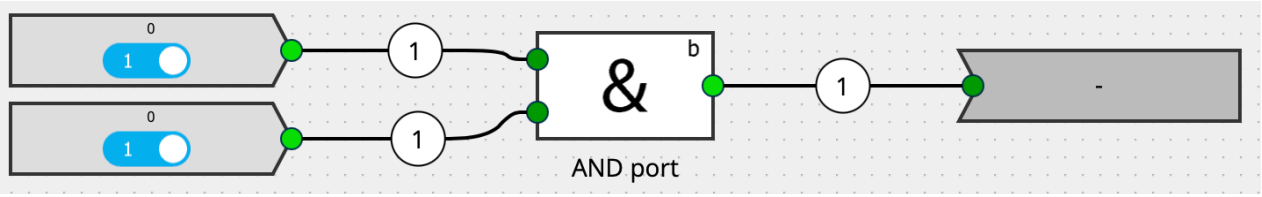
xxter logic manual

Version 1.7 – October 2025

xxter logic introduction

With the xxter logic module you can create and test logic schematics in an intuitive graphical environment, and apply it in the home or building automation.



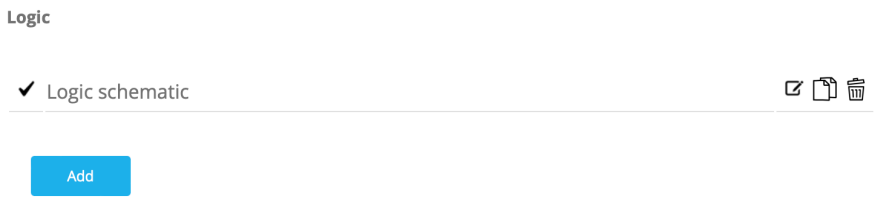
There are many different logical blocks available, varying from simple AND ports to virtual dimmers. By combining multiple blocks in one schema, you can create complex automations.

This manual explains how a logic schematic can be created, how you can test it and how to commission it. As an addendum to this manual an overview is included of all the existing logic elements, with a complete explanation how they can be used.

More information about xxter can be found on our website www.xxter.com and on our forum forum.xxter.com. On our website, you can also find our installation and user manuals.

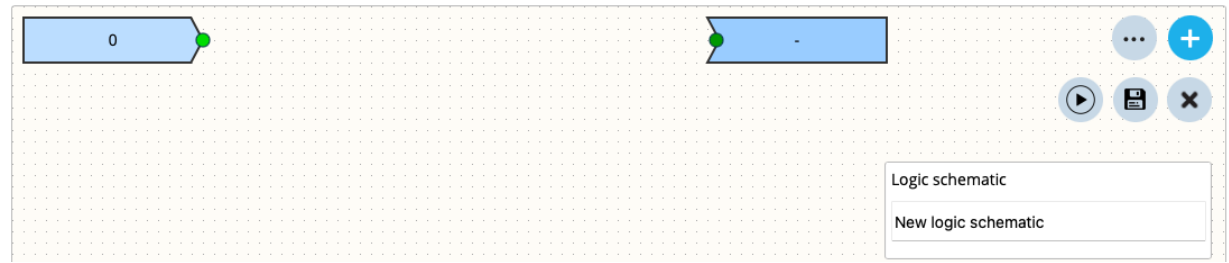
Configuring logic

The xxter logic configuration is done online, in the *My xxter* environment (https://my.xxter.com/). Login with your professional account and select the xxter project for which you want to configure a logic schematic. Select the “Logic” option on the left-hand side.



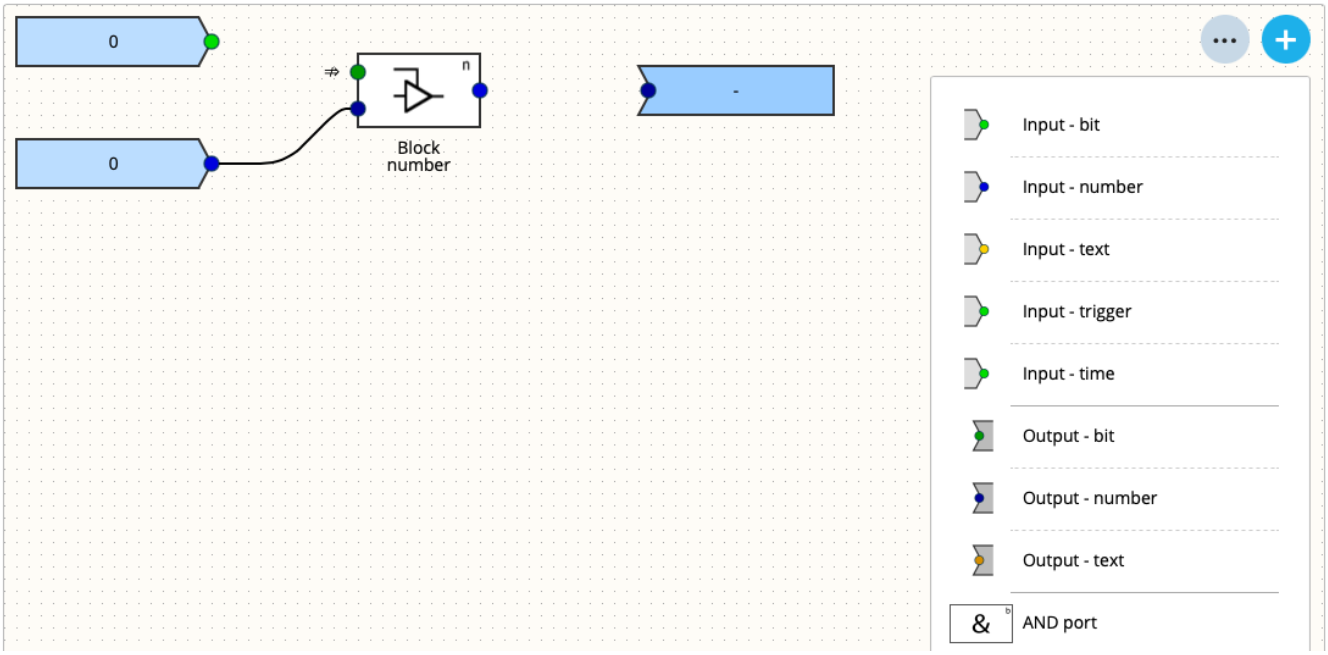
On this page you can add a new logic schematic and edit, duplicate and delete existing schematics. You can disable a schematic (so it will not be executed) by clicking on the checkbox in front of the name.

When you add or edit a schematic, the schema editor will open. In a new schematic, by default a binary input and binary output will be displayed. By clicking on the plus icon on the top right of the screen, you can add logic elements. By clicking on the three dots, you can rename a schematic, save it and close the editor. From this menu, you also start the schematic simulator.

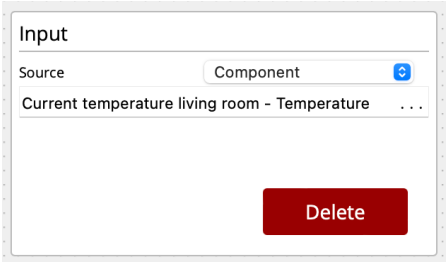


Every input, output or logical block is clearly colour coded on the connectors, indicating which data type can be connected. Green means a binary input (1 or 0), blue a numeric value and yellow a textual value. Outgoing connectors are lighter in colour and an incoming connector is darker in colour. Logical blocks can only be connected to connectors of the same type. You can connect up to 10 lines to one connector (incoming or outgoing).

You can connect elements by dragging your mouse from one connector to the other. It is not mandatory to always connect all connectors of a logical block. If no connector is connected, that input is simply ignored.



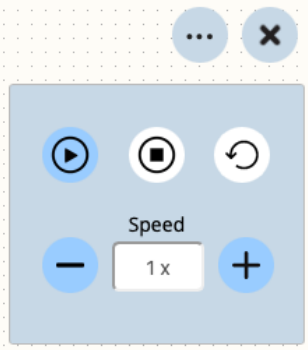
Every input, output or logical block has parameters. For instance, an input can be set up as a constant, or be linked to a component from the automation.



By clicking on the element, the parameters will be shown on the right-hand side of the screen. From this panel you can also delete an element. An overview of all elements and their parameters is included at the end of this manual.

When you have created a logic schematic, don't forget to save it, using the icon with the three dots. From this menu, you can also close the schematic editor.

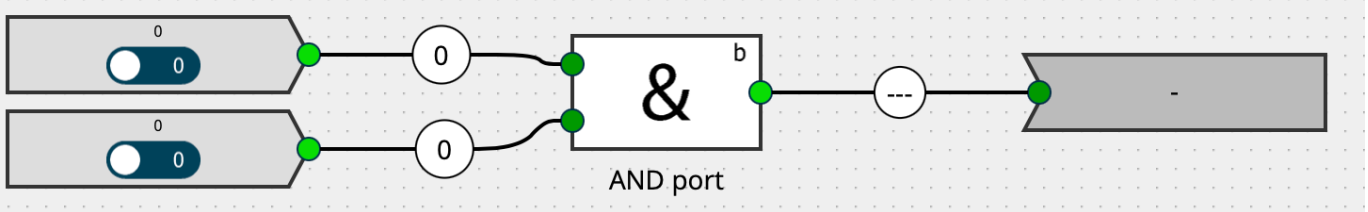
Testing logic



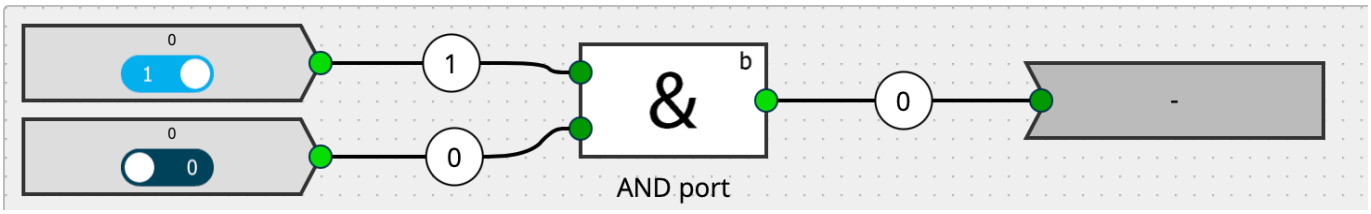
Before commissioning logic in the automated home or building, we advise to always test the schematic first. By clicking on the icon with the three dots and then on the play icon, the schematic will reopen in simulation mode. In this mode it is not possible to make changes to the schematic. By pressing the cross icon in the top right corner, you will return to edit mode.

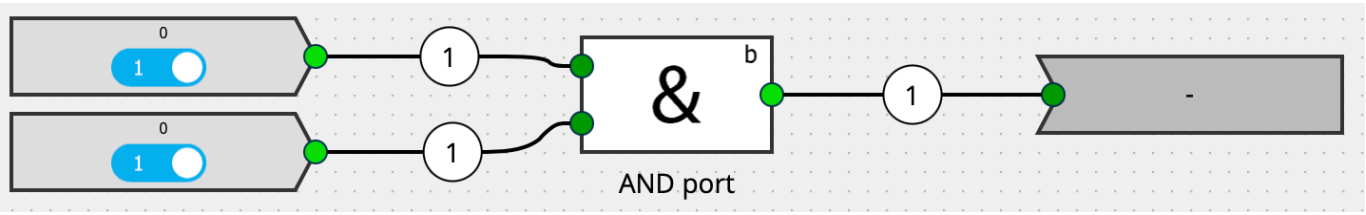
In simulation mode, you can use the menu with the three dots to start and stop the simulation, reset the simulation or change the speed of the simulation. The speed is particularly relevant for logical blocks that perform actions over time, like for instance a light timer, watchdog or delay module.

In simulation mode, you can manually assign virtual values to all inputs. Initially all values are always 0.



When you change a value of an input, the connected logical block is activated and the logic is performed. On the outgoing connector(s) the value will be shown, which is the result of the logic. This way, you can test if the logic schematic works as intended.





Commissioning logic

When you are happy with the logic schematic you created, don't forget to save it, before closing the editor.

By loading the appropriate xxter project, for which you have created the schematic, on the xxter controller, the logic schematic will become active.

To do this, log in on the xxter controller and press "Load configuration".

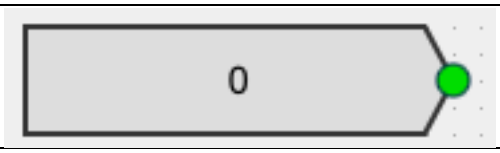
You can verify how the logic is running real time, by enabling the user log for logic, on the *Basic – Settings* page of the xxter controller. When you open the user log, you can see that the input changes for any logical block are logged as well as the resulting output. Every logical block has a unique ID, which can be found in the online logic editor in the *Parameters* window.

Addendum: Logic elements

There are many different logic elements that can be used in a logic schematic. In this addendum all existing logic elements are listed with an explanation how they work, which parameters are available and what the available inputs and outputs are.

Inputs

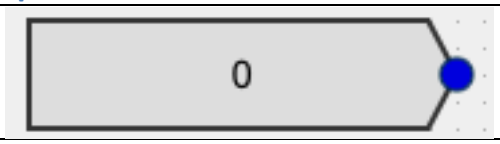
1. Input - bit

	Binary input that can be used as an input for other logic blocks.

Parameters:
- Type:
  ○ Component
  ○ Constant value
- Constant value – Depending type: Constant for the input, either 1 / ON or 0 / OFF
- Component – Depending type: Binary value from the project	

IN: - n/a	OUT: - Binary value, depending on parameters

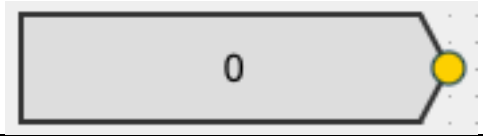
2. Input - number

	Number input that can be used as an input for other logic blocks.

Parameters:
- Type:
  ○ Component
  ○ Constant value
- Constant value – Depending type: Constant for the input, numeric value
- Component – Depending type: Numeric value from the project	

IN: - n/a	OUT: - Numeric value, depending on parameters

3. Input – text

	Text input that can be used as an input for other logic blocks.

Parameters:
- Type:
  ○ Component
  ○ Constant value
- Constant value – Depending type: Constant for the input, textual value
- Component – Depending type: Textual value from the project	

IN: - n/a	OUT: - Textual value, depending on parameters

4. Input – trigger

	Binary input that can be used as an input for other logic blocks. The binary output will be 1 / ON if the trigger is active. Only the Artnet trigger can also give a 0 / OFF as a trigger.

Parameters:

- Type:
  - HTTP trigger – Activated HTTP trigger
  - SIP trigger – Activated SIP trigger
  - DoorBird trigger – Activated trigger from a DoorBird intercom
  - Artnet trigger – Artnet trigger, gives a 1 when the value is greater than 0, otherwise 0
  - Presence detection – Detected presence of one or more persons
  - Page opened – Opened page in the visualization
  - Location trigger – Detected presence based on an iBeacon
- Trigger settings – Depending type

IN:	OUT:
- n/a	- Binary value, gives 1 / ON if the condition is met

5. Input – time

	Binary input that can be used as an input for other logic blocks. The binary output will be set to 1 / ON at the start time and 0 / OFF on the end time.

Parameters:

- Weekdays – Which weekdays (Monday to Sunday) the time trigger should work
- Start time – Start time or the time before or after sunrise or sunset, when the binary output should become 1 / ON
- End time – End time or the time before or after sunrise or sunset, when the binary output should become 0 / OFF

IN:	OUT:
- n/a	- Binary value, gives 1 / ON and 0 / OFF depending on start and end time.

Outputs

6. Output– bit

	Binary output that can be used to process the result of the logic blocks.

Parameters:

- Destination:
  - Component – Component from the project
  - Scene – Scene from the project, with optional action depending on value
  - Script – Script from the project, with optional action depending on value
  - Command – Command from the project (2 options, depending on value)
  - Presence simulation – Start, Stop or Record of the simulation, depending on value
  - Alert service – Alert service, to which a value can be passed
- Details destination – Depending on destination, 1 or 2 parameters

IN:	OUT:
- Binary input	- n/a

7. Output – number

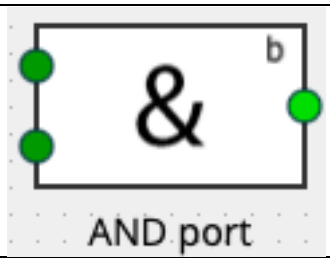
	Numeric output that can be used to process the result of the logic blocks.
Parameters: - Destination: ○ Component - Component from the project ○ Command - Command from the project, to which a value can be passed ○ Alert service - Alert service, to which a value can be passed - Details destination - Depending on destination	
IN: - Numeric input	OUT: - n/a

8. Output – text

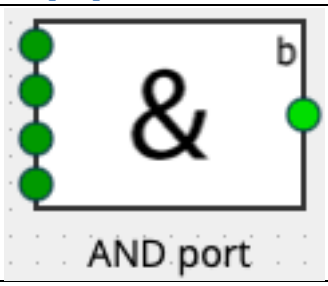
	Textual output that can be used to process the result of the logic blocks.
Parameters: - Destination: ○ Component - Component from the project ○ Alert service - Alert service, to which a value can be passed - Details destination - Depending on destination	
IN: - Textual input	OUT: - n/a

Basic blocks

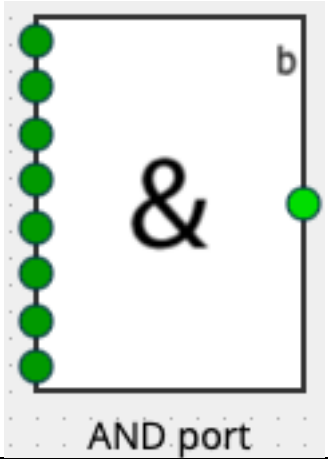
9. AND port

	Twofold AND port, giving output 1 / ON when all inputs are 1 / ON and in all other cases 0 / OFF. The output can also be inverted.
Parameters: - Send output ○ On any input - Every new value on an input produces a new output value ○ On any input change - Only if one of the input values change, an output value is sent ○ On output change - Only if the output value changes, it is sent - Output value ○ Normal - Gives “1 / ON” if the logic is valid ○ Invert - Gives “0 / OFF” if the logic is valid	
IN: - 2x Binary input	OUT: - Binary output

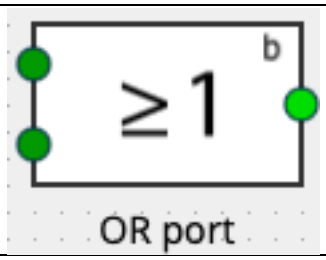
10. AND port [4v]

<table>
<tr><td> 
</td><td>Fourfold AND port, giving output 1 / ON when all inputs are 1 / ON and in all other cases 0 / OFF. The output can also be inverted.</td></tr>
<tr><td colspan="2">Parameters:
- Send output
○ On any input - Every new value on an input produces a new output value
○ On any input change - Only if one of the input values change, an output value is sent
○ On output change - Only if the output value changes, it is sent
- Output value
○ Normal - Gives “1 / ON” if the logic is valid
○ Invert - Gives “0 / OFF” if the logic is valid</td></tr>
<tr><td>IN:
- 4x Binary input</td><td>OUT:
- Binary output</td></tr>
</table>

11. AND port [8v]

<table>
<tr><td> 
</td><td>Eightfold AND port, giving output 1 / ON when all inputs are 1 / ON and in all other cases 0 / OFF. The output can also be inverted.</td></tr>
<tr><td colspan="2">Parameters:
- Send output
○ On any input - Every new value on an input produces a new output value
○ On any input change - Only if one of the input values change, an output value is sent
○ On output change - Only if the output value changes, it is sent
- Output value
○ Normal - Gives “1 / ON” if the logic is valid
○ Invert - Gives “0 / OFF” if the logic is valid</td></tr>
<tr><td>IN:
- 8x Binary input</td><td>OUT:
- Binary output</td></tr>
</table>

12. OR port



Twofold OR port, giving output 1 / ON when one the inputs is 1 / ON and if all the inputs are 0 / OFF, the output will also be 0 / OFF. The output can also be inverted.

Parameters:

- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent
- Output value
  - Normal - Gives "1 / ON" if the logic is valid
  - Invert - Gives "0 / OFF" if the logic is valid

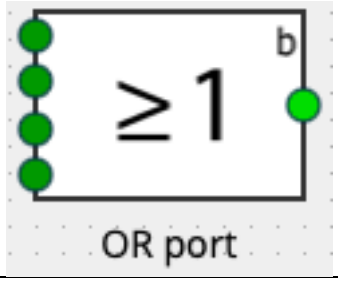
IN:

- 2x Binary input

OUT:

- Binary output

13. OR port [4v]



Fourfold OR port, giving output 1 / ON when one the inputs is 1 / ON and if all the inputs are 0 / OFF, the output will also be 0 / OFF. The output can also be inverted.

Parameters:

- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent
- Output value
  - Normal - Gives "1 / ON" if the logic is valid
  - Invert - Gives "0 / OFF" if the logic is valid

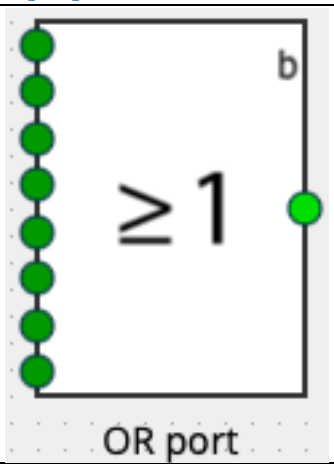
IN:

- 4x Binary input

OUT:

- Binary output

14. OR port [8v]



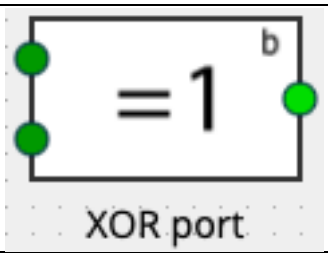
Eightfold OR port, giving output 1 / ON when one the inputs is 1 / ON and if all the inputs are 0 / OFF, the output will also be 0 / OFF. The output can also be inverted.

Parameters:

- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent
- Output value
  - Normal - Gives "1 / ON" if the logic is valid
  - Invert - Gives "0 / OFF" if the logic is valid

IN:	OUT:
- 8x Binary input	- Binary output

15. XOR port



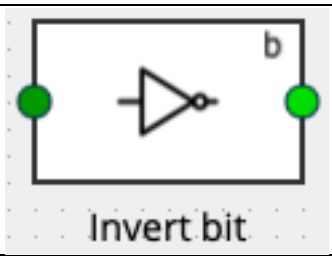
Twofold exclusive OR port, giving output 1 / ON when exactly one the inputs is 1 / ON and the other input is 0 / OFF. Otherwise, the output will be 0 / OFF. The output can also be inverted.

Parameters:

- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent
- Output value
  - Normal - Gives "1 / ON" if the logic is valid
  - Invert - Gives "0 / OFF" if the logic is valid

IN:	OUT:
- 2x Binary input	- Binary output

16. Invert bit

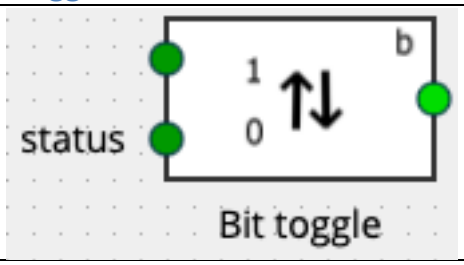


Binary inverter, giving output 1 / ON if the input is 0 / OFF and 0 / OFF if the input is 1 / ON.

Parameters:
- Send output
  - On any input - Every new value on an input produces a new output value
  - On output change - Only if the output value changes, it is sent

IN:	OUT:
- 1x Binary input	- Binary output

17. Bit toggle

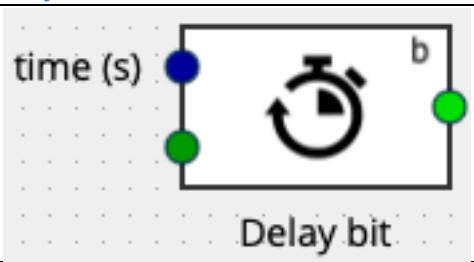


Toggle, changing the output (1 / ON > 0 / OFF or 0 / OFF > 1 / ON) on every binary input or pulse. The status can be provided as additional input to be inverted on toggle.

Parameters:
- Toggle on
  - 0 (OFF) - Only change the output if a 0 is received
  - 1 (ON) - Only change the output if a 1 is received
  - 0 and 1 (OFF/ON) - Always change the output, regardless whether a 0 or 1 is received
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

IN:	OUT:
- Binary input providing the pulse, leading to the toggle - Binary status, that should be inverted at toggle	- Binary output, giving the inverted value of the status input, according to the parameters

18. Delay bit



Delayed binary output, where the incoming telegram or pulse is sent on after the provided time.

Parameters:

- Delay
  - 0 (OFF) - Only change the output if a 0 is received
  - 1 (ON) - Only change the output if a 1 is received
  - 0 and 1 (OFF/ON) - Always change the output, regardless if a 0 or 1 is received
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent
- Restart timer
  - "-" - A new value on the input while the timer is running, will be ignored
  - On input - A new value on the input will always reset the timer

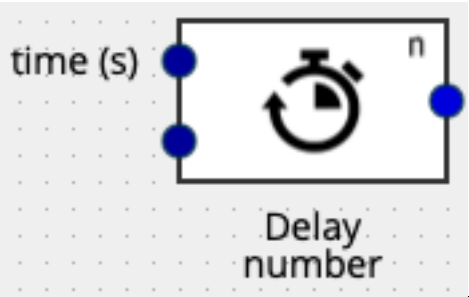
IN:

- Numeric input, providing the delay to be used in seconds
- Binary input, that is to be delayed

OUT:

- Binary output, that will be the same as the binary input, after the provided time has passed

19. Delay number



Delayed numeric output, where the incoming numeric value is sent on after the provided time.

Parameters:

- Send output
  - On any input - Every new value on an input produces a new output value
  - On output change - Only if the output value changes, it is sent
- Restart timer
  - "-" - A new value on the input while the timer is running, will be ignored
  - On input - A new value on the input will always reset the timer

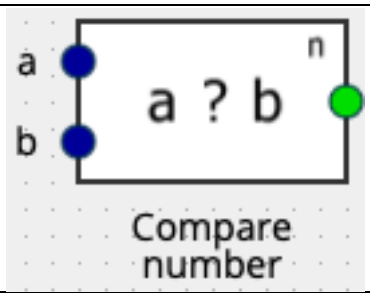
IN:

- Numeric input, providing the delay to be used in seconds
- Numeric input, that is to be delayed

OUT:

- Numeric output, that will be the same as the numeric input, after the provided time has passed

20. Compare number



Compares both incoming values according to the set parameter and provides the result as a binary output. For example, if the parameter is set as “A=B” and values A and B are both the same, the output will be 1 / ON, and otherwise 0 / OFF.

Parameters:

- Compare value
  - A < B - A must be smaller than B
  - A ≤ B - A must be smaller or equal to B
  - A = B - A must be equal to B
  - A > B - A must be greater than B
  - A ≥ B - A must be greater or equal to B
  - A <> B - A must not be equal to B
- Send output
  - On any input - Every new value on an input produces a new output value
  - On output change - Only if the output value changes, it is sent

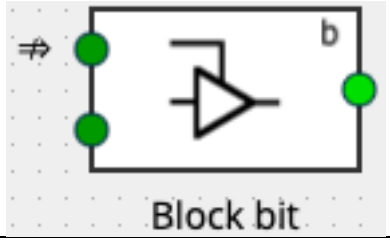
IN:

- Numeric value A, to compare
- Numeric value B, to compare

OUT:

- Binary output, providing the result of the comparison

21. Block bit



Only passes the incoming binary value through as an output when it is not blocked.

Parameters:

- Block input if
  - Block = 0 - Block the input, if the blocking is BIT is set to 0 / OFF
  - Block = 1 - Block the input, if the blocking is BIT is set to 1 / ON
- Send when start blocking - Option to send an output when the input is blocked: disabled, 0 / OFF or 1 / ON
- Send when stop blocking - Option to send an output when the input is no longer blocked: either disabled, 0 / OFF, 1 / ON or the current input value
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

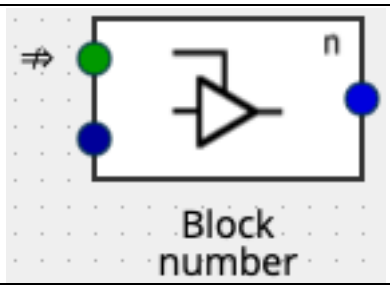
IN:

- Binary input, with the blocking BIT
- Binary input, to pass through

OUT:

- Binary output, passing on the incoming binary value if it is not blocked

22. Block number



Only passes the incoming numeric value through as an output when it is not blocked.

Parameters:

- Block input if
  - Block = 0 - Block the input, if the blocking is BIT is set to 0 / OFF
  - Block = 1 - Block the input, if the blocking is BIT is set to 1 / ON
- Send when stop blocking - Option to send an output when the input is no longer blocked: either disabled or the current input value
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

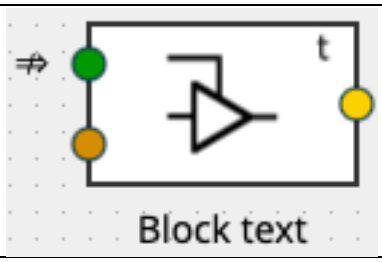
IN:

- Binary input, with the blocking BIT
- Numeric input, to pass through

OUT:

- Numeric output, passing on the incoming number if it is not blocked

23. Block text



Only passes the incoming text value through as an output when it is not blocked.

Parameters:

- Block input if
  - Block = 0 - Block the input, if the blocking is BIT is set to 0 / OFF
  - Block = 1 - Block the input, if the blocking is BIT is set to 1 / ON
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

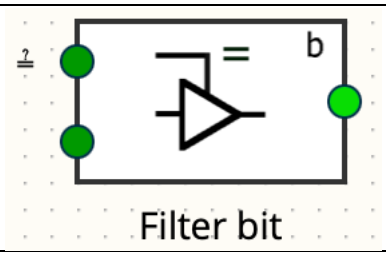
IN:

- Binary input, with the blocking BIT
- Text input, to pass through

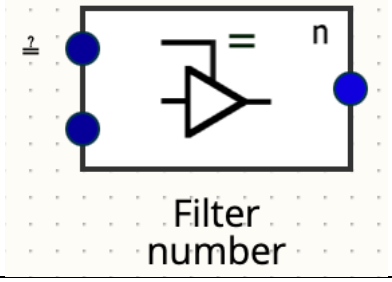
OUT:

- Text output, passing on the incoming text if it is not blocked

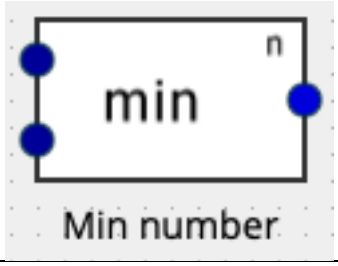
24. Filter bit

<table>
<tr><td> 
</td><td>Only passes the incoming (bottom) binary value through as an output when it is the same as the filter (top) value.</td></tr>
<tr><td colspan="2">Parameters:
N/A</td></tr>
<tr><td>IN:
- Binary input, with the filter value BIT
- Binary input, to pass through</td><td>OUT:
- Binary output, passing on the incoming binary value if it is the same as the filter value</td></tr>
</table>

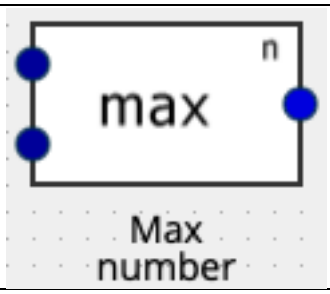
25. Filter number

<table>
<tr><td> 
</td><td>Only passes the incoming (bottom) numeric value through as an output when it is the same as the filter (top) value.</td></tr>
<tr><td colspan="2">Parameters:
- N/A</td></tr>
<tr><td>IN:
- Numeric input, with the filter value
- Numeric input, to pass through</td><td>OUT:
- Numeric output, passing on the incoming number if it is the same as the filter value</td></tr>
</table>

26. Min number

<table>
<tr><td> 
</td><td>Provides the minimal value of both numeric inputs as an output.</td></tr>
<tr><td colspan="2">Parameters:
- Send output
○ On any input - Every new value on an input produces a new output value
○ On any input change - Only if one of the input values change, an output value is sent
○ On output change - Only if the output value changes, it is sent</td></tr>
<tr><td>IN:
- 2x Numeric input</td><td>OUT:
- Numeric output, equal to the lowest value of the inputs</td></tr>
</table>

27. Max number



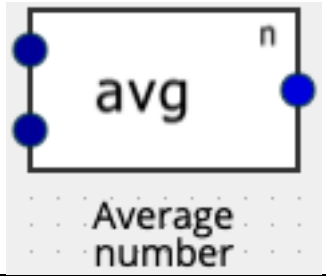
Provides the maximum value of both numeric inputs as an output.

Parameters:
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

IN:
- 2x Numeric input

OUT:
- Numeric output, equal to the highest value of the inputs

28. Average number



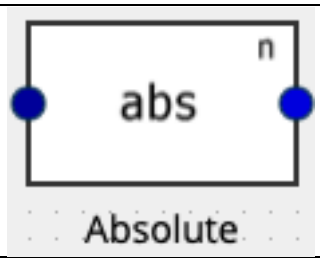
Provides the average value of both numeric inputs as an output.

Parameters:
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

IN:
- 2x Numeric input

OUT:
- Numeric output, equal to the average value of the inputs

29. Absolute



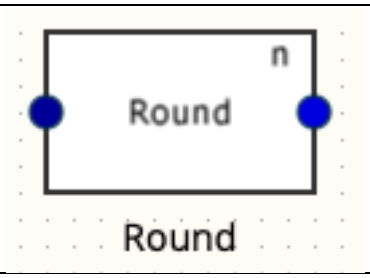
Provides the absolute value of the numeric input as an output. So, a negative value is made positive. For instance, -20 becomes 20. Positive values remain unchanged.

Parameters:
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

IN:
- Numeric input

OUT:
- Numeric output, equal to the absolute value of the input

30. Round number

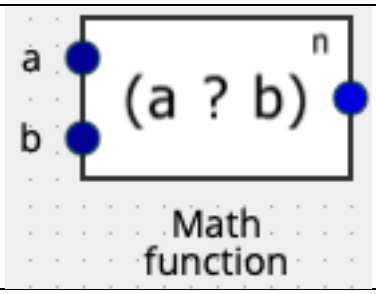


Rounds the numeric input to the provided amount of decimals. For instance, input 1.3889 can be rounded to “1”, “1.4” or “1.39”.

Parameters:
- Decimals - Amount of decimals behind the . where “0” will round the number to an integer.

IN:	OUT:
- Numeric input	- Numeric output, rounded

31. Math function

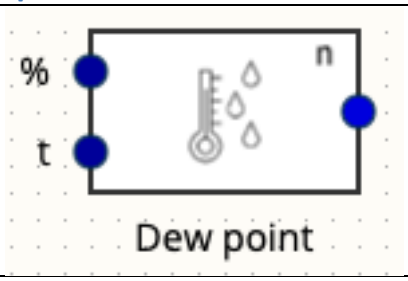


Performs the set math function on both inputs and provides it as an output.

Parameters:
- Function
  - a + b - Add
  - a - b - Subtract
  - a x b - Multiply
  - a / b - Divide
  - a ^ b - Exponentiation
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

IN:	OUT:
- 2x Numeric input	- Numeric output, as a result of the math function on both inputs

32. Dew point



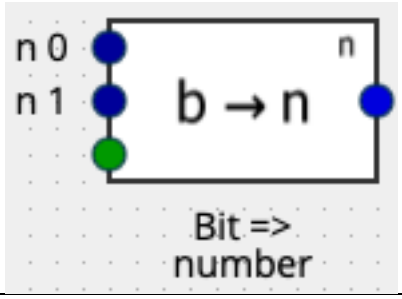
Calculates the dew point temperature based on the relative humidity and current temperature, using ‘Sonntags formula’.

Parameters:
- n/a

IN:	OUT:
- Numeric input with the relative humidity - Numeric input with the temperature	- Numeric output with the dew point (temperature)

Conversion

33. Bit => number



Convert a binary value into a numeric value, based on the provided inputs.

Parameters:

- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

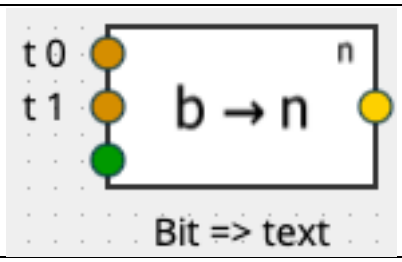
IN:

- Numeric input with the value for binary input 0 / OFF
- Numeric input with the value for binary input 1 / ON
- Binary input

OUT:

- Numeric output, equal to input n 0 if the binary input is 0 / OFF and equal to n 1 if it is 1 / ON

34. Bit => text



Convert a binary value into a text value, based on the provided inputs.

Parameters:

- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

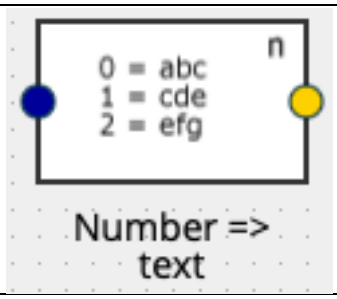
IN:

- Text input with the value for binary input 0 / OFF
- Text input with the value for binary input 1 / ON
- Binary input

OUT:

- Text output, equal to input t 0 if the binary input is 0 / OFF and equal to t 1 if it is 1 / ON

35. Number => text

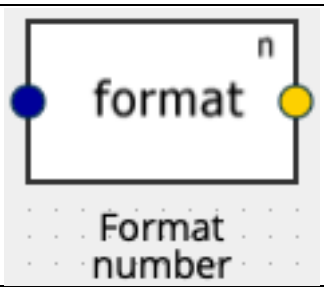


Convert a numeric value into a text value, based on the set parameters. An unknown numeric value is ignored and does not lead to a new output.

Parameters:
- 20x [Numeric value]
  - Text value - Translation table of 20 numeric values to be converted into text

IN:	OUT:
- Numeric input	- Text output, based on the translation table in the parameters

36. Format number

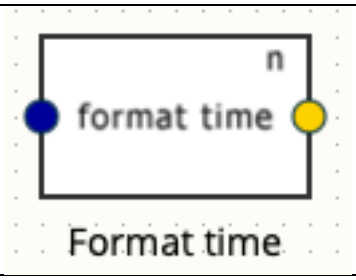


Formats a numeric value as a text, for instance the number 19,3512 as “Max 19,35 KWh”.

Parameters:
- Decimals - Amount of digits after the comma, to which should be rounded down
- Prefix text - Text to be placed in front of the number
- Suffix text - Text to be placed behind the number

IN:	OUT:
- Numeric input	- Text output, displaying the numeric value as a formatted text

37. Format time



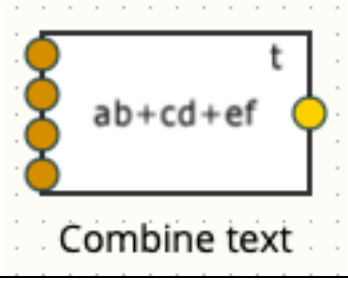
Formats a numeric value to a timestamp, for instance 55270 seconds into “16:21:10”.

The output format always uses 24 hour clock.

Parameters:
- Input is - Value of input is in [msec, sec, min] i.e. in milliseconds, seconds or minutes
- Output format - Text is shown in [h:m, h:m:s, h.m, h.m.s]

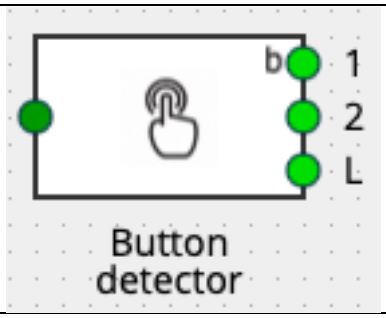
IN:	OUT:
- Numeric input	- Text output, displaying the numeric timestamp as a formatted text

38. Combine text

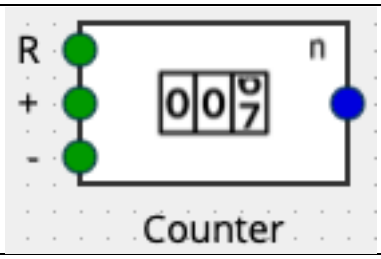
	Combines the 4 text inputs into 1 output, for instance the inputs “There is “ - “no” - “ presence since ” - “14:35” will produce output “There is no presence since 14:35”.
Parameters: - n/a	
IN: - 4x Text input	OUT: - Text output, combining all input text

Intelligent elements

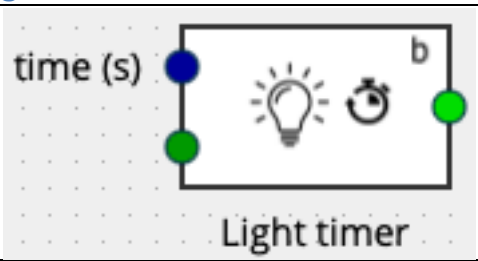
39. Button detector

	Interprets the pulse from a push button into three possible outputs: single push, double push and long push. Depending on the detected pulse it produces a 1 / ON for one of the three outputs.
Parameters: - n/a	
IN: - Binary input	OUT: - Binary output for single push - Binary output for double push - Binary output for long push

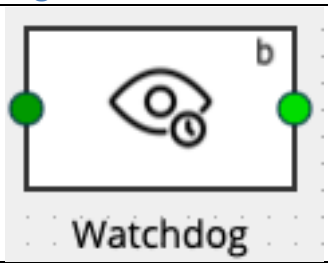
40. Counter

	Provides a numeric output based on the amount of counted pulses on the binary inputs.
Parameters: - n/a	
IN: - Binary input to reset the counter - Binary input to increase the counter - Binary input to decrease the counter	OUT: - Numeric output with the count value

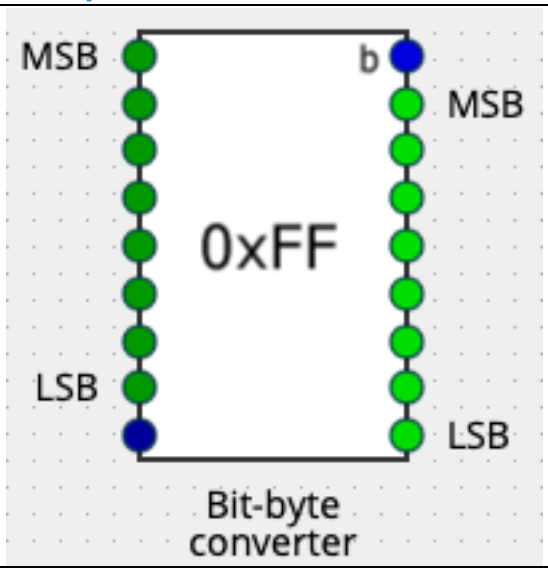
41. Light timer

<table>
<tr><td> 
</td><td>Provides a delayed shut off for the provided amount of seconds after a 1 / ON binary input. Optionally, a 0 / OFF value can cancel the timer and shut off the output immediately.</td></tr>
<tr><td colspan="2">Parameters:
- Allow immediate off
○ Yes - An input of 0 / OFF immediately sets the output to 0 / OFF as well
○ No - An input of 0 / OFF is ignored</td></tr>
<tr><td>IN:
- Numeric input with the delay time in seconds
- Binary input</td><td>OUT:
- Binary output providing 1 / ON for the duration of the provided delay time</td></tr>
</table>

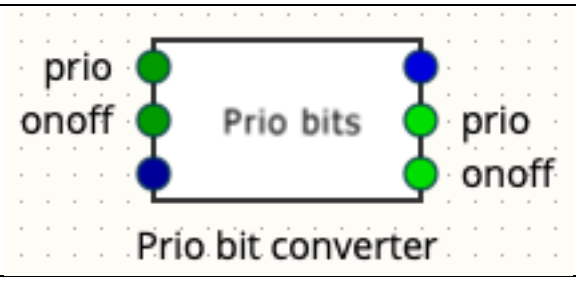
42. Watchdog

<table>
<tr><td> 
</td><td>Gives an alarm on the binary output when there has been no new input (either 0 / OFF or 1 / ON) within the provided period of time.</td></tr>
<tr><td colspan="2">Parameters:
- Watchdog timeout (s) - Amount of seconds wherein a new pulse is required on the binary input
- Send on Alarm
○ 0 - Send a 0 / OFF to the binary output as an alarm
○ 1 - Send a 1 / ON to the binary output as an alarm</td></tr>
<tr><td>IN:
- Binary input to be monitored</td><td>OUT:
- Binary output where the alarm is sent</td></tr>
</table>

43. Bit-byte converter

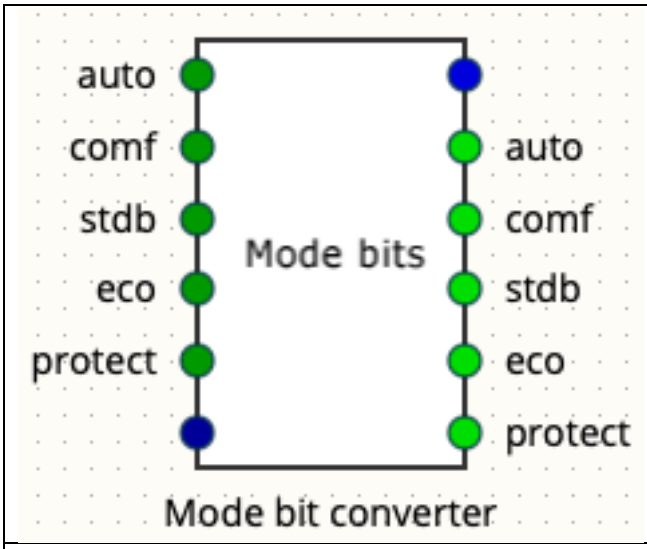
<table>
<tr><td> 
</td><td>Converts either 8 bits into 1 byte, or 1 byte into 8 bits. The order of bits is the most significant bit (128) on top and the least significant bit below.</td></tr>
<tr><td colspan="2">Parameters:
- n/a</td></tr>
<tr><td>IN:
- 8x Binary inputs from MSB (Most Significant Bit) to LSB (Least Significant Bit)
- 1x Numeric input (Byte value)</td><td>OUT:
- 1x Numeric output (Byte value)
- 8x Binary outputs from MSB (Most Significant Bit) to LSB (Least Significant Bit)</td></tr>
</table>

44. Prio bit converter

<table>
<tr><td> 
</td><td>Converts either a prio bit into 2 bits, or 2 bits into a prio bit. The prio bit should be connected to numeric input/output.

The prio bit has value “0” for no priority, “2” for priority off and “3” for priority on.</td></tr>
<tr><td colspan="2">Parameters:
- n/a</td></tr>
<tr><td>IN:
- 1x Bit input priority
- 1x Bit input on/off value
- 1x Numeric input (Prio bit value)</td><td>OUT:
- 1x Numeric output (Prio bit value)
- 1x Bit output priority
- 1x Bit output on/off value</td></tr>
</table>

45. Mode bit converter



Converts either 5 HVAC mode bits into 1 HVAC mode byte, or 1 HVAC mode byte into 5 HVAC mode bits. This allows to extract or combine bits in/from a mode object for the RTC or thermostat.

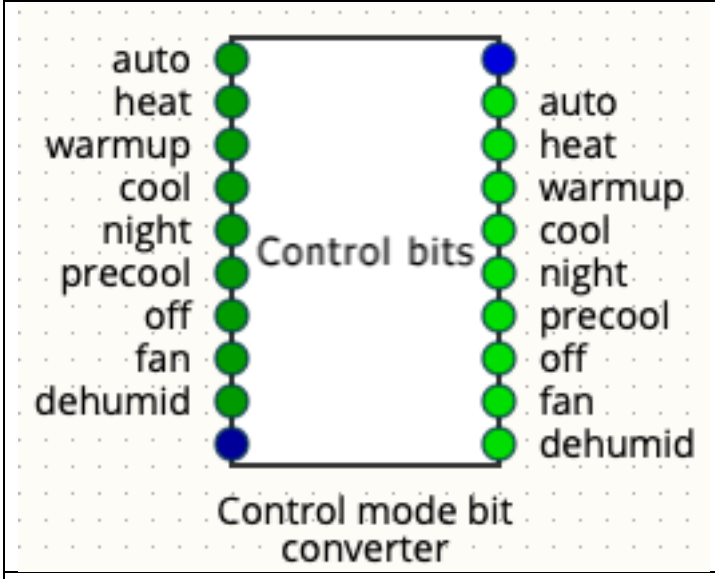
The HVAC mode byte (DPT 20.102) is connected to the numeric input/output.

Parameters:
- n/a

IN:
- 5x Binary inputs for the different modes: auto, comfort, standby, eco and building protect
- 1x Numeric input (DPT 20.102, HVAC mode)

OUT:
- 1x Numeric output (DPT 20.102, HVAC mode)
- 5x Binary outputs for the different modes: auto, comfort, standby, eco and building protect

46. Control mode bit converter



Converts either 9 HVAC control mode bits into 1 HVAC control mode byte, or 1 HVAC control mode byte into 9 HVAC control mode bits. This allows to extract or combine bits in/from a control mode object for the RTC or air-conditioning unit.

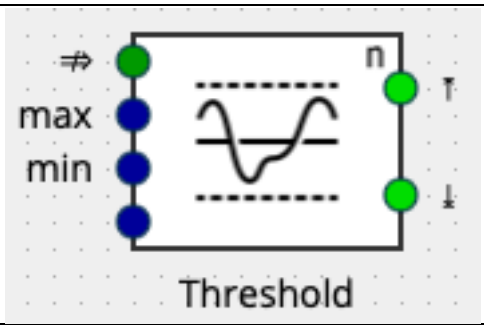
The HVAC control mode byte (DPT 20.105) is connected to the numeric input/output.

Parameters:
- n/a

IN:
- 9x Binary inputs for the different modes: auto, heating, warming up, cooling, night, pre-cooling, off, fan or dehumidification
- 1x Numeric input (DPT 20.105, HVAC control mode)

OUT:
- 1x Numeric output (DPT 20.105, HVAC control mode)
- 9x Binary outputs for the different modes: auto, heating, warming up, cooling, night, pre-cooling, off, fan or dehumidification

47. Threshold



Validates the bottom numeric input value in regard to the threshold values and sends a value to the corresponding binary output if it is out of range. The threshold validation can be blocked with a binary input.

Parameters:

- Use threshold
  - Upper - Only the upper threshold is validated
  - Lower - Only the lower threshold is validated
  - Upper and lower - Both thresholds are validated
- Block input when
  - Block = 0 - Block the input when the blocking BIT is 0 / OFF
  - Block = 1 - Block the input when the blocking BIT is 1 / ON
- Send on upper threshold
  - 0 - When the upper threshold is exceeded, send a 0 / OFF
  - 1 - When the upper threshold is exceeded, send a 1 / ON
- Send on lower threshold
  - 0 - When the lower threshold is exceeded, send a 0 / OFF
  - 1 - When the lower threshold is exceeded, send a 1 / ON
- Send output
  - On any input - Every new value on an input produces a new output value
  - On output change - Only if the output value changes, it is sent

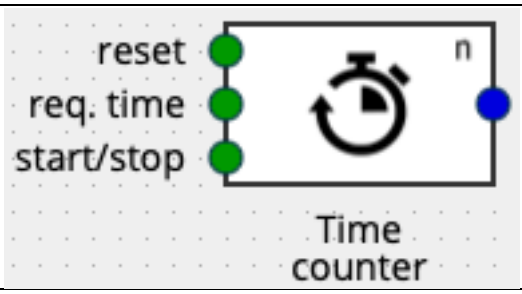
IN:

- Binary input, with the blocking BIT
- Numeric input with the upper threshold
- Numeric input with the lower threshold
- Numeric input that is to be validated

OUT:

- Binary output when the upper threshold is exceeded
- Binary output when the lower threshold is exceeded

48. Time counter



Stopwatch function, counting the time in seconds that the binary input “start/stop” is 1 / ON. The counted time is sent on the numeric output at a start/stop event, on request and if desired also cyclical. With the reset binary input, the counter can be set to 0.

Parameters:

- Send time on
  - Start, Stop and Req. - Send the time on start/stop and on request
  - S, S, R and Cyclic - Send the time on start/stop, on request and cyclic (see parameter)
- Cyclic time (in sec) - If set to cyclic, period in seconds to send the counted time

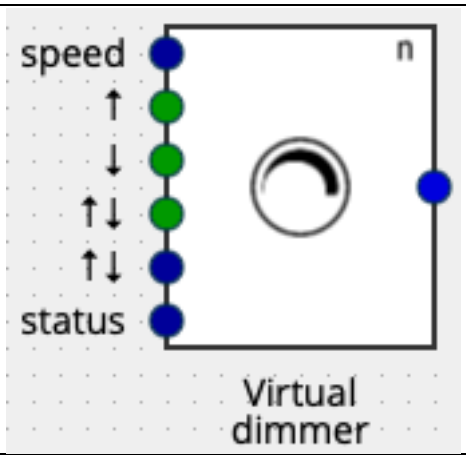
IN:

- Binary input to reset the counter
- Binary input to request the current counted time
- Binary input to start / stop the counter

OUT:

- Numeric value with the counted time in seconds

49. Virtual dimmer



Virtual dimmer allowing several different ways to increase or decrease a dimmer. You can use a binary input / pulse button (up, down or toggle) or a numeric input, to increase and decrease the dimmer. When using the numeric input, it is only relevant whether the value is negative or positive, not the actual value itself.

The speed of dimming is set with a separate numeric input, in seconds from 0 to 100.

Parameters:

- Minimum dim value - Minimal dim value that is sent out
- Under minimum dim value set to 0
  - On - Send a 0 for all values below the minimum dim value
  - Off - Sends no value under the minimum, until the value is truly 0

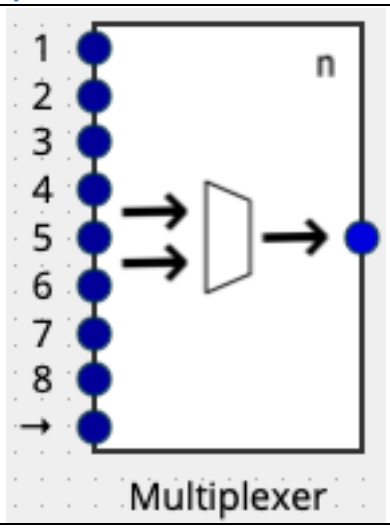
IN:

- Numeric input with the desired speed in seconds to dim from 0 to 100
- Binary input to dim up
- Binary input to dim down
- Binary input with toggle to dim up/down
- Numeric input, increasing or decreasing the dim value depending if the value is positive or negative
- Numeric input, with the current dim value (status, 0-100%)

OUT:

- Numeric output with dim value (0-100%)

50. Multiplexer



With the multiplexer, you can use the bottom numeric value as a search index to select one of the 8 other numeric input values as the output value. Using the parameters, you can configure which input refers to which search index.

When the search index matches one of the parameters of the input values, the numeric value of that input is passed through to the numeric output.

Parameters:

- [1 – 8] Input values - The values that should match the search index
- Input nr 8 behavior
  - As value - Input 8 has its own index, as set by the parameter
  - Default (all other) - Input 8 is seen as default, when the search index doesn't match any of the values set for 1 to 7

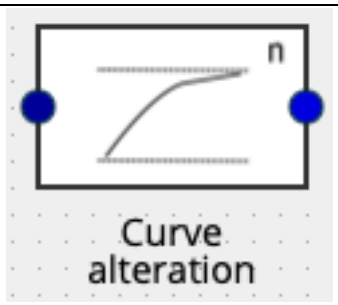
IN:

- 8x Numeric input for input 1 to 8
- Numeric input used as search index

OUT:

- Numeric output with the selected value from one of the inputs

51. Curve alteration



With curve alteration, input values can be translated to different output values to achieve a steeper or flatter curve, for instance for a dimmer. By creating a mapping of 10 points to new values, the module will automatically calculate all intermediate values.

In the parameter graph, the altered curve is displayed

Parameters:

- 10x Input – Output mapping - Table with 10 translate values, to adjust the curve as desired

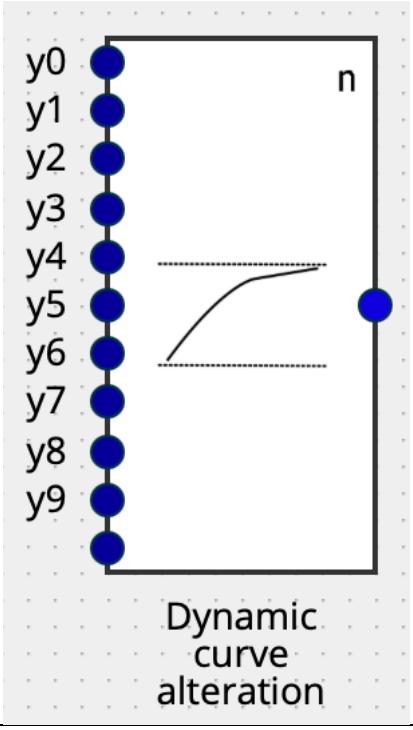
IN:

- Numeric input

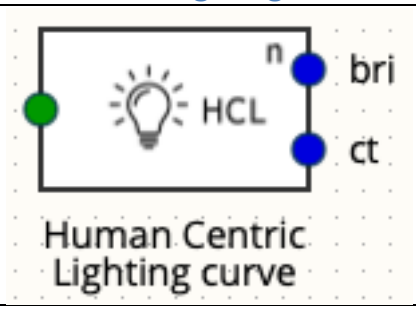
OUT:

- Numeric output

52. Dynamic curve alteration

<table>
<tr><td> 
</td><td>With the dynamic curve alteration, the bottom numeric value is translated to an output value to achieve a steeper or flatter curve, for instance for a dimmer. The mapping is created by setting 10 points that are mapped to the top 10 numeric inputs y0 to y9, to allow a dynamic alteration of the curve.</td></tr>
<tr><td colspan="2">Parameters:
- 10x Input – Output mapping - Table with 10 translate values mapped to numeric inputs y0 to y9, to adjust the curve as desired</td></tr>
<tr><td>IN:
- 10x Numeric input y0-y9 to adjust the curve
- Numeric input, that requires the curve alteration</td><td>OUT:
- Numeric output with the adjusted curve</td></tr>
</table>

53. Human Centric Lighting curve



With the HCL curve, you can generate a dynamic brightness and color temperature for lighting, matching natural lighting conditions. When activated, it will gradually change the light brightness and color during the day.

Parameters:

- Min. brightness - Minimal brightness to output
- Max. brightness - Maximum brightness to output
- Min. color temperature - Minimal color temperature to output
- Max. color temperature - Maximum color temperature to output
- Time base
  - Sunrise / sunset - Uses current sunrise and sunset times to calculate start, peak and end times
  - Time based - Uses manual entry of start, peak and end times
- Start time - Start moment to calculate the curve (before this time, output remains unchanged) *
- Peak time brightness - Moment of maximum brightness *
- Peak time color temperature - Moment of coolest color temperature *
- End time - End moment to calculate the curve (after this time, output remains unchanged) *

* Time input fields are ignored when time base = “sunrise / sunset”

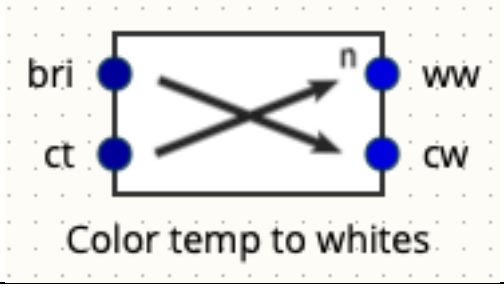
IN:

- Binary input to start / stop the curve calculation

OUT:

- Numeric output with the calculated brightness
- Numeric output with the calculated color temperature

54. Color temperature to whites



With the color temp to whites, you can control two lights, one warm white, one cool white, to create one balanced colored light, based on a brightness and color temperature input.

Parameters:

- Color temperature warm white - Color temperature of the warm white light
- Color temperature cool white - Color temperature of the cool white light

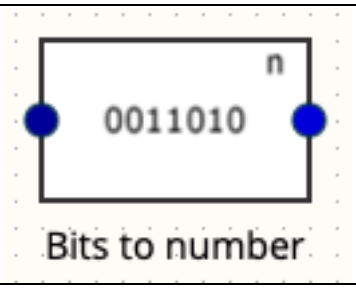
IN:

- Numeric input with brightness
- Numeric input with color temperature

OUT:

- Numeric output with the warm white brightness
- Numeric output with the cool white brightness

55. Bits to number



With bits to number, you can take a binary subset from a number, for instance to get a part of a larger Modbus address value. For instance, if you want to use the first 4 bits of a 16-bit number, the value "**1101**111110101110" (decimal 57262), will give the output "1101" (decimal 13).

Parameters:

- Total bits input [8,16,32] - Defines the bit-length of the input value
- Start bit - Bit from where to take the subset (where bit 1 is the first bit)
- Number of bits to use - Bit length of the subset (cannot be longer than total amount of bits)
- Output value is
  - Unsigned - Defines the binary output to be unsigned (> 0)
  - Signed - Defines the binary output to be signed (<= 0 or > 0)
- Send output
  - On any input - Every new value on an input produces a new output value
  - On any input change - Only if one of the input values change, an output value is sent
  - On output change - Only if the output value changes, it is sent

IN:

- Numeric input with the complete value

OUT:

- Numeric output with the binary subset