UDS ATOM

Dedicated software
for OT/IT convergence

Reference Manual

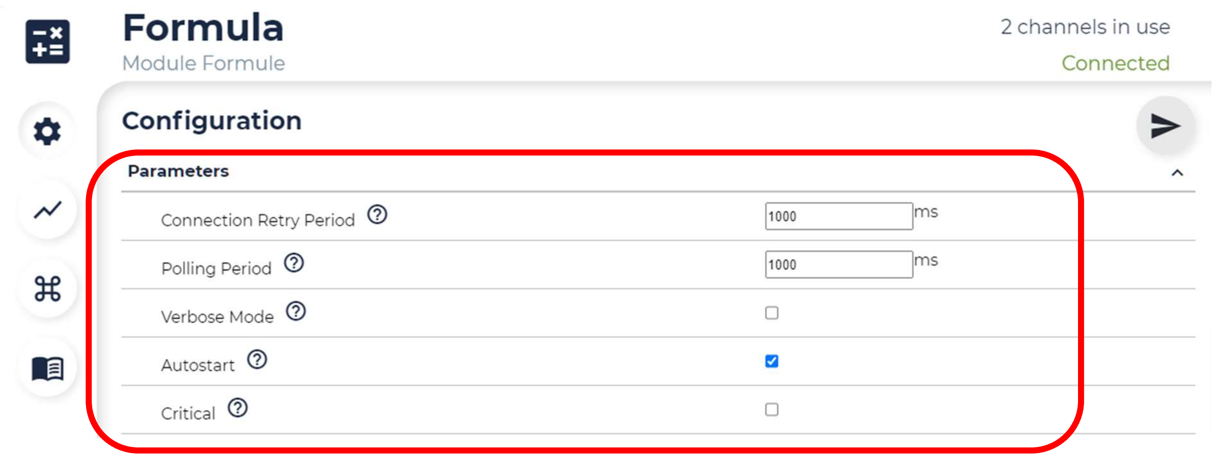
Module Formula

Index

1 Introduction

The formula module enables custom logic, advanced calculations, and data aggregation for operational data analysis at the Edge. For instance, formula allows for new channels to be obtained and calculated from other channels.

2 General configuration



Formula

Module Formule

2 channels in use

Connected

Configuration

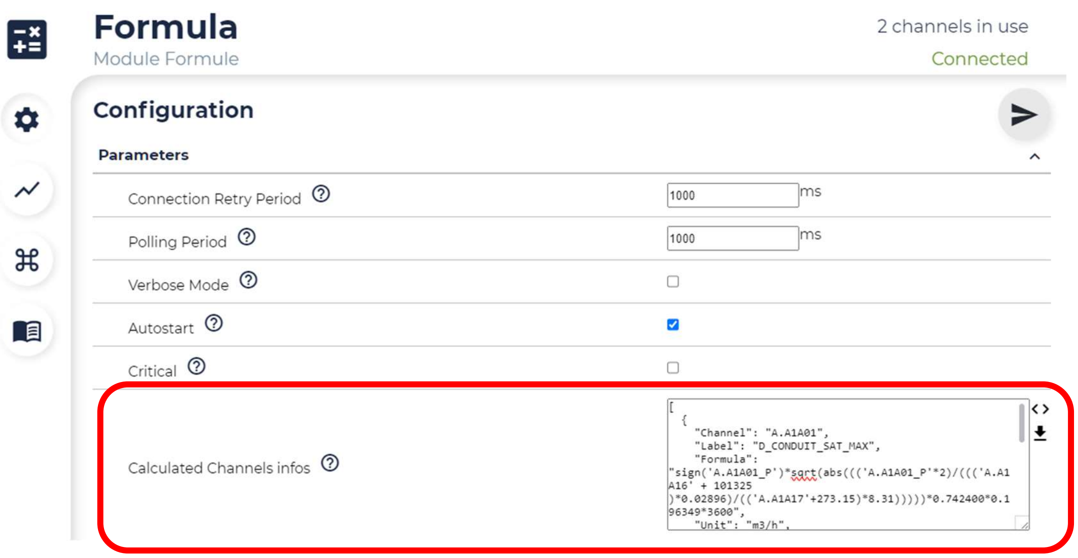
Parameters

Connection Retry Period	1000	ms
Polling Period	1000	ms
Verbose Mode	☐	
Autostart	☑	
Critical	☐	

Every module has a common set of parameters (highlighted in red on the above figure)

- Connection Retry Period: Period of time before the pusher tries to connect after a disconnection.
- Polling Period: Determines the amount of time that passes before the pusher communicates updated values to UDS Atom
- Verbose Mode: Option available that provides additional details as to what the pusher is doing (see [Logbook](../Logbook.md))
- Autostart: When *Autostart* is set to true, the pusher will start automatically after UDS Atom startup. If set to false, the sorucer will be in *Stand-by* state.
- Critical: When set to true, the pusher will send a notification when it's down.

3 Formula configuration



Connection Retry Period 1000 ms

Polling Period 1000 ms

Verbose Mode □

Autostart ☑

Critical □

Calculated Channels infos

```
[
  {
    "Channel": "A.A1A01",
    "Label": "D_CONDUIT_SAT_MAX",
    "Formula":
"sign('A.A1A01_P')*sqrt(abs((('A.A1A01_P'*2)/((('A.A1
A16' + 101325
)*0.02896)/(('A.A1A17'+273.15)*8.31)))))*0.742400*0.1
96349*3600",
    "Unit": "m3/h",
```

Formulas are defined in a JSON array of objects like the following:

```
[
  {
    "Channel": "A.A1A01",
    "Label": "D_CONDUIT_SAT_MAX",
    "Formula": "sign('A.A1A01_P')*sqrt(abs('A.A1A01_P'*2))*0.74",
    " Unit": "m3/h"
    " HighPrio": false
  }
]
```

Each formula is defined by:

Parameter	Type	Required	Description
Channel	string	NO	Unique Id of the channel. if this field is missiing, Channel = **Formula_%02d** value where *%02d* is an autoincrement formatted with 3 digits
Label	string	NO	Caption of the channel. A more human readable name for the channel
Formula	string	**YES**	Formula of the virtual channel. All channels used in formula must be between simple quote and be prefixed by the name of the sourcer followed by a point (example: sourcerName.channelName)
Unit	string	NO	Unit of the virtual channel
HighPrio	boolean	NO	If **true** result of the virtual channel will be compute at high priority by pushers

All arythmetical operator such as +, -, *, / are allowed. Authorized functions in formulas are the folliwing:

Mnemonic	Corresponding function	Description
abs(x)	Absolute value	Returns the absolute value of x.
acos(x)	Arccosine	Computes the arccosine of x in radians.
acosh(x)	Hyperbolic arccosine	Computes the hyperbolic arccosine of x.
asin(x)	Arcsine	Computes the arcsine of x in radians.
asinh(x)	Hyperbolic arcsine	Computes the hyperbolic arcsine of x.
atan(x)	Arctangent	Computes the arctangent of x in radians.
atan2(y,x)	Arctangent (2 inputs)	Computes the arctangent of y/x in radians.
atanh(x)	Hyperbolic Arctangent	Computes the hyperbolic arctangent of x.
ceil(x)	Round to +infinity	Rounds x up to the next integer value (smallest integer > or equal to x.)
cos(x)	Cosine	Calculates the cosine of x, where x is expressed in radians.
cosh(x)	Hyperbolic cosine	Computes the hyperbolic cosine of x.
cot(x)	Cotangent	Computes the cotangent of x (1/tan(x)), with x in radians.
csc(x)	Cosecant	Computes the cosecant of x (1/sin(x)), with x in radians.
exp(x)	Exponential	Computes the value of e raised to the x power.
expm1(x)	Exponential (Arg) – 1	Calculates the value of e raised to the x power and subtracts 1 from the result ((e^x) | – 1).
floor(x)	Round to –infinity	Reduces x to the lower integer value (largest integer less or equal to x.)
getexp(x)	Mantissa and exponent	Returns the exponent of x.
getman(x)	Mantissa and exponent	Returns the mantissa of x.
int(x)	Round to nearest integer	Rounds x to nearest integer.
intrz(x)	-	Rounds x to the nearest integer between x and zero.
ln(x)	Natural logarithm	Computes the natural logarithm of x (in base e)
lnp1(x)	Natural logarithm (Arg +1)	Computes the natural logarithm of (x + 1).
log(x)	Logarithm in base 10	Computes the logarithm of x (in base 10).
log2(x)	Logarithm in base 2	Computes the logarithm of x (in base 2).
max(x,y)	Max. and min.	Compares x and y and returns the larger value.
min(x,y)	Max. and min.	Compares x and y and returns the smaller value.
mod(x,y)	Quotient and remainder	Computes the remainder of x/y, when the quotient is rounded towards –infinity.
pow(x,y)	Power of X	Computes x raised to the power of y.
rand( )	Random number (0 – 1)	Produces a floating-point number between 0 and 1 exclusively.
rem(x,y)	Quotient and Remainder	Computes the remainder of x/y, when the quotient is rounded towards –infinity.
sec(x)	Secant	Computes the secant of x, where x is expressed in radians (1/cos(x))
sign(x)	Sign	Returns 1 if x is greater than 0, returns 0 if x is equal to 0, and returns –1 if x is less than 0.
sin(x)	Sinus	Calculates the sine of x, where x is expressed in radians.
sinc(x)	Sinc	Computes the sine of x divided by x (1/sin(x)/x), with x in radians.
sinh(x)	Hyperbolic sine	Computes the hyperbolic sine of x.
sizeOfDim(ary,di)	-	Returns the size of the specified dimension di for the array ary

sqrt(x)	Square Root	Computes the square root of x.
tan(x)	Tangent	Computes the tangent of x, with x in radians.
tanh(x)	Hyperbolic Tangent	Computes the hyperbolic tangent of x.

UDS ATOM

Dedicated software
for OT/IT convergence

Sales Contact

sales@udsatom.com

Technical Support

support@udsatom.com