

Topic: Adjustment of Main Line capacity
Author: Søren Haxvig
Date: 2023.07.26
To: OEM Partners
Purpose: Demonstrate how to use the logic engine to adjust the main line capacity via API in case more Aqualogic 500 controllers share a main line, and the 3rd party system needs to adjust the distribution of capacity among the controllers.

Description:

The scenario is multiple controllers share a common water source (POC). Each controller can be setup with a main line capacity which is a fraction of the total capacity of the water source. E.g., four controllers share 1000 l/m water source and they each have a main line with 250 l/m as starting point. If the water requirement changes over time (day to day, season to season etc), then the 3rd party system needs to adjust the fraction each controller can use. The logic engine can be used to ensure a controlled change of the main line capacity even if the controller is irrigating at the time of the change.

Adjustment of capacity:

The setup below shows a logic engine which adjust the main line capacity to a set point.

Variables:

	Id	Description	State	Type	Source id	Source sub id	Value
	1	Main 1 set capacity (dl/m)	Ok	Local	0	0	50000
	2	Main 1 actual capacity (dl/m)	Ok	Local	0	0	60000
	3	Water engine state	Ok	We engine state	0	0	3
	4	Running state	Ok	Local	0	0	0

Variable #1 is the set point.

Variable #2 is the actual/old set point.

Variable #3 is the state of the water engine.

Variable #4 is local for the old state of the water engine.

The new set point is made using the API:

```
PATCH /le/var/1/value=50000
{"result":"ok"}
```

Block:

```
▼ if: $1 != $2
    log info: Change capacity from $2 to $1 dl/m in state: $3
    ▼ if: ($3 == 2) || ($3 == 1)
        set: $4 = $3
        action: PATCH /we/engine/targetState=3
        action: PATCH /we/pipe/1/capacity_dl_min=$1
    ▼ if: $4
        action: PATCH /we/engine/targetState=$4
        set: $2 = $1
        set: $4 = 0
```

The first “if” statement checks that the new set point is different from the old.

The second “if” statement check if the water engine is in auto or manual and pause it (targetState=3).

Then the pipe is set to the new capacity.

If the water engine was in pause or manual, it will be set back to the old state (targetState=\$4).

The pause and resume of the water engine ensures the flow management is calculated correctly.

The logic could also be programmed into the 3rd party system but using the logic engine ensures the task is completed after the one API call even if the communication between the 3rd party system and the controller being lost.

A run:

PATCH /le/var/1/value=50000

The log:

Timestamp	Severity	Type	Data
Wed, 26 Jul 2023 17:07:59	Info	le::user::log	logStr: Change capacity from 60000 to 50000 dl/m in state: 2

The variables:

	Id	Description	State	Type	Source id	Source sub id	Value
	1	Main 1 set capacity (dl/m)	Ok	Local	0	0	40000
	2	Main 1 actual capacity (dl/m)	Ok	Local	0	0	40000

The states of the water engine:

```
GET /we/engine/stateEnum
{
  "result": [
    {
      "id": 0,
      "name": "init"
    },
    {
      "id": 1,
```

```
    "name": "manual"
  },
  {
    "id": 2,
    "name": "auto"
  },
  {
    "id": 3,
    "name": "paused"
  },
  {
    "id": 4,
    "name": "raining"
  },
  {
    "id": 5,
    "name": "twNotOperational"
  }
]
}
```