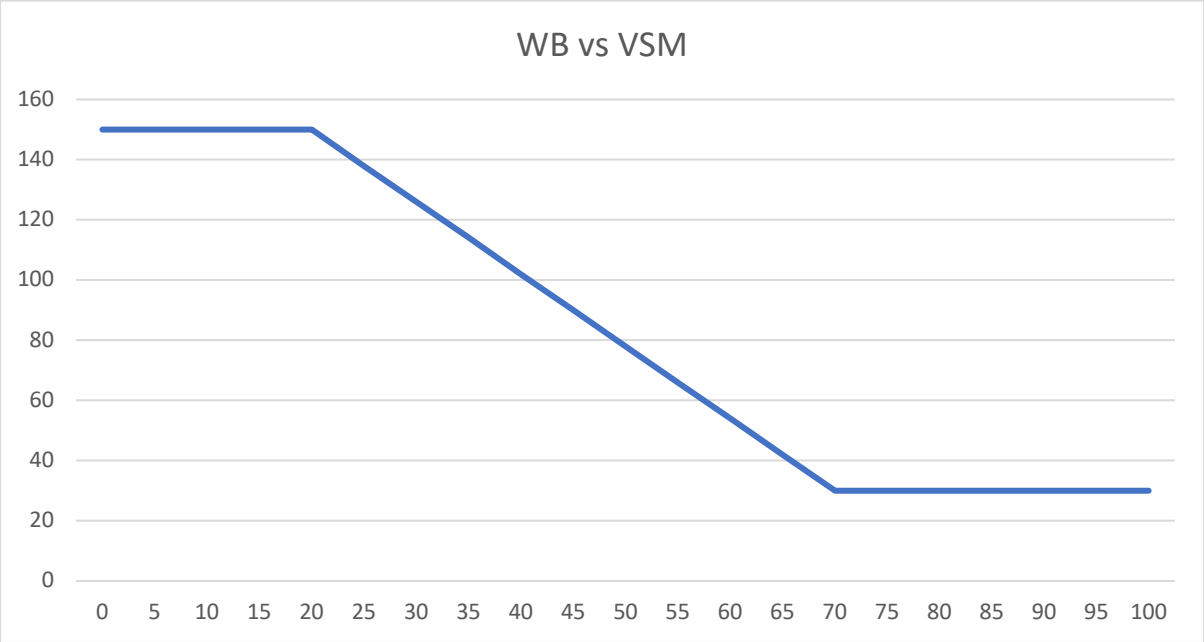
Topic:	Adjust WB on a program based on VSM% reading
Author:	Søren Haxvig
Date:	2023.10.11
To:	OEM Partners
Purpose:	Demonstrate how to use the logic engine to adjust the WB on a program based on a linear adjustment with upper and lower limits.
FW:	Minimum 2.05

Setup:

The adjustment of the WB for a program is based on a VSM% reading via a sensor decoder, which provides new values via interrupt. The adjustment is made in accordance with the graph shown below.



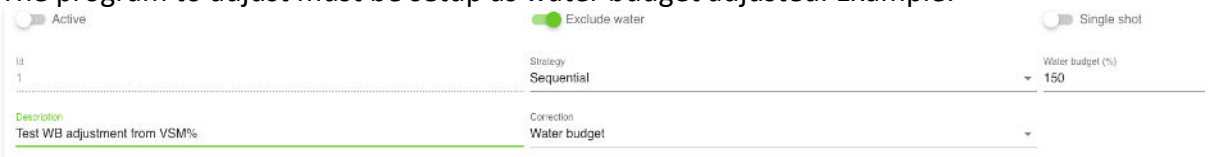
Below 20% VSM the WB is fixed to 150%. Above 70% VSM the WB is fixed to 30%.

Formula:

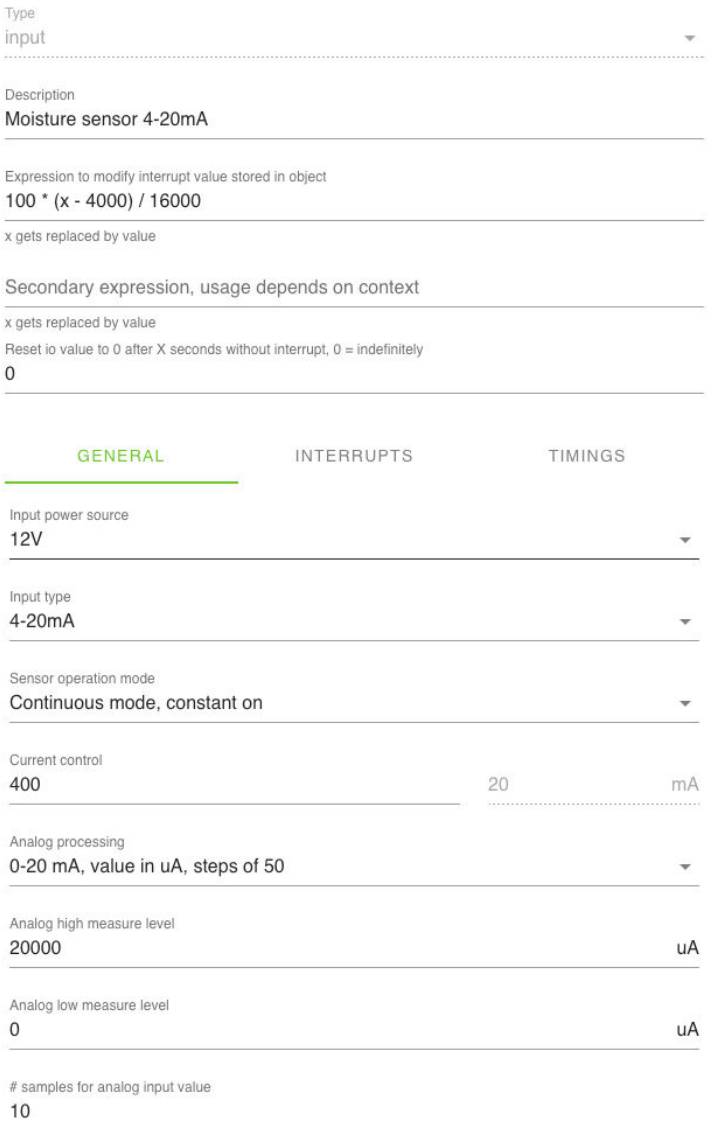
actualWB = ((minWB – maxWB)/(maxVSM – minVSM)) * (actualVSM – minVSM) + maxWB

Prerequisite:

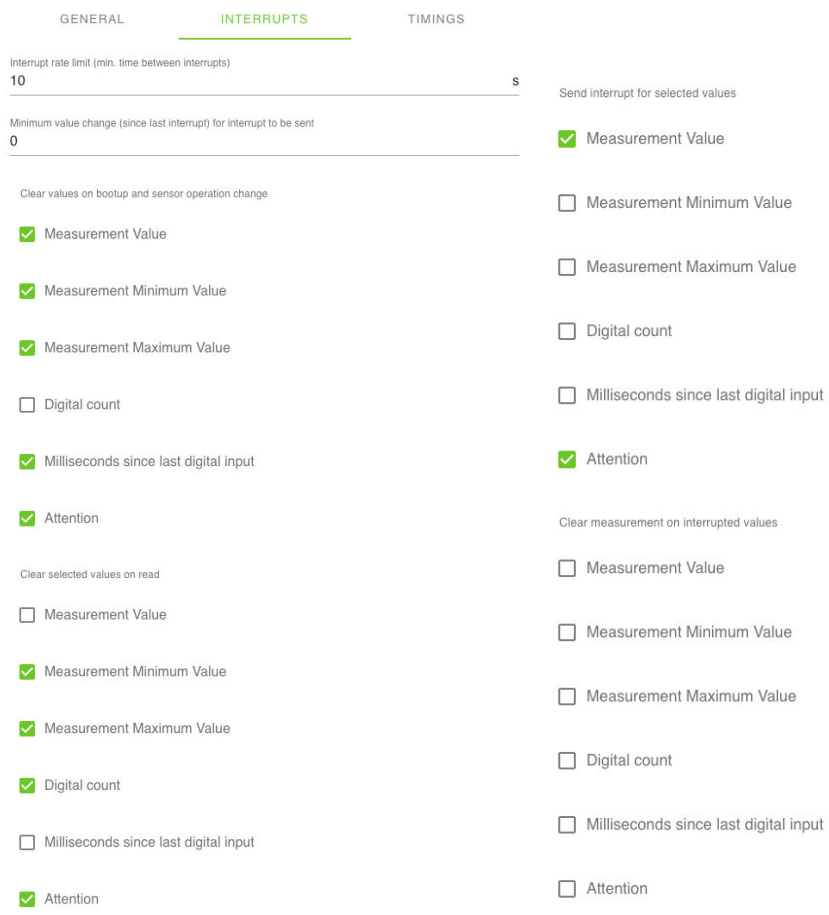
The program to adjust must be setup as water budget adjusted. Example:



The sensor decoder setup below is shown as a 4-20mA reading, but can also be a voltage input or SDI-12 type. The important is that the sensor decoder delivers new value via an interrupt.



The expression converts 4,000uA – 20,000uA into 0-100%.



The interrupt setup above ensure changes in the value are sent, but not more frequent than every 10 seconds.

Variables:

	Id	Description	State	Type	Source id	Source sub id	Value
	1	Minimum VSM %	Ok	Local	0	0	20
	2	Maximum VSM %	Ok	Local	0	0	70
	3	Minimum WB %	Ok	Local	0	0	30
	4	Maximum WB %	Ok	Local	0	0	150
	5	Program to adjust	Ok	Local	0	0	1
	6	Calculated WB %	Ok	Local	0	0	49
	7	New VSM %	Ok	IO value	679088867	1	62
	8	Last VSM %	Ok	Local	0	0	62

Variable #1 is the minimum VSM%. Must be < maximum VSM% and >= 0.
Variable #2 is the maximum VSM%. Must be > minimum VSM% and <= 100.
Variable #3 is the minimum WB%. Must be < maximum WB% and >= 1.
Variable #4 is the maximum WB%. Must be > minimum WB% and >= 300.
Variable #5 is the program number to adjust.
Variable #6 is the calculated WB.
Variable #7 is the new reading from the sensor decoder.

Variable #8 is the last VSM.

The water budget adjustment is made with a variable for easy modifications on the variable page or via the API. Using the API:

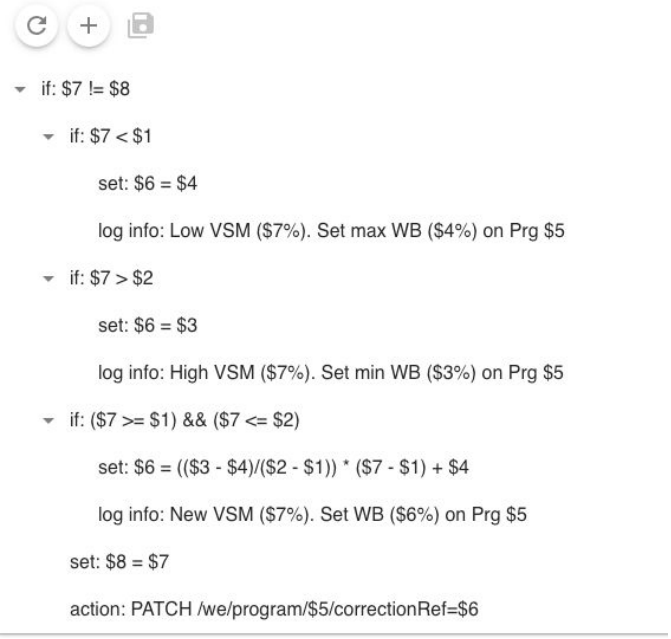
```
PATCH /le/var/1/value=5
{"result":"ok"}
Means set minimum VSM% to 5.
```

The sensor decoder reading the moisture must be set with Source id (address) rather than the value. Via API:

```
PATCH /le/var/1/sourceId= 679088867
{"result":"ok"}
Means set address to 679088867.
```

The other non-internal variables could be embedded in the block code, but it is more convenient to set them via variables.

Block:



The first if statement ensures that it only make changes if there is a new value from the sensor decoder.
The three sub-if statements ensure the calculated WB follows the minimum and maximum including the linear conversion between these two limits.
The PATCH at the end sets the new WB for the program defined by $5. This could also be hardcoded in the block. Especially if the same moisture sensor shall adjust more programs by the same formula. Then just repeat the PATCH for each program.
Note that the WB change also affects running programs. The adjustment will take place from the next step in the program.

An alternative to adjust single programs is to adjust the global water budget which affects all water budget-controlled programs. This alternative command is in the block above:

```
PATCH /we/engine/waterBudget=$6
{"result":"ok"}
```

Log:

Example of log information:

Timestamp	Severity	Type	Data
Wed, 11 Oct 2023 07:14:48	Info	le::user::log	logStr: Low VSM (17%). Set max WB (150%) on Prg 1
Wed, 11 Oct 2023 07:14:28	Info	le::user::log	logStr: New VSM (25%). Set WB (138%) on Prg 1
Wed, 11 Oct 2023 07:14:18	Info	le::user::log	logStr: New VSM (34%). Set WB (116%) on Prg 1
Wed, 11 Oct 2023 07:14:09	Info	le::user::log	logStr: New VSM (46%). Set WB (87%) on Prg 1
Wed, 11 Oct 2023 07:13:59	Info	le::user::log	logStr: New VSM (63%). Set WB (46%) on Prg 1
Wed, 11 Oct 2023 07:13:39	Info	le::user::log	logStr: High VSM (75%). Set min WB (30%) on Prg 1
Wed, 11 Oct 2023 07:13:30	Info	le::user::log	logStr: New VSM (65%). Set WB (42%) on Prg 1