

9. Блоки додаткової обробки

9.1 SCALING (сімейство *Conditioning*)

Даний функціональний блок призначений для масштабування числових величин. Він реалізовує лінійну залежність вихідної величини (*OUT*) від вхідної (*IN*) за формулою:

$$OUT = (IN - in_min) \cdot \frac{(out_max - out_min)}{(in_max - in_min)} + out_min \quad (9.1)$$

Графічно залежність виходу *OUT* від входу *IN* показана на рис.9.1. Мінімальні та максимальні вхідні (*in_min*, *in_max*) та вихідні (*out_min*, *out_max*) величини, відносно яких проводиться масштабування, задаються у вхідному параметрі *PARAM* типу *Para_SCALING*. Тип даних *Para_SCALING* включає 4-ри поля типу *REAL* для завдання вхідних та вихідних меж, а також одне поле "*clip*" типу *BOOL* для визначення необхідності обмеження вихідної величини (див. рис.4.44).

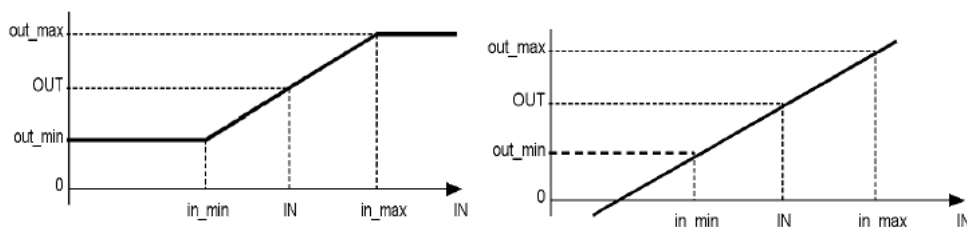


Рис.9.1. Залежність масштабованого вихідного значення (*OUT*) від вхідного (*IN*) при обмеженні на вихідний сигнал (зліва, *Clip*=1) та без обмеження (праворуч, *Clip*=0) для блока *SCALING*.

Функціонування блоку продемонструємо на прикладі масштабування вхідного аналогового сигналу від датчика температури з діапазоном $0-150^{\circ}\text{C}$, який підключений до *%IW0.1.1*. Результат масштабування необхідно записати в змінну *T1_R*.

По замовченню, при опитуванні, сигнали від універсальних аналогових вхідних модулів перетворюються в діапазон $0-10000$. Тобто вхідні межі будуть $0-10000$, а вихідні $0-150$. Для параметрів масштабування створюємо змінну *T_PARA* типу *Para_SCALING*, властивості *VALUE* для полів заповнюємо відповідно до рис.9.2 (зверху). Присвоїмо поле *clip:=TRUE* для обмеження по мінімуму та максимуму вихідної (масштабованої величини).

Вигляд програми користувача на FBD показаний на рис.9.2 (внизу). Створюється екземпляр функціонального блоку *SCALE_T1* типу *SCALING*. Попередньо *%IW0.1.1* перетворюється в тип *REAL*, відповідно до типу параметру *IN* функціонального блоку *SCALING*. Вихідний параметр *STATUS* потрібен для контролю за помилками, в прикладі не використовується.

Name	Type	Value	Comment
T_PARA	Para_SCALING		
in_min	REAL	0.0	мінімальне вхідне значення
in_max	REAL	10000.0	максимальне вхідне значення
out_min	REAL	0.0	мінімальне масштабоване значення
out_max	REAL	150.0	максимальне масштабоване значення
clip	BOOL	true	активувати обмеження по виходу

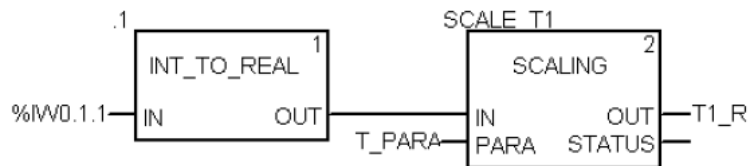


Рисунок 9.2– Приклад використання функціонального блока SCALING для масштабування аналогового вхідного сигналу: зверху – опис структурної змінної T_PARA типу Para_SCALING, знизу – приклад програми на FBD.

Аналогічним чином можна проводити масштабування вихідної величини.

9.2. Ланка транспортного запізнювання *DTIME* (сімейство *Conditioning*)

Функціональний блок *DTIME* призначений для реалізації ланки чистого (транспортного) запізнювання між входом *IN* та виходом *OUT*. Час запізнювання визначається значенням T_DELAY(рис.9.3)

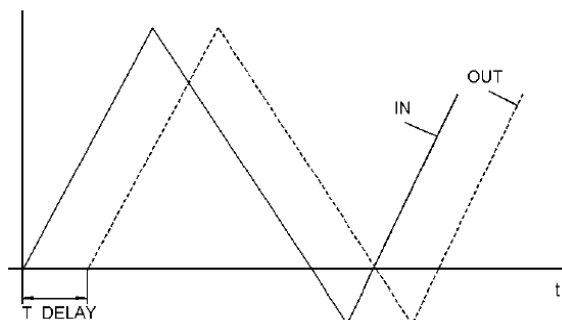


Рисунок 9.3– Діаграма роботи DTIME.

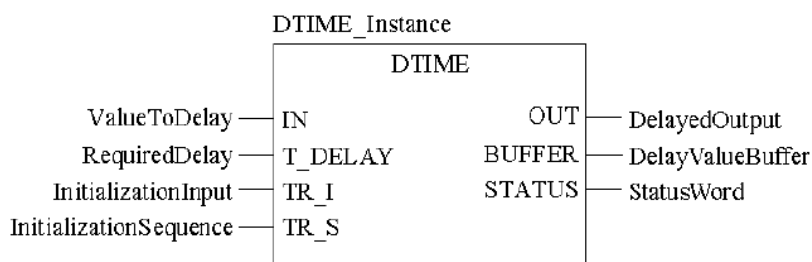


Рисунок 9.4– Приклад виклику блоку DTIME.

Таблиця 9.1 Параметри блоку DTIME

Вхідні параметри		
<i>IN</i>	<i>REAL</i>	Вхідне числове значення
<i>T_DELAY</i>	<i>TIME</i>	Час запізнення
<i>TR_I</i>	<i>REAL</i>	Вхід ініціалізації
<i>TR_S</i>	<i>BOOL</i>	Команда ініціалізації
Вихідні параметри		
<i>OUT</i>	<i>REAL</i>	Вихідне значення
<i>BUFFER</i>	<i>ANY</i>	Пам'ять для збереження значень (завжди повинен бути підключений до змінної)
<i>STATUS</i>	<i>WORD</i>	Слово стану

Для розміщення даних транспортного запізнення між входом і виходом використовується змінна, яка підключається до виходу *BUFFER*. Ця змінна має тип *ANY*, враховуючи що розмір буфер може бути різним. Однак в більшості випадків розмір буфера буде більше ніж *REAL*, оскільки кількість значень в буфері буде більше 1-го. Тому в якості змінної рекомендується використовувати масив типу *REAL*. Наприклад, масив *ARRAY [0..10] of REAL* може вміщувати до 11 елементів.

Максимальна величина затримки розраховується за формулою:

$$T_DELAY_{max} = n \cdot T_Period \quad (9.1)$$

де *n* - кількість значень, які можуть бути збережені в *BUFFER*, *T_Period*-інтервал виклику функціонального блоку.

9.3. Аперіодична ланка *LAG_FILTER* (сімейство *Conditioning*)

Функціональний блок *LAG_FILTER* призначений для реалізації аперіодичної ланки (1-го порядку), де вихід розраховується за формулою:

$$OUT = OUT_{old} + \frac{dt}{LAG + dt} \cdot (GAIN \cdot \frac{IN_{old} + IN}{2} - OUT_{old})$$

де змінні з індексами *old* – значення на попередньому виклику, *dt* – інтервал між викликами блоку, інші параметри наведені в таблиці 9.2.

Даний блок повинен обов'язково викликатися на першому циклі ПЛК.

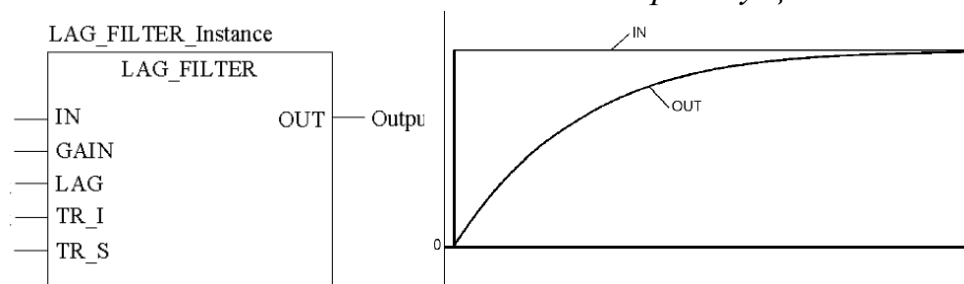


Рис.9.4. Приклад виклику блоку *LAG_FILTER* та діаграма його роботи.

Параметри блоку LAG_FILTER

Вхідні параметри		
<i>IN</i>	<i>REAL</i>	Вхідне значення
<i>GAIN</i>	<i>REAL</i>	Коефіцієнт підсилення
<i>LAG</i>	<i>TIME</i>	Стала часу
<i>TR_I</i>	<i>REAL</i>	Вхід ініціалізації (слідкування)
<i>TR_S</i>	<i>BOOL</i>	1 = режим слідкування 0 = автоматичний режим
Вихідні параметри		
<i>OUT</i>	<i>REAL</i>	Вихід