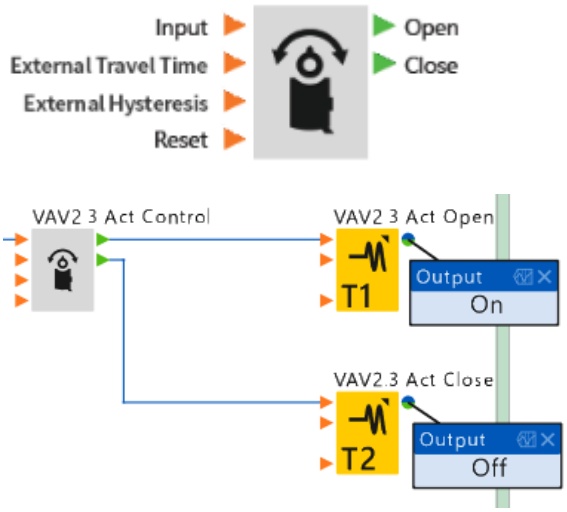
Once the Input has a value greater than zero, then the accumulator will recommence its count per hour. The accumulator can be reset via the reset User Variable which has been configured as a pulse ON/OFF digital.



Note that if the input value becomes negative, the accumulated value can count backwards until the value goes positive, which will cause the total to begin counting up again at that point.

[Back to block tables page]

ACTUATOR CONTROL



The Actuator block converts a position input signal into the travel time required by a Drive Open / Drive Close-style Actuator to reach the desired position.

NODE DESCRIPTORS

Inputs

Input: An analogue value representing actuator position from 0 to 100% where 0%=CLOSE and 100%=OPEN. A value of 0% does permanently drive the CLOSE output, where a value of 100% does permanently drive the OPEN output.

External Travel Time: The external setting of the time (seconds) required for the actuator to travel 90°. If left unconnected the internal value will be used.

External Hysteresis: The external setting of the amount the Input value has to change before the OPEN or CLOSE outputs are be activated. If left unconnected the internal value will be used.

Reset: An external digital value that re-references the position fully closed to be 0%. If a pulse is applied to Reset, the Close output is active for the amount specified as Travel Time. If Reset is forced to On (a value greater than 0) the Close output will remain on indefinitely.

Outputs

Open: A digital value which is ON if the Actuator position needs to be moved in the OPEN direction.

Close: A digital value which is ON if the Actuator position needs to be moved in the CLOSED direction.

INTERNAL PROPERTIES

Hysteresis Percentage: An adjustable Hysteresis is implemented to avoid nuisance activations of the actuator and therefore increase its life expectancy.

Travel Time Interval (sec): Enter the travel time interval in seconds.

Reset Position: Manual or scheduled calibration can be performed via the Reset input. Select from Open or Close.

Notes

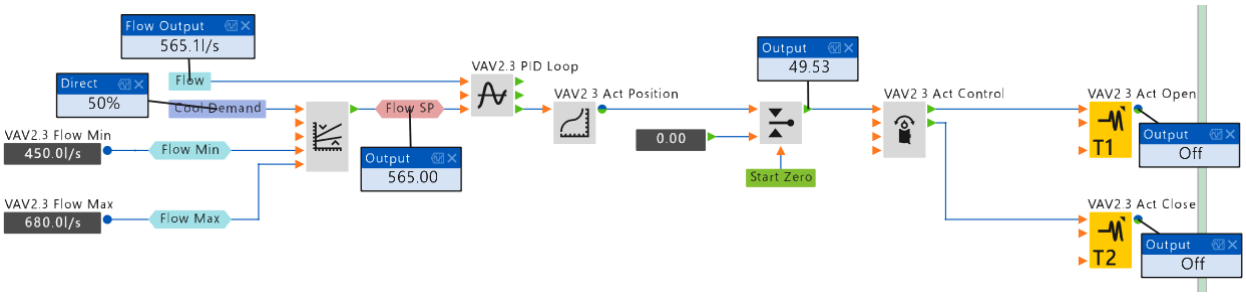
The Hysteresis is symmetrically applied to the current actuator position.

Example: With Position at 50% and Hysteresis at 2%. The thresholds are 52% and 48%.

Every time the Hysteresis is exceeded the actuator position will change for the amount specified by the INPUT but no less than the amount specified as Hysteresis. After the new position has been reached, the Hysteresis is reapplied to that new position.

This block is normally used in conjunction with the linear scale and PID to determine the required actuator positon depending on demand.

Example



In this example, the calculated flow is input into the PID loop. The linear scale determines the flow setpoint required based on the min and max values set and the current cooling demand for this example.

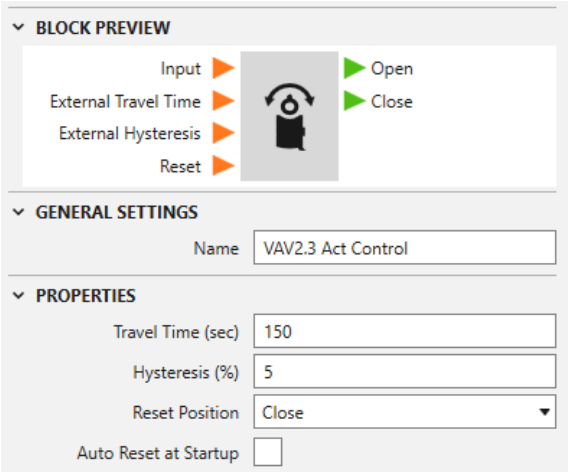
The PID in conjunction with the Unicurve provides the actuator block with the signal to either drive open or closed to maintain the desired flow.

The Unicurve has the internal properties set as follows,

Inputs	Outputs	
-100	0	✕
100	100	✕

The switch block is used to provide a reset or calibration for the actuator block back to a zero or closed state.

The internal properties for the actuator block for this project are set as follows,



[Back to block tables page]

ALARM

The Alarm block allows internal alarms to be generated via the config logic. The status of configured Alarms can be viewed using the integrated Omni webpage. Active alarms also appear on the Omni HMI display (if installed).

256 Alarm blocks can be added to an Omni config in Focus per device.

NODE DESCRIPTORS

Trigger - Digital Input: Point input which is used to trigger the alarm.

If the internal block property Latch is enabled, then the Trigger input only has to be an ON pulse to turn the alarm on and keep it on until it is reset. If the internal block property Latch is disabled, then the alarm is only ON as long as the Trigger input is ON.

Enable - Digital Input: Point input which enables or disables the detection of subsequent alarm trigger events. If nothing is connected to the Enable node, the Alarm will be still output if a Trigger signal is received. If an object is connected to the Enable node, the Alarm will only trigger if the Enable node receives an ON signal input.

Reset - Digital Input: Input point which resets the state of any alarm which has been Latched. It may be a single pulse, or you can keep it ON to effectively disable the alarm block temporarily.

Digital Output: Indicates if the alarm condition is ON or OFF.

INTERNAL PROPERTIES

Delay to ON: This field specifies the delay for triggering the Output to an ON state for an activated alarm. The default is 0 seconds (or immediately).

Delay Units: This field specifies the units (Hours, Minutes or Seconds) that relate to the Delay to On field.

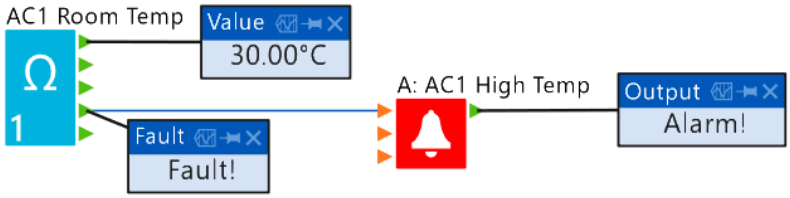
Latched: If this option is enabled, then the alarm only requires a pulse (or a steady ON signal) on the Trigger input to set the alarm and keep it set until it is reset, otherwise the alarm will only be ON as long as the Trigger input is ON.

Display on HMI: The alarm message will be displayed for 5 seconds at a time on the Digital Controller's main display if other alarms are active, or constantly if no other alarms are active.

EXAMPLES

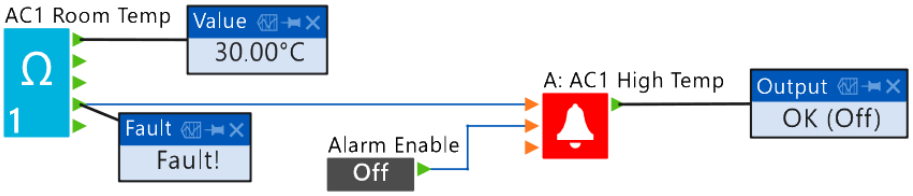
Alarm Trigger

When the alarm block receives a trigger, an alarm is activated and shown at the output.

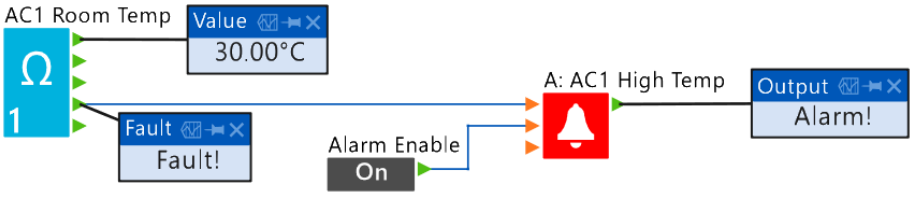


Alarm Enable

When the alarm block enable node is set to **OFF**, the alarm notification is **disabled**. The output will not trigger an alarm event if the trigger input is on.

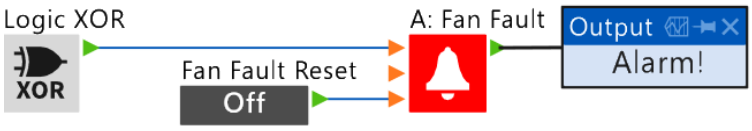


When the alarm block enable node is set to **ON**, alarm notifications are **enabled**. The output will trigger an alarm if the trigger node is active. If the enable node is not used, then the alarm will trigger as normal.



Alarm Reset

When the alarm block receives a reset pulse, any alarm present is cleared. **This is used when the alarm is set to latch**. Note that the **trigger state must be off** for the alarm to clear otherwise the alarm will re instate after the reset.



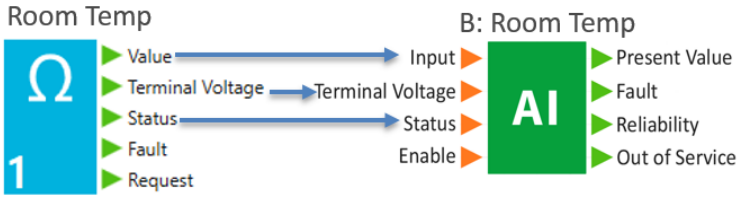
Note that the Reset should be a pulse type to de-latch the Alarm block set to Latch mode.

[Back to block tables page]

ANALOG INPUT

The Analog Input block defines a standardised BACnet object whose properties represent the externally visible characteristics of an analog input for any BACnet client.

This BACnet object is typically used in conjunction with the controller Input block, such as a sensor, as shown below.



This passes the sensor value to the AI input node which is then passed to the AI Present Value node. The terminal voltage and Status nodes that are fed into the AI block and used to generate the fault and reliability of the sensor.

The BACnet client can then monitor the AI output nodes as required.

NODE DESCRIPTORS.

Present Value - The present value node provides the value that has been written to the block at the highest priority, either from the blocks input or a BACnet client. In the block properties, you will notice a relinquish default value. This is the value that will be present on the Present Value node if all the 16 priorities have been relinquished or released, until an input value is detected.

Enable Node - The enable node enables or disables the block as per Innotech block functions. When the enable is On, the block input value is passed through to the Present Value. If the Enable is Off, the input is not passed through and the last value is held by the block, until the relinquish node is triggered to the default value set in the blocks properties.

Note that the enable has no effect if the out of service node has been enabled and activated from a BACnet client.

Out of Service - BACnet clients write directly to the Out of Service property to disable local control of the object, therefore disconnecting the Input from the Present Value and only allowing control via the BACnet network. *(The block Input node is ignored whenever the Out of Service output is TRUE)*

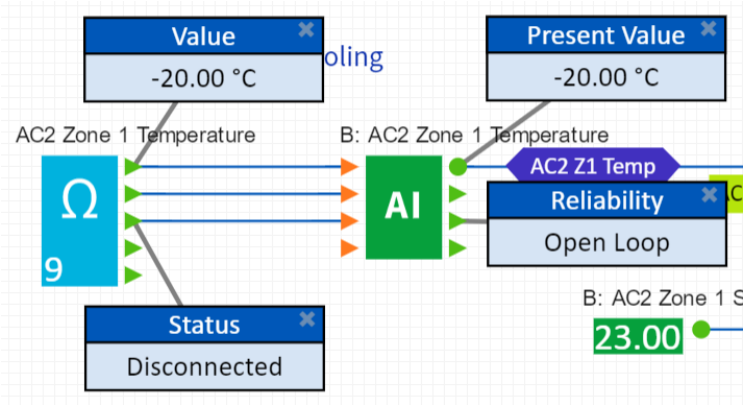
If the internal Allow Override property is not enabled (FALSE), the Out of Service property cannot be written to, thus preventing the external client taking control of this block.

Reliability - The reliability output specifies if the BACnet object is in an ok or fault state. The value on the fault node advises if a fault exists, a value of 1, or not in fault with a value of zero (0).

Valid Reliability Values	
RELIABILITY_NO_FAULT_DETECTED	= 0
RELIABILITY_OVER_RANGE	= 2
RELIABILITY_UNDER_RANGE	= 3
RELIABILITY_OPEN_LOOP	= 4
RELIABILITY_SHORTED_LOOP	= 5

Example reliability value

In this example, the sensor is open circuit. The value, terminal voltage and status are used to generate the BACnet output node fault and reliability states.



INTERNAL BLOCK PROPERTIES

Object Instance - Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Device Type: A text description of the device connected to the Input.

Units: The Units property of the object allows you to select one of the standard BACnet units.

Minimum Present Value: The Minimum Present Value sets the lower end of the desired Present Value output for alarm monitoring.

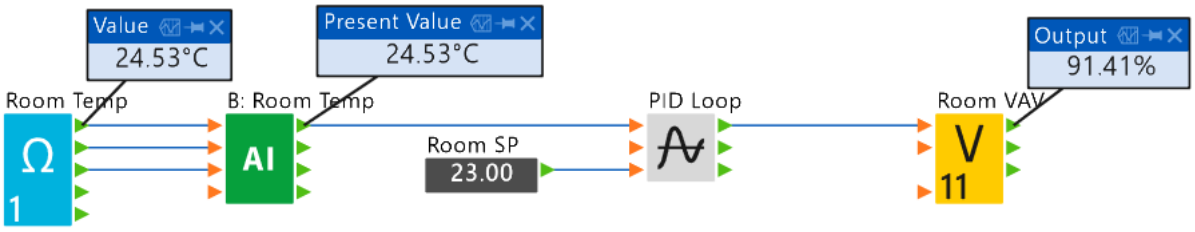
Maximum Present Value: The Maximum Present Value sets the upper end of the desired Present Value output for alarm monitoring.

Change Of Value Increment: The Change of Value (COV) setting allows you to optimise the amount of BACnet traffic to and from your BACnet device.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

EXAMPLE APPLICATION

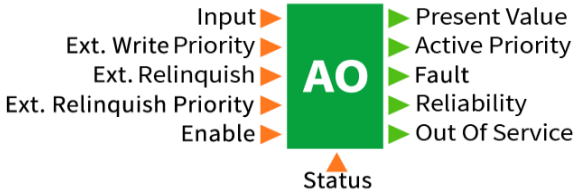
In the example below, the Omni sensor is connected to the Analog Input (AI) which passes through the value and status of the sensor to the AI. The AI is then available globally across the local BACnet network to be monitored by any BACnet client via the AI Present Value.



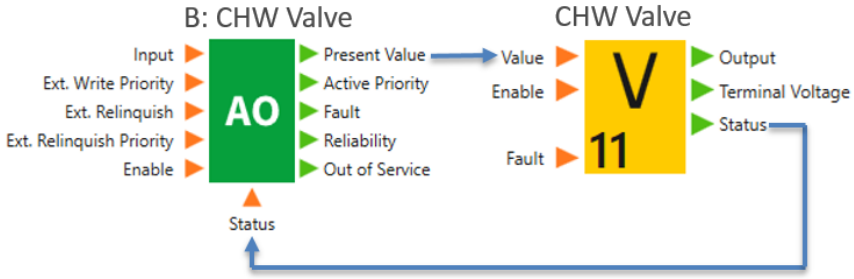
[Back to block tables page]

ANALOG OUTPUT

The Analog Output block defines a standardised BACnet object whose properties represent the externally visible characteristics of an Analog Output for any BACnet client.



This BACnet object is typically used in conjunction with the controller Output, such as a Voltage output, as shown below.



The AO Input node value is passed through to the Present Value of the AO block which in turn is fed into the Voltage Output Value node to be presented on this blocks Output node and corresponding UIO Terminal, in this case 11.

NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active, and the Input value is written to the object at the highest Write Priority.

Ext. Write Priority: This determines the position in the Priority list (Known as the Priority Array) that the input value will be written to. (This equates to a value from 1-16, where 1 is the highest priority) The Input value will be written at the Ext. Write Priority value on a change in value of the Input, or a change in value of the Ext. Write Priority input.

Ext. Relinquish: The Ext. Relinquish node is a digital signal applied to delete a value that has been written to the object at a particular priority. When using the Ext. Relinquish node, you must also indicate the Priority of the value to be relinquished. This is achieved by setting a value from 1-16 on the Ext. Relinquish Priority node. Therefore, you should set the Ext. Relinquish Priority first, and then pulse the digital signal on the Ext. Relinquish node from “Off” to “On”.

Ext. Relinquish Priority - The Ext. Relinquish Priority input allows a user to select any priority (and value) to be relinquished from the Priority Array.

Enable: The Enable input, **when used**, of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Present Value: The Present Value displays the value that was written at the highest Write Priority, either from the input or from a BACnet client. If all priorities have been relinquished, the Relinquish Default internal property will become the Present Value.

Active Priority: The Active Priority indicates the highest priority that was written to the object. (1-16) If all priorities have been relinquished, the Active Priority will be 0 until the next input write and then default to the priority set in the blocks properties.

Fault: The Fault output will be FALSE whenever the Reliability output has a value of NO_FAULT_DETECTED, otherwise it will be TRUE.

Reliability - The reliability output specifies if the BACnet object is in an ok or fault state. The value on the fault node advises if a fault exists, a value of 1, or not in fault with a value of zero (0).

Out Of Service: The value of this property indicates whether or not the object is currently being overridden by a BACnet Client.

INTERNAL BLOCK PROPERTIES

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Device Type: A text description of the device connected to the Input.

Units: The Units property of the object allows you to select one of the standard BACnet units.

Minimum Present Value: The Minimum Present Value sets the lower end of the desired Present Value output for alarm monitoring.

Maximum Present Value: The Maximum Present Value sets the upper end of the desired Present Value output for alarm monitoring.

Relinquish Default: The Relinquish Default setting is the default value the Present Value will display when all Write Priorities are relinquished.

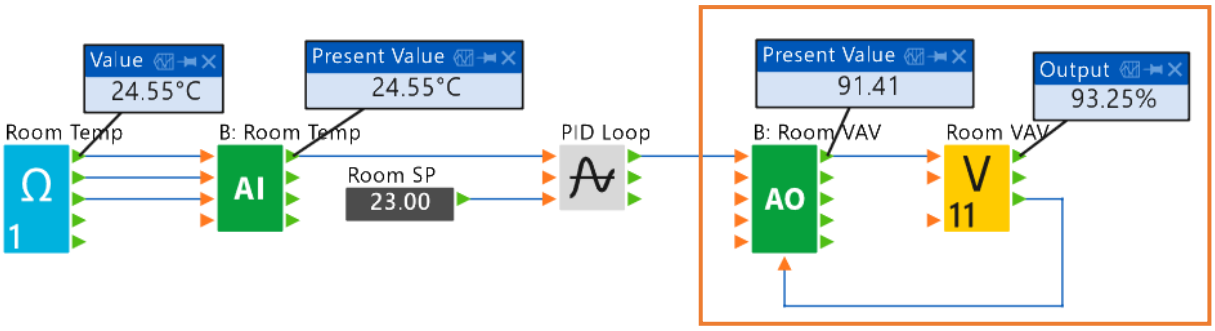
Change Of Value Increment: The Change of Value (COV) setting allows you to optimise the amount of BACnet traffic to and from your BACnet device.

Write Priority: The Write Priority internal property indicates the priority at which the Input value will be written. This property is only used when there is no connection to the Ext. Write Priority input.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

EXAMPLE APPLICATION

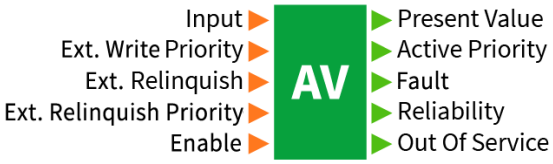
In the example below, the Analog Output (AO) is in line with the PID signal which is then fed into the Voltage output. The AO Present Value is globally available to any BACnet client on the BACnet network. The Status is monitored and if a fault is detected, this is connected to the AO block and is reported via the reliability value.



Back to block tables page

ANALOG VALUE

The Analog Value Block defines a standardised BACnet object whose properties represent the externally visible characteristics of an analog value to any BACnet client.



The AV can be used to provide an analog value to a BACnet client. A value fed in to the Input node is presented on the blocks Present Value.

Analog Value blocks are more suited to items such as setpoints or monitoring rather than control functions within the config.

NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active, and the Input value is written to the object at the highest Write Priority.

Ext. Write Priority: This determines the position in the Priority list (Known as the Priority Array) that the input value will be written to. (This equates to a value from 1-16, where 1 is the highest priority) The Input value will be written at the Ext. Write Priority value on a change in value of the Input, or a change in value of the Ext. Write Priority input.

Ext. Relinquish: The Ext. Relinquish node is a digital signal applied to delete a value that has been written to the object at a particular priority. When using the Ext. Relinquish node, you must also indicate the Priority of the value to be relinquished. This is achieved by setting a value from 1-16 on the Ext. Relinquish Priority node. Therefore, you should set the Ext. Relinquish Priority first, and then pulse the digital signal on the Ext. Relinquish node from “Off” to “On”.

Ext. Relinquish Priority - The Ext. Relinquish Priority input allows a user to select any priority (and value) to be relinquished from the Priority Array.

Enable: The Enable input, **when used**, of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Present Value: The Present Value displays the value that was written at the highest Write Priority, either from the input or from a BACnet client. If all priorities have been relinquished, the Relinquish Default internal property will become the Present Value.

Active Priority: The Active Priority indicates the highest priority that was written to the object. (1-16) If all priorities have been relinquished, the Active Priority will be 0 until the next input write and then default to the priority set in the blocks properties.

Fault: The Fault output will be FALSE whenever the Reliability output has a value of NO_FAULT_DETECTED, otherwise it will be TRUE.

Reliability - The reliability output specifies if the BACnet object is in an ok or fault state. The value on the fault node advises if a fault exists, a value of 1, or not in fault with a value of zero (0).

Out Of Service: The value of this property indicates whether or not the object is currently being overridden by a BACnet Client.

INTERNAL BLOCK PROPERTIES

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Device Type: A text description of the device connected to the Input.

Units: The Units property of the object allows you to select one of the standard BACnet units.

Minimum Present Value: The Minimum Present Value sets the lower end of the desired Present Value output for alarm monitoring.

Maximum Present Value: The Maximum Present Value sets the upper end of the desired Present Value output for alarm monitoring.

Relinquish Default: The Relinquish Default setting is the default value the Present Value will display when all Write Priorities are relinquished.

Change Of Value Increment: The Change of Value (COV) setting allows you to optimise the amount of BACnet traffic to and from your BACnet device.

Write Priority: The Write Priority internal property indicates the priority at which the Input value will be written. This property is only used when there is no connection to the Ext. Write Priority input.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

Back to block tables page

ANALYSE

The Analyse block provides analysis of its input over a time period.

NODE DESCRIPTORS

Input - Analog or Digital Input: This is the input from which the calculations are made. It can be Analog or Digital; the block will calculate its values based on the setting of the block property Analysis Method.

Enable - Digital Input: Point input which enables or disables the block operation.

If this input is not connected, then the block is enabled by default. If the block becomes disabled, accumulation of the input and time are temporarily halted.

Reset - Digital Input: Point input which resets the values of all outputs to zero, and accumulation starts from zero.

Ext. Interval (External Interval) - Analog Input: Point input which can be used to override the internal block property Analyse Interval.

Maximum - Analog Output: Point value with either the maximum analog value input over the last time period specified, or the total number of ON inputs (a transition from OFF to ON) over the time period specified.

Minimum - Analog Output: Point value with either the minimum analog value input over the last time period specified, or the total number of OFF inputs (a transition from ON to OFF) over the time period specified.

Average - Analog Output: Point value with either the average analog value input over the last time period specified, or the average number of ON inputs over the time period specified.

INTERNAL BLOCK PROPERTIES

Interval

Time: The output calculations are based on the last time units (hours, minutes or seconds).

This can be a decimal number and the Units of this interval are set in the Unit field. For example, if this field was 2 and the next field was minutes, the Analyse block will analyse all of the input values over the last 2 minutes.

Units: This sets the Time Interval units in hours, minutes or seconds.

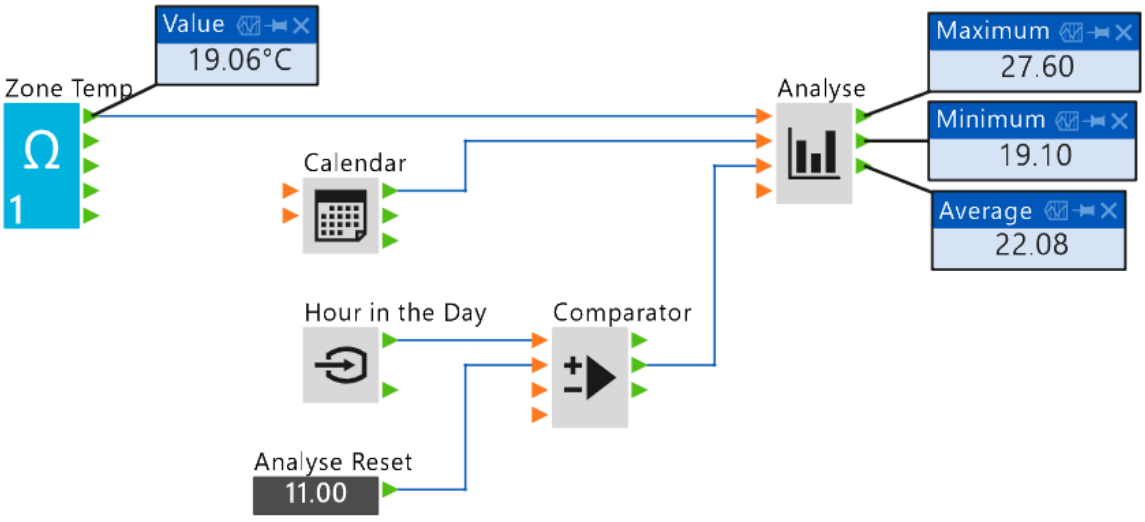
Analyse Type: This is the method used when analysing the results. Select from the following:

Analog: The block will produce the maximum, minimum and average of the input value over the last seconds, minutes or hours (set in the internal settings of the block).

Digital: The block will produce the total number of ON inputs (a transition from OFF to ON), the number of OFF inputs (a transition from ON to OFF) and the average number of ON inputs over the last seconds, minutes or hours (set in the internal settings of the block).

The Maximum value is based on the total number of rising edges in the time period.
The Minimum value is based on the total number of falling edges in the time period.
The Average value is the total duration in the On state in the time period.

EXAMPLE



When the input receives a value, the analyse block determines the maximum minimum and average of the recorded values. The blocks units and time affect the output average.

In this example, the zone temperature is being monitored between a time range while the Calendar event is On. The Analyse will reset when the time of day is equal to the reset variable value, in this case when the hour equals 11.

Back to block tables page

BACNET COMMS INPUT

The BACnet Comms Input is used to read back BACnet data values from any BACnet device and its Objects located on the local BACnet network. **Note that 200 BCI Blocks can be added per Omni device in Focus.**

NODE DESCRIPTORS

Value: This is the value of the selected object property.

Status Flag: The BACnet Status Flags property of the Object selected. Bit Values: In-alarm (0), Fault (1), Overridden (2), Out of Service (3). Any value other than 0 indicates there is an issue for the object.

Comms Fault: The Comms Fault indicates the communications status with the BACnet device. (0=OK, 1=Error, 2=Device Offline, 3=Internal Error)

- If valid communication is established with the external device, the output will be 0 - OK
- If communication is established with the external device but an error is received, this output will be 1 – Error
- If the external device is considered offline, this output will change to 2 - Device Offline. ·
    - A device is considered as Offline dependent on the BACnet protocol Retries and Timeout properties.
- If the Omni internal Comms server fails to register the BACnet Object, this is classed as an Internal Comms Fault. In this condition the Comms Fault output will be 3 – Internal Error.

INTERNAL BLOCK PROPERTIES

Name: Specify a name for this block. The default name is BACnet Comms Input.

Default Value: The value that is put on the Value node when the internal property Decommission is set to True or when there is a Comms Error.

Decommission: Check the box to decommission the block. This prevents further read values being performed for the object being read.

Network Number: Choose from Global, Local or Specific. We recommend specific to optimise network communications.

Device Instance: This is the unique BACnet address of the Device on which the BACnet object resides.

Object Type: Choose the BACnet object type. The selection determines what items are available in the Object Property combo.

Object Instance: This is the unique BACnet address for the selected object. Every Object Type must have a unique address on the BACnet Device. The default Object Instance is 0.

Object Property: Choose the BACnet object type Property. The items shown are based on the Object Type combo selection.

Update Method: Choose the Update Method: Polling, DS-COV, DS-COVP, DS-COVU, DS-COVPU.

- **Polling** - Based on the Polling Rate property. The Value of the selected Object Property is continually read / polled at the selected Polling Rate. **Polling Rate**: Number of seconds to wait between requests for the remote point value if the update method is set to Polling.
- **DS-COV** - **Change of Value (default)**, the object tells the input that if it has changed based by more or less than the block's COV Increment value. The device requests a **Confirmed Change Of Value notification** from the other device. The other device determines when it notifies recipients of value changes based on its own internal settings. If the other device doesn't support COV then an error will be shown. This is a **BACnet confirmed service**, meaning that the other device will retry sending the notification if no response is received.
- **DS-COVU** - **Change of Value Unconfirmed**, the object tells the input that if it has changed based by more or less than the block's COV Increment value. If COVU is selected, a notification is sent out but it doesn't expect acknowledgement that the notification was received. The device requests an **Unconfirmed Change Of Value notification** from the other device. The other device determines when it notifies recipients of value changes based on its own internal settings. If the other device doesn't support COV then an error will be shown. This is a **BACnet unconfirmed service**, meaning that the other device will not retry sending the notification if it fails.
- **DS-COVP** - **Change of Value Property** (Registers an Object Property), the object tells the input that if it has changed based by more or less than the block's COV Increment value. You can also specify a different COV Increment from the Comms Input block. The device requests a **Confirmed Change Of Value notification** from the other device for the selected Object Property. The other device determines when it notifies recipients of value changes based on its own internal settings or the specified COV increment given. If the other device doesn't support COV then an error will be shown. This is a **BACnet confirmed service**, meaning that the other device will retry sending the COV notification is no response is received.
- **DS-COVPU - Change of Value Unconfirmed Property** (Registers an Object Property), the object tells the input that if it has changed based by more or less than the block's COV Increment value. You can also specify a different COV Increment from the Comms Input block. If COVPU is selected, a notification is sent out but it doesn't expect acknowledgement that the notification was received. The device requests an **Unconfirmed Change Of Value notification** from the other device for the selected Object Property. The other device determines when it notifies recipients of value changes based on its own internal settings or the specified COV increment given. If the other device doesn't support COV then an error will be shown. This is a **BACnet unconfirmed service**, meaning that the other device will not retry sending the COV notification if it fails.

COV Lifetime: (visible when DS-COVx is selected as the Update Method) The number of seconds between each subsequent re-registration request sent to the external controller. After the time listed here (in seconds) the COV Lifetime restarts automatically.

Object Property: (visible when DS-COPx is selected as the Update Method) The property to be read from the target object.

Specify COV Increment: (visible when DS-COPx is selected as the Update Method) Check to specify the COV increment from the Comms Input block rather than the target block.

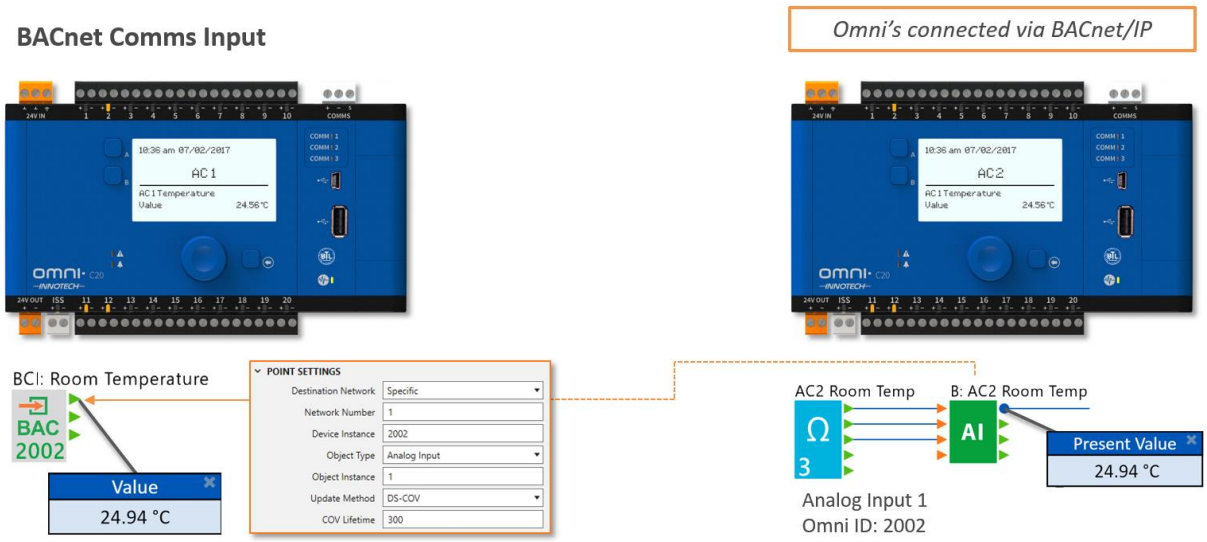
COV Increment: (visible when DS-COPx is selected as the Update Method) The amount the value needs to change by, either positive or negative before the external device will send a new value.

Notes

The key fields here are: Device Instance and Object instance. This is the BACnet Device you are connecting to (by unique BACnet Device Instance number) and the BACnet Object you are connecting to on that device (by BACnet Object type). In BACnet you can have multiple objects with the same BACnet Object number, **as long as they are different types**. You use the Object Type field to advise which type, for example Analog Value.

These objects do not appear in iComm as the green BACnet object, as they are not true objects but read-write accessors.

EXAMPLE USAGE



Essentially the BACnet Comms Input can be used to read back any BACnet object on any compatible BACnet devices. This value can then be used as required in your configuration.

[Back to block tables page]

BACNET COMMS OUTPUT

The BACnet Comms Output is used to write a value to any compatible BACnet objects, such as a setpoint, on a BACnet device located on the local BACnet network. **Note that 200 BCO blocks can be added per Omni in Focus.**

NODE DESCRIPTORS

Input: Value that is written to the external BACnet object

Ext. Priority: Priority at which the value is written (overrides block Priority property if connected)

Value: This is the value of the selected object property that will be written to the remote BACnet Object

Comms Fault: The Comms Fault indicates the communications status with the BACnet device. (0=OK, 1=Error, 2=Device Offline, 3=Internal Error)

Refer to the comms input for more information on the comms fault.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is BACnet Comms Output.

Decommission: Check the box to decommission the block. This prevents further write values being performed for the object being read.

Network Number: Choose from Global, Local or Specific. We recommend specific to optimise network communications.

Device Instance: This is the unique BACnet address of the Device on which the BACnet object resides.

Object Type: Choose the BACnet object type. The selection determines what items are available in the Object Property combo.

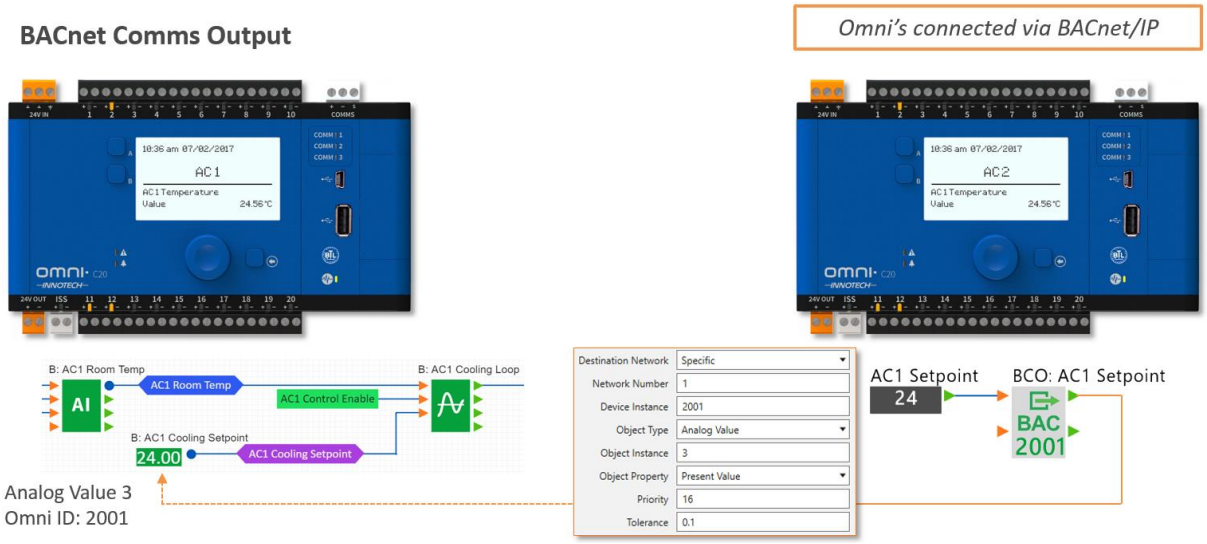
Object Instance: This is the unique BACnet address for the selected object. Every Object Type must have a unique address on the BACnet Device. The default Object Instance is 0.

Object Property: Choose the BACnet object type Property. The items shown are based on the Object Type combo selection.

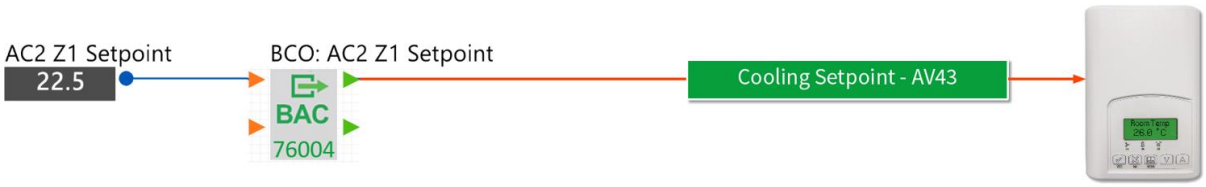
Tolerance: Input value will be written to the target object when it changes by more than the tolerance.

EXAMPLE USAGE

In the example below, the BACnet comms output is writing to the other Omni setpoints present value which is a BACnet user variable.



In this example, the BACnet comms output is writing the setpoint value that can be adjusted via the HMI or webserver, to the BACnet stat setpoint value present value. For this device, this is Analog Value 43.



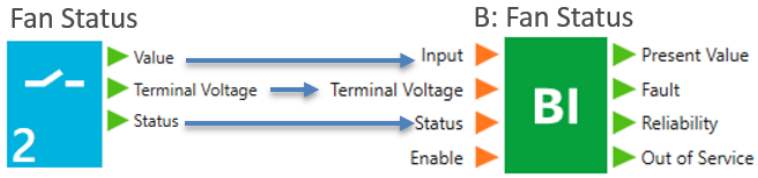
Essentially, any compatible BACnet object can be written to using this block. When using this block to perform a BACnet write to an object, the correct priority must be used. This will need to be checked if a third party device is used.

[Back to block tables page]

BINARY INPUT

The Binary input block can be used to provide a digital (binary) status to the BACnet network.

This BACnet object is typically used in conjunction with the controller Input, such as a fan status, as shown below.



This will pass the status value to the BI input node which is then passed to the BI Present Value node. The terminal voltage and Status nodes that are fed into the BI block and used to generate the fault and reliability of the input.

The BACnet client can then monitor any of the BI output nodes as required.

NODE DESCRIPTORS.

Present Value - The present value node provides the value that has been written to the block at the highest priority, either from the blocks input or a BACnet client. In the block properties, you will notice a relinquish default value. This is the value that will be present on the Present Value node if all the 16 priorities have been relinquished or released, until an input value is detected.

Enable Node - The enable node enables or disables the block as per Innotech block functions. When the enable is On, the block input value is passed through to the Present Value. If the Enable is Off, the input is not passed through and the last value is held by the block, until the relinquish node is triggered to the default value set in the blocks properties.

Note that the enable has no effect if the out of service node has been enabled and activated from a BACnet client.

Out of Service - BACnet clients write directly to the Out of Service property to disable local control of the object, therefore disconnecting the Input from the Present Value and only allowing control via the BACnet network. *(The block Input node is ignored whenever the Out of Service output is TRUE)*

If the internal Allow Override property is not enabled (FALSE), the Out of Service property cannot be written to, thus preventing the external client taking control of this block.

Reliability - The reliability output specifies if the BACnet object is in an ok or fault state. The value on the fault node advises if a fault exists, a value of 1, or not in fault with a value of zero (0).

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Binary Input.

Object Instance - Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

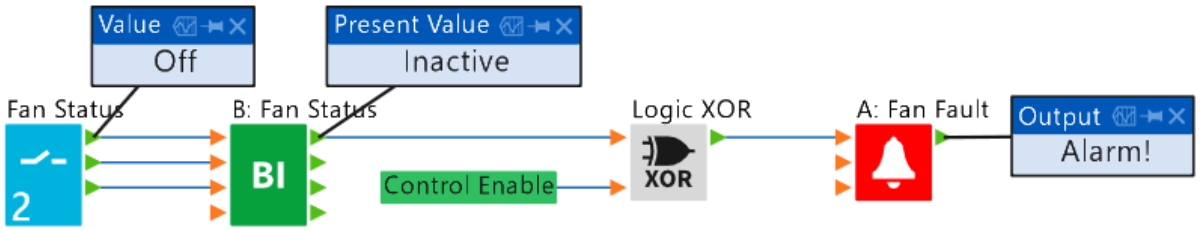
Device Type: A text description of the device connected to the Input.

Polarity: Select from Normal or Reverse.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

EXAMPLE APPLICATION

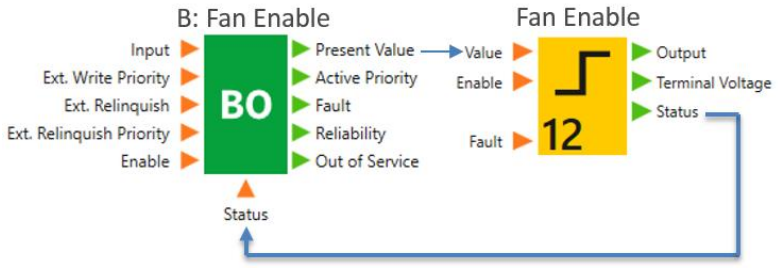
In the example below, the Fan status input on the Omni is connected to the Binary Input (BI). The value from the digital input is fed into the BI then the value is sent to the BI Present Value node. This can then be monitored by any BACnet client on the BACnet network.



Back to block tables page

BINARY OUTPUT

The Binary Output block defines a standardised BACnet object whose properties represent the externally visible characteristics of a binary output.



Typically, this block is used to pair a Innotech digital output block to provide its value and status to the BACnet network, as shown above.

NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active, and the Input value is written to the object at the highest Write Priority.

Ext. Write Priority: This determines the position in the Priority list (Known as the Priority Array) that the input value will be written to. (This equates to a value from 1-16, where 1 is the highest priority) The Input value will be written at the Ext. Write Priority value on a change in value of the Input, or a change in value of the Ext. Write Priority input.

Ext. Relinquish: The Ext. Relinquish node is a digital signal applied to delete a value that has been written to the object at a particular priority. When using the Ext. Relinquish node, you must also indicate the Priority of the value to be relinquished. This is achieved by setting a value from 1-16 on the Ext. Relinquish Priority node. Therefore, you should set the Ext. Relinquish Priority first, and then pulse the digital signal on the Ext. Relinquish node from “Off” to “On”.

Ext. Relinquish Priority - The Ext. Relinquish Priority input allows a user to select any priority (and value) to be relinquished from the Priority Array.

Enable: The Enable input, **when used**, of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Present Value: The Present Value displays the value that was written at the highest Write Priority, either from the input or from a BACnet client. If all priorities have been relinquished, the Relinquish Default internal property will become the Present Value.

Active Priority: The Active Priority indicates the highest priority that was written to the object. (1-16) If all priorities have been relinquished, the Active Priority will be 0 until the next input write and then default to the priority set in the blocks properties.

Fault: The Fault output will be FALSE whenever the Reliability output has a value of NO_FAULT_DETECTED, otherwise it will be TRUE.

Reliability - The reliability output specifies if the BACnet object is in an ok or fault state. The value on the fault node advises if a fault exists, a value of 1, or not in fault with a value of zero (0).

Out Of Service: The value of this property indicates whether or not the object is currently being overridden by a BACnet Client.

INTERNAL BLOCK PROPERTIES

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Device Type: A text description of the device connected to the Input.

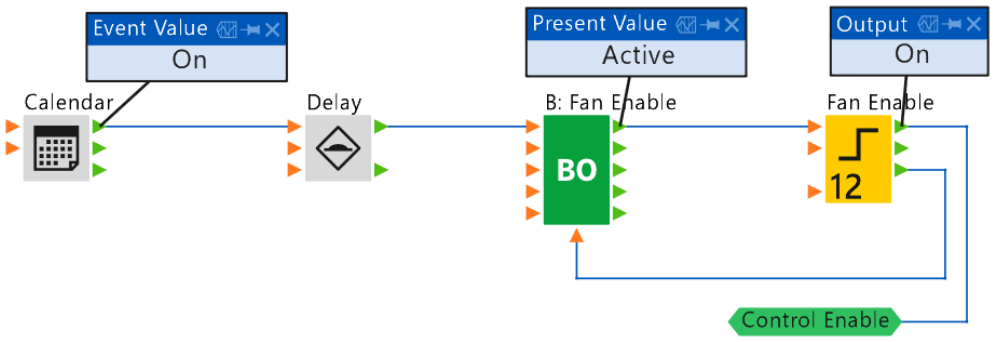
Relinquish Default: The Relinquish Default setting is the default value the Present Value will display when all Write Priorities are relinquished.

Write Priority: The Write Priority internal property indicates the priority at which the Input value will be written. This property is only used when there is no connection to the Ext. Write Priority input.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

EXAMPLE APPLICATION

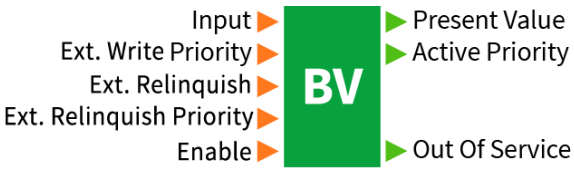
In the example below, the Binary Output (BO) is connected in line with the control signal for the fan control. This is then connected to the Omni Digital Output for control. The Digital Outputs status is connected to the BO to pass on the fault condition to the BO reliability. The BO can be monitored by any BACnet client on the BACnet network.



Back to block tables page

BINARY VALUE

The Binary Value block defines a standardised BACnet object whose properties represent the externally visible characteristics of a binary value.



The Binary Value block can be used to provide any digital signal connected to it to be exposed to the BACnet network. Any BACnet client can monitor this blocks Present Value.

NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active, and the Input value is written to the object at the highest Write Priority.

Ext. Write Priority: This determines the position in the Priority list (Known as the Priority Array) that the input value will be written to. (This equates to a value from 1-16, where 1 is the highest priority) The Input value will be written at the Ext. Write Priority value on a change in value of the Input, or a change in value of the Ext. Write Priority input.

Ext. Relinquish: The Ext. Relinquish node is a digital signal applied to delete a value that has been written to the object at a particular priority. When using the Ext. Relinquish node, you must also indicate the Priority of the value to be relinquished. This is achieved by setting a value from 1-16 on the Ext. Relinquish Priority node. Therefore, you should set the Ext. Relinquish Priority first, and then pulse the digital signal on the Ext. Relinquish node from “Off” to “On”.

Ext. Relinquish Priority - The Ext. Relinquish Priority input allows a user to select any priority (and value) to be relinquished from the Priority Array.

Enable: The Enable input, **when used**, of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Present Value: The Present Value displays the value that was written at the highest Write Priority, either from the input or from a BACnet client. If all priorities have been relinquished, the Relinquish Default internal property will become the Present Value.

Active Priority: The Active Priority indicates the highest priority that was written to the object. (1-16) If all priorities have been relinquished, the Active Priority will be 0 until the next input write and then default to the priority set in the blocks properties.

Out Of Service: The value of this property indicates whether or not the object is currently being overridden by a BACnet Client.

INTERNAL BLOCK PROPERTIES

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Device Type: A text description of the device connected to the Input.

Relinquish Default: The Relinquish Default setting is the default value the Present Value will display when all Write Priorities are relinquished.

Write Priority: The Write Priority internal property indicates the priority at which the Input value will be written. This property is only used when there is no connection to the Ext. Write Priority input.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

Back to block tables page

CALCULATOR ADD

The Calculator Add block provides an addition calculation based on two inputs.

NODE DESCRIPTORS

Input A: The first of two values used for the addition calculation.

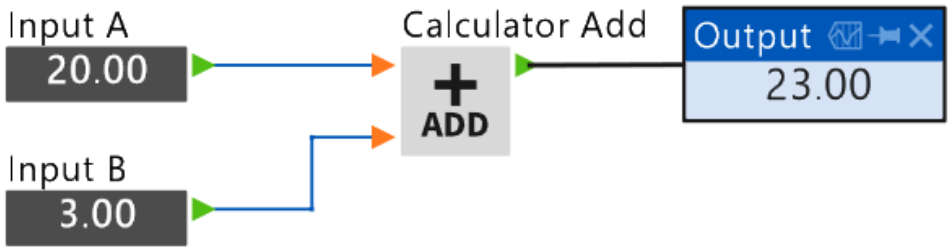
Input B: The second of two values used for the addition calculation

Output: The calculated result of Input A + Input B.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator Add.

USAGE EXAMPLE



Essentially the inputs are added together to provide the final output value.

Back to block tables page

CALCULATOR DIVIDE

The Calculator Divide block provides a division calculation based on two analog inputs.

NODE DESCRIPTORS

Input A: The first of two values used for the division calculation.

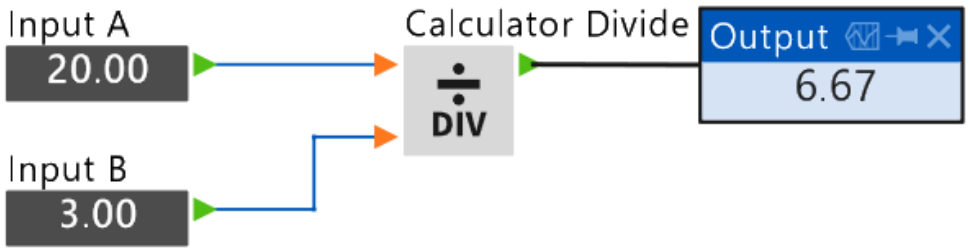
Input B: The second of two values used for the division calculation.

Output: The calculated result of Input A divided by Input B.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator Divide.

USAGE EXAMPLE



Essentially the inputs are divided to provide the final output value.

Back to block tables page

CALCULATOR INTEGER DIVIDE

The Calculator Integer Divide block provides an Integer division calculation based on two analog inputs.

The Integer Divide calculation divides two numbers. The output value is the whole number only of the division calculation. For example: 18 divided by 4 = 4 with 2 remainder. The Output result is the **whole number only** of 4 because the remainder of 2 is dropped.

NODE DESCRIPTORS

Input A: The first of two values used for the integer division calculation.

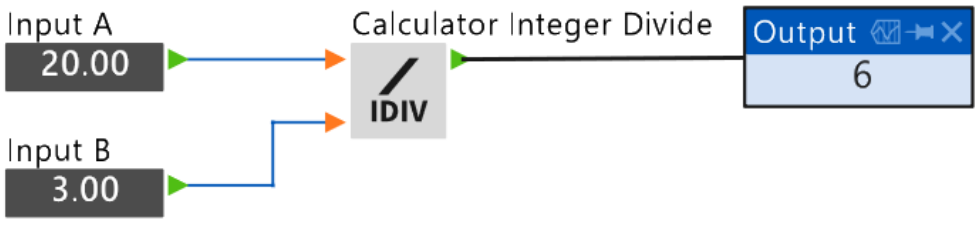
Input B: The second of two values used for the integer division calculation.

Output: The calculated result of Input A divided by Input B. The Output is equal to the whole part of Input A divided by Input B. For example, 57 div 6 = 9.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator Integer Divide.

USAGE EXAMPLE



Essentially the inputs are divided to provide the final **whole number** output value.

Back to block tables page

CALCULATOR LOGARITHM

The Calculator Logarithm block provides a logarithm calculation based on two analog inputs.

The Logarithm calculation is basically a reverse Power calculation. For example, input A is 1000, Input B is 10 as the base value, Log output will be 3. This is because 10 to the power of 3 (10 x 10 x 10 = 1000).

NODE DESCRIPTORS

Input A: The first of two values used for the logarithm calculation. **This is the value to be converted.**

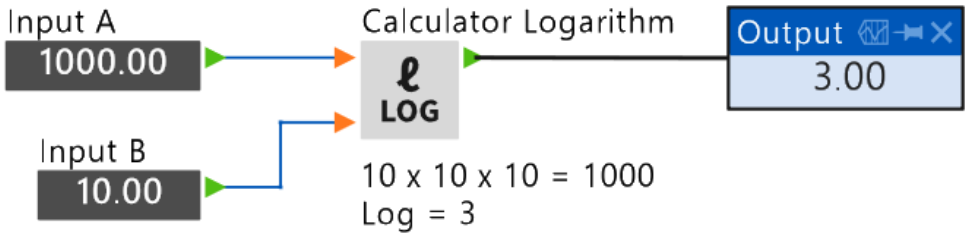
Input B: The second of two values used for the logarithm calculation. This is referred to as the Base value. **For decimal this will be 10**.

Output: The calculated result of Log Input A to Base Input B.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator logarithm.

USAGE EXAMPLE



The Calculator evaluates the values at input A and B and presents the calculated value at the output.

Back to block tables page

CALCULATOR MODULUS

The Calculator Modulus block provides a modulus calculation based on two analog inputs.

The Modulus calculation divides two numbers and the output value is the remainder only of the division calculation.

For example: 10 Mod 5 = 0, this is because 10 divided by 5 has no remainder. 7 Mod 5 = 2, because 7 divided by 5 equals 1 with 2 remainder.

NODE DESCRIPTORS

Input A: The first of two values used for the Modulus calculation.

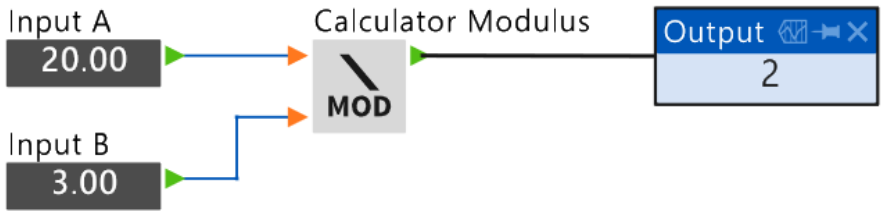
Input B: The second of two values used for the Modulus calculation.

Output: The calculated result of the remainder when Input A divided by Input B.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator Modulus.

USAGE EXAMPLE



Back to block tables page

CALCULATOR MULTIPLY

The Calculator Multiply block provides a multiplication calculation based on two analog inputs.

NODE DESCRIPTORS

Input A: The first of two values used for the multiplication calculation.

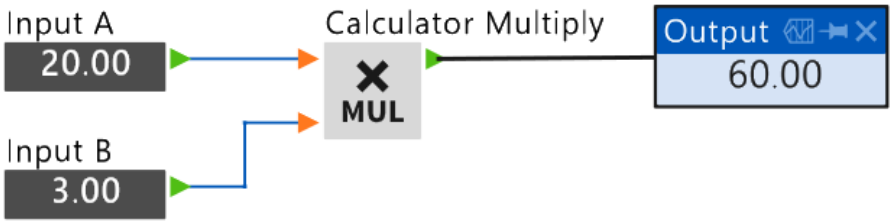
Input B: The second of two values used for the multiplication calculation.

Output: The calculated result of the remainder when Input A multiplied by Input B.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator Multiply.

USAGE EXAMPLE



Back to block tables page

CALCULATOR POWER

The Calculator Power block provides a power of calculation based on two analog inputs.

The Power calculation multiplies the number at Input A by itself the number of times specified in Input B.

For Example: Input A is 20, Input B is 3, Output result: 20 x 20 x 20 = 8000.

NODE DESCRIPTORS

Input A: The first of two values used for the power calculation.

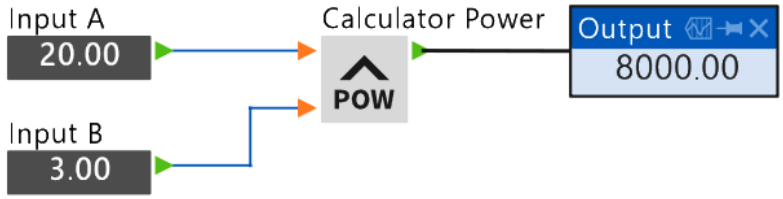
Input B: The second of two values used for the power calculation.

Output: The calculated result of Input A to the Power of Input B.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator Power.

USAGE EXAMPLE



Back to block tables page

CALCULATOR SUBTRACT

The Calculator Subtract block provides a subtraction calculation based on two analog inputs.

NODE DESCRIPTORS

Input A: The first of two values used for the subtraction calculation.

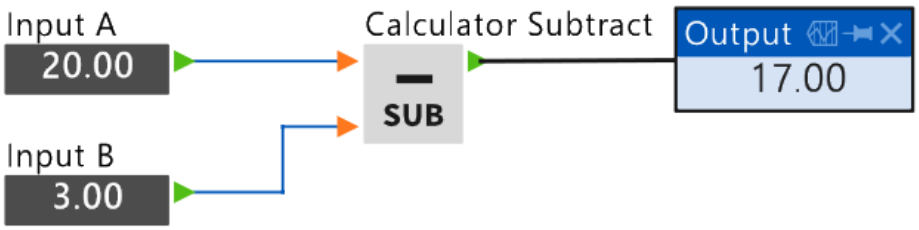
Input B: The second of two values used for the subtraction calculation.

Output: The calculated result of Input A - Input B.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calculator Subtract.

USAGE EXAMPLE



Back to block tables page

CALENDAR

The Calendar block performs the function of a standard calendar object with built-in iCal standard support. Calendar supports scheduling of on and off events during a day or across multiple days and allows for recurring events. The calendar block supports both analog and digital calendars.

Note that 100 Calendars can be added per Omni in Focus. Each Calendar can have up to 100 events with a total of 2000 events per device.

NODE DESCRIPTORS

Override Value: The value of the calendar block if it is overridden.

Calendar Override: The input determines if the calendar should be overridden.
Note: Typically, an exception calendar or after hours' switch would be connected to the calendar input.

Event Value: Outputs the value of the calendar event, ON or OFF. This node is used for standard events to enable equipment.

Time Until: Outputs a countdown to the next calendar event or the end of the current event and then turns the Output ON or OFF.

Optimum Run: Connect this node to the Optimum Start block's Enable node to execute a calendar event at the optimum time determined by the connected Optimum Start block. If no event is active, it shows the value of the next event in this calendar. If an event is active (including an override event) it shows the value of the current event.

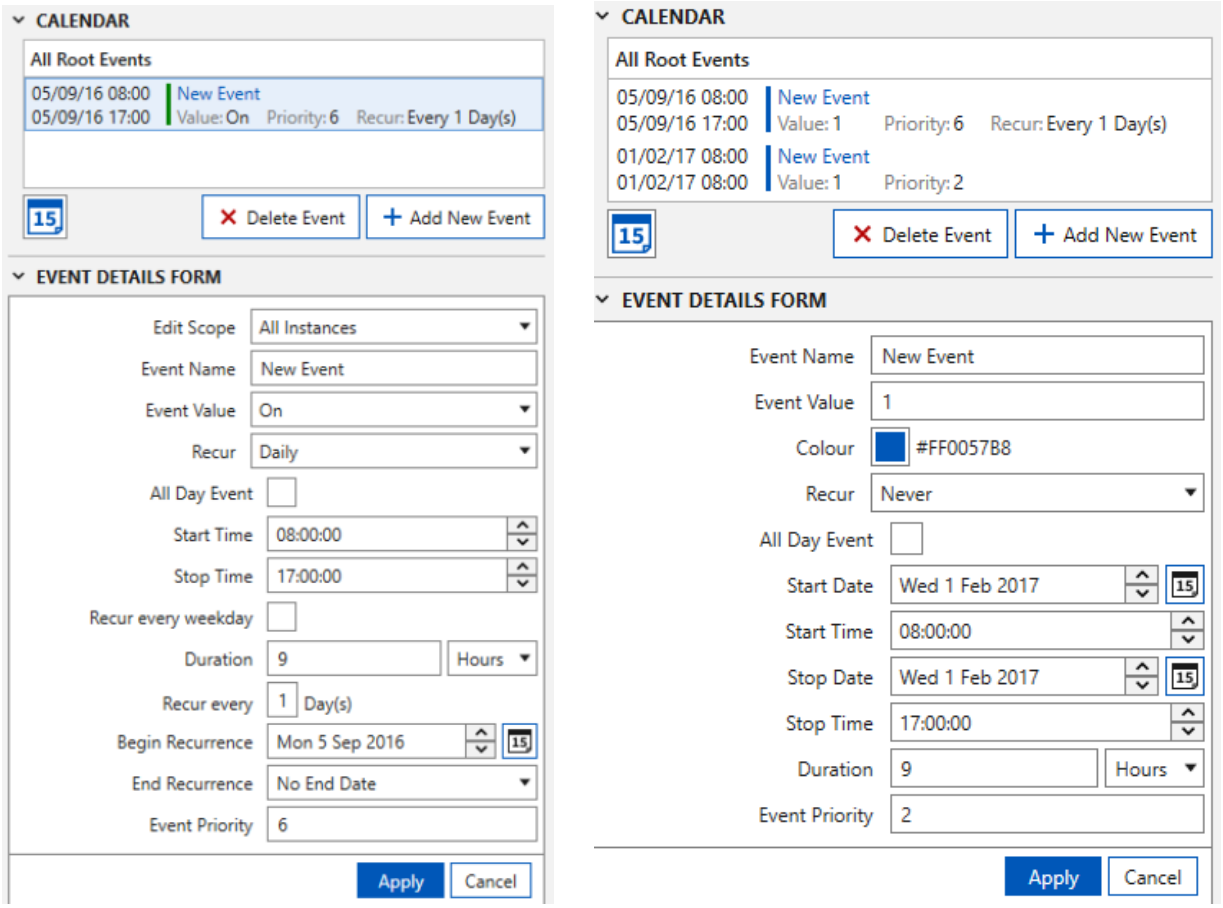
INTERNAL PROPERTIES

Name: Specify a name for the calendar. The default name is Calendar.

Calendar Event Type: Select Analog or Digital. Digital calendars are always OFF when no event is active.

Inactive Event Value: [Analog Calendar Types only] Enter the default value for the Event Value node when no event is active.

Add New Event: Click the button to create a new event. After a new event is created, the Event Details Form will be shown for customising your new event.



Add new event examples above. Digital left and analog right. The set analog can be used to provide a time switched setpoint value.

Delete Event: Click the button to delete the currently selected event.

Edit Scope: [Recurring Events Only] This value is used when editing an event from the CAD calendar as you can get a single instance of an occurring event.

Event Name: Give the calendar event a name.

Event Value: For Digital calendars, select On or Off. For Analog calendars, a value is used to determine how the event is handled.

Colour: [Analog Type Only] Select a colour for your calendar event.

Recur: Select a recurrence option: Daily, Weekly, Monthly or Yearly.

All Day Event: Check the box to make the event an all-day event. If this is selected, start and stop times and duration are not required to be set.

Start Time: Set the event start time.

Stop Time: Set the event stop time.

Start Date: [Doesn't show if a recur interval is selected] Set the event start date.

Stop Date: [Doesn't show if a recur interval is selected] Set the event stop date.

Recur every. Select recurrence interval based on recurrence frequency selected.

Duration: Specify a duration rather than selecting a stop time.

Begin Recurrence: Select a date to start the recurrence from.

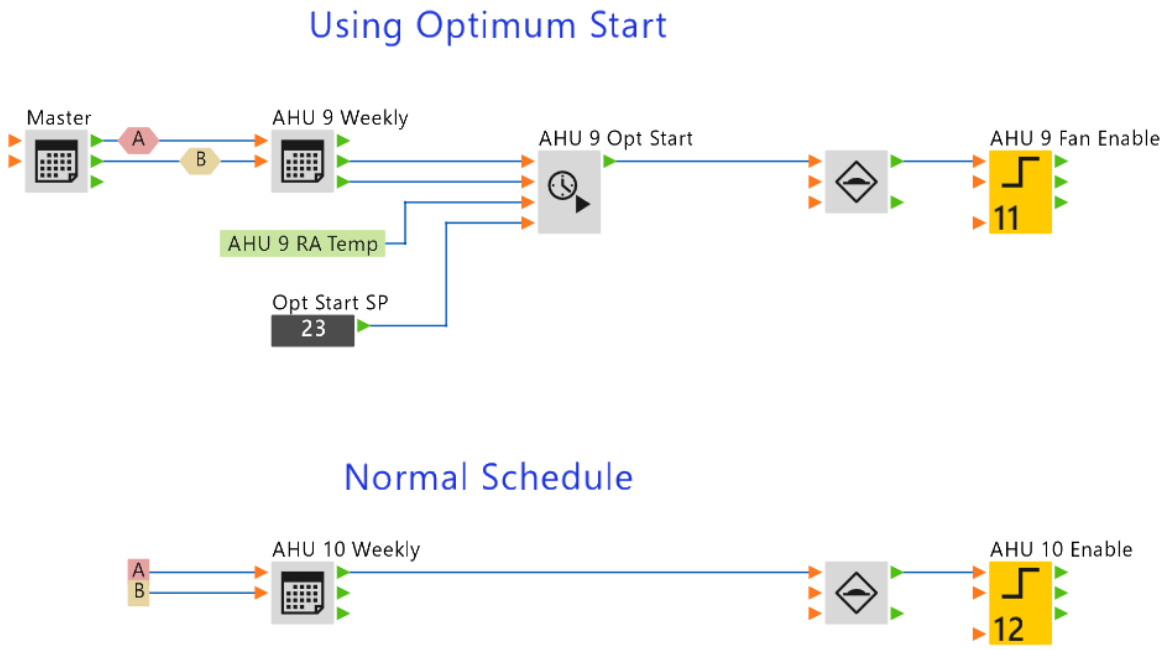
End Recurrence: Select when to finish the recurrence of the event. Select from No End Date, By Date or By Count.

Event Priority: When multiple events overlap, priority will be given to the event with the lowest number.

Apply: Click the Apply button to save any changes made to new or existing calendars.

Cancel: Cancel changes to a new or existing calendar.

USAGE EXAMPLES



Back to block tables page

CALENDAR - BACNET

The Calendar object defines a standardised object used to describe a list of calendar dates, which might be thought of as "holidays," "special events," or simply as a list of dates.

NODE DESCRIPTORS

Present Value: Indicates the current value of the calendar. Present Value is TRUE if the current date is in the Date List, otherwise False.

Time Until: Outputs a countdown to the next calendar event or the end of the current event and then turns the Output ON or OFF.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Calendar.

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

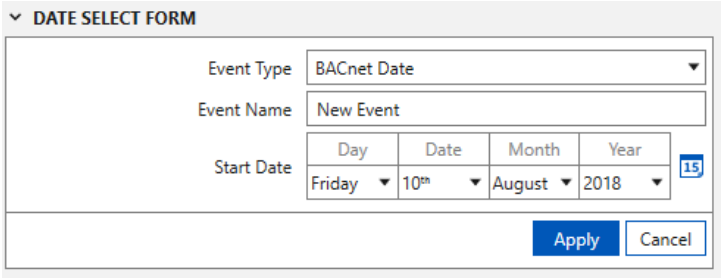
Add Date: Add a date to the BACnet calendar.

Delete Date: Delete selected date from BACnet calendar.

Event Type: Select from BACnet Date (individual date), BACnet Date Range (selection of dates) or BACnet WeekNDay (select a week of dates starting from the specified day).

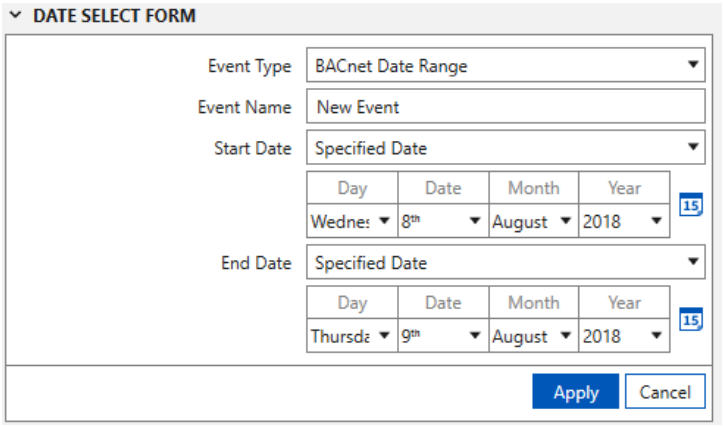
EXAMPLES

BACnet Date Example



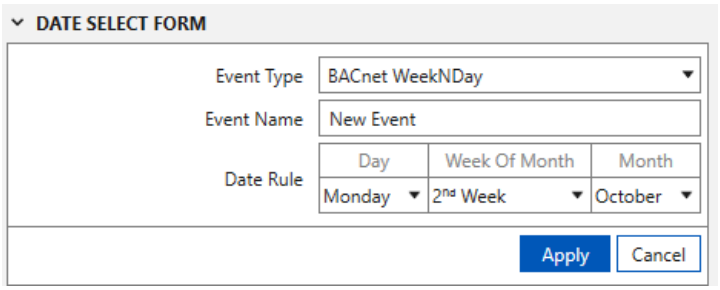
This will provide an ON signal for the selected day.

BACnet Date Range Example



The BACnet calendar provides an ON event for the date range selected.

BACnet WeekNDay Example

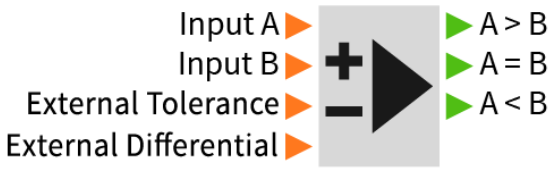


The BACnet calendar provides an ON event for the day of the week and month selected.

[Back to block tables page]

COMPARATOR

The Comparator block takes two analog inputs and compares them using comparison and differential.



NODE DESCRIPTORS

Input A: The first of two inputs used by the block's comparison function.

Input B: The second of two inputs used by the block's comparison function.

External Tolerance: Point input which can be used to override the internal block property Tolerance.

External Differential: Point input which can be used to override the internal block property Differential. Avoid setting the External Tolerance value less than the internal block property Differential value.

A > B: Point which is the ON state if Input A is greater than Input B, as per the comparison function.

A = B: Point which is the ON state if Input A is equal to Input B, as per the comparison function.

A < B: Point which is the ON state if Input A is less than Input B, as per the comparison function.

FUNCTIONALITY

For example, read through the following scenario. Let's assume we have a comparator block with a Differential value of 2 and a Tolerance value of 6.

- If INPUT A and INPUT B were both equal to 20, then the A=B output would be ON and the other two outputs would be OFF. The A=B output would remain on as long as INPUT A was within the Tolerance Band of INPUT B, so if INPUT B remained at 20, then INPUT A could be between 17 and 23 for the A=B output to stay ON (the difference between 17 and 23 is the Tolerance value of 6).

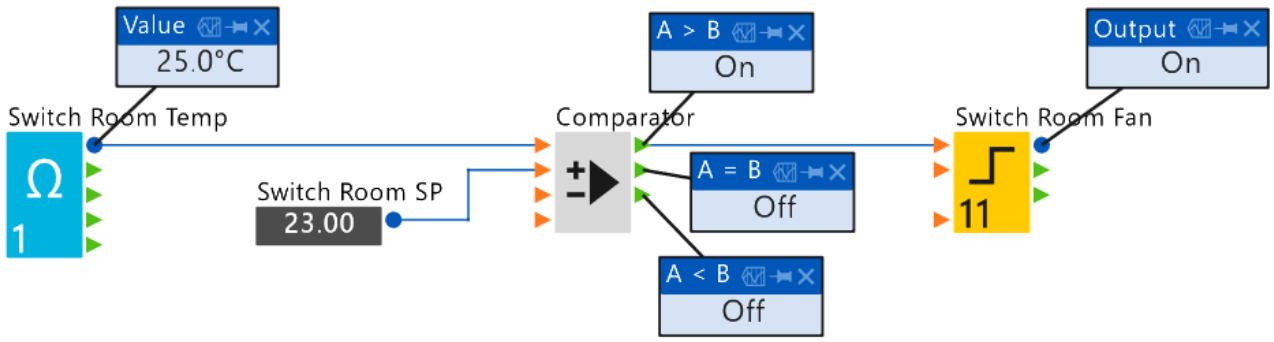
- As the value of INPUT A falls, the A=B output would turn OFF when INPUT A became less than or equal to 17 and the A<B output would turn ON.

- Finally, as the value of INPUT A rose again, the A=B output would not turn back on until the Differential Band had been exceeded - in this case, INPUT A would have to become greater than or equal to 19 (which is equal to 17 + the Differential value) before the A=B output turned on and the A<B output turned OFF.

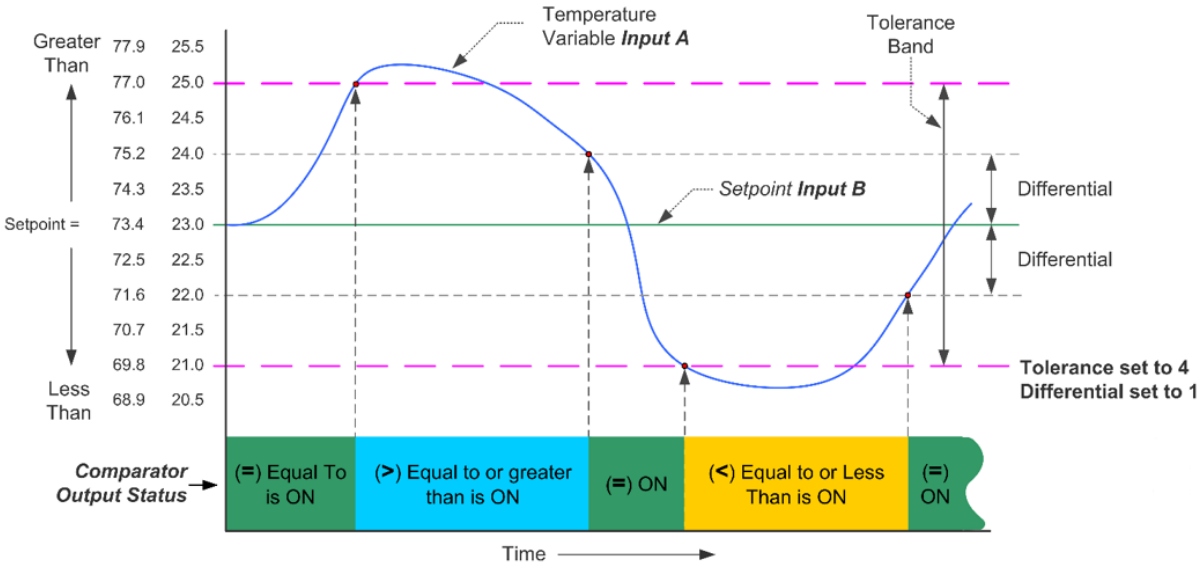
The A>B output works in a similar manner except at the other end of the scale.

The previous example utilises the full power of the Comparator block, however you can simplify the operation by setting either or both of the Tolerance and Differential values to zero. If the Differential value is zero, then the A=B output will turn OFF immediately that INPUT A fell outside of the tolerance band from INPUT B and it would turn back ON immediately when it fell within the same band. If the Tolerance value is zero, then the Comparator block would perform a normal comparison operation of INPUT A and INPUT B with no tolerance or differential functions.

CONFIG EXAMPLE



Comparator Functional Graph



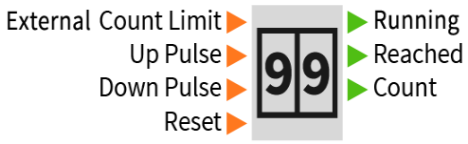
Note that the differential value set in the block properties is equal in value either side of the setpoint in the above example. The total Tolerance value is split in half either side of the setpoint in the above example.

A useful tip is to fully test the comparator in the simulator to ensure that the intended result is produced.

Back to block tables page

COUNTER

The Counter block provides a counting function from a digital signal. The counter can count up and down, and will produce a signal when a count limit has been reached. The counter can be reset at any time and provides the current value of the counter on an output.



NODE DESCRIPTORS

External Count Limit: Point value representing the count at which the Reached output is triggered. This input overrides the block property Count Limit.

Once the block property Count Limit has been reached this value, the Reached output will turn ON until the current Count value is less than the Count Limit. This value must be greater than or equal to zero, and is an external value which overrides the block property Count Limit.

Up Pulse Digital Input: Point input which increments the current Count value whenever it makes a transition from OFF to ON.

Down Pulse Digital Input: Point input which decrements the current Count value whenever it makes a transition from OFF to ON.

Reset Digital Input: Point input which resets the current Count whenever it makes a transition from OFF to ON. The Counter will not begin counting until the Reset input is OFF.

Running Digital Output: Point which is ON whenever the current Count value is non-zero.

Reached Digital Output: Point which is ON whenever the current Count value is greater than or equal to the Count Limit.

Count Analog Output: Point which is the current Count value.

INTERNAL PROPERTIES

Count Limit: The internal Count limit. This value can be overridden by the Count Limit Input if connected. The Count Limit must be greater than or equal to zero.

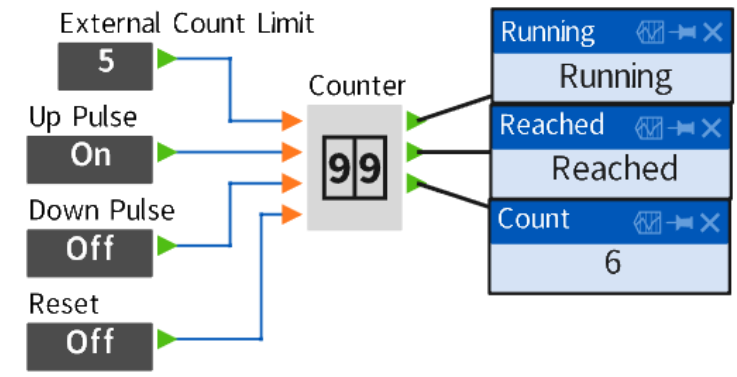
Initial Value: This is the starting Count value which will be used at the output of the counter.

When configuration is read back from the device the Initial Value will be set to the Count value at the output of the block at the time of the configuration transfer. This is useful if configuration needs to be retransferred to the device (e.g. for maintenance) because the Count value will not be reset to zero but instead will start counting from the last known Count value. For example, you may be using the block as a people counter, but when sending the configuration, you may need to specify that the counter has already counted 350 people.

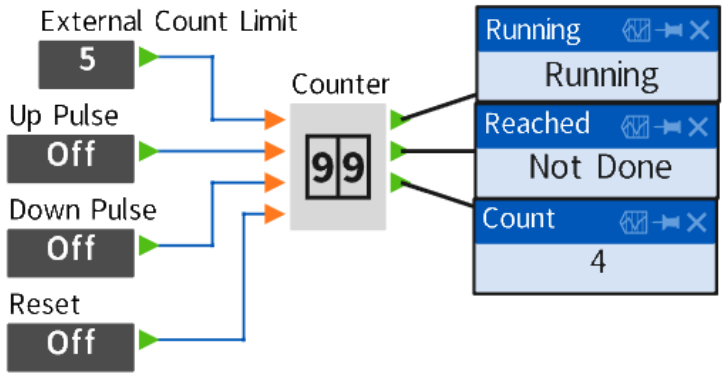
EXAMPLES

The Counter counts up and down pulses. Pulses will continue to be counted even after the limit is reached, the output will be turned on until reset of the count is below the limit.

Counter - Limit Reached

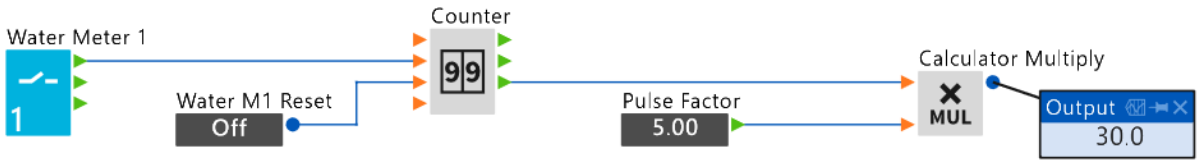


Counter - Limit Not Reached



The Counter in the example below will monitor the status of Digital Input #1, Water Meter 1 which is a momentary pulse type meter. Upon each pulse of this Input, the counter counts 1. This is added to the Counter total until reset, which can be achieved via the Water M1 Reset User Variable. The final reading is provided via the Calculator block, which multiplies the Counter total by the meter conversion factor provided by the Meter Pulse Factor User Variable.

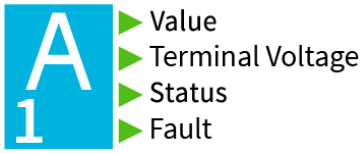
This is a common use of the counter block. The counter block for this example is set to the default configuration.



Back to block tables page

CURRENT LOOP INPUT

The Current Loop Input block returns the translated value of the applied analog loop current back to the internal processing of the digital controller.



NODE DESCRIPTORS

Value: The translated, calibrated and alpha smoothed signal.

Terminal Voltage: The current input voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Fault: Indicates that the Output point is outside the min/max properties.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Current Loop Input.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Alpha: Percentage of output difference added to the previous output to generate a new one.

Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Minimum Input Value: Specify the minimum input value.

Maximum Input Value: Specify the maximum input value.

Calculate Gain & Offset: Click to calculate the Gain and Offset. When the box is checked you can enter the Omni Reading and meter Measured Values to adjust the gain and offset for the controller. This setting is used to adjust the values on the Omni compared to a meter reading if required.

Omni Reading #1 & #2: The value that the Omni controller shows on the HMI or the on-board web-server.

User Gain Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

User Offset Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

Suppression: Power line frequency suppression, select from None, 50Hz and 60 Hz.

[Back to block tables page]

CURRENT LOOP OUTPUT

The Current Loop Output block provides an analog output current (0-20mA) based on the internal processing of the digital controller.

NODE DESCRIPTORS

Value: The type of input depends on the settings applied for the internal block property Uni Curve Points for Function.

Enable Digital Input: Used to enable or disable the block's operation. If the Enable input is not connected, the block will be enabled by default.

Fault Digital Input: If set to the On state, sets the internal block property Fault Value as the output value.

Output: Value which is the value sent to the ASIC.

Terminal Voltage: Value which is the current output voltage on the pin.

Status Information Output: Indicates the current status of the block.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Current Loop Output.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Disable Value: This value is set to the Output point when the Enable point input is FALSE.

Fault Value: This value is set to the Output when the Fault input is True.

Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Calculate Gain & Offset: Click to calculate the Gain and Offset. When the box is checked you can enter the Omni Reading and meter Measured Values to adjust the gain and offset for the controller. This setting is used to adjust the values on the Omni compared to a meter reading if required.

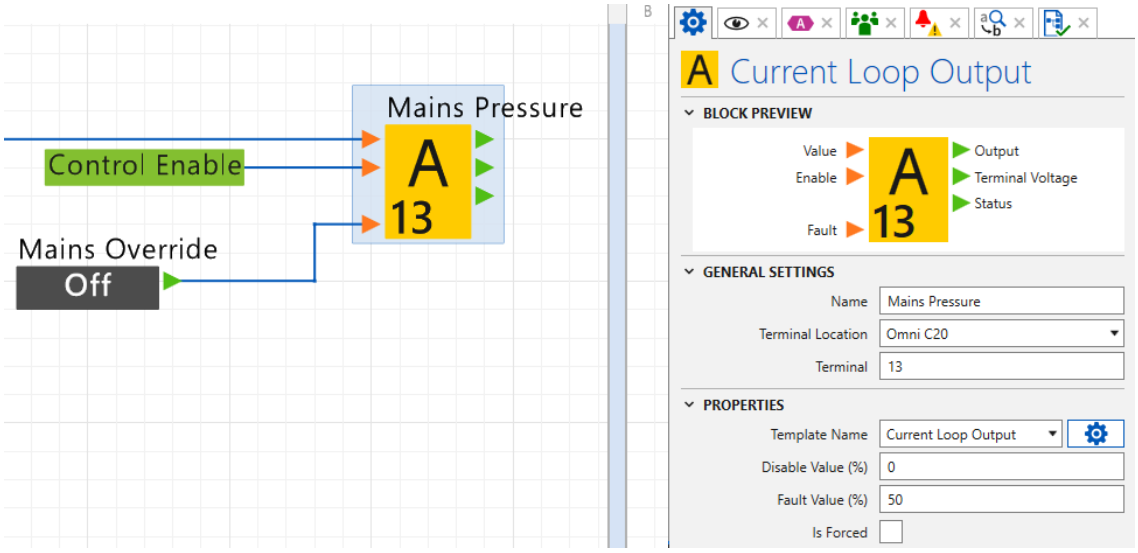
Omni Reading #1 & #2: The value that the Omni controller shows on the HMI or the on-board web-server.

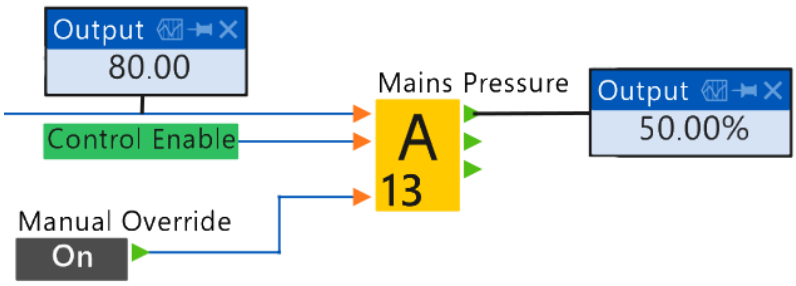
User Gain Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

User Offset Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

Operation of the Enable and Fault nodes

The **Enable node**, if connected, allows the block to provide a signal on the output node for normal operation. If this enable signal is in an OFF state, then the output signal will be given at the enable percentage value entered via the blocks properties with zero (0) being the default. If set to anything above zero, the output provides a signal at the nominated % value when the enable is OFF, regardless of what value is on the input Value node.





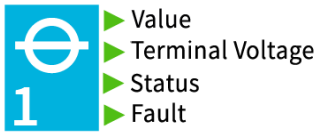
The **Fault node** allows you to connect any required digital signal to provide an ON state to drive the Output signal to the required percentage. The default is set to Zero (0) but can be any value from 0 to 100%. In this example this has been set to 50% to drive the Output open for a period of time via the Pulsed User Variable for 120 Seconds. This will override the Value Node input signal and the Enable Node if connected.

[Back to block tables page]

CURRENT/VOLTAGE INPUT

The Current/Voltage Input (Current Voltage Transducer Input (CVT)) block is designed to be used as a physical measurement interface from a Current Transformer (CT) producing a safe output voltage of 333mVAC RMS rather than the typical 0-5A AC current signal.
This means:

- A CVT has the burden resistor integrated, a CT has not.
- Disconnecting the secondary of a powered CVT is always safe.
- Disconnecting the secondary of a powered CT can be fatal.



NODE DESCRIPTORS

Value: Value which is the value sent to the ASIC.

Terminal Voltage: Value which is the current output voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Fault: Indicates if the block processing logic is correct or outside of the tolerance.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Current/Voltage Input.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Alpha: Percentage of output difference added to the previous output to generate a new one.

Line Frequency Selection:

- 50 Hz.
- 60 Hz.

Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Minimum Input Value: Specify the minimum input value.

Maximum Input Value: Specify the maximum input value.

Calculate Gain & Offset: Click to calculate the Gain and Offset. When the box is checked you can enter the Omni Reading and meter Measured Values to adjust the gain and offset for the controller. This setting is used to adjust the values on the Omni compared to a meter reading if required.

Omni Reading #1 & #2: The value that the Omni controller shows on the HMI or the on-board web-server.

Measured Value #1 & #2: The actual meter measured value.

User Gain Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

User Offset Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

Trim Value (Volts): Voltage offset correction at zero Amperes.

USAGE NOTES

Any brand of CVT device producing 333mVRMS output can be connected.

Such devices are normally designed as split core units, allowing easy installation. The UIO is utilising the 0-625mVDC range. The CVT signal is processed by internal filter algorithms and supplied to the block output as Millivolts RMS. This output is now ready for scaling into Amperes via the translation curve found in the block template.

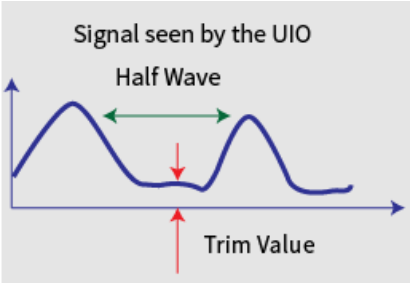
Installation:

1. Ensure the correct CVT sensor template has been applied in the configuration.
2. To compensate for the wiring influence, connect the CVT to the UIO terminals. Ensure no AC current is flowing through the CVT.
3. Monitor the block Terminal Voltage node.
4. Enter the Terminal Voltage into the “Trim Value” field.
5. For sites with multiple equivalent CVTs wired in the same manner, the “Trim Value” can be copied across, thus providing a suitable starting point.
6. Verify your setup with current flowing through the CVT.

CONNECTION OF STANDARD CTs INSTEAD OF CVTs TO ANY UIO TERMINAL WILL CAUSE IRREPAIRABLE DAMAGE TO THE OMNI BEMS CONTROLLER AND PRESENT RISK OF ELECTROCUTION.

By default, Input and Output blocks will snap to the left or right of the page they are placed on. They can be moved freely within the Focus configuration. Click the block, hold Ctrl and move the block into position.

Block considerations



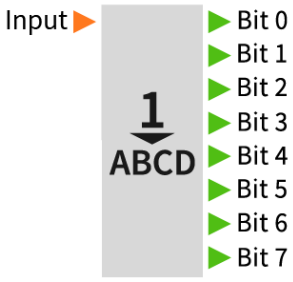
The CT Input block is targeted for symmetrical loads.

Meaning that high power half wave rectified loads such as half wave rectified power supplies and thyristor controlled loads may not produce accurate results. This is because the load current seen by the CVT could be negative only and therefore not seen with this setup. Should this be the case, reverse the CVT orientation on the phase or reverse the polarity at the UIO terminal.

Back to block tables page

DECODER

The Decoder block provides a binary decoding function, allowing you to convert the analog value range between 0 and 255 to an 8-bit binary number represented by 8 outputs of the block. The Decoder block is usually used in conjunction with the Encoder block.



NODE DESCRIPTORS

Input: The is the value in the range from 0 to 255 which will be converted into a binary number.

If the Input value is outside the valid range (0 - 255) it will default to either 0 or 255 depending on the actual value fed to Input i.e. if Input value is less than 0 it will default to 0 and if the Input value is larger than 255 it will default to 255.

Bit 0..7 Digital Outputs: The eight outputs represent the 8-bit binary result of the conversion from the analog value fed into the Input.

Byte Offset: Specify the byte offset. This allows you to shift the start point of the decoding sequence. A value of 1 shifts the decoder 8 bits. This is normally used when two or more decoders are used for larger numbers.

APPLICATIONS

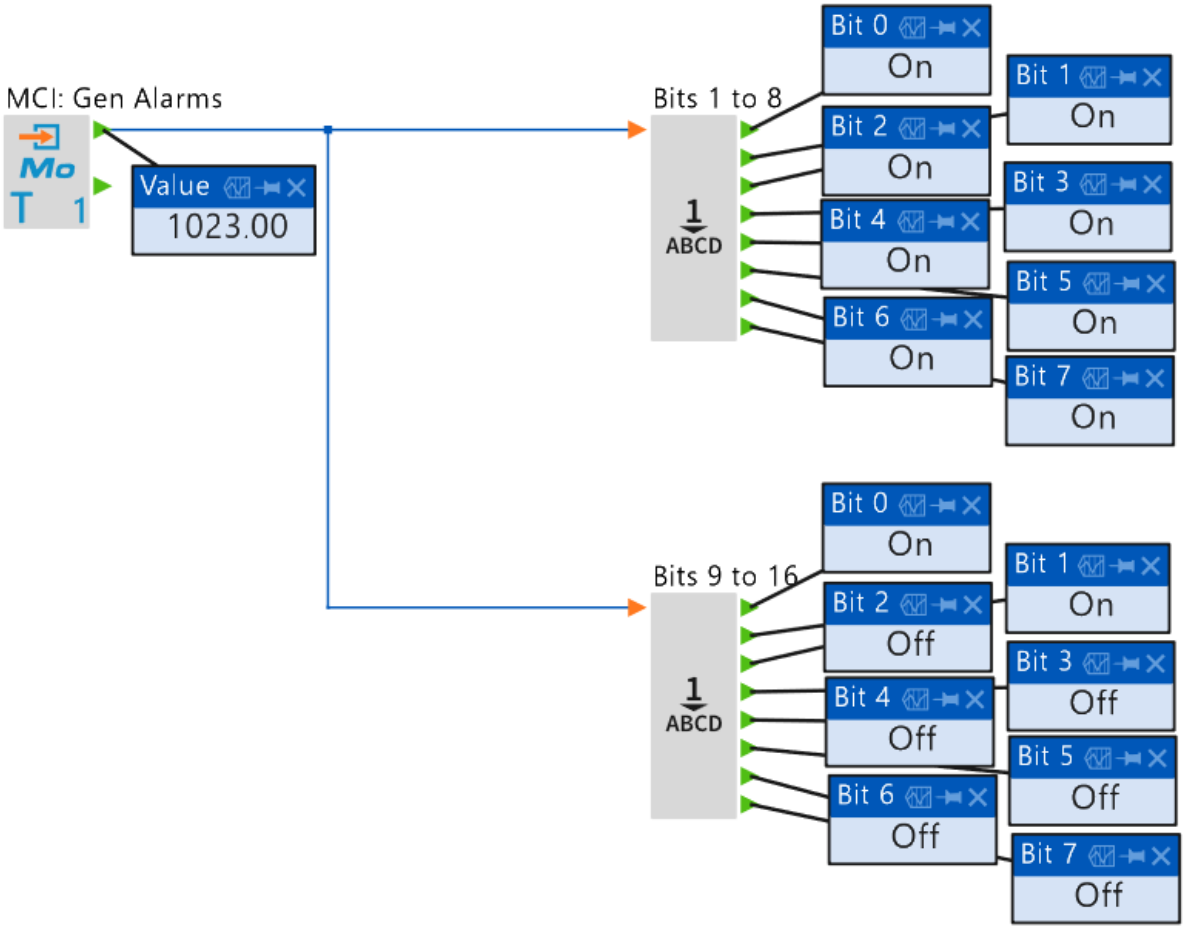
One of the useful applications of Encoder and Decoder blocks is to reduce the quantity of traffic on the network.

This is achieved by encoding up to eight binary values that would need to be transferred to other devices on the global network, sending that value over the global network and then decoding it on the receiving side.

For example, only one Communications Output block needs to be used to transfer the same information that, without using binary encoding, would require eight blocks to be used.

Using binary encoding a network configuration can reduce the traffic on the global network as well as reduce the resources used in the configuration.

Decoding a Modbus Bit Packed Register value



In this example, two Decoders are used to break apart the decimal value to the corresponding binary bit values for a 16-bit number held in the Modbus register.

The lower decoders byte offset is set to 1 to shift the reading to bits 9 to 16.

[Back to block tables page]

DELAY

The Delay block is a conventional delay timer with the provision to set the delay times externally.

NODE DESCRIPTORS

Input: Used to stimulate the Delay block.

Ext. Delay to On: Value specifying how long to wait, after the Input is turned ON, before turning the Output ON.

The units of this delay (hours, minutes or seconds) is set within internal block property On Unit. If this input is not connected, the internal block property Delay to On Value will be used, otherwise this value will override the internal setting.

Ext. Delay to Off: Value specifying how long to wait, after the Input is turned OFF, before turning the Output OFF.

The units of this delay (hours, minutes or seconds) is set within the internal block property Off Unit. If this input is not connected, the internal block property Delay to Off Value will be used, otherwise this value will override the internal setting.

Output: Represents the block state as ON or OFF.

Timer: The amount of time until the Output state becomes ON or OFF.

The units of this value (hours, minutes or seconds) are set in the internal block properties for Time Value Units (On Units and Off Units).

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Delay.

Delay to On: The internal on delay time, which can be overridden by the On Delay input. Delay On Value uses the related internal block setting On Unit for the unit type.

Delay to On Units: Specifies the time units to use for Delay On Value or On Delay input.

Delay to Off: The internal off delay time, which can be overridden by the Off Delay input. Delay Off Value uses the related internal block setting Off Unit for the unit type.

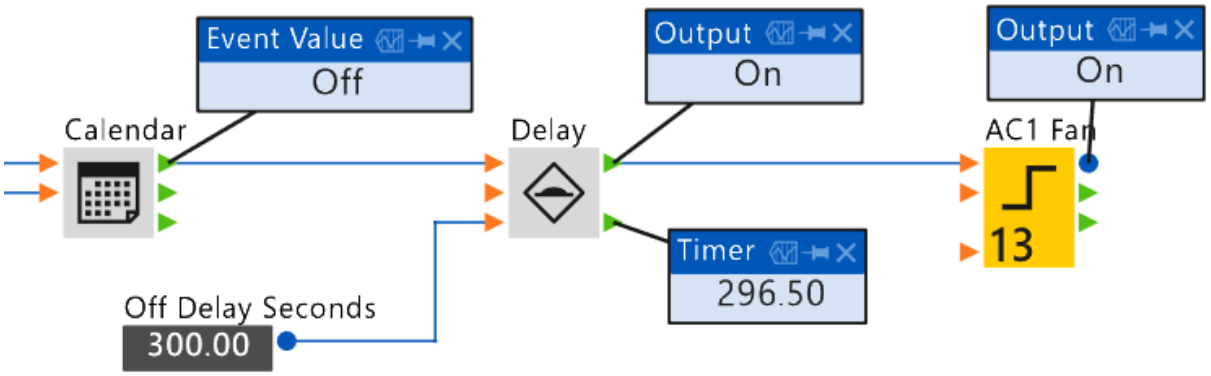
Delay to Off Units: Specifies the time units to use for Delay Off Value or Off Delay input.

EXAMPLE

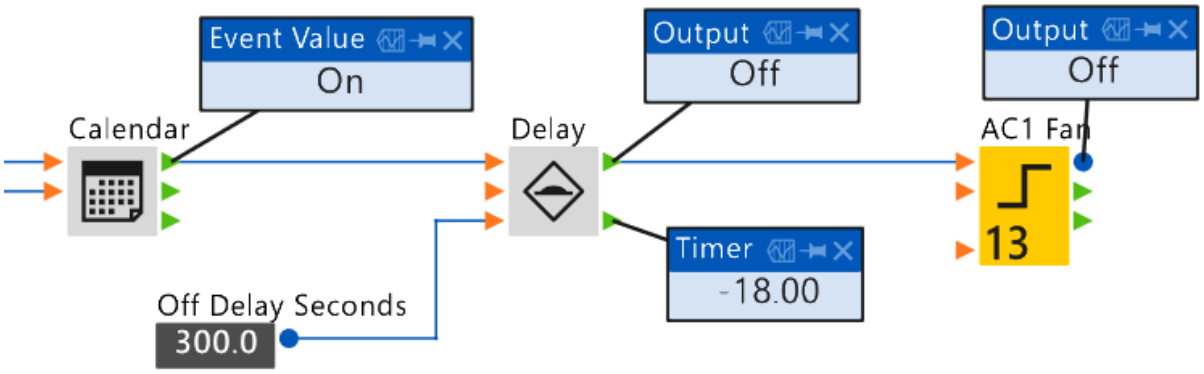
The delay block has been added to provide an off delay timer for when the calendar event transitions to off. The delay time can be user adjustable with the use of a User Variable as shown. Once the off delay has timed out, the fan will control off.

This delay also has an On delay which delays the start of the fan via the On delay time value. The default is 30 seconds.

OFF Delay



ON Delay

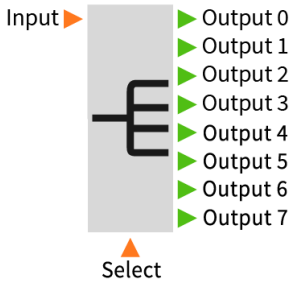


Once the On delay has timed down, the fan will be controlled on.

[Back to block tables page]

DEMULTIPLEXOR

The Demultiplexor block provides a reverse signal switching function, allowing you to switch an input signal between different output points.



NODE DESCRIPTORS

Input: Point input which will be switched to one of the available Output points.

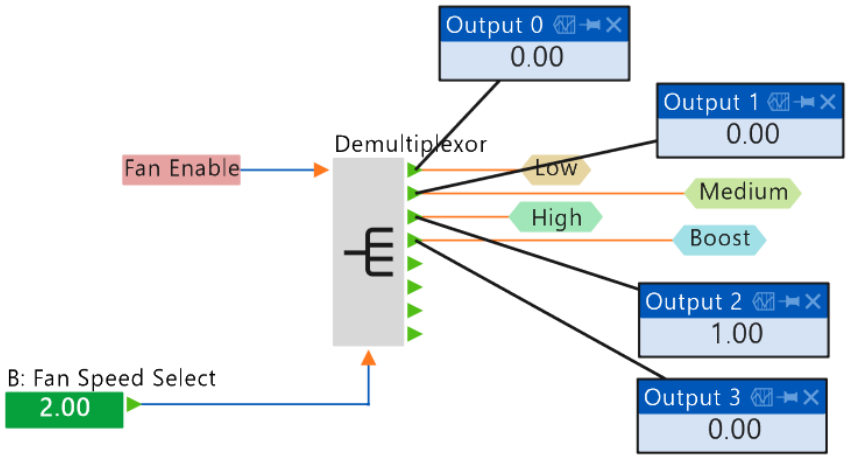
Select: Used to select which of the Output points is mapped to the Input.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Demultiplexor.

EXAMPLE USE

In the example below, the fan speed user variable is used to select the fan speed required. In this case high speed is being selected.



Back to block tables page

DIGITAL INPUT (CONTACT)

Allows you to connect a dry digital contact into the Omni UIO terminal.

NODE DESCRIPTORS

Value: The Output point value of the Digital Input (Contact) block, and will be a digital ON or OFF (a value of 0 or 1 respectively).

Terminal Voltage: The current input voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Digital Input (Contact).

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Back to block tables page

DIGITAL INPUT (VOLTAGE)

Allows you to connect a Digital wet contact into the Omni UIO terminal.

NODE DESCRIPTORS

Value: The Output point value of the Digital Input (Voltage) block, and will be a digital ON or OFF (a value of 0 or 1 respectively).

Terminal Voltage: The current input voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Digital Input (Voltage).

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Back to block tables page

DIGITAL OUTPUT

The Digital Output block provides a digital output signal from the internal processing of the digital controller.

NODE DESCRIPTORS

Value: The Output point value of the Digital Input (Voltage) block, and will be a digital ON or OFF (a value of 0 or 1 respectively).

Enable: Used to enable or disable the block's operation. If the Enable point input is not connected, the block will be enabled by default.

Fault: Sets the internal block property Fault Value as the Output point value.

Output: The value sent to the ASIC.

Terminal Voltage: The current input voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Digital Input (Voltage).

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Disable Value: This value is set to the Output point when the Enable point input is False.

Fault Value: The value that becomes the output value of the block if the FAULT input to the block is set to ON.

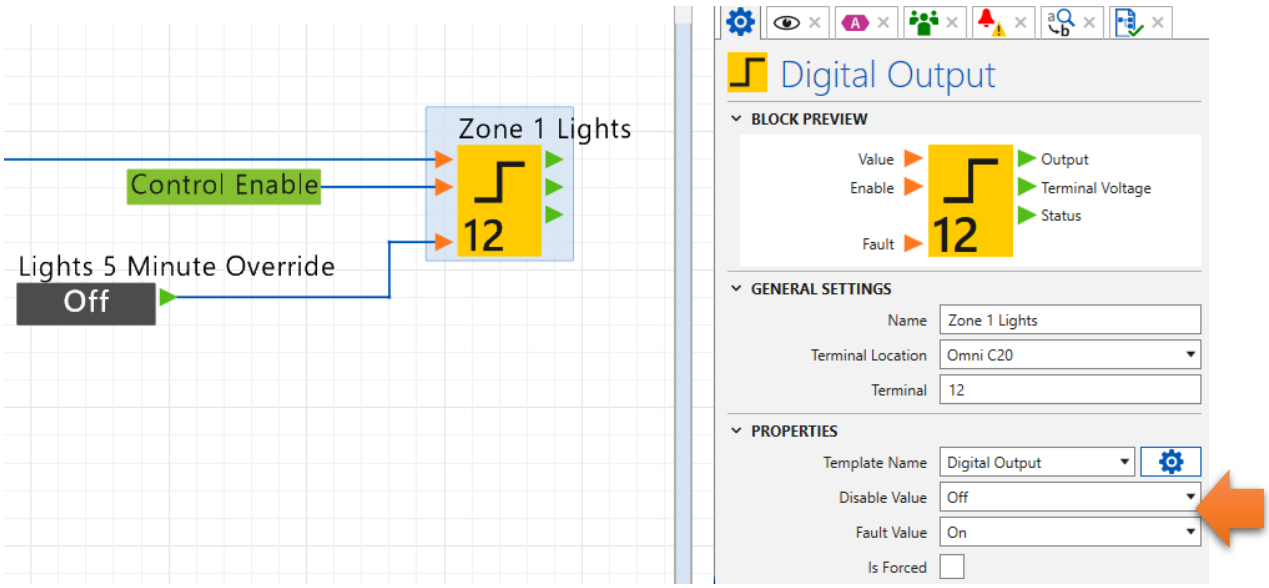
Forced: Set this internal block property to TRUE.

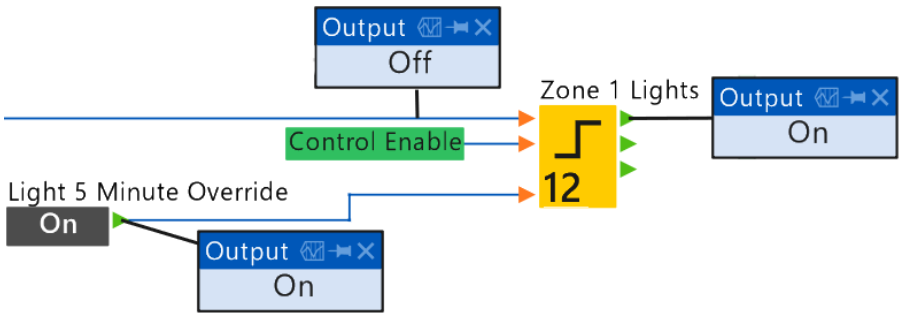
Template information: Switch Mode:

- High Only: 12V when on, floating otherwise
- Low Only: 0V when off, floating otherwise
- Auto (12V when on, 0V when off)

Operation of the Enable and Fault nodes

The **Enable node**, if connected, allows the block to provide a signal at the output node for normal operation. If this enable signal is in an OFF state, then the output signal will be given at the enable state via the blocks properties with OFF being the default. This can be set to ON to switch the output ON when the Enable Node is OFF, although this would interfere with the input node operation if connected.





The **Fault node** allows you to connect any required digital signal to provide an ON or OFF state to switch the Output signal to the required state. The default is set to OFF. In this example this has been set to ON to drive the Lights ON for a period of time via the Pulsed User Variable set at 300 seconds. This will override the Value Node input signal and the Enable Node if connected.

[Back to block tables page]

DUTY CYCLE OUTPUT

The Duty Cycle Output block provides a fixed frequency square wave signal (2 digital states) to the terminals, where the ON / OFF ratio is determined by the internal processing of the digital controller.

NODE DESCRIPTORS

Value: A % value for the block's calculation. Output duty cycle is 0-100%, where properties define 0% = Minimum PWM and 100% = Maximum PWM.

Enable: Used to enable or disable the block's operation. If the Enable point input is not connected, the block will be enabled by default.

Fault: Sets the internal block property Fault Value as the Output point value.

Output: The value sent to the ASIC.

Terminal Voltage: The current output voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Digital Input (Voltage).

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Disable Value: This value is set to the Output point when the Enable point input is False.

Fault Value: The value that becomes the output value of the block if the FAULT input to the block is set to ON.

Forced: Set this internal block property to TRUE.

Template information: Switch Mode:

- High Only: 12V when on, floating otherwise
- Low Only: 0V when off, floating otherwise
- Auto (12V when on, 0V when off)

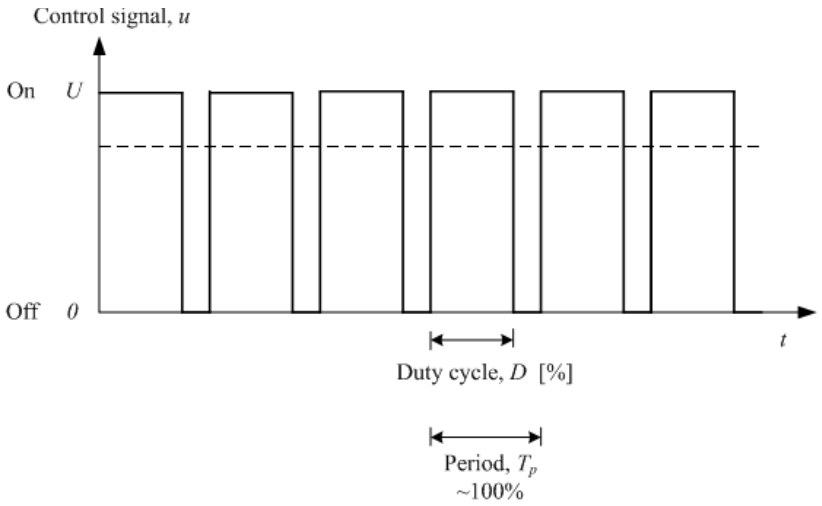
Frequency

- 0.06Hz
- 0.119Hz
- 0.238Hz
- 0.477Hz
- 0.953Hz
- 1.9Hz
- 3.8Hz
- 7.6Hz
- 15.3Hz
- 30.5Hz
- 61Hz
- 122Hz
- 244Hz
- 488Hz
- 976Hz

What is Pulse Width Modulation?

A control system with a pulse width modulated (PWM) control signal is a modulation technique that controls the width of the pulse, formally the pulse duration, based on modulator signal information.

The purpose of PWM is to obtain an approximately continuous or smooth control signal using a binary or on/off actuator or PWM element which typically is an SSR element (solid state relay). The principle of PWM is to keep the element in the on state (and in an off state) so that the resulting average control signal is as specified. PWM elements operate with a given fixed period, Tp, e.g. 1 sec. The part of the period where the PWM element is in an on-state is denoted the duty cycle (D), which is measured in percent.



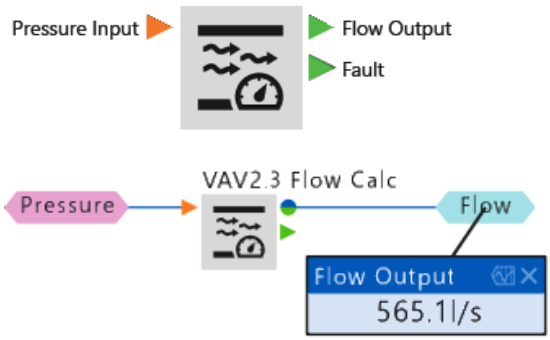
Operation of the Enable and Fault nodes

The **Enable node**, if connected, allows the block to provide a signal on the output node for normal operation. If this enable signal is in an OFF state, then the output signal will be given at the enable percentage value entered via the blocks properties with zero (0) being the default. If set to a value of 30 this will apply a 30% output signal when the enable is off, regardless of what value is on the input Value node.

The **Fault node** allows you to connect any required digital signal to provide an ON state to drive the Output signal to the required percentage. The default is set to Zero (0) but can be any value from 0 to 100%. For example, if this has been set to 100% it will provide a full signal at the output to drive the equipment while the node is ON. This will override the Value Node input signal and the Enable Node if connected.

Back to block tables page

FLOW CALCULATOR



The Flow Calculator block reads the on board differential pressure sensor and calculates the volume of airflow measured.

NODE DESCRIPTORS

Pressure Input: The value input at this node is used with the K-factor setting to calculate the Flow Output.

Flow Output: The air flow value is calculated from the differential pressure sensor readings and K-factor setting.

Fault: Digital Output: Indicates that the Output is outside the Minimum or Maximum Flow Fault Limit properties.

INTERNAL PROPERTIES

Unit: Select from Metric or Imperial units.

Minimum Flow Output (L/s): Specify the Minimum Flow Output.

Maximum Flow Output (L/s): Specify the Maximum Flow Output.

Calibration Settings

Calculate K Factor: Check the box to enable the K-Factor calculation.

Calculated Flow (L/s): Specify the Calculated Flow.

Measured Flow (L/s): Specify the Measured Flow.

K-Factor: Specify a K-Factor value.

Notes

The air flow value is calculated from the differential pressure sensor input and K-factor setting using the following formula:

Flow = K-Factor x SQRT(Pressure in Pa)

Pressure must be in Pascals but flow may be in imperial or metric units depending on the K-Factor.

For example: Flow is 800L/s and Pressure is 140Pa. K-Factor = 800 / SQRT(140) = 67.61
Flow is 1695cfm and Pressure is 140Pa K-Factor = 1695 / SQRT(140) = 143.25

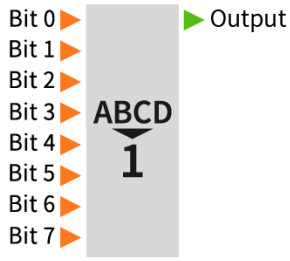
K-Factor Settings: This value is the conversion factor that is used to calculate the flow input from the measured pressure input. It encompasses the pitot tube K-Factor, duct area and unit conversions.

K-Factor = “Pitot Tube K-Factor” x “Duct Area” x “ Conversion Factor” .

Back to block tables page

ENCODER

The Encoder block provides a binary decoding function, allowing you to encode the data from up to 8 binary inputs into one analog value ranging from 0 to 255. The Encoder block is usually used in conjunction with the Decoder block.



NODE DESCRIPTORS

Bit 0..7 Digital Inputs: The four inputs will be converted into an analog value between 0 and 255. If the Inputs are not connected, the Output will default to 0.

Output: The is the result of the binary conversion from the signals provided to the inputs.

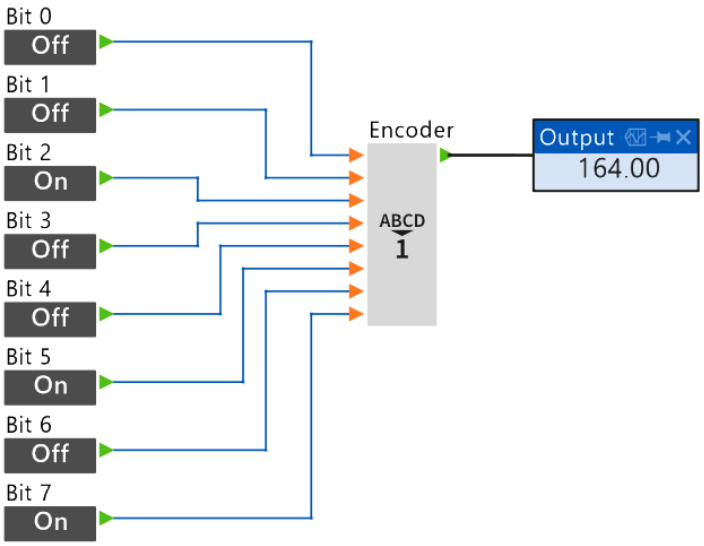
INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Encoder.

Byte Offset: Specify the Byte Offset.

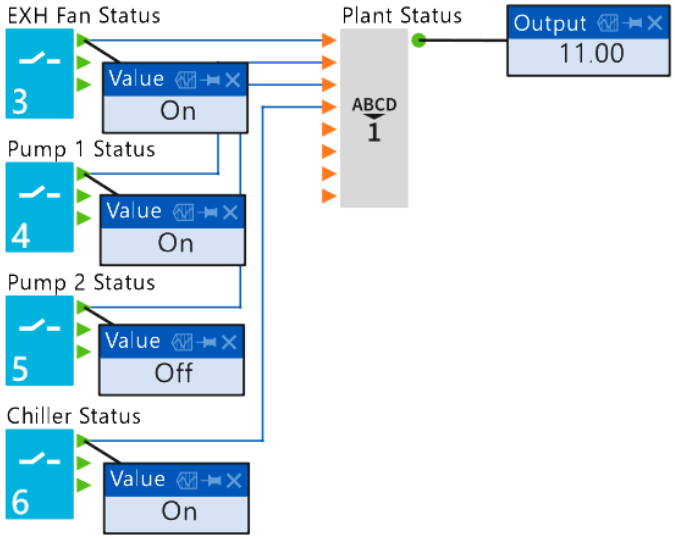
EXAMPLE

In the application below, the encoder provides a decimal value of the binary number input into the input nodes.

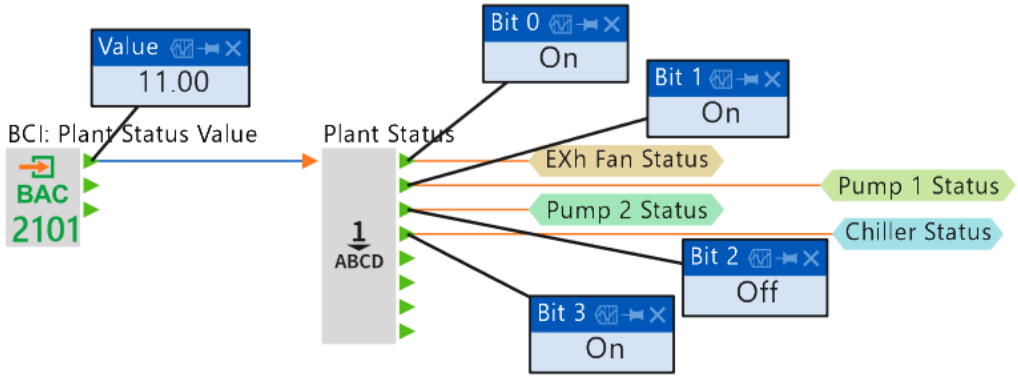


Extended example

The chiller plant Omni has the digital inputs connected to the Encoder block. This provides a binary conversion value for the active bits on the input nodes. In this case, bits 0,1 and 3 are on providing the binary to decimal value of 11 at the blocks output node. This node has a BACnet Protocol Watch added for the receiving Omni, or other BACnet clients to monitor and decode.



The Omni that wishes to decode the encoded value above, simply reads back the BACnet Analog Input that was assigned via the Protocol Watch via the BCI, then decodes this to reveal the bit states as shown. This can then be used as required in the controller logic.



Note that this is fine for monitoring, but be aware that splitting control over multiple controllers can be an issue when the network experiences communications issues or the LAN is severed or down.

[Back to block tables page]

GLOBAL INPUT

The Global Input block will receive and output a global value which has been broadcast by another controller over the network via a Global Output Block either via the RS-485 network or Ethernet via TCP if this option is selected.

NODE DESCRIPTORS

Value: This is the value of the corresponding remote Global Output block. It may be digital or analog - it depends on what the corresponding Global Output block is connected to.

Fault: This is a digital signal indicating that no Global Output point could be found corresponding to this Global Input point. This may be because no such Global Output block exists on the network or because there has been a device or network failure. The VALUE output will be set to the internal Fail Value in this case.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Global Input. Ensure the specified name is 20 characters or less.

Fail Value Type: Digital / Analog.

Fault Value: Fault value that will appear on the VALUE output in the event of a failure to receive this global input.

[Back to block tables page]

GLOBAL OUTPUT

The Global Output block allows you to broadcast any value on the local controller to other controllers on the network either via the RS-485 network or Ethernet via TCP.

100 Innotech Global Outputs can be added to a config per Omni in Focus.

NODE DESCRIPTORS

Value: This is the value of the corresponding remote Global Output block. It may be digital or analog - it depends on what the corresponding Global Output block is connected to.

Fault: This is a digital signal indicating that no Global Output point could be found corresponding to this Global Input point. This may be because no such Global Output block exists on the network or because there has been a device or network failure. The VALUE output will be set to the internal Fail Value in this case.

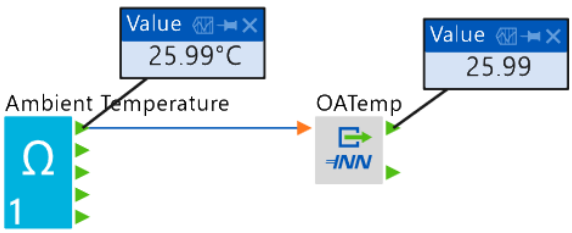
INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Global Input. Ensure the specified name is 20 characters or less.

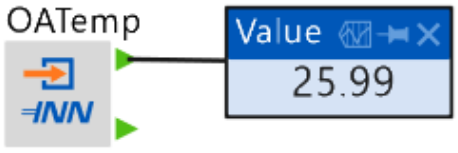
Fail Value Type: Digital / Analog.

Fault Value: Fault value that will appear on the VALUE output in the event of a failure to receive this global input.

EXAMPLE



Controller #1 transmits the Value



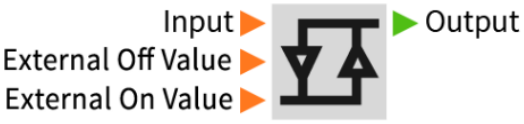
Controller #2 Receives the Value

In this example, the ambient temperature is being broadcast over the network to all other controllers that wish to receive this value. **Note that the label used for the Global Input, must be identical to the Global Output.**

[Back to block tables page]

HYSTERESIS

The Hysteresis block performs the function of a conventional staging relay. It produces a digital ON output when the input value moves past the On Value, and a digital OFF when the input value moves past the Off Value. The On Value and the Off Value are set internally but can be overridden with external values.



NODE DESCRIPTORS

Input. An **analogue input** value used as the basis of the hysteresis comparison.

External Off Value. An analogue input value used as the hysteresis Off Value. If this is connected, it overrides any internal Off Value setting.

External On Value. An analogue input value used as the hysteresis On Value. If this is connected, it overrides any internal On Value setting.

OUTPUT. This is the result of the hysteresis comparison, which is a digital ON or OFF value.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Hysteresis.

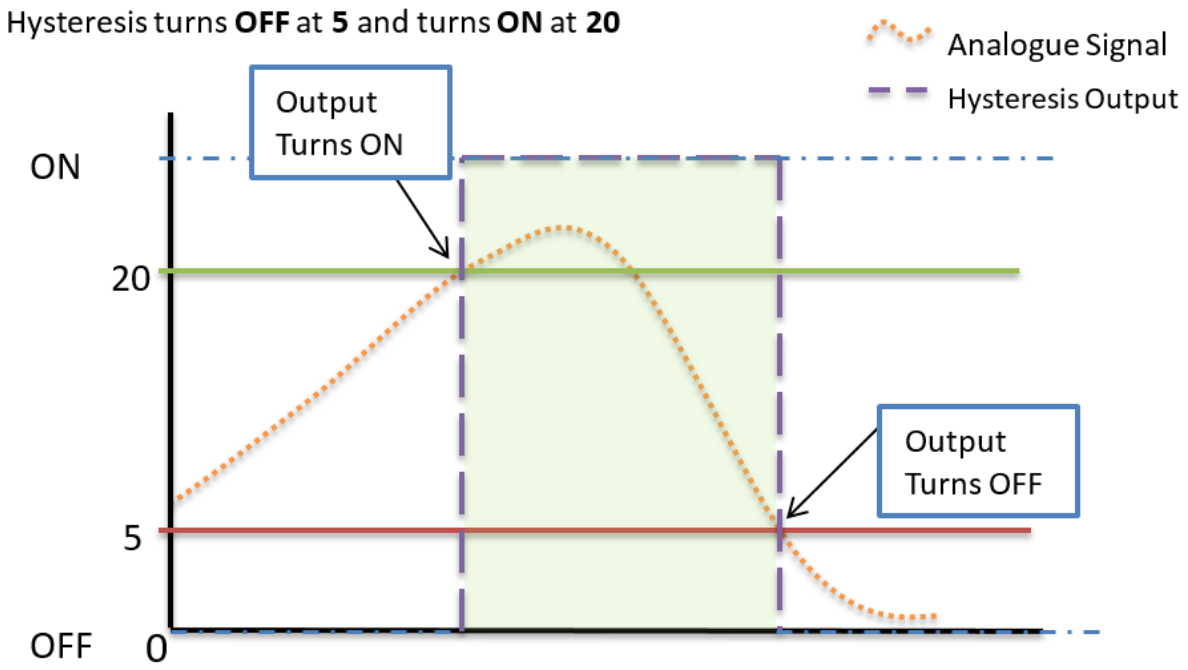
Off Value: The internal hysteresis OFF value which may be overridden by the input Ext Off, if that input is connected.

On Value: The internal hysteresis ON value which may be overridden by the input Ext On, if that input is connected.

When used with a 0 to 100 % signal, such as a PID loop signal, the **Output turns off at must be set above zero**, and the **Output turns on at, must be set below 100**. The input signal must exceed or drop below these values to function.

EXAMPLE USAGE

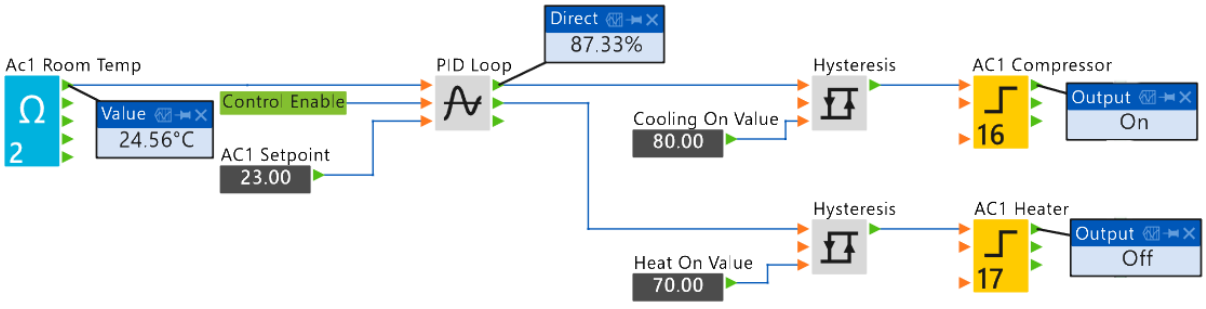
In this example the Off value is set to 5 and the on value is set to 20.



Note for this example, that the ON level if set to 20, would need to be exceeded to actually change the output to an ON state. The signal would need to drop below the OFF level of 5 to change the output to an OFF state.

PRACTICAL EXAMPLE

In the example below, the Hysteresis blocks are used to convert the analog PID signal value to a digital On/Off value at the set limits. In this case, the user is able to set the On values via the User Variables connected to the external value nodes of the Hysteresis blocks. The off levels have been set to 15 for both blocks.



[Back to block tables page]

ISS INPUT

The ISS Input block will receive and output a value which has been received via the ISS Terminal on the controller. Only for use with Omni C20 and C40 controllers.

NODE DESCRIPTORS

Value: This is the value of the input selected by the properties.

Fault: The Fault indicates the communications status with the BACnet device. 0 = OK / 1 = Error / 2 = Device Offline / 3 = Internal Error.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is ISS Comms Input.

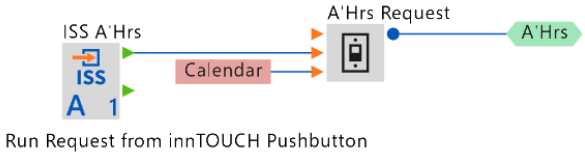
Device Type: Select from Unassigned or a specific ISS device that has been added to your configuration's network.

Input Type: Select an input type based on the ISS model you have selected in the Device Type combo.

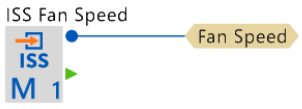
- ISS01 - Select from After Hours, Fan Speed, Temperature or Set Point.
- ISS02 - Select from After Hours, Fan Speed, Temperature or Set Point.
- ISS03 - Select from After Hours, Fan Speed, Temperature, Set Point or one of four Universal Inputs.
- ISS04 - Select from After Hours, Fan Speed, Temperature, Set Point or one of four Universal Inputs.

Fault Value: Fault value that will appear on the VALUE output in the event of a failure

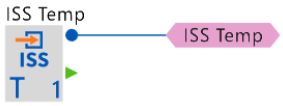
EXAMPLE USAGE



Run Request from innTOUCH Pushbutton



Fan Speed Selection from innTOUCH Pushbutton



Temperature from innTOUCH

Back to block tables page

ISS COMMS OUTPUT

The ISS Output block will send a value to the ISS Terminal on the controller. Only for use with Omni C20 and C40 controllers.

NODE DESCRIPTORS

Value: This is the value to be written to the ISS output.

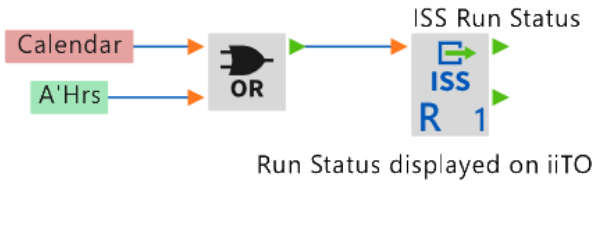
Fault: The Fault indicates the communications status with the BACnet device. 0 = OK / 1 = Error / 2 = Device Offline / 3 = Internal Error.

INTERNAL PROPERTIES

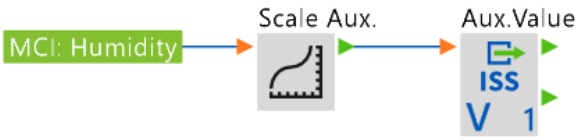
Device Type: Select from Unassigned or a specific ISS device that has been added to your configuration's network.

Output Type: Select an output type from the combo. Select from Disable, Run Status, Set Point or Auxiliary Value.

EXAMPLE USAGE



Run Status displayed on iiTOUCH



innTOUCH secondary display. Scale for imperial, CO2, Hum, Pressure etc
In this case 0-100 = 0-100 (Set Uni-curve and ISS Min the same)

Back to block tables page

LARGE ANALOG VALUE

The Large Analog Value block defines a standardised BACnet object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Large Analog Value object to make any kind of double-precision data value accessible to other BACnet devices.

This block is typically used to provide a 64bit value to a BACnet client.

NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active, and the Input value is written to the object at the highest Write Priority.

Ext. Write Priority: This determines the position in the Priority list (Known as the Priority Array) that the input value will be written to. (This equates to a value from 1-16, where 1 is the highest priority) The Input value will be written at the Ext. Write Priority value on a change in value of the Input, or a change in value of the Ext. Write Priority input.

Ext. Relinquish: The Ext. Relinquish node is a digital signal applied to delete a value that has been written to the object at a particular priority. When using the Ext. Relinquish node, you must also indicate the Priority of the value to be relinquished. This is achieved by setting a value from 1-16 on the Ext. Relinquish Priority node. Therefore, you should set the Ext. Relinquish Priority first, and then pulse the digital signal on the Ext. Relinquish node from “Off” to “On”.

Ext. Relinquish Priority: The Ext. Relinquish Priority input allows a user to select any priority (and value) to be relinquished from the Priority Array.

Enable: The Enable input of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Present Value: The Present Value displays the value that was written at the highest Write Priority, either from the input or from a BACnet client. If all priorities have been relinquished, the Relinquish Default internal property will become the Present Value.

Active Priority: The Active Priority indicates the highest priority that was written to the object. (1-16) If all priorities have been relinquished, the Active Priority will be 0.

Fault: The Fault output will be FALSE whenever the Reliability output has a value of NO_FAULT_DETECTED, otherwise it will be TRUE.

Reliability: The Reliability output for this type of object indicates the following:
2 = Over the pre-set Maximum Present Value.
3 = Under the pre-set Minimum Present Value.

Out Of Service: The value of this property indicates whether or not the object is currently being overridden by a BACnet Client.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Large Analog Value.

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Units: The Units property of the object allows you to select one of the standard BACnet units.

Minimum Present Value: The Minimum Present Value sets the lower end of the desired Present Value output for alarm monitoring.

Maximum Present Value: The Maximum Present Value sets the upper end of the desired Present Value output for alarm monitoring.

Relinquish Default: The Relinquish Default setting is the default value the Present Value will display when all Write Priorities are relinquished.

Change Of Value Increment: The Change of Value (COV) setting allows you to optimise the amount of BACnet traffic to and from your BACnet device.

Write Priority: The Write Priority internal property indicates the priority at which the Input value will be written. This property is only used when there is no connection to the Ext. Write Priority input.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

Back to block tables page

LATCH

The Latch block provides a general-purpose digital latching mechanism with a Reset/Toggle mode and a built-in timed reset facility. In the Reset mode, the Latch block will latch the INPUT, and will only reset when the RESET input is ON. In the Toggle mode, the Latch block toggles the OUTPUT ON and OFF alternately when it receives an input pulse. If you enable the internal timer, the output will turn OFF after the set period, regardless of whether the block is in Reset or Toggle mode. When running, the timer value is output as the number of time units until the timer expires.

NODE DESCRIPTORS

Ext Value: Provides an override to the internal block property Timer Value. The internal block property Time Value Unit determines the time units to use (hours, minutes or seconds). If Ext Value is not connected, the internal block property Timer Value will be used instead.

Input: Used to latch the Output ON or OFF, depending on the internal block properties. The Toggle mode of the block relies on the Input turning OFF before it turns back ON again to toggle the Output state.

Reset: Can be used to keep the Latch block disabled. Whenever the Reset input is ON, the block timer is set to zero and the Output is set to OFF. This input may, for example, be connected to the output of the Daily Schedule, so that the Latch block is disabled whenever the schedule is currently running one of its schedules.

Output: Indicates the active ON or OFF state of the block.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Latch.

Mode: This enables you to set the block into Toggle mode, Reset mode or Remember Value (see below Usage Notes for details).

Reset Value: This is the internal timer setting. When latched, the block will turn on the Output for this amount of time, unless the input Ext Value is connected which will override this internal value. The units of the time value are set by the Time Units property. If the Timer Value is set to zero, then the timer will not take effect and the Output will remain ON unless toggled OFF or Reset.

Time Units: This field selects the units (hours, minutes or seconds) which apply to the internal block property Timer Value, or if connected to the input External Value.

Initial Value: [Remember Value Mode Only] Specify the initial value.

Notes.

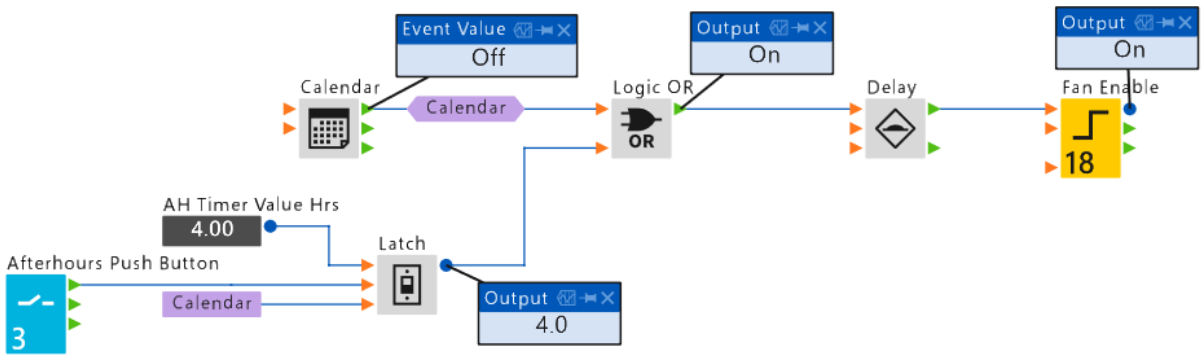
- An input pulse refers to the INPUT turning OFF before turning back ON again.
- If you set the Timer Value to zero, the timer will have no effect on the OUTPUT of the block, and the block can then be used as a conventional toggle or latching mechanism.
- Each time a pulse is received on the input, the internal timer is reset regardless of the current Mode of the block (Toggle or Reset).
- In Reset mode, the output will latch ON when an INPUT pulse is received. If the Timer Value is greater than zero, the OUTPUT will turn OFF after this time, unless another pulse was received at the input (which starts the timer from the beginning). This means that once latched, the OUTPUT will stay on as long as input pulses are received at the INPUT before the time is up.
- In Toggle mode, beware if you have a Timer Value set. For example, you may expect that after toggling the OUTPUT ON, the next input pulse will toggle the output OFF. But if the timer expired in the meantime which turns the OUTPUT OFF, then the next input pulse will actually toggle the OUTPUT back ON again!

EXAMPLE

In this example below we have a classic use of the Latch block as an afterhours run timer in an air-conditioning control system. The latch has been configured with a Timer Value of 4 hours. When the afterhours **Input #3** is triggered (momentary), the **Latch Output will turn ON (1)** for the **Timer Value**. Once the Timer has reached the Value, after 4 hours, the **Latch Output will turn off**.

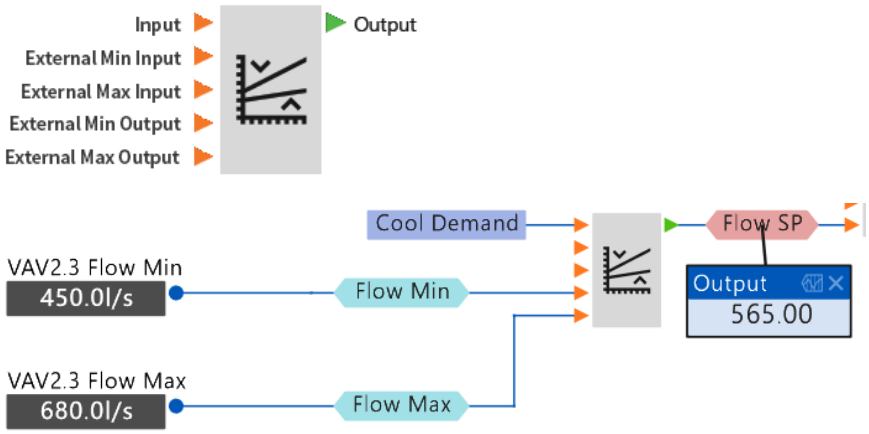
If the AC1 Weekly schedule is active, the Latch resets and the output will turn OFF (0), even if the Timer Value is currently counting as the time schedule becomes active. This could happen if the start time of the schedule is set to 7:00 am and the afterhours push button was pressed at 6:00 am. The Latch counter would have only counted 60 minutes but will be cancelled due to the active Weekly schedule start at 7:00 am.

The above description is true if the Latch is set to Reset mode.



Back to block tables page

LINEAR SCALE



The Linear Scale block calculates the output values for the given input values using a 2-point unicurve, where the maximum and minimum range of inputs and outputs can be defined by the internal block properties.

Essentially, when an input value such as the cooling demand is input into the block, it will be scaled within the min and max values to provide a flow setpoint output.

NODE DESCRIPTORS

Input: Provides an Input Value to the block.

External Min Input: Provides an external Minimum Input value to the block, which overrides any value which may be present in the block properties.

External Max Input: Provides an external Maximum Input value to the block, which overrides any value which may be present in the block properties.

External Min Output: Provides an external Minimum Output value to the block, which overrides any value which may be present in the block properties.

External Max Output: Provides an external Maximum Output value to the block, which overrides any value which may be present in the block properties.

Output: For input values within the defined input range, the Output will be the proportionally scaled value within the output range.

[Back to block tables page]

LOG

The Log block allows data logging of input signals to non-volatile RAM on the controller, which may be extracted and analysed at any time. **Note that 250 Log blocks can be added per Omni.**

NODE DESCRIPTORS

Input: This input type is configured using internal block property Is Digital Logging and is the value which will be logged (along with a time/date stamp).

Enable: Input point used to enable the Logging block. If this is not connected, then the Log block will be enabled by default.

Trigger: Input pulse (transition from OFF to ON) used to force an immediate logging of the Input regardless of the current interval time.

The interval time is reset when a log is triggered. For example, if the interval time is one minute and a log is triggered, then the next log event will happen one minute from when the triggered log occurred (unless another triggered log happens in the meantime).

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Log.

Logging Type: This tells the log block whether or not it is logging a digital or an analog value.

If you select the Is Digital Logging check box, then the log block will make an entry in the log whenever the Input changes state. That is, if the Input turns OFF, a log entry will be made, and if it turns ON, a log entry will also be made. If you do not select the Is Digital Logging check box, then the log block will only make an entry when it is triggered or when the time interval has elapsed.

Logging Frequency: [Analog Type Only] The unit of time for the logging. This is used in conjunction with the Interval time. This can be set to seconds, minutes, hours, days or auto. When set to Auto, the Tolerance value specifies when a new value will be logged. A new value is logged if the value change is greater than or equal to the Tolerance specified. However, if the Tolerance is 0, a new value is only logged if the value change is greater than the Tolerance only.

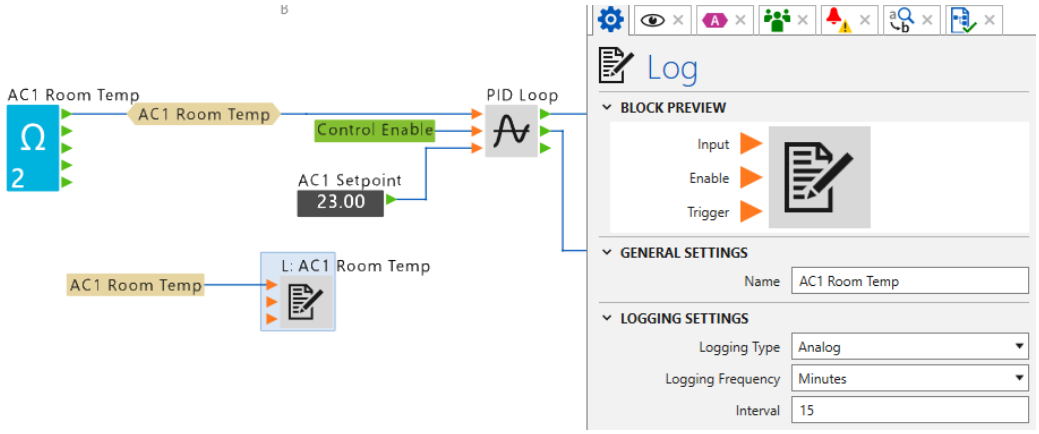
Interval: [Analog Type Only] This is the interval between logging events.

Interval defines a logging event interval. If the block's Input and Trigger inputs are not connected, then no interval logging will occur. If you do not want any interval-based logging, choose an interval of 0 - then logging will only occur on a triggered event or a digital value event.

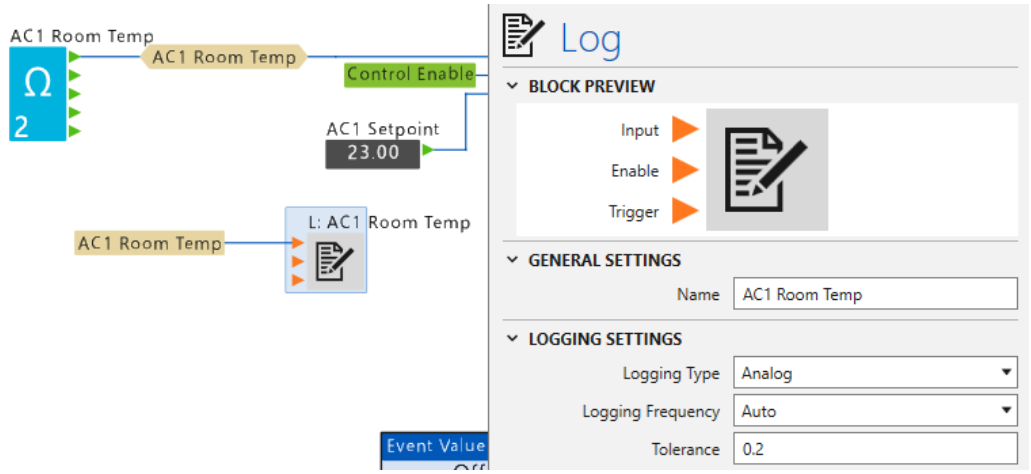
The Log block will automatically log on the hour IF the logged value has not changed in the previous 59 minutes.

EXAMPLES

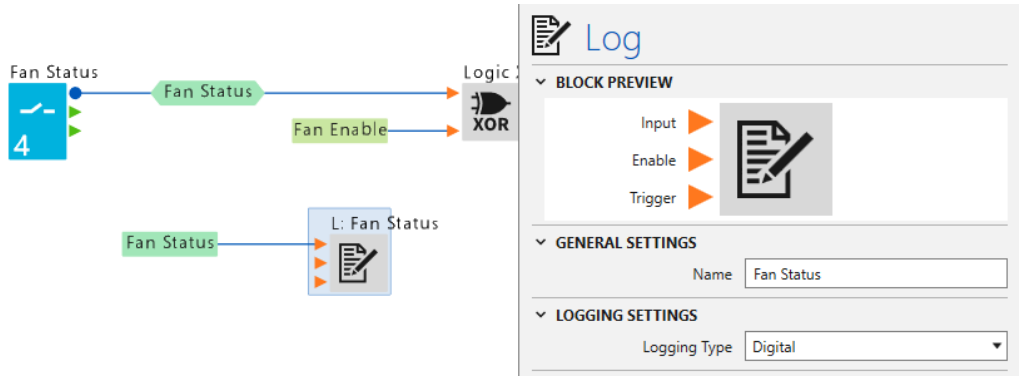
In the example below, the log block is set to log the room temperature every 15 minutes.



The log can also be set to auto and a tolerance specified. In this case the tolerance is set to 0.2. as the temperature changes by more than the set tolerance, the value will be logged.



For a digital value, the log is set to digital and logs the fan status on a change of state.



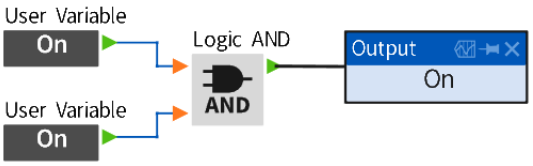
Back to block tables page

LOGIC AND

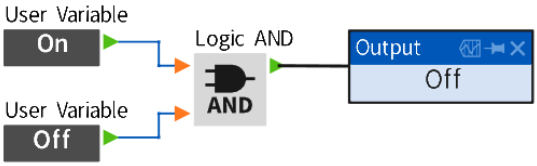
The Logic AND block provides a logical AND function based on two digital inputs. Both input A and B need to be on to provide an ON state at the output.

EXAMPLES

Logic AND - On



Logic AND - Off

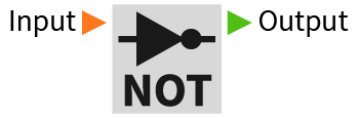


Back to block tables page

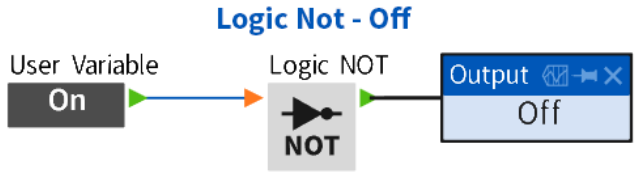
LOGIC NOT

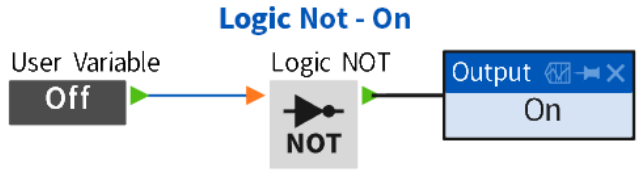
The Logic NOT block provides a logical inversion function for a single digital input.

Any signal connected to the input will be inverted at the output node.



EXAMPLES





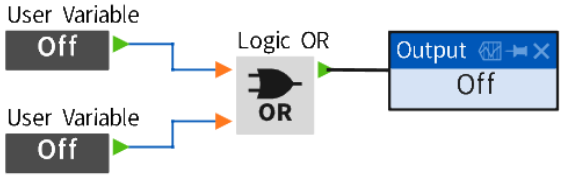
Back to block tables page

LOGIC OR

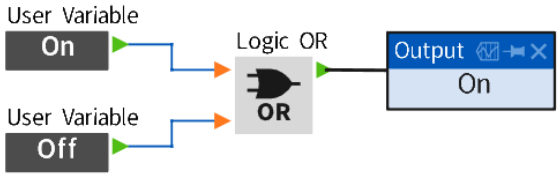
The Logic OR block provides a logical OR function based on two digital inputs. Either Input A OR Input B OR Both need to be on to turn the Output on.

EXAMPLES

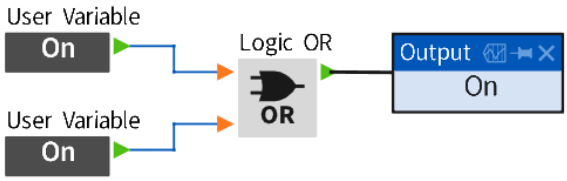
Logic Or - Off/Off Input



Logic Or - On/Off Input



Logic Or - On/On Input



Back to block tables page

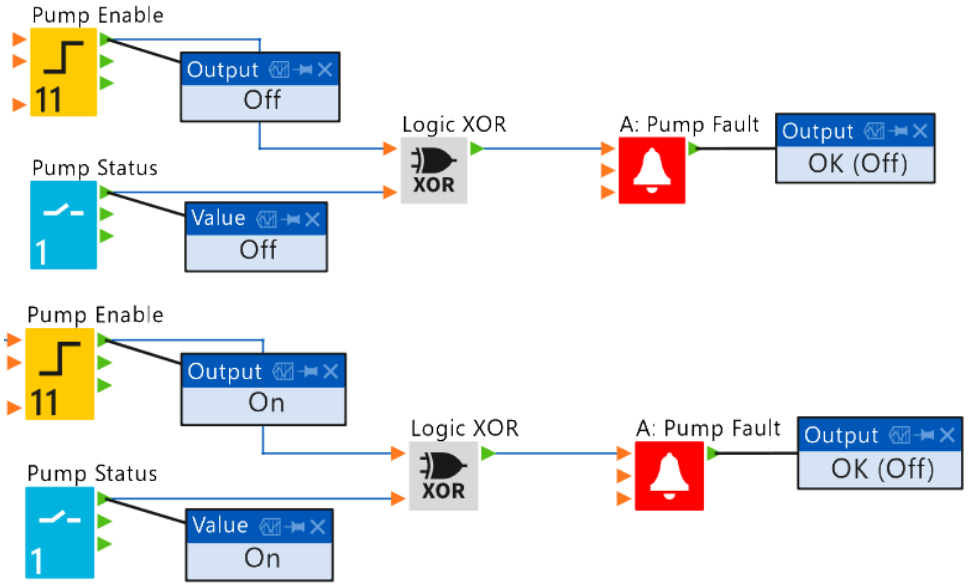
LOGIC XOR

The Logic XOR block provides a logical exclusive XOR function based on two digital inputs.

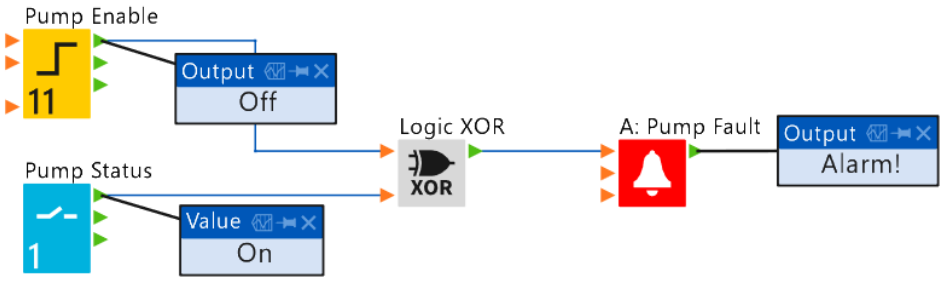
CONFIG EXAMPLES

The XOR is typically used as a mismatch state to monitor equipment control as shown in the examples below.

If both XOR inputs are in the same state, such as in this example both are OFF or ON, then the Alarm will not activate.



When either input is does not match the state of the other input, as in this example, the enable is OFF but the status is ON, thus the Alarm is activated. This is also true in the reverse state if the enable is On and the status is OFF an Alarm is also Activated.



Back to block tables page

LOOK UP

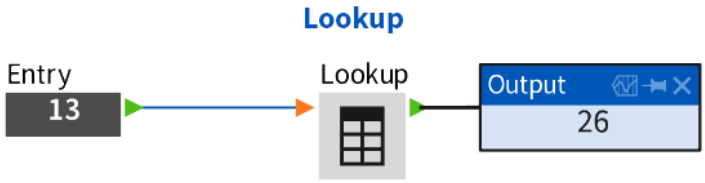
The Lookup block provides a 32-value lookup table which allows you to reference a valve in the table to the entry value. **Note that 50 Lookup blocks can be added per Omni.**

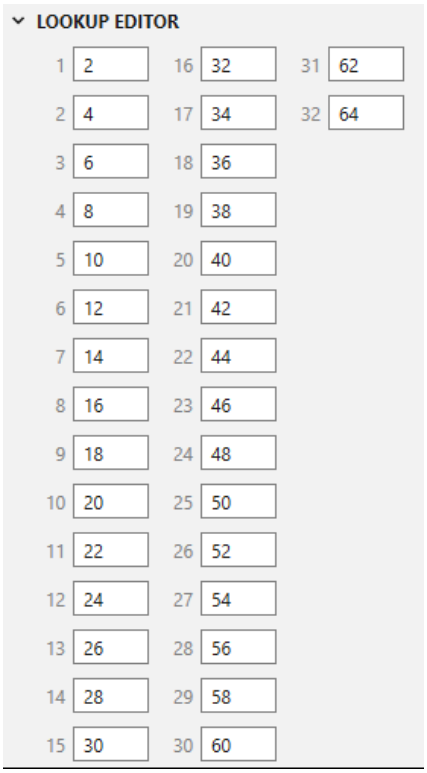
NODE DESCRIPTORS

Entry: input used to specify which of the 32 entries from the Lookup Table is to be used as the Value output. The Entry input must be from 1 to 32; any other number will produce an output Value of 0.

Output: Point which is the value of the Lookup Table entry for the Entry input value.

EXAMPLE





The Lookup block properties lookup editor determines the output based on the input. Up to 32 input values and be referenced via the input value.

The input value is based on whole numbers only.

Back to block tables page

LOOP BLOCK (BACNET)

The Loop Block defines a standardised BACnet object whose properties represent the externally visible characteristics of any form of feedback control loop.

NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active.

Enable: The Enable input of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value will cause a new value to be written to the Present Value only if the Enable Input is active.

Ext. Set Point: External setpoint for the PID control loop (overrides block setpoint property if connected)

Present Value: The Present Value displays the value that was written either from the input or from a BACnet client.

Fault: The Fault output will be FALSE whenever the Reliability output has a value of NO_FAULT_DETECTED, otherwise it will be TRUE.

Reliability: The Reliability output for this type of object

Out Of Service: The Out of Service output indicates the override state of the object. (TRUE or FALSE) BACnet clients write directly to the Out of Service property to disable local control of the object, therefore disconnecting the Input from the Present Value and only allowing control via the BACnet network. (The block Input node is ignored whenever the Out of Service output is TRUE)

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Loop Block.

Set Point: The required temperature that should be reached at the occupancy time.

Allow Override: Read/Write – The Allow Override pushbutton dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override pushbutton is ON, the Out of Service property of the BACnet object is writable from other BACnet clients. If the pushbutton is OFF, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value.

Alpha: Percentage of difference added to the previous value to generate the new one.

Object Instance: Unique BACnet address identifier for the object.

AI Object Instance: Control value object instance.

AO Object Instance: Manipulated value object instance.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Output Units: Select the output units.

Action: Select Direct or Reverse.

Proportional Constant: Specify the Proportional Constant Value.

Units: Select a unit of measure.

Derivative Constant: Specify the Derivative Constant Value.

Units: Select a unit of measure.

Bias: Specify a Bias value.

Maximum Output: Specify a maximum output value.

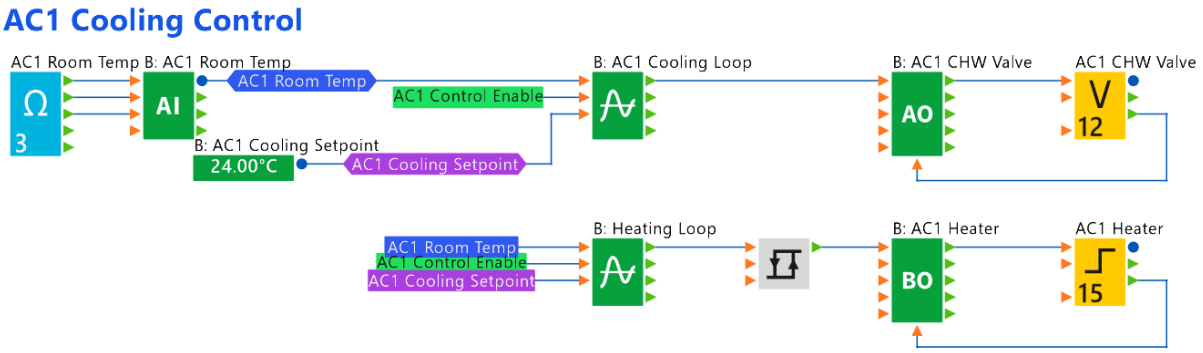
Minimum Output: Specify a minimum output value.

Change Of Value Increment: The Change of Value (COV) setting allows you to optimise the amount of BACnet traffic to and from your BACnet device.

Dead Band: Specify a dead band value.

EXAMPLE

In this config, the BACnet PID loops are used to provide cooling and heating for this unit. **Note that two loops must be used as each can only be a cooling type or heating type.**



Back to block tables page

MODBUS COMMS INPUT

The Modbus Comms Input allows you read back a Modbus register value on a slave device connected to the RTU or TCP networks. **Note that a total of 250 MCI or MCO blocks can be added per Omni and can be a mix of both.**

NODE DESCRIPTION

Value: This is the value of the register as read from the Modbus slave device.

Fault: This provides an error code indicating the reason for a failed read.

- 00 - No Fault
- 01 - Illegal Function (function code not supported by slave device)
- 02 - Illegal Address (address not valid for slave device)
- 03 - Illegal Data Value (data value not valid for slave device)
- 04 - Slave Device Failure
- 05 - Acknowledge (slave device is processing request)
- 06 - Slave Device Busy
- 07 - Negative Acknowledge
- 08 - Memory parity error
- 09 - No response from slave (after timeout and retries - not an exception code

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Modbus Comms Input.

Protocol Type: TCP / RTU

Slave Device Identifier: Enter a value for the to identify the Slave Device.

IP Address: [TCP Protocol Type Only] Enter an IP address.

Port: [TCP Protocol Only] Specify a port number.

Modbus Function: Choose from Read Discrete Coil (0x01) / Read Discrete Input (0x02) / Read Holding Register (0x03) / Read Input Register (0x04).

Register Address: Specify a register address.

Data Type: The items available here are dependent on the value selected in the Modbus Function combo.

Data Length: This is a fixed informational value that will change depending on the selection in the Data Type combo.

Swap 16: This item is active or inactive depending on the selection in the Data Type combo.

Swap 32: This item is active or inactive depending on the selection in the Data Type combo.

Swap 64: This item is active or inactive depending on the selection in the Data Type combo.

Multiplier: Specify a multiplier value.

Offset: Specify an offset amount.

Exponent: Specify an exponent value.

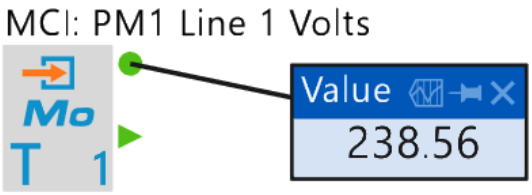
Polling Period: Specify a polling period. The polling period is the number of seconds to wait between requests.

Fault Value: Fault value that will appear on the VALUE output in the event of a failure.

EXAMPLE

The MCI block is configured to read back the voltage register for the slave device on the TCP network.

The value node has a BACnet Protocol watch to expose the register value to the BACnet network.



Back to block tables page

MODBUS COMMS OUTPUT

The Modbus Comms Output allows you to write to a slave device register with the required value. The slave device can be on the RTU or TCP networks. **Note that a total of 250 MCI or MCO blocks can be added per Omni and can be a mix of both.**

NODE DESCRIPTION

Value: This is the value to be written to the Modbus slave device.

Fault: This provides an error code indicating the reason for a failed write.

- 00 - No Fault
- 01 - Illegal Function (function code not supported by slave device)
- 02 - Illegal Address (address not valid for slave device)
- 03 - Illegal Data Value (data value not valid for slave device)
- 04 - Slave Device Failure
- 05 - Acknowledge (slave device is processing request)
- 06 - Slave Device Busy
- 07 - Negative Acknowledge
- 08 - Memory parity error
- 09 - No response from slave (after timeout and retries - not an exception code

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Modbus Comms Output.

Protocol Type: TCP / RTU

Slave Device Identifier: Enter a value for the to identify the Slave Device.

IP Address: [TCP Protocol Type Only] Enter an IP address.

Port: [TCP Protocol Only] Specify a port number.

Modbus Function: Choose from Read Discrete Coil (0x01) / Read Discrete Input (0x02) / Read Holding Register (0x03) / Read Input Register (0x04).

Register Address: Specify a register address.

Data Type: The items available here are dependent on the value selected in the Modbus Function combo.

Data Length: This is a fixed informational value that will change depending on the selection in the Data Type combo.

Swap 16: This item is active or inactive depending on the selection in the Data Type combo.

Swap 32: This item is active or inactive depending on the selection in the Data Type combo.

Swap 64: This item is active or inactive depending on the selection in the Data Type combo.

Multiplier: Specify a multiplier value.

Offset: Specify an offset amount.

Exponent: Specify an exponent value.

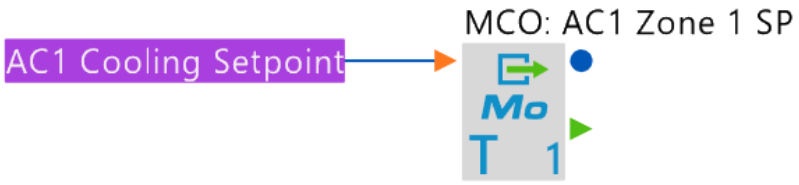
Change of Value Tolerance: Specify a value for the change of tolerance for this block.

Enable Minimum Write: Check the box to specify a minimum write period.

Minimum Write Period: [Available only when Enable Minimum Write is checked] Specify a value.

EXAMPLE

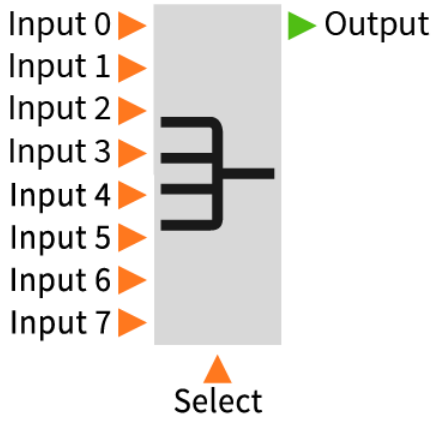
In this example, the User Variable is being used to write a setpoint value to the Modbus slave register for zone 1.



Back to block tables page

MULTIPLEXOR

The Multiplexor block provides a signal switching function, allowing you to switch the Output of any one of a set of analog Input signals.



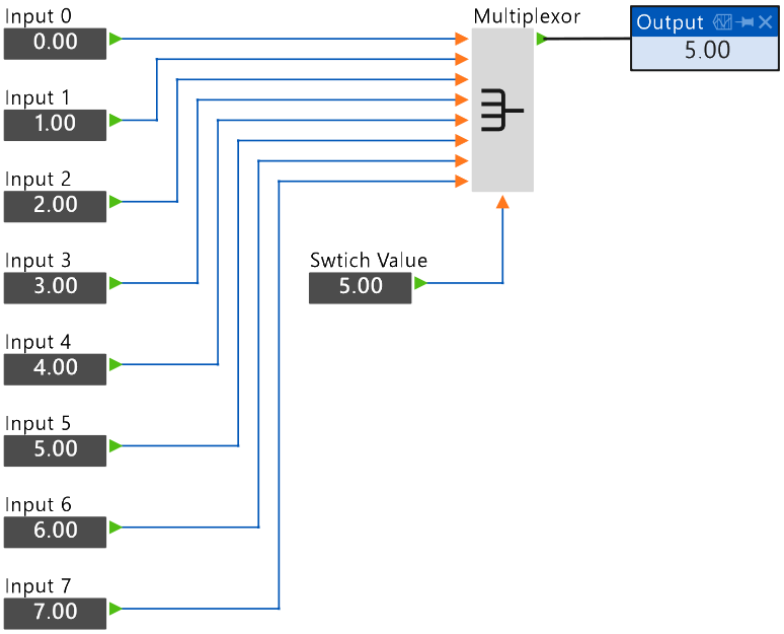
NODE DESCRIPTORS

Input 0..7: Point inputs, one of which will be switched to the Output based on the value of the Control input.

Select: Point input which is used to select which of the Input points is mapped to the Output.

Output: Point which is available to be mapped to the Input signal. The activated Input can be selected using the select input value.

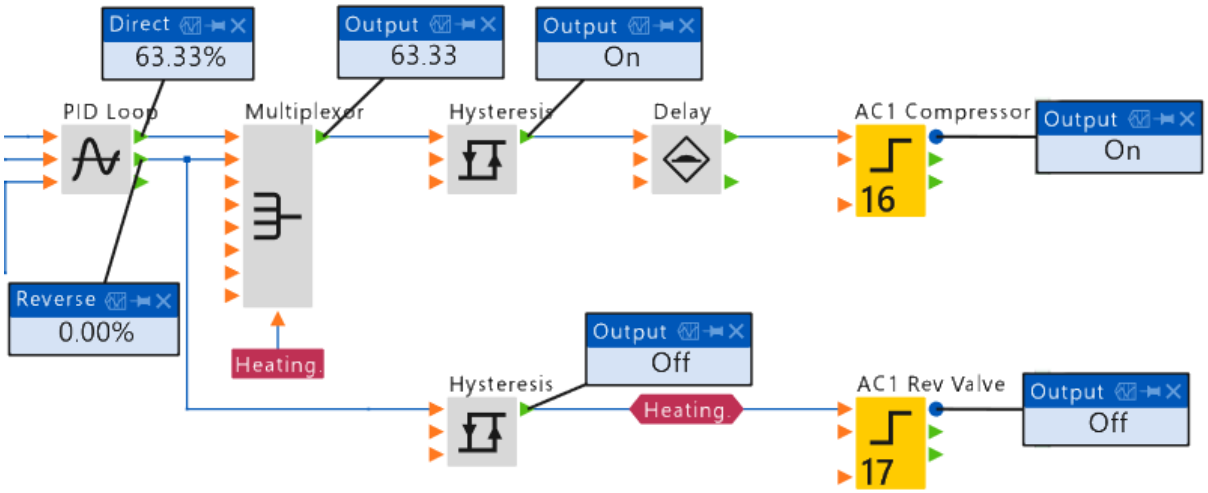
EXAMPLE



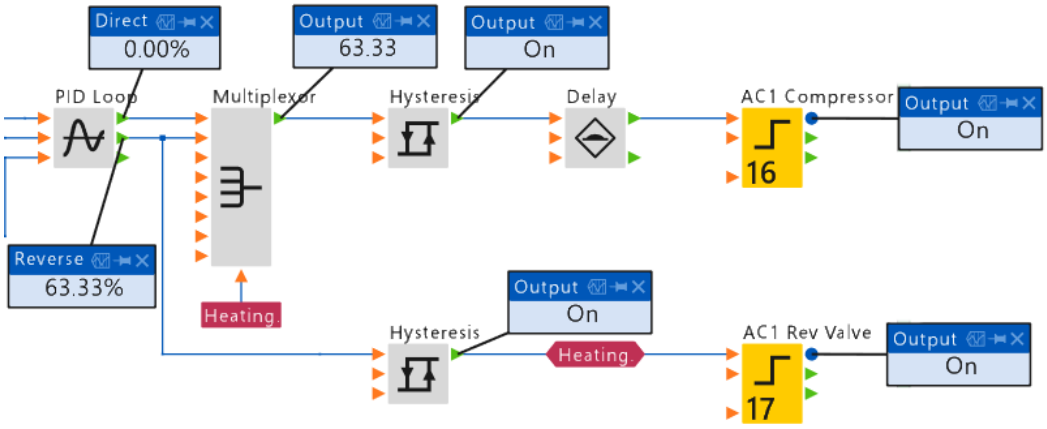
This example shows that a value of 5 on the select node. This selects input 5 (node 6), which is then fed to the multiplexors output node. Note that the input nodes start at 0 to 7.

The Multiplexor Switch in the example below selects between the Direct (Cooling) and Reverse (Heating) Acting outputs of the PID loop.

When the Heating mode signal (Binary) on the **Switch input** of the Multiplexor is **OFF (0)**, the Direct Acting signal of the PID is switched to the Multiplexor output.



When the Heating Mode is **ON (1),** the Reverse Acting signal of the PID is switched through to the Multiplexor Output.

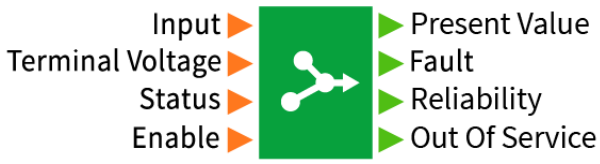


You can think of the Multiplexor as a selectable gate switch.

[Back to block tables page]

MULTISTATE INPUT

The Multistate Input block defines a standardised BACnet object whose Present Value represents the result of an algorithmic process within the BACnet Device in which the object resides.



NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet)

Terminal Voltage: The current input voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Enable: The Enable input of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value will cause a new value to be written to the Present Value only if the Enable Input is active.

Present Value: The Present Value displays the value that was written either from the input or from a BACnet client.

Fault: The Fault output will be FALSE whenever the Reliability output has a value of NO_FAULT_DETECTED, otherwise it will be TRUE.

Out Of Service: The Out of Service output indicates the override state of the object. (TRUE or FALSE) BACnet clients write directly to the Out of Service property to disable local control of the object, therefore disconnecting the Input from the Present Value and only allowing control via the BACnet network. (The block Input node is ignored whenever the Out of Service output is TRUE)

Reliability: The Reliability output for this type of object.

INTERNAL PROPERTIES

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

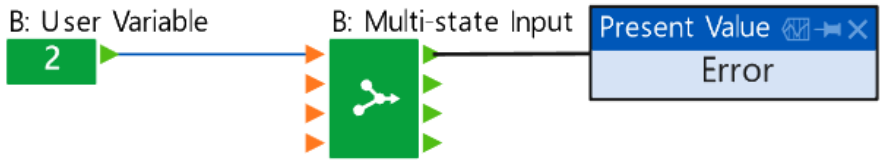
Device Type: A text description of the device connected to the Input.

State Text: The State Text is an array of values, each line ended with a carriage return is a new index in the array. The first item has an index of 1, the second line is 2 and so on. By default, index 1 is output by the block at the Present Value node. If a User Variable or other relevant input is connected to the Input node, this value can specify which line in the array is output at the Present Value node. Up to 100 lines with 127 characters each can be used.

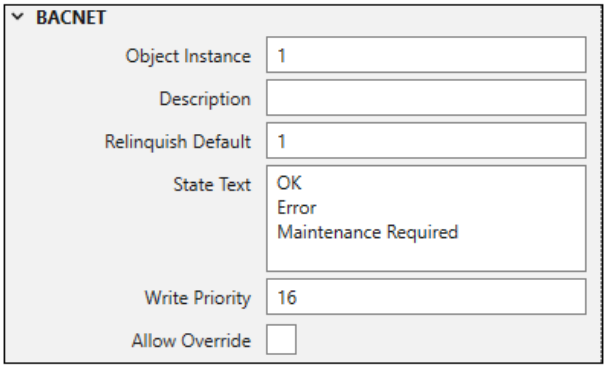
Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

EXAMPLE

BACnet Multi-State Input: Present Value Output



The Present Value node output is based on what items are listed in the State Text Property and the value input at the input node. In this case, the value of two (2) outputs the second item in the list, for this example, is Error. Note that the state text starts at a value of 1.

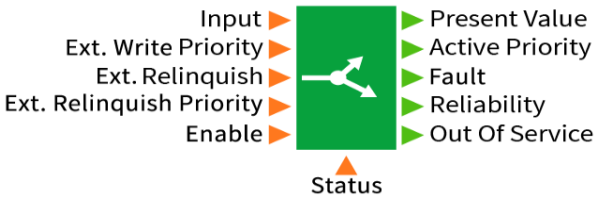


Value	State Text Selected
1	Ok
2	Error
3	Maintenance required

Back to block tables page

MULTISTATE OUTPUT

The Multistate Output block defines a standardised BACnet object whose properties represent the desired state of one or more physical outputs or processes within the BACnet Device in which the object resides.



NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active, and the Input value is written to the object at the highest Write Priority.

Ext. Write Priority: This determines the position in the Priority list (Known as the Priority Array) that the input value will be written to. (This equates to a value from 1-16, where 1 is the highest priority) The Input value will be written at the Ext. Write Priority value on a change in value of the Input, or a change in value of the Ext. Write Priority input.

Ext. Relinquish: The Ext. Relinquish node is a digital signal applied to delete a value that has been written to the object at a particular priority. When using the Ext. Relinquish node, you must also indicate the Priority of the value to be relinquished. This is achieved by setting a value from 1-16 on the Ext. Relinquish Priority node. Therefore, you should set the Ext. Relinquish Priority first, and then pulse the digital signal on the Ext. Relinquish node from “Off” to “On”.

Ext. Relinquish Priority: The Ext. Relinquish Priority input allows a user to select any priority (and value) to be relinquished from the Priority Array.

Enable: The Enable input of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Present Value: The Present Value displays the value that was written at the highest Write Priority, either from the input or from a BACnet client. If all priorities have been relinquished, the Relinquish Default internal property will become the Present Value.

Active Priority: The Active Priority indicates the highest priority that was written to the object. (1-16) If all priorities have been relinquished, the Active Priority will be 0.

Fault: The Fault output will be FALSE whenever the Reliability output has a value of NO_FAULT_DETECTED, otherwise it will be TRUE.

Reliability: The Reliability output for this type of object.

Out Of Service: The value of this property indicates whether or not the object is currently being overridden by a BACnet Client.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Multi-State Output.

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Device Type: A text description of the device connected to the Input.

Relinquish Default: The Relinquish Default setting is the default value the Present Value will display when all Write Priorities are relinquished.

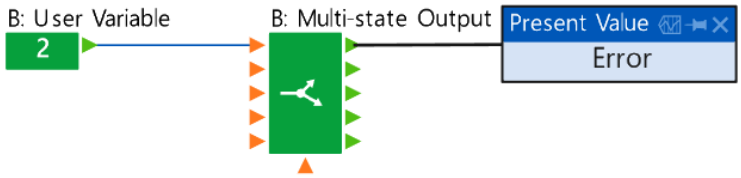
State Text: The State Text is an array of values, each line ended with a carriage return is a new index in the array. The first item has an index of 1, the second line is 2 and so on. By default, index 1 is output by the block at the Present Value node. If a User Variable or other relevant input is connected to the Input node, this value can specify which line in the array is output at the Present Value node. Up to 100 lines with 127 characters each can be used.

Write Priority: The Write Priority internal property indicates the priority at which the Input value will be written. This property is only used when there is no connection to the Ext. Write Priority input.

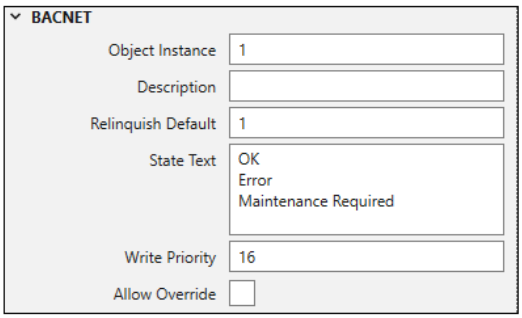
Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

EXAMPLE

BACnet Multi-State Output: Present Value Output



The Present Value node output is based on what items are listed in the State Text property and what value is input at the input node. Note that the state text starts at a value of 1.

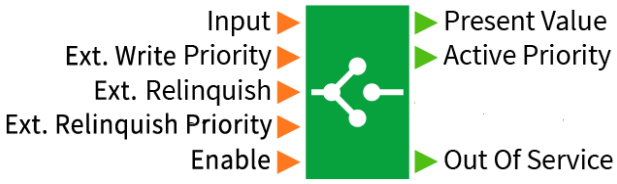


Value	State Text Selected
1	Ok
2	Error
3	Maintenance required

Back to block tables page

MULTISTATE VALUE

The Multistate Value block defines a standardised BACnet object whose properties represent the externally visible characteristics of an analog value.



NODE DESCRIPTORS

Input: The Input node is the value written from within the configuration software to create a desired outcome at the Present Value. (If no other intervention takes place via BACnet) The Input value will only become relevant to the Present Value if the Enable node is active, and the Input value is written to the object at the highest Write Priority.

Ext. Write Priority: This determines the position in the Priority list (Known as the Priority Array) that the input value will be written to. (This equates to a value from 1-16, where 1 is the highest priority) The Input value will be written at the Ext. Write Priority value on a change in value of the Input, or a change in value of the Ext. Write Priority input.

Ext. Relinquish: The Ext. Relinquish node is a digital signal applied to delete a value that has been written to the object at a particular priority. When using the Ext. Relinquish node, you must also indicate the Priority of the value to be relinquished. This is achieved by setting a value from 1-16 on the Ext. Relinquish Priority node. Therefore, you should set the Ext. Relinquish Priority first, and then pulse the digital signal on the Ext. Relinquish node from “Off” to “On”.

Ext. Relinquish Priority: The Ext. Relinquish Priority input allows a user to select any priority (and value) to be relinquished from the Priority Array.

Enable: The Enable input of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Present Value: The Present Value displays the value that was written at the highest Write Priority, either from the input or from a BACnet client. If all priorities have been relinquished, the Relinquish Default internal property will become the Present Value.

Active Priority: The Active Priority indicates the highest priority that was written to the object. (1-16) If all priorities have been relinquished, the Active Priority will be 0.

Out Of Service: The value of this property indicates whether or not the object is currently being overridden by a BACnet Client.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Multi-State Value.

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

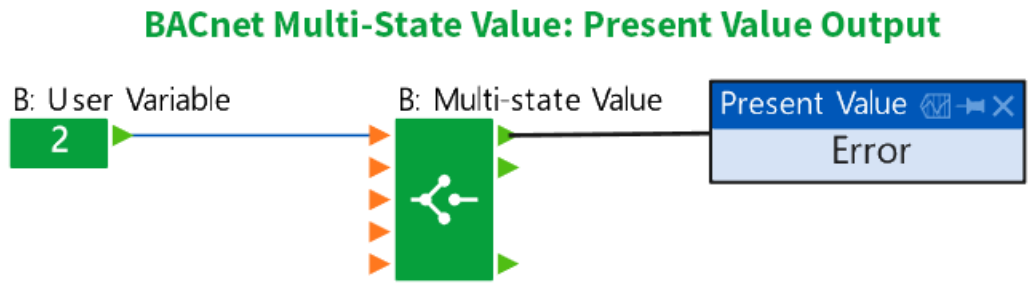
Relinquish Default: The Relinquish Default setting is the default value the Present Value will display when all Write Priorities are relinquished.

State Text: The State Text is an array of values, each line ended with a carriage return is a new index in the array. The first item has an index of 1, the second line is 2 and so on. By default, index 1 is output by the block at the Present Value node. If a User Variable or other relevant input is connected to the Input node, this value can specify which line in the array is output at the Present Value node. Up to 100 lines with 127 characters each can be used.

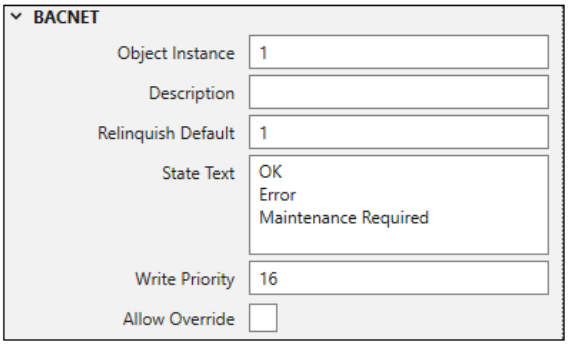
Write Priority: The Write Priority internal property indicates the priority at which the Input value will be written. This property is only used when there is no connection to the Ext. Write Priority input.

Allow Override: The Allow Override checkbox dictates whether or not a BACnet client is allowed to write to the Out Of Service property of the object. If the Allow Override checkbox is ticked, the Out of Service property of the BACnet object is writable from other BACnet clients. If unchecked, the Out of Service property is ready only, meaning no BACnet clients can disconnect the Input node from the Present Value output.

EXAMPLE



The Present Value node output is based on what items are listed in the State Text property and what value is input at the input node. Note that the state text starts at a value of 1.



State Text	Value
Ok	1
Error	2
Maintenance required	3

Back to block tables page

OPTIMUM START

The Optimum Start block estimates time required to heat/cool an enclosed area by averaging past recorded times.

NODE DESCRIPTORS

Time Until: This input is normally driven directly from the output of a Calendar block. It is used in the optimum start calculation to determine how many minutes there are remaining before the normal scheduled start.

Enable: This node is connected to the Calendar Block's Optimum Run node.

Temperature: This value, normally the temperature from a sensor, is used in the calculation to decide whether an optimum start cycle should be commenced.

Ext. Setpoint: This is the required value that the Input should reach (within the specified internal block property Dead Band) by the start of normal scheduled operation.

Time Output: Output changes depending on whether an optimum start cycle is in progress and also depending on the relation to the occupancy time:

- Before Occupied Time: Negative when showing minutes before A/C start. Positive when running, counting down to occupancy start.
- During Occupied Time: Time remaining to occupancy end during occupied time (as per schedule block output).

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Optimum Start.

Initial Settings

- **Heat Value** (Dg/h): This is the initial value of the ratio of required increase in temperature (T) and the time taken to heat the enclosed area by T to achieve the Setpoint.
- **Cool Value** (Dg/h): This is the initial value of the ratio of required increase in temperature (T) and the time taken to cool the enclosed area by T to achieve the Setpoint.

Max Start Time (min): This is the maximum time the A/C should run before the occupancy time (even if the estimated time to reach the Setpoint is bigger and thus the required temperature will not be reached at occupancy time).

Setpoint: This is the required temperature that should be reached at the occupancy time, which is used only if the Setpoint input to the block is not connected. You don't have to connect the Setpoint input, but doing so allows you to modify the Setpoint (via an Editable Watch on a User Variable) during DDC operation.

Dead Band: This is the total range of the dead band for the optimum start algorithm. This is the range of values (centred about the Setpoint) in which optimum start algorithm will stop operating.

Alpha (%): This is a smoothing constant (percentage) used in the integral and derivative calculations for the block. It is a representation of how fast the Start Time output value will change to meet the required value.

An Alpha of 100% will change immediately to the required value, whereas an Alpha of 50% will only move half way toward the required value each time the block is processed. If Alpha is set to 0%, the block behaves the same as if it was set to 100% (otherwise the block output value would never change).

USAGE

Overview of Optimum Start Algorithm

Refer below explanation of Optimum Start algorithm. Note that temperatures referenced are in Celsius (°C), however the algorithm functions in the same way when using Fahrenheit (°F).

The time required is based on a linear approximation of T (required change in temperature) versus t (time taken to changed ambient temperature by T to achieve set point. This ratio is expressed as °C/Hr. A separate ratio is calculated for heating and cooling.

Errors will be induced due to the linear approximation of a non-linear characteristic. I.E: On a hot day it may take 40 minutes to cool a room from 32°C to 22°C. On a cool day it may take a quarter of the time (10 minutes) to cool the same room by half that amount (27°C to 22°C). Also, it may take 75 minutes to go from 32°C to 20°C. (The relationship is not directly proportional and different set points will yield different results).

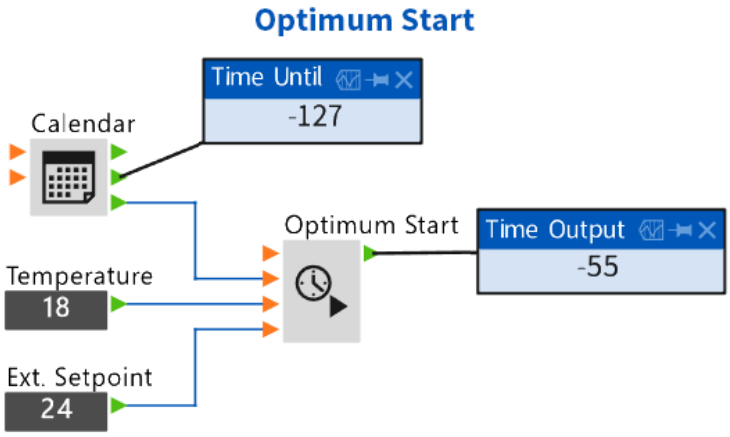
This block uses the slope of the line T/t (called S) to determine the estimated time required to achieve set point. As heating a room will be entirely different to cooling, two values for S are calculated and recorded in the block's memory. There is an option to save these values to the Non-volatile memory area.

S is determined from the previous attempt to reach set point. Each time an attempt is made, S is recalculated, filtered via alpha value supplied by user, and updated, hence it will require several attempts before getting close to its target. A larger Alpha value i.e. 100% will allow it to move to the real result faster, but may create instability and oscillation during the steady state.

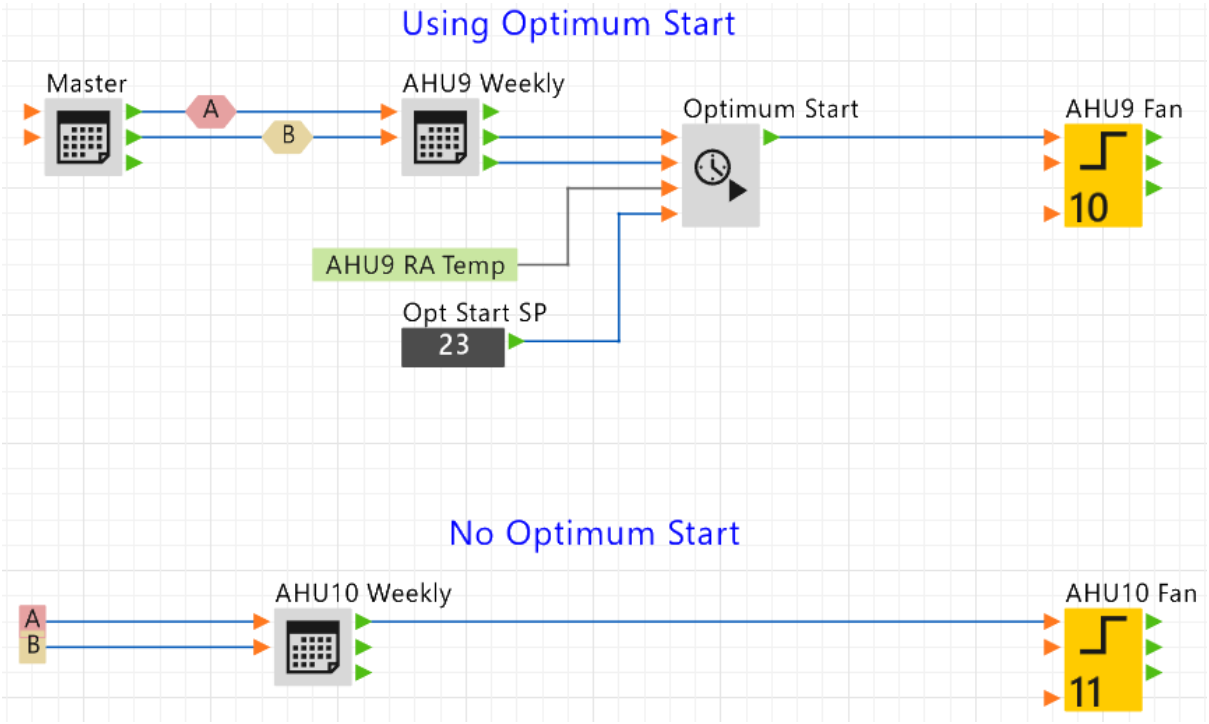
Care has to be taken in choosing the Initial Heat and Cool values in the internal block settings. If they are set too large then S will never be calculated and the Optimum Start block will not work. The initial values should be chosen to best represent the space being controlled and the equipment used. If these values are unknown, then small values should be used to start with so the Optimum Start block can calculate the best Heat and Cool constants for the system over time.

During this discussion, the Input value and Set Point are referred to as temperatures, along with the internal tolerance parameter and initial cooling and heating constants. Whilst the temperature will be the most common input type, it is possible to use other types of inputs. An example of an alternative application would be for a humidifying plant. The input value and set point value would deal with relative humidity, and therefore the internal parameters should also be thought of as dealing with relative humidity.

EXAMPLE



The Optimum Start block determines the optimum time to start the equipment based on the time until the system is meant to start, the current temperature and the setpoint.



[Back to block tables page]

PID LOOP

The PID Loop block is a proportional plus integral and derivative control loop. It is one of the primary control blocks used for maintaining constant temperature, humidity or a controlled variable. The three components of PID calculations are: Proportional Band, Integral Constant and Derivative Constant. Correct tuning of these three parameters will minimise the time for a system to reach the Setpoint, and to achieve system stability at the Setpoint.

NODE DESCRIPTORS

Input: This may be a percentage, voltage, temperature or any other such value and is used as the primary input to the block.

Enable: Enables processing and output from the PID Loop block. If the Enable input is not connected, the block will be enabled by default.

Ext. Setpoint: The external Setpoint to which the PID Loop will control. If this is not connected, the internal block property Setpoint will be used.

Output: A positive value indicates the amount of control required to bring the Input Value down to meet the Setpoint. A negative value indicates the amount of control required to bring the Input Value up to meet the Setpoint, and should to be read without the negative sign. Output functions as both the Direct and Reverse outputs rolled into the one point, and can be used separately or in conjunction with the Direct and Reverse outputs.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is PID Loop.

Set Point: This is the Setpoint of the PID Loop to which the Input to the block should finally settle, which is used only if the Setpoint input to the block is not connected.

Dead Band Value: This is the total range of the Dead Band for the PID Loop. This is the range of values (centred about the Setpoint) in which no Direct or Reverse acting control is used.

Alpha Value: This is a smoothing constant (percentage) used in the integral and derivative calculations for the block. It is a representation of how fast the output value will change to meet the required value. An Alpha of 100% will change immediately to the required value, whereas an Alpha of 50% will only move half way toward the required value each time the block is processed.

Proportional Band: Set the Proportional Band setting for the direct or reverse acting part of the PID Loop.

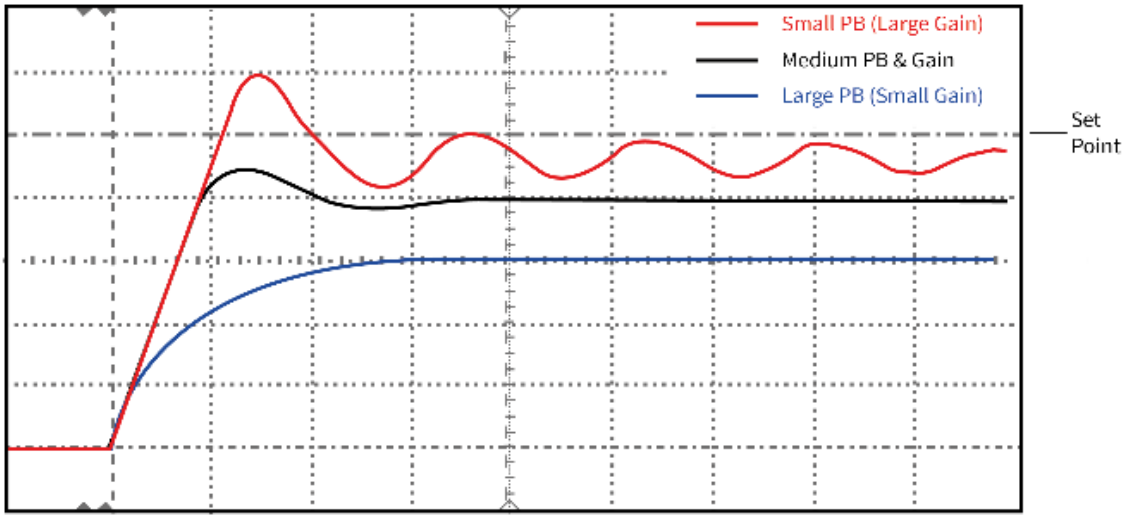
Integral Constant: Set the Integral Constant setting for the direct or reverse acting part of the PID Loop.

Derivative Constant: Set the Derivative Constant setting for the direct or reverse acting part of the PID Loop.

PID USAGE AND DESCRIPTIONS

Proportional Band

The Proportional Band is used to calculate the rate at which the system can be moved from the current value to the Set Point. The following illustration provides an example of possible results from different Proportional Band settings:



The red line represents a small value set for the Proportional Band. Note that it quickly moves from the initial value towards the Setpoint. However, note also that it dramatically overshoots the Setpoint and has to compensate. This oscillation continues around the Setpoint, which results that the red line example never achieves stability on the Setpoint, and this oscillation continues for the remainder of the graph.

Setting a Proportional Band value too low for the situation may stress equipment, as the output never achieves stability on the Set Point, which may result in equipment constantly switching between being On and Off.

The blue line represents a large value set for the Proportional Band. The rate of ascent is much smoother, yet much slower. Whilst the blue line achieves horizontal stabilisation, it may never actually reach the Setpoint.

Setting a Proportional Band Value too high for the situation may make the speed of change too slow, or your equipment may be working constantly to reach the Setpoint, without actually reaching the Setpoint.

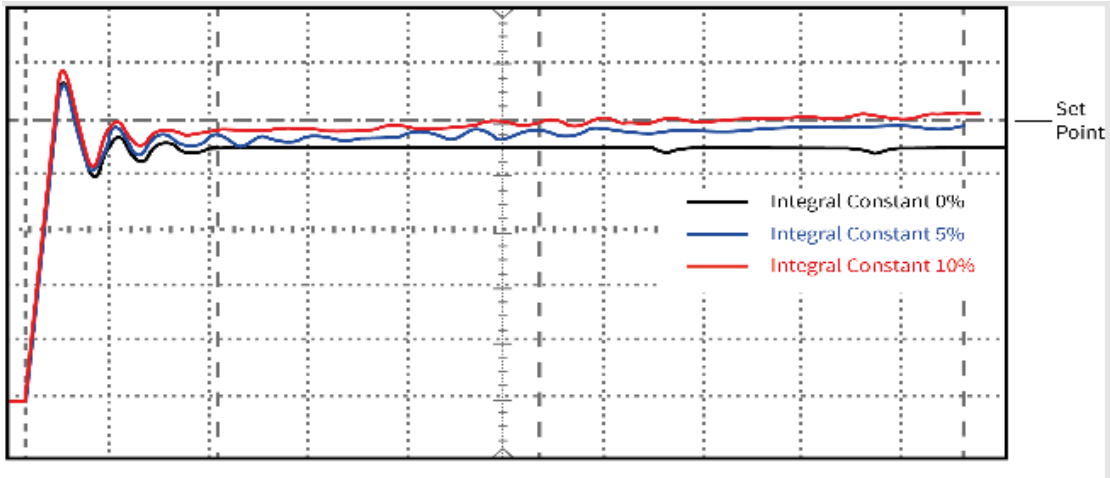
The black line is perhaps the optimal Proportional Band setting for this example. The rate of change is comparable to the red line, yet overshoot is minimised. Stabilisation at the Setpoint is achieved quite quickly with minimal oscillations.

The Proportional Band is the most important parameter of the PID Loop block, and should typically be set as small as possible, without causing excessive oscillations above and below the Setpoint value.

***Proportional**: **Looks at the present**. It uses the current error to determine output. Typically, it is in charge of most of the error reduction*.

Integral Constant

The Integral Constant works to ensure the Setpoint is reached once stabilisation has been reached, and to maintain the Setpoint value. The figure below provides three examples of different settings for the Integral Constant:



The black line has a value of zero for the Integral Constant. With no value set, the resulting stabilisation point of the PID Loop is horizontal, yet retains its offset from the real Setpoint.

The blue line has a small value set for the Integral Constant. In this scenario, the output value is slowly pushed toward the Setpoint once stabilisation has been reached. This will slowly reduce the offset value over time.

The red line has a large value set for the Integral Constant. In this scenario, the output value rises quickly toward the real Setpoint once stabilisation has been reached. However, too large an Integral Constant value will introduce additionally oscillations around the Setpoint once it has been reached.

The Integral Constant will eliminate Proportional Band Offset, at the expense of an increased System Time.

Use of the Integral Constant will have the effect of amplifying your setting for the Proportional Band. In real terms, it will have the effect that the Proportional Band has been set lower than it appears. Therefore, you will need to adjust the Proportional Band value to account for the Integral Offset value's contribution to the calculation.

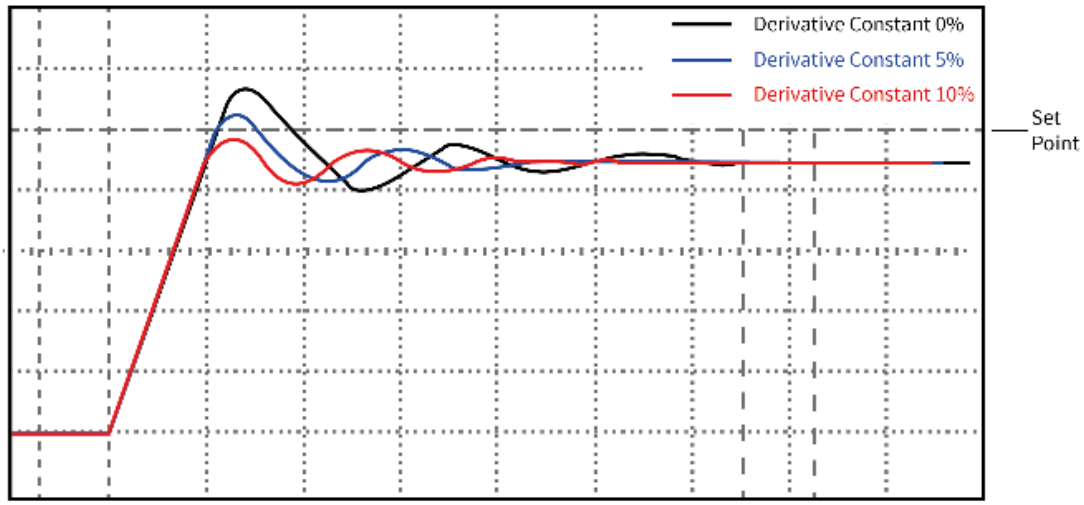
Using too large a value for the Integral Constant will introduce additional oscillations once the Set Point is reached, which could stress machinery.

The Integral Constant is used to eliminate Offset within the PID Loop calculations, which is critical in some systems. Any non-zero value will remove the Offset eventually. Larger values will remove the Offset more quickly, but may cause oscillations around the Set Point.

***Integral**: **Looks at the past**. It uses a sum of past errors to eliminate long term error.*

Derivative Constant

The Derivative Constant is used to squash calculation oscillations, and is often referred to as damping. The figure below provides three examples of different settings for the Derivative Constant:



The black line is an example of a value of zero set for the Derivative Constant. In this scenario, Overshoot and oscillations around the Setpoint are large until stabilisation at the Setpoint has been reached.

The blue line is an example of a small value set for the Derivative Constant. In this scenario, overshoot and oscillations have been reduced, and stabilisation at the Setpoint has been reached quicker.

The red line is an example of a larger value set for the Derivative Constant. In this scenario, overshoot and oscillations have been greatly reduced, and the Setpoint is reached very quickly.

Advantages of the Derivative Constant:

- Increased Stability
- Faster Setting Time
- Doesn't increase offset
- Less overshoot
- Slightly shorter peak time
- Allows for further reduction and tweaking of the Proportional Band value

Disadvantages of the Derivative Constant:

- Any noise present in the system will be amplified with the Derivative Constant. This could lead to quite incorrect values in certain situations.
- Too large a Derivative Constant value can result in a slow-acting, over-damped response.

The Derivative Constant is the least important value in a PID Loop calculation, and is often set to zero. It is best used in small amounts for improving system stability and reducing overshoot. It can be increased during fine-tuning to allow for improved controller performance.

Derivative: Looks at the future *- Trying to predict where the signal is going, it can be used to prevent both overshooting SetPoints and output ringing.*

EXAMPLES

Each use of the PID Loop block will benefit from tweaking and adjusting the Proportional Band, Integral and Derivative Constant.

The following illustrations are two possible starting points for your adjustments for situations where the PID Loop block is for used for Static Pressure or Temperature adjustment.

Static Pressure Example

Field	Direct Acting	Reverse Acting
Proportional Band	375.00	375.00
Integral Constant	2.00%	2.00%
Derivative Constant	0.00%	0.00%
Set Point	250	250
Dead Band	0.00	0.00
Alpha	100%	100%

Supply Air Temperature Example

Field	Direct Acting	Reverse Acting
Proportional Band	18.00	18.00
Integral Constant	2.00%	2.00%
Derivative Constant	0.00%	0.00%
Set Point	12	12
Dead Band	0.50	0.50
Alpha	100%	100%

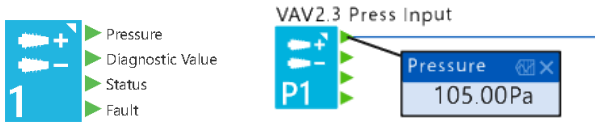
These are examples only; settings for initial PID Loop Block settings. Actual settings will depend on the job, and will benefit from onsite tunning.

Notes

1. When using Integral and Derivative control, Both the direct and Reverse acting settings should be set the same on both direct and reverse.
2. All PID settings should be checked or tunned on site to establish the correct operational settings.

[Back to block tables page]

PRESSURE INPUT



The Pressure Input block reads the on-board differential pressure sensor on an Omni C8V or C10V controller.

NODE DESCRIPTION

Pressure: Shows the value from the differential pressure sensor on the controller.

Diagnostic Value: Diagnostic pressure reading from the pressure sensor.

Status: indicates the current status of the block.

Fault: indicates that the output point is outside the min/max values.

INTERNAL BLOCK PROPERTIES

Name: Specify a name for this block. The default name is Pressure Input.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Pressure Sensor: The physical terminal on which the sensor resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager.

Alpha: Percentage of output difference added to the previous output to generate a new one.

Is Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Fault Monitoring

Minimum Input Value: Specify the minimum input value.

Maximum Input Value: Specify the maximum input value.

Calibration

Calculate Gain & Offset: Click to calculate the Gain and Offset. When the box is checked, you can enter the Omni Reading and meter Measured Values to adjust the gain and offset for the controller. This setting is used to adjust the values on the Omni compared to a meter reading if required.

Omni Reading #1 & #2: The value that the Omni controller shows on the HMI or the on-board web-server.

Measured Value #1 & #2: The actual meter measured value.

User Gain Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

User Offset Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

Suppression: Power line frequency suppression, select from None, 50Hz and 60 Hz.

Pressure Cutoff (Pa): Specifies the threshold that forces the pressure output to zero.

Back to block tables page

PSYCHROMETRIC

The Psychrometric block calculates a number of useful psychrometric functions based on two inputs. The block can produce one of the following calculations based on the Output Function selection and the Secondary Input selection.

NODE DESCRIPTORS

Dry Bulb: The dry-bulb temperature of the system in Degrees Celsius (°C).

Secondary Input: Select Wet Bulb or Relative Humidity from the Secondary Input combo box in the block properties.

Output Function: The single output, whose output type is selected using internal block property Function Type.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Psychrometric.

Unit: Select Metric or Imperial units of measure.

Barometric Pressure (kPa): This is the atmospheric pressure of the system. Under normal circumstances, the default setting for Barometric Pressure will be sufficient, but at abnormal altitudes, for example, you may need to adjust this value.

Output Function: The internal differential value. Select from the following options:

- Dew Point
  - Secondary Input: Relative Humidity: Select this option with Secondary Input as a Relative Humidity (%) input, and Output function as Relative Humidity (%).
  - Secondary Input: Wet Bulb: Select this option with Secondary Input as a Wet Bulb input, and Output function as Dew Point.
- Humidity Ratio
  - Secondary Input: Relative Humidity: Select this option with Secondary Input as a Relative Humidity (%) input, and Output function as Humidity Ratio (kg/kg).
  - Secondary Input: Wet Bulb: Select this option with Secondary Input as a Wet Bulb input, and Output function as Humidity Ratio (kg/kg).
- Enthalpy
  - Secondary Input: Relative Humidity: Select this option with Secondary Input as a Relative Humidity (%) input, and Output function as Enthalpy (kJ/kg).
  - Secondary Input: Wet Bulb: Select this option with Secondary Input as a Wet Bulb input, and Output function as Enthalpy (kJ/kg).
- Wet Bulb: Selecting this option configures the Secondary Input as Relative Humidity (%).
- Relative Humidity: Selecting this option configures the Secondary Input as Wet Bulb.

Secondary Input: Select from Wet Bulb or Relative Humidity for the secondary input.

Psychrometric Calculation Comparison Summary Innotech Approximations vs. ASHRAE Publication

For psychrometric calculations using dry bulb temperature and relative humidity, the calculated results are accurate over the input ranges listed below:

Calculation	Dry Bulb Range	Relative Humidity Range
Dew Point	0 to 60°C (32 to 140°F)	5 to 100%
Humidity Ratio (kg/kg)	0 to 60°C (32 to 140°F)	0 to 100%
Enthalpy (kJ/kg)	0 to 40°C (32 to 104°F)	0 to 100%

For psychrometric calculations using dry bulb and wet bulb temperatures, the calculated results are accurate over the input ranges listed below:

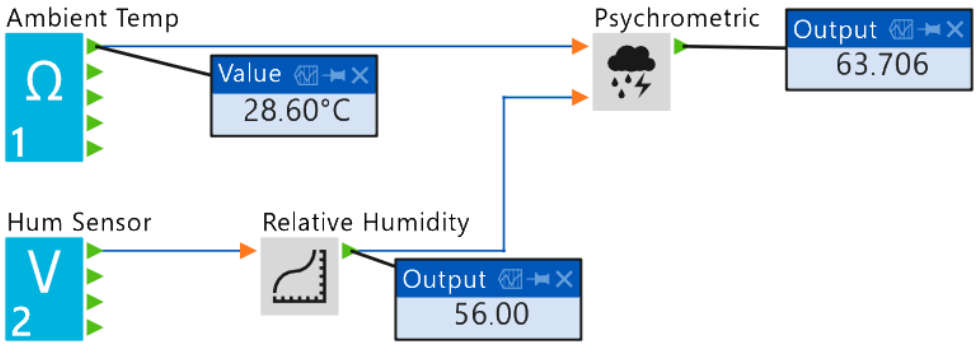
Calculation	Dry Bulb Range	Wet Bulb Range
Dew Point	0 to 60°C (32 to 140°F)	Values for RH ranging from 5 to 100%
Humidity Ratio (kg/kg)	0 to 60°C (32 to 140°F)	Values for RH ranging from 5 to 100%
Enthalpy (kJ/kg)	0 to 40°C (32 to 104°F)	Values for RH ranging from 5 to 100%
Relative Humidity (%)	0 to 80°C (32 to 176°F)	Values for RH ranging from 5 to 100%

The wet bulb ranges are expressed as ranges of relative humidity. This is because it is not possible to specify one range of wet bulb values (for each calculation) that will be valid for all dry bulb temperatures. Use a psychrometric chart to look up the wet bulb temperature range given the required dry bulb temperature and the relative humidity range (5 to 100%).

The Humidity Ratio Output mode provides a very rough estimate of the humidity ratio, and is not suitable for any operation which requires the accurate measurement or recording of humidity ratio levels.

EXAMPLE

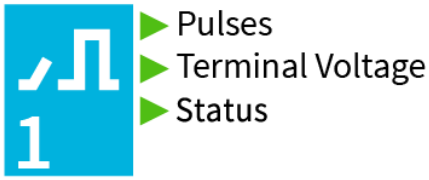
The Psychrometric block requires both inputs to be able to calculate an Output. This example has the Secondary input set to Relative Humidity and the Output function set to Enthalpy.



Back to block tables page

PULSE INPUT (CONTACT)

The Pulse Input (Contact) block outputs a count of OFF to ON transitions.



NODE DESCRIPTORS

Pulses: Provides the number of pulses since the last block cycle.

Terminal Voltage: The current input voltage on the pin.

Status Information Output: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Pulse Input (Contact).

Terminal: The physical terminal on which the block resides.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

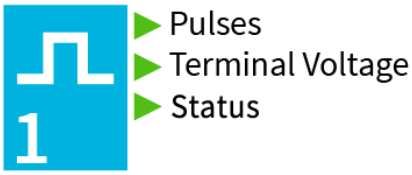
Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Is Forced: Set this internal block property to TRUE or FALSE as needed to Force or not force the block.

Back to block tables page

PULSE INPUT (VOLTAGE)

The Pulse Input (Voltage) block outputs a count of OFF to ON transitions.



NODE DESCRIPTORS

Pulses: Provides the number of pulses since the last block cycle.

Terminal Voltage: The current input voltage on the pin.

Status Information Output: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Pulse Input (Contact).

Terminal: The physical terminal on which the block resides.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Is Forced: Set this internal block property to TRUE or FALSE as needed to Force or not force the block.

Range: Range by hardware:

- None
- 5V (2.5V ON / 1.25V OFF)
- 10V (5V ON 2.5V OFF)
- 20mA (10mA ON / 5mA OFF)

Differential: 1 = Differential / 0 = 0V (Default)

Back to block tables page

PULSE OUTPUT

When the trigger input changes from OFF to ON, the block will output a single pulse (i.e.: turn ON for a specified duration and then OFF again). The duration (width) of the pulse is specified in the block properties and is accurately controlled by the controller. The use of this block is beneficial if you need a pulse generated at the UIO that is shorter than the cycle time of the controller.

NODE DESCRIPTORS

Value: The type of input depends on the settings applied for Internal Block Property Uni Curve Points for Function.

Trigger: Activates the pulse on a 0 to 1 transition.

Enable: Used to enable or disable the block's operation. If the Enable input is not connected, the block will be enabled by default.

Fault: If activated will use the internal block property Fault Value for the Output pulse value.

Output: The value sent to the ASIC.

Terminal Voltage: The current output voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Pulse Output.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Disable Value: This value is set to the Output value when the Enable input is False.

Fault Value: This value is set to the Output value when the Fault input is True.

Forced Trigger: Set this internal block property to TRUE to set the related Force Value to the Output point.

Forced Value: This value is set to the Output point when the internal block property Forced is set to TRUE.

Pulse Duration: Duration of the generated output pulse in seconds.

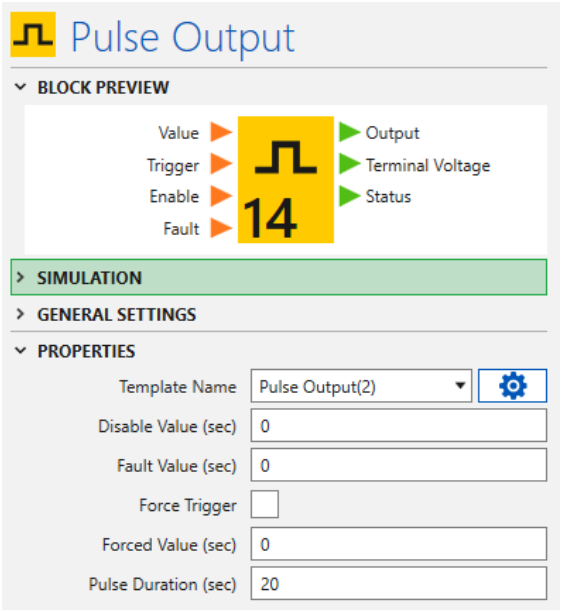
Switch Mode:

- Auto: 12V when on, 0V when off. (Default)
- High Only: 12V when on, floating otherwise.
- Low Only: 0V when off, floating otherwise.

Maximum Pulse Width: Duration of the longest output pulse (in seconds).

Pulse Output function

The following Pulse Output description is with the properties set as shown. The maximum Pulse duration in the duplicated Template is set to 20. The default template has a 10 second maximum.



The pulse output is triggered via the Pulse Trigger node and is a digital signal form off to on then off again. Once pulsed, this provides an output signal on the Output node of ON for the pulse time then turns OFF after the time period has expired.

This is also forms a relationship with the Value node signal. If the Value signal is at or above the pulse duration maximum set in the template, in this case this is a value of 20, then the full pulse output will be ON for the total period of time of 20 seconds.

This then scales back in proportion to the signal value and the maximum pulse duration. For example, if the value signal is 10, then the pulse when triggered will be 10 seconds of the maximum 20 seconds. If the signal is zero, then only a brief pulse will be present at the Output node.

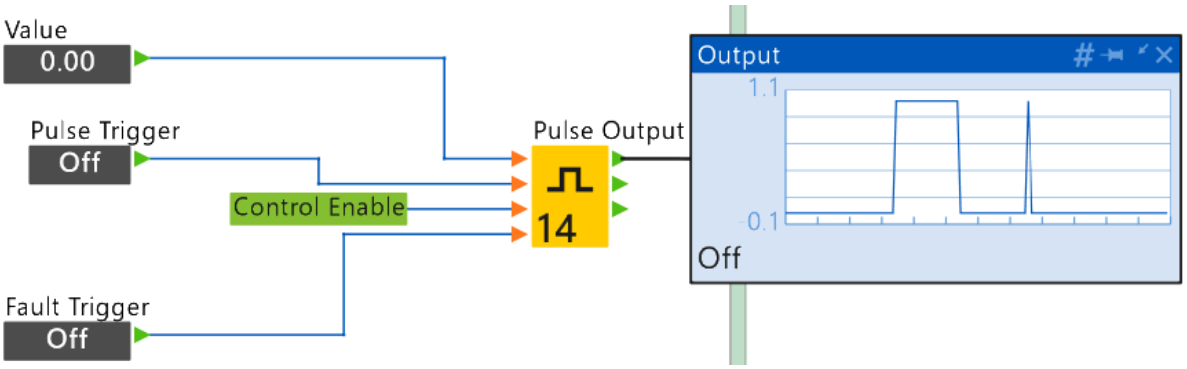
Pulse Output Vs Input Value Table

Pulse duration is set to 20 seconds in the block properties.

Value input signal	Output pulse in seconds
0	Instant Pulse
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
Up to 100	20

As shown in the table above, the input value forms a relationship with the Pulse output duration in accordance with the pulse duration setting set via the block properties.

Example



The example above shows the pulse output when the value input is at 10 and 0 respectively with the pulse duration set to 20 as per the table above.

[Back to block tables page]

RANDOM NUMBER

The Random Number block provides a functionality of a random number generator. The Random block continually outputs random numbers unless a Trigger input is connected. Every time there is a digital pulse on the Trigger input to the block a new random number is generated and fed to the Output of the block. The Output of the block is a number between Min and Max values specified as internal block properties of the block.

NODE DESCRIPTORS

Trigger: Pulse input (transition from OFF to ON states) used to force the generation of a new random number.

Output: This is the generated random number, between the Min and Max values as specified in the internal block properties.

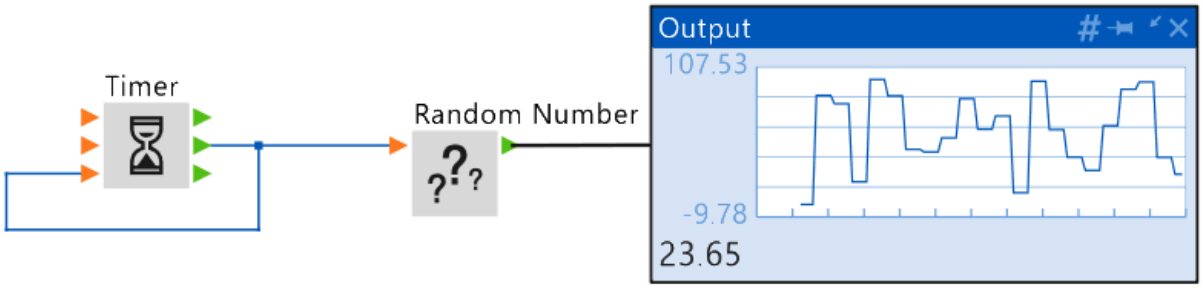
INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Random Number.

Min Value: Specify the minimum value for the random number generator.

Max Value: Specify the maximum value for the random number generator.

EXAMPLE



Each time an impulse is detected on the Trigger node, a random number between the specified minimum and maximum values is sent to the Output node.

Back to block tables page

RECIPE

The Recipe block is a sophisticated multi-stage sequential Setpoint generator. **Note that 50 Recipe blocks can be added per Omni.**

NODE DESCRIPTORS

Enable: Point input which enables the Recipe block. If the Enable input is not connected, then the Recipe block will be enabled by default.

Restart: Point input which restarts the recipe processing from the beginning.

Step: Pulse input which is used to force the recipe processing to the next stage in the internal block property Recipe Editor.

Holding: Point input which should be set to ON when the conditions based on the recipe output has reached the desired point. When the recipe block is at a flat line, with Holding ON the timer will continue. With Holding OFF, the recipe will stop (be held) until Holding returns to ON.

Value: Point which is the output value for the current stage of the recipe.

Stage: Point representing the current stage in the recipe which is being processed. There are 10 possible stages so this number will be from 1 to 10.

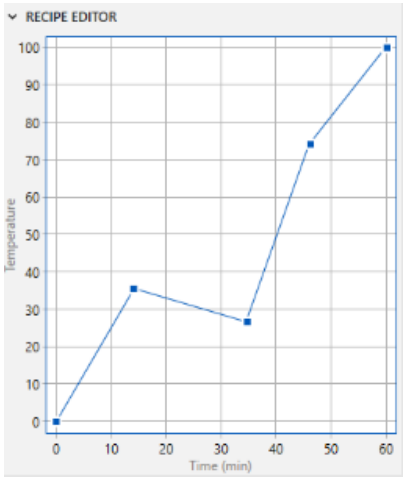
Progress: Point which is the percentage (0..100) of the current stage which has been completed.

Done: Point indicating whether or not the recipe has finished processing.

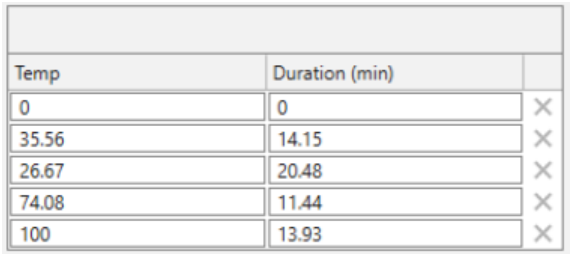
INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Recipe.

Recipe Editor - The Recipe Editor provides two methods to configure the operation of the Recipe Block. The Recipe Editor Chart Control allows for a simple drag and drop method to create your control algorithm. Simply click on the line anywhere to add a new node, and drag to the needed position.



For more granular control of the Recipe Block's control algorithm, edit the values in the Input To Output Values table. From here you can also delete entries, and any changes made are updated in real time in the chart control.



Temp	Duration (min)	
0	0	✕
35.56	14.15	✕
26.67	20.48	✕
74.08	11.44	✕
100	13.93	✕

Usage Notes General

With the Recipe block, you are able to implement a scenario such as the following:

- Starting at 0°C, raise the output Setpoint to 21°C.
- Leave the Setpoint at 21°C and maintain it for 30 minutes.
- Gradually Increase the Setpoint to 34°C degrees over a period of 120 minutes.
- Hold the Setpoint at 34°C degrees and maintain it for 10 minutes.
- Raise the Setpoint to 95°C degrees.
- Gradually increase the Setpoint to 100°C degrees over a period of 25 minutes.
- Maintain the set point for 37 minutes then signal completion.

The internal block properties configuration of the Recipe block determines the recipe.

Usage Notes Holding Input

If you were using the Recipe block to control temperature, the Holding input would be connected to the output of a Comparator block which compared the desired temperature (value) and the actual temperature (from a temperature sensor).

The Holding input is only used for a recipe where the output from the recipe block must be reached before you want the elapsed time to start decrementing: the recipe block will not start decreasing the time remaining for the current stage until the Holding input is ON.

The holding temperature must be specified as a horizontal line on the internal block property Recipe Editor, otherwise the Holding input will be ignored. An example of this is described in the Recipe Editor usage notes below.

Usage Notes Recipe Editor

For a usage example, read through the following scenario:

Temp / Time (min) fields:

The temperature and time boxes are on the left of the Recipe Editor box. For each temperature on the left, you specify the duration, in minutes, for the recipe block to raise or lower the output to that temperature. Note that the times are cumulative. For example, given the following three points:

- Stage 1: Temp 25°C, Time 30 minutes.
- Stage 2: Temp 35°C, Time 50 minutes.
- Stage 3: Temp 35°C, Time 60 minutes.

This means that you want the recipe process to reach the Setpoint of 25°C in 30 minutes, and then to increase to 35°C over a period of 20 minutes. Finally, the Setpoint of 35°C must be maintained (Holding input must be ON) for 10 minutes before Done output becomes ON. The holding temperature is represented as a horizontal line on the recipe graph.

Recipe Graph

You use the graph to add and remove points:

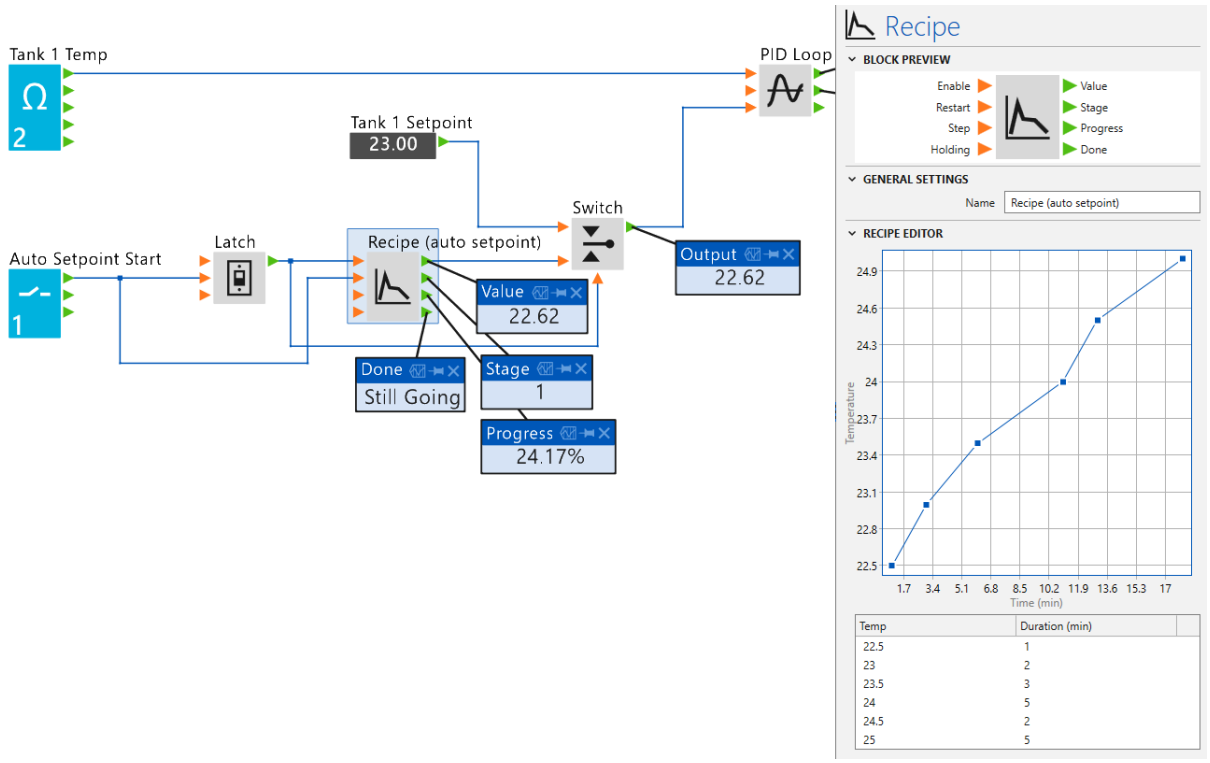
- Click on the graph with the left mouse button to add a new point.
- Click on an existing point with the right mouse button to remove that point.

You can also move points on the graph by pressing and holding the left mouse button on an existing point and dragging it to its new location on the graph.

EXAMPLE

Create a recipe by modifying the graph in the block properties. Multiple stages can be added. Each stage can have a different value and duration. After one stage has completed, the next stage begins. The Done node will only have an output when all stages are complete.

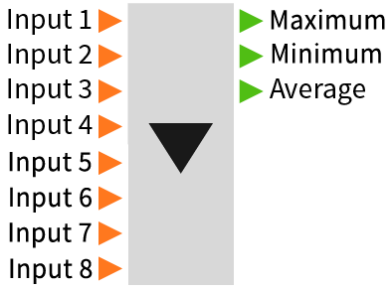
In the example below, when the recipe block is trigged via the auto setpoint push button, the latch enables the start of the recipe staged process. This simply resets the setpoint at set time periods until all stages are complete. This overrides the user setpoint until the latch timer expires.



[Back to block tables page]

SELECTOR

The Selector block provides the function of a Minimum, Maximum or Average signal select unit.



NODE DESCRIPTORS

Input 1..8: Used in the internal calculation of maximum, minimum and average.

Maximum: The maximum value from all the connected Inputs to the block.

Minimum: The minimum value from all the connected Inputs to the block.

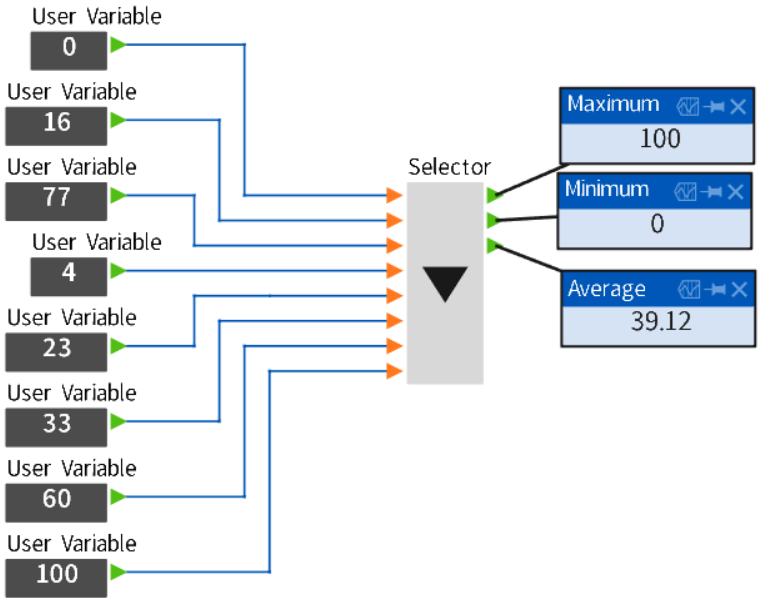
Average: The calculated average value of all the connected Inputs to the block.

INTERNAL PROPERTIES

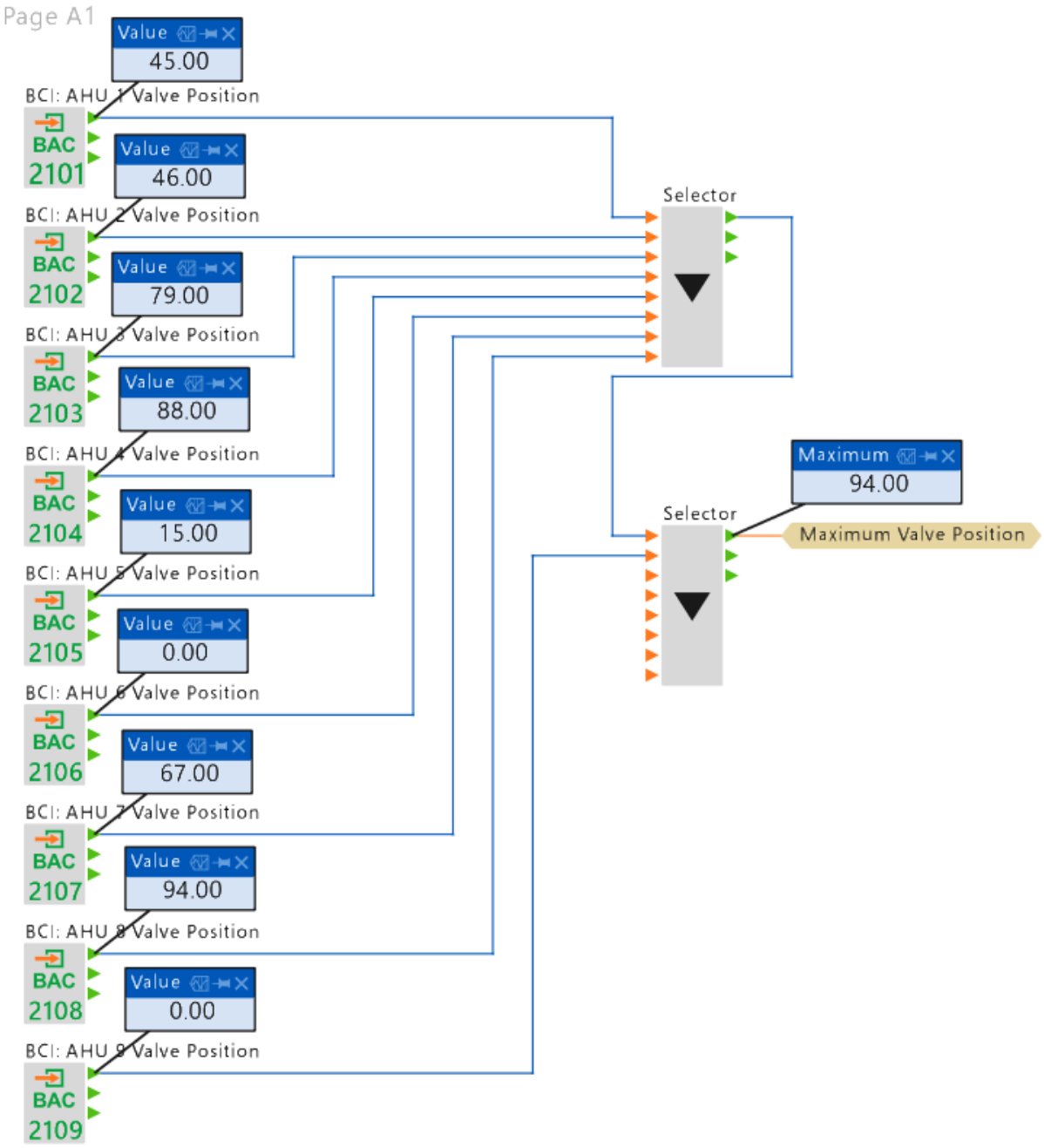
Name: Specify a name for this block. The default name is Selector.

EXAMPLE

The Selector block outputs a maximum, minimum and average for the input values.



In this example, the selector is monitoring nine AHU's valve positions. The maximum valve position is then used to produce the cooling call to the chiller plant.



Back to block tables page

SENSOR INPUT

The Sensor Input block is an analog input which measures resistive type signals and the input unit to the translator is always in Ohms.

NODE DESCRIPTORS

Value: The translated, calibrated and alpha smoothed signal.

Terminal Voltage: The current input voltage on the pin.

Fault: Indicates if the Output is outside the min/max properties.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Request: Provides After Hours input detection without disturbing the translated signal for a period of time.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Sensor Input.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Alpha: Percentage of output difference added to the previous output to generate a new one.

Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Minimum Input Value: Specify a minimum input value.

Maximum Input Value: Specify a maximum input value.

Calculate Gain & Offset: Click to calculate the Gain and Offset. When the box is checked you can enter the Omni Reading and meter Measured Values to adjust the gain and offset for the controller. This setting is used to adjust the values on the Omni compared to a meter reading if required.

Omni Reading #1 & #2: The value that the Omni controller shows on the HMI or the on-board web-server.

Measured Value #1 & #2: The actual meter measured value.

User Gain Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

User Offset Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller. It is used to manually adjust the output value, plus or minus a specified amount.

Suppression: Power line frequency suppression. Choose from None, 50Hz or 60Hz.

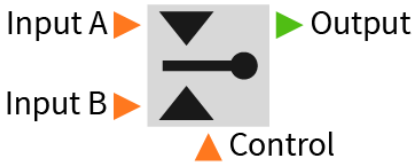
Request Delay Time (sec): Minimum time before request triggered.

Fault Delay Time (sec): Maximum time before fault is reported.

Back to block tables page

SWITCH

The Switch block provides a simple 2-channel switching function.



NODE DESCRIPTORS

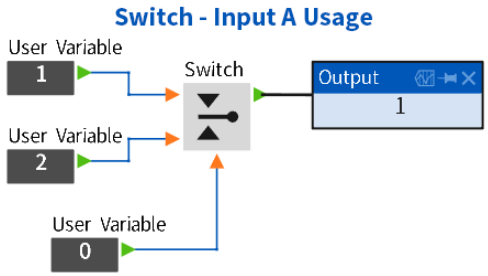
Input A: The first of two inputs which can be switched through to the Output depending on the Control input.

Input B: The second of two inputs which can be switched through to the Output depending on the Control input.

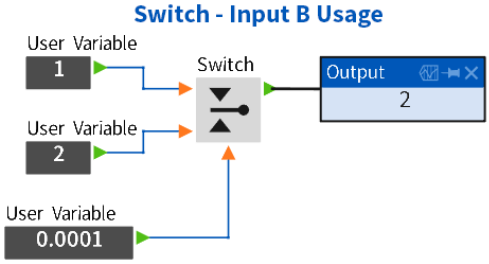
Control: This point is used to determine which Input to switch to the Output. If the value of Control is less than or equal to zero (or digital OFF), then Input A will be switched to the Output. If the value of Control is greater than zero (or digital ON), then Input B will be switched to the Output.

Output: This point is the value which has been switched.

EXAMPLES



When the value sent to the control input is zero, Input A is selected.



If the control input value is above zero by any amount, Input B is selected.

Back to block tables page

SYSTEM INPUT

The System Input block provides access to system information, current time, IO Module Temperature, Power Supply Temperature, System 12V etc.

NODE DESCRIPTORS

Output: The value as configured using internal block property Input Origin.

Fault: Diagnostic status output for the following system parameters (if selected): IOMODULETEMP, POWERSUPPLYTEMP, SYSTEM12V, CURRENTSENSE, SYSTEM5V. Can be one of 3 values (OK, Warning or Error). Other system parameters will always be OK.

INTERNAL PROPERTIES AND VALUES

Name: Specify a name for this block. The default name is System Input.

Input Origin Type: **Clock**

Input Origin: Set the input source for the System Input block. Select from the following:

- Current Month
- Current Year
- Date
- Date & Time
- Day of Month
- Day of Week
- Daylight (Daylight hours' output is displayed in seconds)
- Hour in the Day
- Minute in the Hour
- Sunrise (output is displayed in seconds from midnight)
- Sunset (output is displayed in seconds from midnight)
- Time

Input Origin Type: **Diagnostics**

Input Origin: Set the input source for the System Input block. Select from the following:

- Current Sense
- IO Module Temperature
- Power Supply Temperature
- System 12V
- System 5V

Back to block tables page

TIMER

The Timer block provides the functionality of a conventional electric timer with the provision to be able to set the time limit externally.

NODE DESCRIPTORS

Ext. Time: If connected, overrides the internal block property Timer Value. The units of this value (hours, minutes or seconds) are set in internal block property Time Value Unit.

Enable: A digital enable signal which, if connected, enables or disables the Timer block but does not reset any internal values. If the Enable input is not connected, then the Timer block is enabled by default.

Reset: Pulse which resets the current Timer block internal values back to zero - it basically restarts the timer from the beginning.

Running: Point which is ON whenever the Time value is greater than zero.

Done: Point which is ON when the timer has reached the internal block property Timer Value. The timer keeps counting so that you can then re-adjust the external time limit to be greater than the current time value - which will turn the Done output OFF.

Time: The current timer value.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Timer.

Timer Value: The internal timer limit/value. This value is overridden by the input External Limit, if connected.

Time Units: The units of the timer value. You can choose from hours, minutes or seconds.

Initial Timer Value: This is the starting timer value which will be used at the Time output of the block.

Usage Notes

When configuration is read back from the device the Initial Timer Value will be set to the timer value at the Output of the block at the time of the configuration transfer.

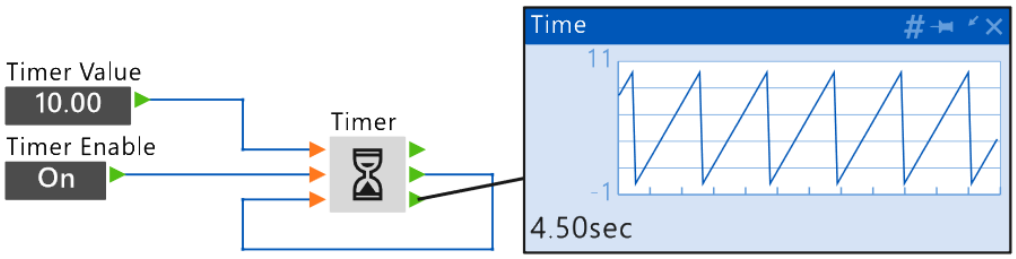
This is useful if configuration needs to be retransferred to the device (e.g. for maintenance) because the timer value will not be reset to zero but instead will start timing from the last known timer value.

For example, you may be using the block to log the hours run on a compressor, but when sending the configuration, you may need to specify that the compressor already has a run time of 150 hours.

EXAMPLE

In the example below, the timer is being used as a self-resetting timer for triggering other items in the config. As long as the enable is ON the timer will continue to count and reach the value, in this case the timer value is set via the external user variable and is set to 10 seconds.

Once the 10 seconds are done, the reset is triggered to reset the timer and start the process over again. This simple config has many applications in logic control.



Back to block tables page

TREND LOG (BACnet)

The Trend Log object monitors a property of a referenced object and, when predefined conditions are met, saves ("logs") the value of the property and a timestamp in an internal buffer for subsequent retrieval. The data may be logged periodically, upon a change of value or when "triggered" by a write to the Trigger property.

The logged data can then be extracted by a BACnet client and viewed as required.

NODE DESCRIPTORS

Input: The Enable input of the object controls whether or not the value from the Input is passed to the Present Value output of the object. A change in Input value, or change in the Write priority value will cause a new value to be written to the Present Value at the selected Write Priority only if the Enable Input is active.

Ext. Enable: This enables a programmer to Enable and Disable the Trend Log block based on any condition within the configuration.

Ext. Trigger: If set to Trigger mode, the Input is logged to the block and updated onto the Value output.

Present Value: Shows the value of the Input.

Reliability: This shows the health status of the Trend Log object, as far as the device can determine.

Is Logging: This reflects whether the Trend Log is currently acquiring data and creating records.

This output is ACTIVE when:

- The actual time is within the StartTime and StopTime
- And the Ext.Enable is ON (If connected)
- And the Buffer is not full, when "Stop When Full" is ticked

Therefore, when the block is not enabled, or is not within in the Start and Stop Times, or the Buffer is full and the "Stop when Full" is ticked, the "Is Logging" output will be FALSE (Off)

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Trend Log.

Object Instance: This is the Trend Log's BACnet Object Instance.

AI Object Instance: This is an Analog Input created by the Trend Log to allow a BACnet system to read the current Input value.

Description: Optional description of the block.

LOGGING SETTINGS

Log View: This determines the log view in the web-server Logging view - Digital or Analog.

Specify Start Date/Time: When unticked, the start date and time is unspecified (Start now).

Start Date: [visible when Specify Start Date/Time is ticked] - Specify a Start Date.

Start Time: [visible when Specify Start Date/Time is ticked] - Specify a Start Time.

Specify Stop Date/Time: When unticked, the stop date and time is unspecified (never stops).

Stop Date: [visible when Specify Stop Date/Time is ticked] - Specify a Stop Date.

Stop Time: [visible when Specify Stop Date/Time is ticked] - Specify a Stop Time.

Buffer Size: Total number of records.

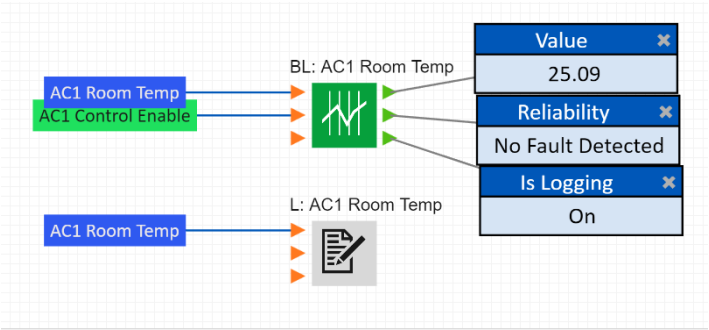
Stop When Full: When unticked and the Record Count = Total Allowable Records (Buffer Size is full) the first log in the database will be removed to allow the new entry, and so on. (First in, first out)
When ticked and the Record Count = Total Allowable Records (Buffer Size is full), no more logs will be saved to the database until the Buffer is cleared.

Logging Type: Polled - Logging occurs at a time interval specified by the Logging Frequency and Log Interval settings. Change of Value (COV) - Logging occurs when the value has changed by the value specified in the COV Increment setting. Triggered - A new record is written when the External Trigger Block Input is activated.

Logging Frequency: [visible when Polled Logging Type is selected] - Select the Polled logging frequency, seconds, minutes or hours.

Log Interval: [visible when Polled Logging Type is selected] - Specify a timed value between logs. The interval type will be what is specified in the Logging Frequency setting.

EXAMPLE



The BACnet Trend Log is logging the AC1 Room temp in parallel with the Innotech Log block. Note that the BACnet Trend Log must be enabled to Log the point value.

Back to block tables page

TRIAC DIGITAL OUTPUT

NODE DESCRIPTORS

Value: The input value to the block.

Enable: Used to enable or disable the block's operation. If the Enable input is not connected, the block will be enabled by default.

Fault: If set to the On state, sets the internal block property Fault Value as the output value.

Output: The output value is determined based on the internal block properties and inputs.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is TRIAC Digital Output.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager.

Disable Value: This value that becomes the output value of the block if the block is disabled.

Fault Value: The value that becomes the output value of the block if the fault input to the block is set to on.

Is Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

[Back to block tables page]

TRIAC PULSE OUTPUT

NODE DESCRIPTORS

Value: Sets the input value of the Triac Pulse Output block.

Enable: Used to enable or disable the block's operation. If the Enable input is not connected, the block will be enabled by default.

Fault: If set to the On state, sets the internal block property Fault Value as the output value.

Output: The value which is sent to the TRIAC.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is TRIAC Pulse Output.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager.

Disable Value: This value that becomes the output value of the block if the block is disabled.

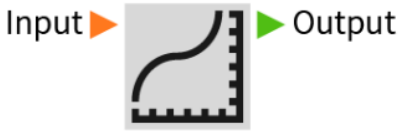
Fault Value: The value that becomes the output value of the block if the fault input to the block is set to on.

Is Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Back to block tables page

UNIVERSAL CURVE

The Universal Curve block converts an analog input to an analog output based on a graph which you define in the internal block properties. **Note that 100 Universal Curve blocks can be added per Omni.**



NODE DESCRIPTORS

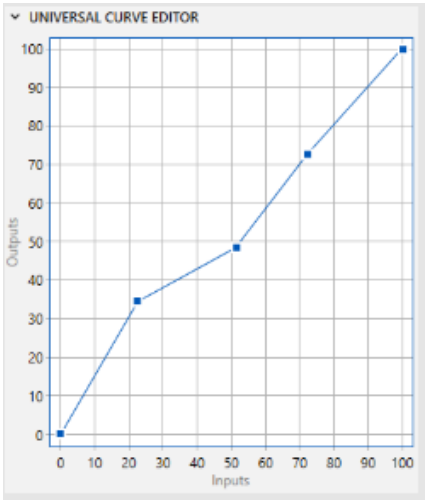
Input: Point input which will be converted to the Output based on the internal block properties.

Output: Point which is derived from a combination of both the Input value and the internal block properties.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Universal Curve.

Universal Curve Editor - The Universal Curve Editor provides two methods to configure the operation of the Universal Curve Block. The Universal Curve Chart Control allows for a simple drag and drop method to create your control algorithm. Simply click on the line anywhere to add a new node, and drag to the needed position.

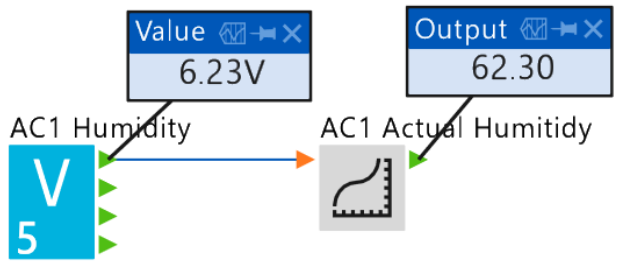


For more granular control of the Universal Curve Block's control algorithm, edit the values in the Input To Output Values table. From here you can also delete entries, and any changes made are updated in real time in the chart control.

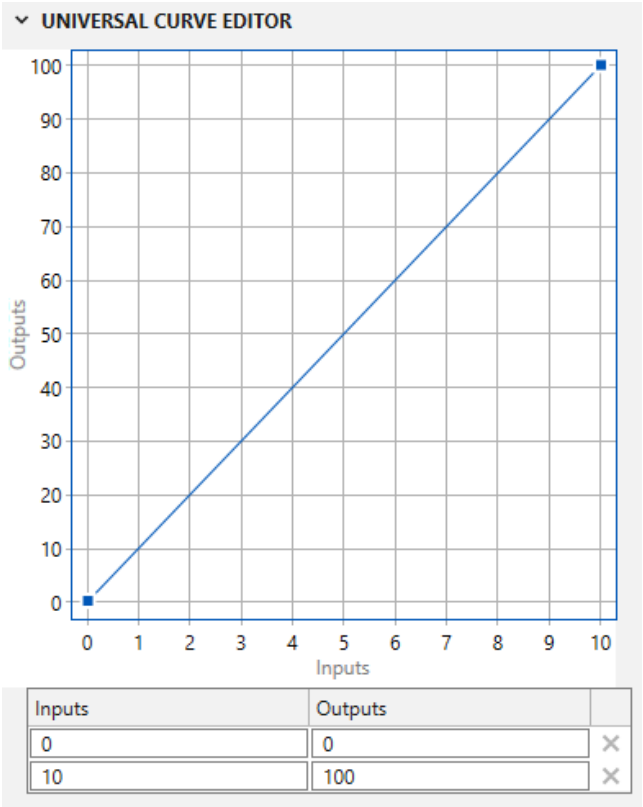
Inputs	Outputs	
0	0	✕
22.63	34.53	✕
51.33	48.45	✕
72.41	72.74	✕
100	100	✕

EXAMPLE

In the example below, the voltage input is being rescaled via the Universal Curve to provide a real world value based on the voltage input.



The Input and Output values need to be entered into the table for the scaling to take effect as desired. In this case, the 0 to 10 volts signal provides a 0 to 100 % RH output.



[Back to block tables page]

USER VARIABLE

The User Variable block allows the user to provide a value as an input to a block and to edit that value from the Digital Controller's HMI (Human-Machine Interface) or supported Innotech software programs.

NODE DESCRIPTORS

Output: Point which may be used as the input to other blocks. The Output type is configured in internal block property Type.

INTERNAL PROPERTIES

Type: This is the type of variable which is being used, which can be a Value, Digital, Digital Pulse or Selector. A Digital type produces an ON or an OFF output, the Value type can produce any decimal value, Digital Pulse can produce and ON / OFF value and specified Pulse Duration and the Selector type can output ON, OFF or Auto which is equivalent to 0, 1 or 2 decimal output.

Value: The value settings are determined by the setting type for the user variable: Value, Digital or Auto. Select from the following:

- For a Value User Variable, enter a numeric value. The value must be within the specified Range settings for Min and Max value.
- For a Digital User Variable, select from OFF or ON.
- For a Digital Pulse, select from OFF or ON.
- For a Selector User Variable, select from OFF, ON or Auto.

Max Value: Enter the maximum allowed numeric value for the User Variable (Default is 100).

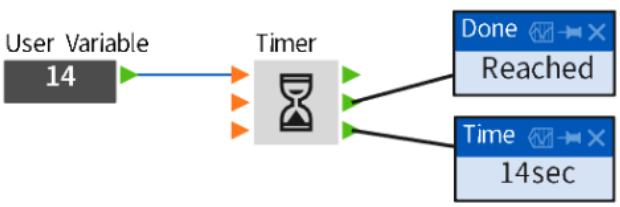
Min Value: Enter the minimum allowed numeric value for the User Variable (Default is 0).

The Value settings are only applicable for a User Variable of type Value. Range settings are not visible for the other User Variable types.

USER VARIABLE TYPES

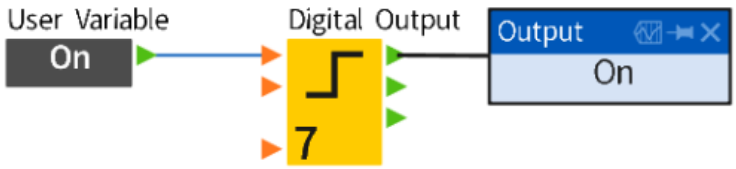
Value

When the User Variable is set to value, the specified value is sent to the output, in this case as the external value for the Timer block.



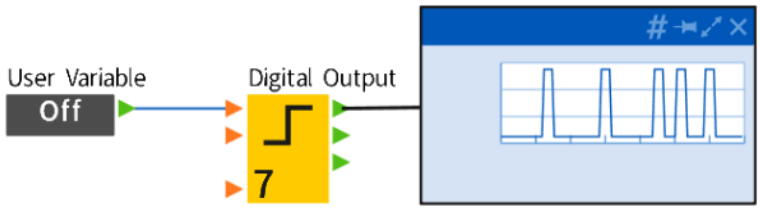
Digital

When set to a digital type, the value can be set to On or Off.



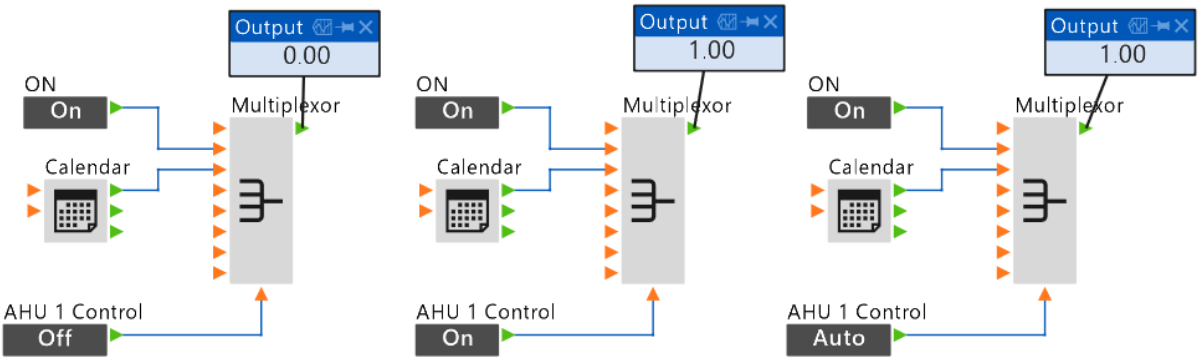
Digital Pulse

When set to digital pulse, the User variable when turned on, will then stay on for a brief period then turn off again. This is repeated each time the User Variable is turned on. This is useful for resets or triggering other logic manually by a user.



Selector

The selector mode acts like a software Auto / On / Off switch. Each press of the User Variable selects the mode in a cyclic order from off to auto. Note that each mode has a corresponding value as explained below.



Multiplexor Input 1 is selected Off = 0. Nothing is connected so zero is output

Multiplexor Input 2 is selected as On = 1, which is on via the User Variable

Multiplexor Input 3 is selected as Auto = 2. This provides Calendar control.

Note that the AOM selector mode outputs a zero (0) for OFF, a one (1) for ON and a two (2) for Auto. This allows you to Select the multiplexor input for the required operation.

Back to block tables page

USER VARIABLE (BACNET)

The User Variable block allows the user to provide a value as an input to a block and to edit that value from the Digital Controller's HMI (Human-Machine Interface), supported Innotech software programs or any BACnet client.

Note that the BACnet type for this object is an Analog Value and is writable by default. You do not need to specify a priority to write to the Present Value as it does not exist for this block.

NODE DESCRIPTOR

Present Value: The Present Value displays the value that was last written either from a User change (via a Watch placed on the node), or from a BACnet client or Innotech Client over the communication network. The Present Value is limited to values within the range of the Minimum Present Value and Maximum Present Value properties.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is User Variable.

Object Instance: Unique BACnet address identifier for the object.

Description: A simple explanation of the object. The name would usually be changed, so the descriptor indicates the type of object that is selected.

Type: Value / Digital / Digital Pulse / Selector

Value: The value of the user variable based on the Type selection.

Min Value: The minimum value when the Value Type is selected.

Max Value: The maximum value when the Value Type is selected.

Pulse Duration: Pulse duration in seconds when the Type is set to Digital Pulse.

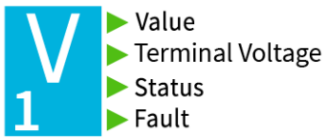
Units: The Units property of the object allows you to select one of the standard BACnet units.

Change Of Value Increment: The Change of Value (COV) setting allows you to optimise the amount of BACnet traffic to and from your BACnet device.

[Back to block tables page]

VOLTAGE INPUT

The Voltage Input block returns the translated value of the applied analog terminal voltage back to the internal processing of the digital controller.



NODE DESCRIPTORS

Value: The output has been translated, calibrated and has alpha applied to the signal.

Terminal Voltage: The current input voltage on the pin.

Status: Indicates the current status of the block. The following outputs are provided:

- OK
- Disconnected
- Overload

Fault: Turns ON when input is greater that Maximum Value or less than Minimum Value specified in block properties.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Voltage Input.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Alpha: 0-100% Resolution 0.01%. Default to 100%.

Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Minimum Input Value: Specify a minimum input value.

Maximum Input Value: Specify a maximum input value.

Calculate Gain & Offset: Click to calculate the Gain and Offset. When the box is checked you can enter the Omni Reading and meter Measured Values to adjust the gain and offset for the controller. This setting is used to adjust the values on the Omni compared to a meter reading if required.

Omni Reading #1 & #2: The value that the Omni controller shows on the HMI or the on-board web-server.

Measured Value #1 & #2: The actual meter measured value.

User Gain Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

User Offset Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

[Back to block tables page]

VOLTAGE OUTPUT

The Voltage Output block enables an analog output voltage (0 - 10VDC) to be provided to the terminals, based on the internal processing of the digital controller.

NODE DESCRIPTORS

Value: 0 to 100%, or a value. Depending on the IO Template.

Enable: Used to enable or disable the block's operation. If the Enable input is not connected, the block will be enabled by default.

Fault: If set to the On state, sets the internal block property Fault Value as the output value.

Output: The value sent to the ASIC.

Terminal Voltage: The current output voltage on the pin.

Status: Indicates the current status of the block.

INTERNAL PROPERTIES

Name: Specify a name for this block. The default name is Voltage Output.

Terminal Location: Select your device from the combo. The combo will list your selected controller, any attached REMs and Unassigned. Configurations cannot be sent to the controller is an Input or Output is unassigned. A block will be unassigned if you have placed more blocks than the controller has physically and when copy and pasting blocks.

Terminal: The physical terminal on which the block resides.

Template Name: The preselected template for the block. This can be edited by using the Template Manager. For more information, refer to the template manager section.

Disable Value (%): This value is set to the Output when the Enable input is False.

Fault Value (%): This value is set to the Output when the Fault input is True.

Terminal Voltage Tolerance: Analog value which is the limit for diagnostic purposes.

Is Forced: Set this internal block property to TRUE to set the related Force Value to the Output point.

Calculate Gain & Offset: Click to calculate the Gain and Offset. When the box is checked you can enter the Omni Reading and meter Measured Values to adjust the gain and offset for the controller. This setting is used to adjust the values on the Omni compared to a meter reading if required.

Omni Reading #1 & #2: The value that the Omni controller shows on the HMI or the on-board web-server.

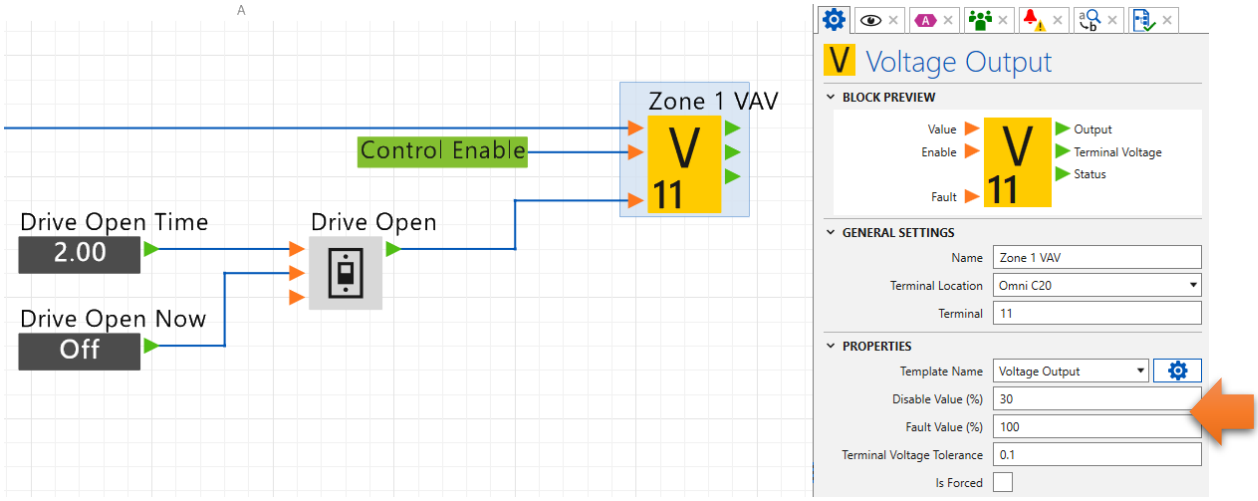
Measured Value #1 & #2: The actual meter measured value.

User Gain Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

User Offset Calibration: This reading is automatically calculated when Omni Readings and Measured Values are entered. The field can be manually edited if required. Transfer your configuration to send the settings to the controller.

Operation of the Enable and Fault nodes

The **Enable node**, if connected, allows the block to provide a signal on the output node for normal operation. If this enable signal is in an OFF state, then the output signal will be given at the enable percentage value entered via the blocks properties with zero (0) being the default. In this example, this has been set to a value of 30 which will apply a 30% output signal when the enable is off, regardless of what value is on the input Value node.



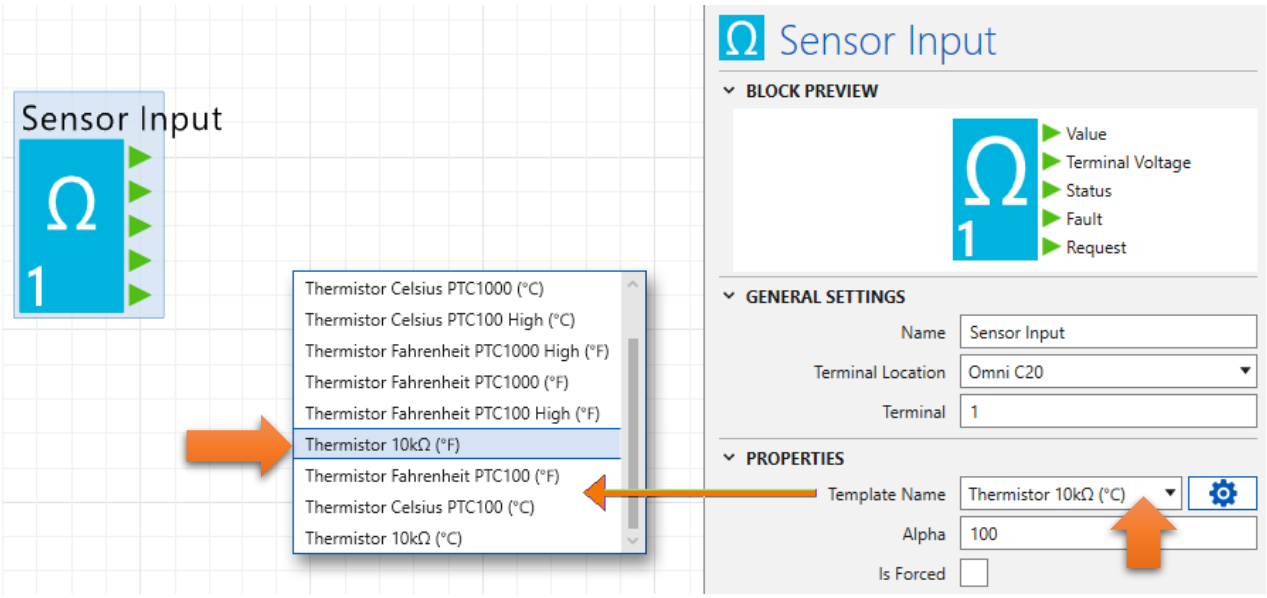
The **Fault node** allows you to connect any required digital signal to provide an ON state to drive the Output signal to the required percentage. The default is set to Zero (0) but can be any value from 0 to 100%. In this example this has been set to 100% to drive the VAV open for a period of time via the Latch block. This will override the Value Node input signal and the Enable Node if connected.

[Back to block tables page]

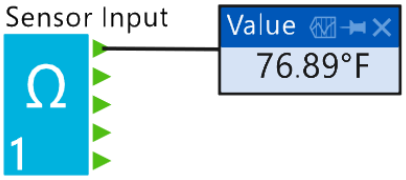
UIO TEMPLATE MANAGER

The UIO Templates Manager is used to customise the default block settings. Select the UIO Templates Manager or click the settings cog in the block's properties to open the UIO Templates Manager to edit template settings.

For example, if a sensor input is added to the config, the default template can be altered to provide the temperature reading in Fahrenheit, or many other options, rather than the default Celsius. This is selected via the drop list by clicking on the template list.

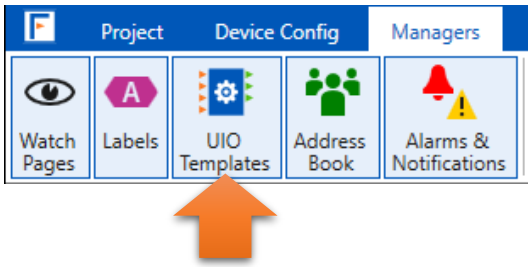


In this case, the sensor will now report the temperature in degrees Fahrenheit at the blocks output node rather than using a universal curve to perform the conversion.

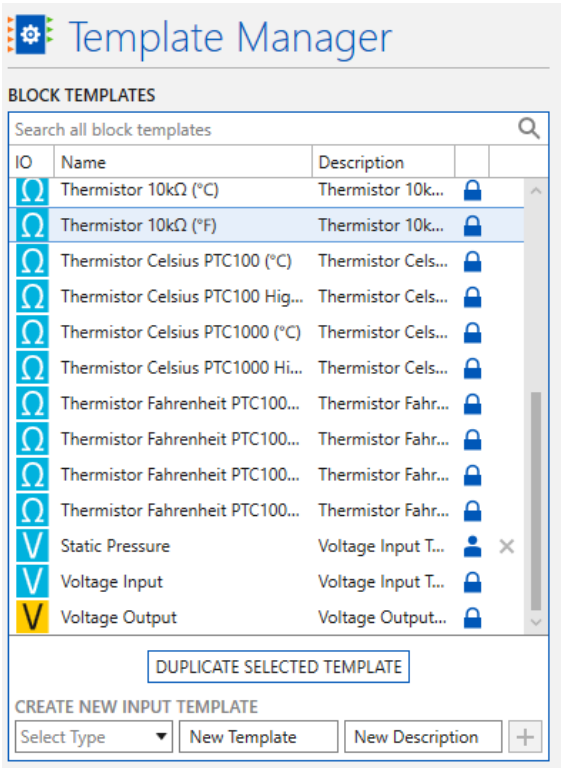


Accessing the Template Manager

Clicking the Template icon, or accessing the UIO Templates button in the Managers menu, opes the template manager.



The template manger lists and manages all the available templates for this project.



Focus comes preloaded with the standard templates upon installation.

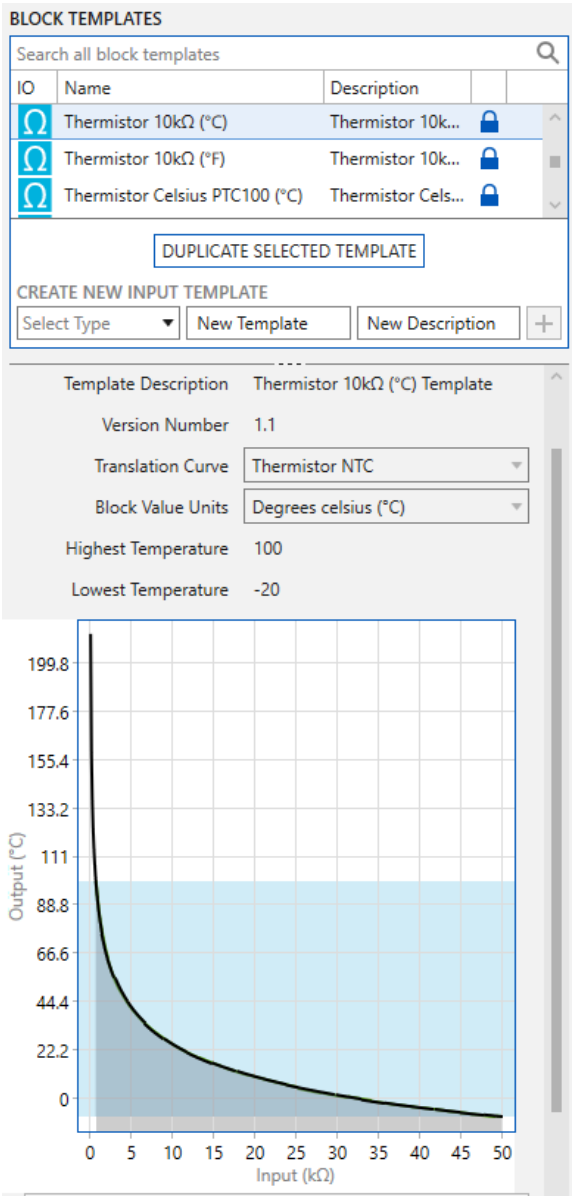
All locked templates cannot be edited or modified.

Any template can be duplicated to enable custom templates to be created.

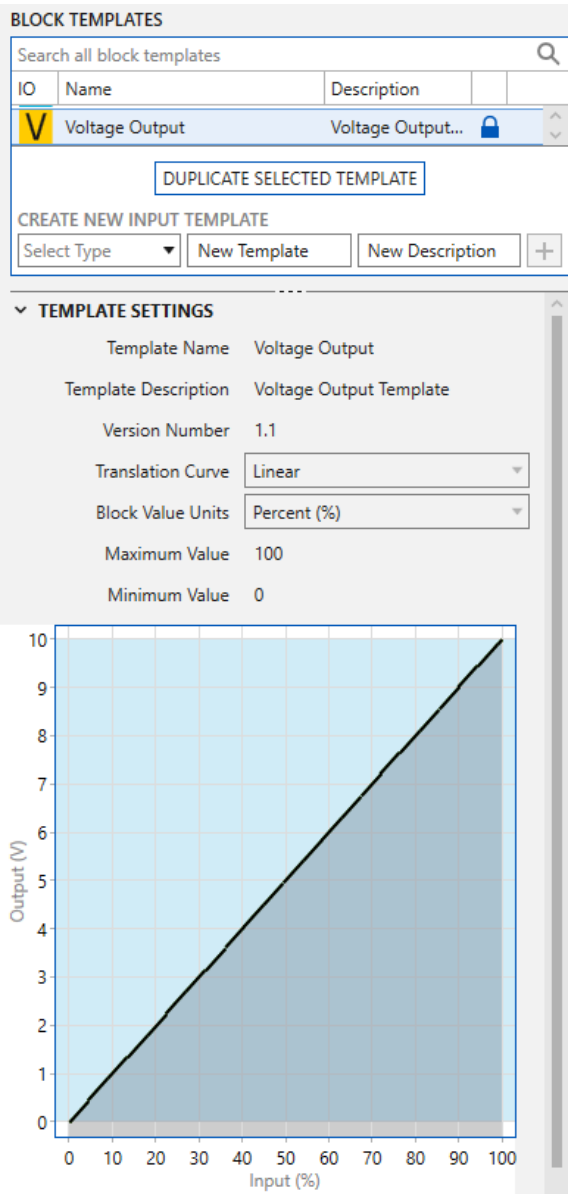
The icon beside the template name indicates the type of template created.

- This icon indicates a factory template that cannot be modified
- A project template that can be applied to all devices in the project
- A template that is part of a user collection

Clicking on a template shows the details applied to the UIO block.

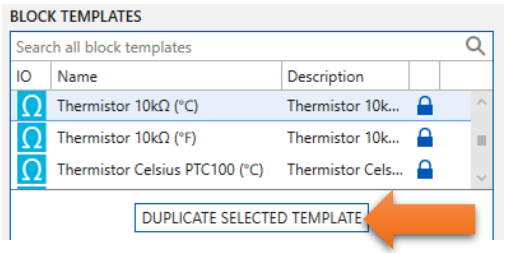


Default Thermistor °C template



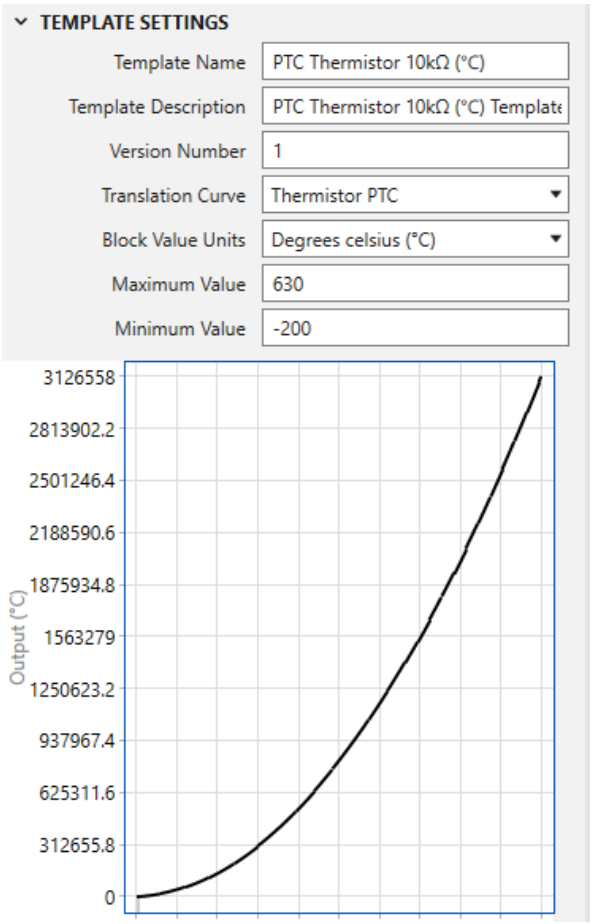
Default Voltage Output Template

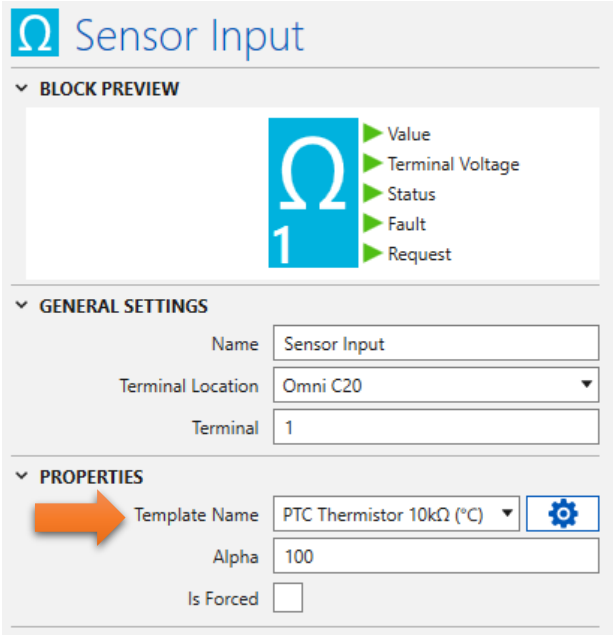
To create a custom template, click the duplicate button to copy the selected template and enable editing.



Once duplicated, the template becomes editable and you can alter the available fields as required for your custom requirements.

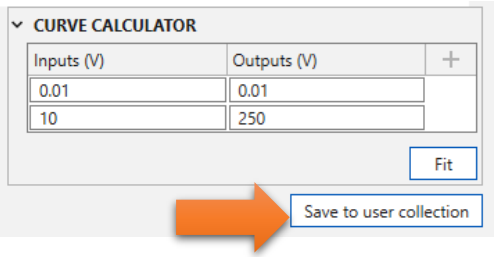
This new template can then be applied to the required UIO block and the operation checked.





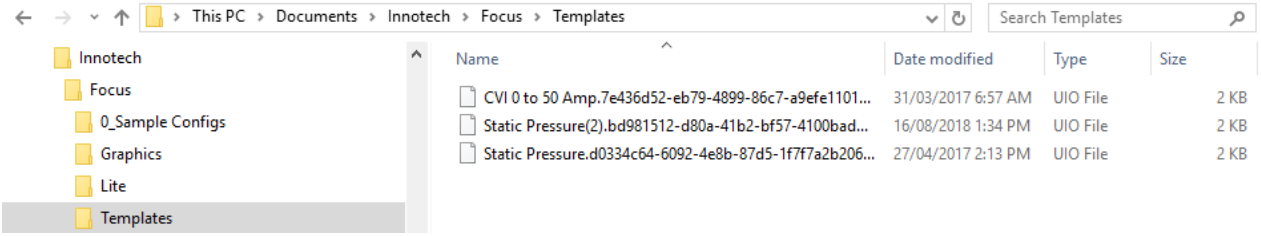
Template user collections

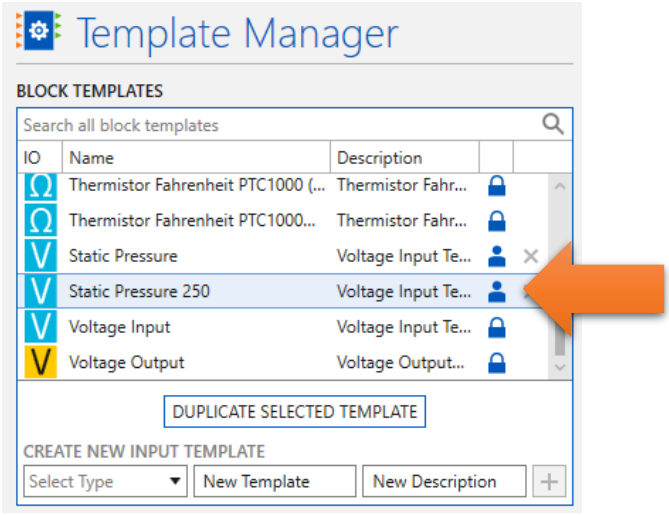
To add Templates to a user collection, it must first be duplicated as shown above. Click the Save to user collection button at the bottom of the template to save the duplicated template.



Templates are saved to: C:\Users\[USERNAME]\Documents\Innotech\Focus\Templates. Files are saved with the extension .uio.

Focus checks this directory when loading and adds the templates in this folder to the Template Manager.





These files can be shared with others and new shared files can be added by pasting the .uio file into this folder.

When a template is used in a Focus project, it is also saved with the .focus project file. If you share the focus file to another user, when the file is opened they will see the unique template in the project and can save it to their own collection. Custom user templates can be deleted directly from the Template Manager (by clicking X) or by deleting from the C:\Users\[USERNAME]\Documents\Innotech\Focus\Templates directory.

Note: If you delete a template from the directory while Focus is running, the template list will be updated next time Focus is run.

[Back to block tables page]

Product Support

Direct support and product information can be obtained via Internet, Email, Fax or Mail.

Internet: www.innotech.com
Email: sales@innotech.com
Phone: +61 7 3421 9100
Fax: +61 7 3421 9100
Address: PO Box 292
Sunnybank
QLD, 4109
Australia