

Topic: Filter flush
Author: Søren Haxvig
Date: 2023.07.14
To: OEM Partners
Purpose: Demonstrate how to use the logic engine to control a filter flush based on pressure and a fixed flush time.

Differential pressure input:

This setup is based on one sensor decoder reading a differential pressure on the filter.

Variables:

	Id	Description	State	Type	Source id	Source sub id	Value
	1	Two wire state	Ok	Tw state	0	0	2
	2	Diff. Pressure Transducer value	Ok	IO value	679088867	1	9125
	3	Flush valve value	Ok	IO value	671088775	1	0
	4	Timer	Ok	Local	0	0	46

Variable #1 is the state of the 2-wire. It is important to make sure the 2-wire is operational while reading and reacting on sensors.

Variable #2 is the input from a sensor decoder. In this example the conversion via the IO configuration is made to convert the 4-20mA signal into mBar (See conversion setup at the end of this document). The flush trigger is set to 5000mBar. The flush time is 5 minutes.

Variable #3 is the output decoder controlling the flush valve.

Variable #4 is a timer used to make sure the sensor reading is stable before doing any actions.

Block:



```

▼ if: $1 == 2 && (($2 && ($2 > 5000)) || $3)
    set: $4 = 60 # Check the pressure IS above for 60 s
    ▼ while: $2 && ($2 > 5000) && !$3 && $4
        set: $4 = $4 - 1
        sleep: 1
    ▼ if: $3 && !$4
        log info: Flush valve manually opened
    ▼ if: !$3 && !$4
        action: PATCH /we/engine/targetState=3 # Pause engine
        sleep: 10
        action: POST /lu/671088775/1/on # Open flush valve
        log info: Flush valve opened at $2 mBar
        sleep: 300 # Flush for 5 minutes
    ▼ if: !$2
        log warning: Unable to read level from pressure transducer!
    ▼ if: $1 != 2
        log info: Two-wire disabled
    ▼ if: $1 == 2 && !$3
        log info: Flush valve manually closed
    ▼ if: $3 && !$4
        action: POST /lu/671088775/1/off # Close flush valve
        action: PATCH /we/engine/targetState=2 # Resume engine
        log info: Flush valve closed at $2 mBar

```



The first “if” statement checks that the 2-wire is operational, and the pressure is above 5000mBar or the flush valve is already open.

The while blocks checks that the pressure stays above the limit for one minute.

The 3rd “if” block starts setting the irrigation engine in pause, wait for that in 10 seconds and then open the flush valve.

Wait 300 seconds for flushing before it continues.

Last “if” block closes the flush valve and set the irrigation engine back to auto mode.

Run of a program:

Timestamp	Severity	Type	Data
Fri, 14 Jul 2023 10:05:37	Info	we::station::stop	id: 18, water_p: 0, emitted_dl: 9, runtime_sec: 545, precipitation_dum: 545, globalExpectedFlow_dl_min: 0
Fri, 14 Jul 2023 10:05:36	Info	we::program::stateChange	id: 4, fromState: running, toState: idling
Fri, 14 Jul 2023 10:01:40	Info	le::user::log	logStr: Flush valve manually closed
Fri, 14 Jul 2023 09:56:32	Info	we::station::start	id: 18, ioAddr: 539873072, ioldx: 2, water_p: 0, expectedFlow_dl_min: 1, globalExpectedFlow_dl_min: 1, programId: 4
Fri, 14 Jul 2023 09:56:29	Info	we::engine::stateChange	fromState: paused, toState: auto
Fri, 14 Jul 2023 09:56:27	Info	le::user::log	logStr: Flush valve closed at 9156 mBar
Fri, 14 Jul 2023 09:51:26	Info	le::user::log	logStr: Flush valve opened at 9125 mBar
Fri, 14 Jul 2023 09:51:17	Info	we::station::stop	id: 18, water_p: 0, emitted_dl: 0, runtime_sec: 57, precipitation_dum: 57, globalExpectedFlow_dl_min: 0
Fri, 14 Jul 2023 09:51:15	Info	we::engine::stateChange	fromState: auto, toState: paused
Fri, 14 Jul 2023 09:50:20	Info	we::station::start	id: 18, ioAddr: 539873072, ioldx: 2, water_p: 0, expectedFlow_dl_min: 1, globalExpectedFlow_dl_min: 1, programId: 4
Fri, 14 Jul 2023 09:50:17	Info	we::station::stop	id: 17, water_p: 0, emitted_dl: 8, runtime_sec: 510, precipitation_dum: 510, globalExpectedFlow_dl_min: 1
Fri, 14 Jul 2023 09:41:47	Info	we::station::start	id: 17, ioAddr: 539873072, ioldx: 1, water_p: 0, expectedFlow_dl_min: 1, globalExpectedFlow_dl_min: 1, programId: 4
Fri, 14 Jul 2023 09:41:44	Info	we::engine::stateChange	fromState: paused, toState: auto
Fri, 14 Jul 2023 09:41:43	Info	le::user::log	logStr: Flush valve closed at 313 mBar
Fri, 14 Jul 2023 09:36:41	Info	le::user::log	logStr: Flush valve opened at 9000 mBar
Fri, 14 Jul 2023 09:36:33	Info	we::station::stop	id: 17, water_p: 0, emitted_dl: 1, runtime_sec: 91, precipitation_dum: 91, globalExpectedFlow_dl_min: 0
Fri, 14 Jul 2023 09:36:30	Info	we::engine::stateChange	fromState: auto, toState: paused
Fri, 14 Jul 2023 09:35:02	Info	we::station::start	id: 17, ioAddr: 539873072, ioldx: 1, water_p: 0, expectedFlow_dl_min: 1, globalExpectedFlow_dl_min: 1, programId: 4
Fri, 14 Jul 2023 09:35:00	Info	we::program::stateChange	id: 4, fromState: idling, toState: running

The program run above demonstrates a run of a program with two steps/stations which is interrupted twice for a filter flush. Both stations are setup up to run 5 minutes.

Individual pressure input on filter inlet and outlet:

This setup is based on two sensor decoders reading a differential pressure on the filter.

Variables:

	Id	Description	State	Type	Source id	Source sub id	Value
	1	Two wire state	Ok	Tw state	0	0	2
	2	Diff. Pressure	Ok	Local	0	0	5375
	3	Flush valve value	Ok	IO value	671088775	1	0
	4	Timer	Ok	Local	0	0	0
	5	Filter In	Ok	IO value	679088854	1	6375
	6	Filter Out	Ok	IO value	679088867	1	1000

Variable #1 is the state of the 2-wire. It is important to make sure the 2-wire is operational while reading and reacting on sensors.

Variable #2 is the deferential pressure between inlet and outlet. The flush trigger is set to 5000mBar. The flush time is 5 minutes.

Variable #3 is the output decoder controlling the flush valve.

Variable #4 is a timer used to make sure the sensor reading is stable before doing any actions.

Variable #5 and #6 are the input from the sensor decoders reading the pressure on the inlet and outlet respectively. In this example the conversion via the IO configuration is made to convert the 4-20mA signal into mBar (See conversion setup at the end of this document).

Block:

▶ if: \$1 == 2 && ((\$2 && (\$2 > 5000)) || \$3)

▼ if: \$5 && \$6

set: \$2 = \$5 - \$6

The first “if” block is the same as in the example with one differential pressure input. The second “if” block above is used to generate the differential pressure.

Altering parameters via API:

If the parameters for trigger level and flush time need to be adjustable via API, then the setup can be changed to the setup below.

Variables:

	Id	Description	State	Type	Source id	Source sub id	Value
	1	Two wire state	Ok	Tw state	0	0	2
	2	Diff. Pressure	Ok	Local	0	0	4187
	3	Flush valve value	Ok	IO value	671088775	1	0
	4	Timer	Ok	Local	0	0	43
	5	Filter In	Ok	IO value	679088854	1	6375
	6	Filter Out	Ok	IO value	679088867	1	2188
	7	Flush trigger level	Ok	Local	0	0	3000
	8	Flush time	Ok	Local	0	0	60

Added #7 for trigger level and #8 for the flush time.

Block:



```
▼ if: $1 == 2 && (($2 && ($2 > $7)) || $3)
    set: $4 = 60 # Check the pressure IS above for 60 s
    ▼ while: $2 && ($2 > $7) && !$3 && $4
        set: $4 = $4 - 1
        sleep: 1
    ▼ if: $3 && !$4
        log info: Flush valve manually opened
    ▼ if: !$3 && !$4
        action: PATCH /we/engine/targetState=3 # Pause engine
        sleep: 10
        action: POST /lu/671088775/1/on # Open flush valve
        log info: Flush open for $8 minutes. DP $2 > $7 mBar
        sleep: $8 # Flush for 5 minutes
    ▼ if: !$2
        log warning: Unable to read level from pressure transducer!
    ▼ if: $1 != 2
        log info: Two-wire disabled
    ▼ if: $1 == 2 && !$3
        log info: Flush valve manually closed
    ▼ if: $3 && !$4
        action: POST /lu/671088775/1/off # Close flush valve
        action: PATCH /we/engine/targetState=2 # Resume engine
        log info: Flush valve closed at DP=$2 mBar
    ▼ if: $5 && $6
        set: $2 = $5 - $6
```

The values for trigger level and flush time can then be adjusted like:

```
PATCH /le/var/7/value=3000
{"result":"ok"}
PATCH /le/var/8/value=60
{"result":"ok"}
```

Sensor decoder setup:**IO configuration:**

Protocol	mk3	
Type	input	
Description	4-20mA differential pressure transducer	
Expression to modify interrupt value stored in object	$10000 * (x - 4000) / 16000$	
x gets replaced by value		
Secondary expression, usage depends on context	x	
x gets replaced by value		
Reset io value to 0 after X seconds without interrupt, 0 = indefinitely	0	

GENERAL	INTERRUPTS	TIMINGS
Input power source		
12V		
Input type		
4-20mA		
Sensor operation mode		
Continuous mode, constant on		
Current control	24.05	mA
Analog processing		
0-20 mA, value in uA, steps of 50		
Analog high measure level	20000	uA
Analog low measure level	0	uA

The expression converts the 4000-20000uA to 0-10000mBar.

The Input power source of 12V might differ depending on the actual sensor.

GENERAL	INTERRUPTS	TIMINGS
Interrupt rate limit (min. time between interrupts)		
5 s		
Minimum value change (since last interrupt) for interrupt to be sent		
0		

The setting above will update the value every 5 seconds regardless of there is any change. A minimum uA change could be defined to avoid too many interrupts on the 2-wire if the sensor fluctuates.

Send interrupt for selected values

- ☒ Measurement Value
- ☐ Measurement Minimum Value
- ☐ Measurement Maximum Value
- ☐ Digital count
- ☐ Milliseconds since last digital input
- ☐ Attention

This setup ensures the sensor decoder sends interrupt when the reading changes.