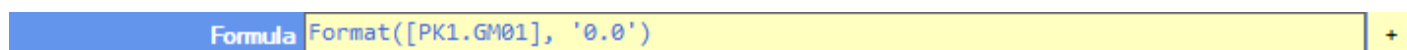


Fórmulas

O uso de fórmulas possibilita expandir como os elementos são configurados, ao invés de uma configuração fixa é possível efetuar operações matemáticas, utilizar funções, operações de lógica, combinar tags, fazer correções, etc.

Os campos que aceitam o uso de fórmula são facilmente identificados pelo fundo amarelo e pelo sinal de + a direita.



Clicando no sinal de + ou dando dois cliques no campo da formula irá abrir um assistente para facilitar a configuração.

Ainda nesse assistente é possível consultar as tags e testar as fórmulas antes de aplicar.

Utilize este assistente para aprender a utilizar as fórmulas e para fazer testes.

Ver: Editor de Fórmulas

Tags

Para representar as Tags sempre coloque entre colchetes [].

A tag pode ser chamada tanto pelo **Nome** como o **Apelido**.

Ex: **Abs** ([Sensor4])

Ex: [Sensor3]

Ex: **Round** (([Temperatura1] + [Temperatura2] + [Temperatura3]) / 3 , 2)

Textos e Números

Números podem ser inseridos diretamente no campo da fórmula, enquanto que textos requerem o uso de aspas simples.

Ex: **68**

Resultado: **68**

Ex: **'Velocidade'**

Resultado: **Velocidade**

Ex: **Velocidade**

operação acima resulta em erro

É possível combinar valores.

Ex: **68 + 2**

Resultado: **70**

Ex: **'Velocidade é ' + '68 km/h'**

Resultado: **Velocidade é 68 km/h**

Não combine textos com números sem antes converter.

ex: **68 + 'km/h'**

operação acima resulta em erro

ex: **68 + '2'**

textos que representam números é aceito mas não recomendado

Resultado: **70**

Converta para o formato apropriado antes.

ex: **Str (68) + 'km/h'**

Resultado: **68km/h**

ex: **68 + Int('2')**

Resultado: **70**

Regras Gerais

- Separador de casas decimais é ponto (.)
- Separador de parâmetros é vírgula (,)
- Utilize parênteses para agrupar operações "**(5 - 2) / 3**"
- Espaçamentos são aceitos "**Cos(pi/6) = Cos (pi / 6)**"
- Funções que utilizam ângulos trabalham em radianos.
- Resultados **NaN** significa que não tem resposta e/ou a fonte da Tag está sem comunicação.
- Resultados em **?** significa que o valor extrapolou o máximo ou resultou em uma divisão por zero.
- Maiúsculo e minúsculo não fazem diferença no nome das funções (**Now()** = **now()** = **NOW()**)
- Maiúsculo e minúsculo **são** diferentes para as Tags "**[tagtemp]** é diferente de **[tagTemp]**"
- Funções requerem o uso de parênteses para os parâmetros, mesmo que não tenha nenhum

"ex: **Now()**"

Operadores

Tipo	Símbolo	Descrição	Exemplo	Resultado
Lógica	or 	ou	1 > 2 or 4 > 3 false or true	true true
Lógica	and &&	e	1 > 2 and 4 > 3 false and true	false false
Relacional	= ==	igualdade	1 = 2 1 == 2	false false
Relacional	!= <>	diferente	1 != 2 1 <> 2	true true
Relacional	<	menor que	1 < 2 1 < 1	true false
Relacional	<=	menor ou igual que	1 <= 2 1 <= 1	true true
Relacional	>	maior que	1 > 2 1 > 1	false false
Relacional	>=	maior ou igual que	1 >= 2 1 >= 1	false true
Adição	+	soma	1 + 2	3
Subtração	-	subtrair	1 - 2	-1
Multiplicação	*	multiplicar	1 * 2	2
Divisão	/	dividir	1 / 2	0.5
Resto	%	resto	1 % 2	1
Bit	&	and	1 & 2	0
Bit		or	1 2	3
Bit	^	xor	1 ^ 2	3
Bit	<<	left shift	1 << 2	4
Bit	>>	right shift	1 >> 2	0

Unário	! not		if (not true, 3, 4) if (not false, 3, 4)	4 3
--------	----------	--	---	--------

Funções Matemáticas

Nome	Descrição	Exemplo	Resultado
Abs	Valor absoluto (sem sinal).	Abs(33) Abs(-78) Abs(-33.45)	33 78 33.45
Acos	Arco cosseno (ângulo em radiano).	Acos(pi/6) Acos(pi/4) Acos(pi/2)	1.01 0.66 NaN
Asin	Arco seno (ângulo em radiano).	Asin(pi/6) Asin(pi/4) Asin(pi/2)	0.55 0.90 NaN
Atan	Arco tangente (ângulo em radiano).	Atan(pi/6) Atan(pi/4) Atan(pi/2)	0.48 0.66 1.00
Ceiling	Maior valor inteiro que seja maior ou igual as valor.	Ceiling(1.4) Ceiling(37.98) Ceiling(-37.98)	2 38 -37
Cos	Cosseno (ângulo em radiano).	Cos(pi/6) Cos(pi/4) Cos(pi/2)	0.86 0.70 0
Exp	Exponencial de "e" (e = 2.7182)	Exp(3) Exp(3.5) Exp(4)	20.08 33.11 54.59
Floor	Maior valor inteiro que seja menor ou igual ao valor.	Floor(1.4) Floor(37.98) Floor(-37.98)	1 37 -38
IEEERemainder	Resto da divisão seguindo as especificações da Sessão 5.1 da ANSI/IEEE Std 754-1985	IEEERemainder(3, 2) IEEERemainder(12, 5)	-1 2
Log	Logarítmico. Par1 = Valor Par2 = Base	Log(1, 10) Log(2, 10) Log(2, 8)	0 0.30 0.33
Log10	Logarítmico base 10.	Log10(1)	0
Max	Retorna o maior valor.	Max(1, 2)	2

Nome	Descrição	Exemplo	Resultado
Min	Retorna o menor valor.	Min(1, 2)	1
Pow	Potencia. Par1 elevado ao Par2	Pow(3, 2) Pow(2, 3)	9 8
Round	Arredonda o numero para a quantidade de casas decimais especificada. Par1 = Valor Par2 = Casas Decimais	Round(3.222, 2) Round(3.55, 1) Round(3.55, 0)	3.22 3.6 4
Sign	Retorna um numero que representa o sinal.	Sign(-10) Sign(0) Sign(54)	-1 0 1
Sin	Seno (ângulo em radiano).	Sin(0)	0
Sqrt	Raiz quadrada.	Sqrt(4) Sqrt(12)	2 3.46
Tan	Tangente (ângulo em radiano).	Tan(0)	0
Truncate	Retorna a parte inteira do número.	Truncate(1.7) Truncate(-1.7) Truncate(23.56)	1 -1 23

Funções Conversão

Nome	Descrição	Exemplo	Resultado
Int	Converte valores para inteiro. Valores fracionários são arredondados. 32-bit (4-byte) -2,147,483,648 a 2,147,483,647.	Int(73.2) Int(73.5) Int(73.8) Int(-73.5) Int('73.5')	73 74 74 -74 74

Double	Converte valores para double. Pode conter valores fracionários. 64-bit (8-byte) - 1.79769313486231570E+308 to - 4.94065645841246544E-324 4.94065645841246544E-324 to 1.79769313486231570E+308	Double('73.5')	73.5
Str	Converte valores para texto.	Str(73.5)	'73.5'
Hex	Converte números para hexadecimal.	Hex(75) Hex(65535)	'4B' 'FFFF'

Condicionais

Nome	Descrição	Exemplo	Resultado
in	Retorna se um valor esta presente. par1 = Valor a ser procurado par2 = Valor a ser testado ... parN = Valor a ser testado	in(1 + 1, 1, 2, 3) in(4, 1, 2, 3) in(4, 1, 2, 3, 4, 5, 6, 7)	true false true
if	Condicional if par1 = Condição par2 = Retorna verdadeiro par3 = Retorna se falso	if(3 > 2, 'sim', 'não') if(3 < 2, 'sim', 'não') if(1 + 1 = 2, 4, 5)	'sim' 'não' 4

Constantes

Nome	Valor	Exemplo	Resultado
pi	3.141592	pi + 3	6.141592
e	2.718281	e + 3	5.718281

Nota: Constantes não requer o uso de parênteses como nas funções.

Funções Auxiliares

Format(x, y)

Efetua a formatação de valor **x** usado as configurações **y**.

Configurações para **y**:

Nota: Regras abaixo podem ser combinadas.

Representa dígito: '0'

Caso o valor contenha casas decimais ele será arredondado.

Format(36.89, '0')	37
Format(12, '0')	12

Casas decimais: '0.00'

Quantidade de zeros após o ponto define quantas casas decimais serão exibidas.

Format(36.89, '0.0')	36.9
Format(12, '0.0')	12.0
Format(36.89, '0.0000')	36.8900

Representa valor opcional: '#'

Se houver um dígito na posição ele exibe, caso contrário é oculto.

Format(36, '0.0#')	36.0
Format(36.8, '0.0#')	36.8
Format(36.89, '0.0#')	36.89
Format(36.894, '0.0#')	36.89
Format(36.8945, '0.0#')	36.89

Zeros a esquerda: '00'

Quantidade de zeros antes do ponto define quantas casas serão exibidas.

Caso o valor tenha mais dígitos eles serão exibidos.

Format(36.89, '00')	37
Format(36.89, '000')	037
Format(36.89, '000.0')	036.9
Format(12, '00000')	00012
Format(234536, '00')	234536

Separador de milhar: '0,0'

Utilize uma virgula entre zeros para habilitar o separador de milhar.

Format(234324232.44343, '0,0')	234,324,232
Format(234324232.44343, '0,0.00')	234,324,232.44
Format(4232.44343, '00000,0.00')	004,232.44

Unidade

Utilize a letra u na formatação para indicar a posição da unidade.

Format([sensor1], '0,0u')	23.4°C
Format([sensor1], '0,0 u')	23.4 °C
Format([sensor1], 'u0,0')	°C23.4

Note que para o recurso de unidade funcionar a tag deve estar cadastrada com a unidade no servidor.

Random(x)

Retorna um valor aleatório de acordo com os parâmetros.

Nota: Valores mínimo e máximo estão inclusos na geração.

Random()

Caso nenhum parâmetro seja passado é assumido um valore entre 0 e 100.

Random()	22
Random()	98
Random()	0

Random(x)

Retorna um valor aleatório entre 0 e **x**

Random(x, y)

Retorna um valor aleatório entre **x** e **y**

Random(120, 130)	128
Random(120, 130)	130
Random(120, 130)	125

Now(x)

Retorna a data e hora atual.

Nota: A geração das datas seguem as configuração regionais do sistema operacional. Devido a isso podem ocorrer diferenças na ordem e apresentação de certas datas.

Now()

Retorna a data e hora atual utilizando as configurações de região do sistema operacional.

Now()	2021/07/09 07:11:32	Data e hora padrão.
-------	---------------------	---------------------

Now(x)

Retorna a data e hora atual utilizando o parâmetro **x**.

Exemplos abaixo são para a data 2021/07/09 07:11:32

Parâmetro	Resultado	Descrição
Now('yyyy')	2021	Ano.
Now('yy')	21	Ano com 2 dígitos.
Now('MMMM')	Julho	Mês por extenso.
Now('MMM')	Jul	Mês abreviado.
Now('MM')	07	Mês.
Now('dddd')	Sexta	Dia da semana.
Now('ddd')	Sex	Dia da semana abreviado.
Now('dd')	09	Dia.

Now('hh')	07	Hora no formato 0 a 12h.
Now('HH')	19	Hora no formato 0 a 24h.
Now('mm')	11	Minutos.
Now('ss')	32	Segundos.
Now('tt')	PM	AM/PM.
Now('zz')	-3	Zona UTC.
Now('"'texto"')	texto	Adicionar Textos. (entre aspas duplas)

Parâmetros acima podem ser combinados para gerar o resultado desejado.

Exemplos	Resultado	Descrição
now('"'Hoje é" dddd')	Hoje é Segunda	Texto + Data
now('yyyy-MM-dd')	2021-07-09	Apenas data
now('HH:mm')	19:11	Apenas horas
now('dd "de" MMMM')	07 de Julho	Data + Texto + Data

ToTime(x, [y])

Converte uma contagem de segundos para o formato de horas. Resultado padrão é Dia.Hora:Minuto:Segundo.
Para formatar o resultado utilize o parâmetro opcional [y].
Na tabela abaixo será utilizado o valor **99999s** como exemplo que resulta em **1dia** e **03:46:39**.

Note que o parâmetro utiliza minúsculo para apenas exibir parte do campo desejado e maiúsculo para exibir to TOTAL acumulado do campo.
Quando combinado utilize preferencialmente o primeiro parâmetro como maiúsculo e os demais minúsculos, assim garante que total está sempre exibido (ver exemplos abaixo).

Parâmetro	Resultado	Descrição
-----------	-----------	-----------

ToTime(99999, 'dd')	1	Total de dias (dias é uma exceção e minuscuro também exibe o total).
ToTime(99999, 'DD')	1	Total de dias.
ToTime(99999, 'hh')	03	Parte das horas.
ToTime(99999, 'HH')	27	Total de horas.
ToTime(99999, 'mm')	46	Parte dos minutos.
ToTime(99999, 'MM')	1666	Total de minutos.
ToTime(99999, 'ss')	39	Parte dos segundos.
ToTime(99999, 'SS')	99999	Total de segundos.

Exemplos	Resultado	Descrição
ToTime(99999)	1.03:46:39	Conversão padrão.
ToTime(9999)	02:46:39	Conversão padrão.
ToTime(999)	00:16:39	Conversão padrão.
ToTime(99999, 'hh:mm:ss')	03:46:39	Exibe apenas a parte do horário, note que ficou 1 dia omitido.
ToTime(99999, 'HH:mm:ss')	27:46:39	Com o HH maiúsculo agora ele exibe o total de horas.
ToTime(99999, 'mm:ss')	46:39	Exibe apenas a parte dos minutos e segundos.
ToTime(99999, 'MM:ss')	1666:39	Exibe o total de minutos e os segundos.
ToTime(99999, 'ddd hh:mm:ss')	1d 03:46:39	Adicionando um caractere 'd' nos dias.
ToTime(99999, 'ddd hhh:mmm:sss')	1d 03h:46m:39s	Adicionando caracteres extras.
ToTime(99999, 'ddd hhh mmm sss')	1d 03h 46m 39s	Adicionando caracteres extras.
ToTime(99999, 'HH horas')	27 horas	Adicionando textos.
ToTime(99999, 'DD dia e hh horas')	1 dia e 03 horas	Adicionando textos.

Revisão #44

Criado Thu, Feb 18, 2021 7:47 PM por [Alessandro Ueki](#)

Atualizado Thu, Mar 23, 2023 7:22 PM por [Alessandro Ueki](#)