

Національний технічний університет
«Дніпровська політехніка»
ТОВ «Шнейдер Електрик Україна»

Авторизований центр навчання
компанії «Шнейдер Електрик»
при кафедрі електропривода

**ПРОГРАМУВАННЯ
ВБУДОВАНОГО ПЛК ATV Logic
перетворювача частоти
Altivar Machine 320**

Методичний посібник
для студентів спеціальності
141 «Електроенергетика, електротехніка та електромеханіка»
та слухачів курсів підвищення кваліфікації

Укладач: Боровик Р.О.

Авторизований
центр навчання



Дніпро
2024

ЗМІСТ

ВСТУП.....	4
1. Загальний огляд.....	5
2. Організація ATV Logic.....	5
Групи параметрів.....	5
Типи завдань.....	6
3. Створення програми FV.....	8
Інтерфейс.....	9
Панель інструментів.....	9
Структура меню.....	10
Набір функціональних блоків.....	10
Вставка функціональних блоків.....	11
Створення віртуального порту.....	11
З'єднання блоків.....	11
3.1. Конфігурація віртуальних входів та виходів.....	12
3.2. Властивості функціональних блоків.....	13
Властивості блоку.....	13
Властивості віртуального порту.....	13
3.3. Параметри відображення.....	14
Розміри поля діаграми.....	14
3.4. Перевірка програми.....	14
3.5. Збереження проєкту / програми FV.....	15
3.6. Налаштування.....	16
3.7. Порядок виконання.....	16
4. Елементи мови FBD.....	17
4.1. Спеціальні функції.....	17
NUM (analog).....	17
NUM (register).....	17
TRUE.....	17
FALSE.....	17
4.2. Логічні функції.....	17
Функція NOT.....	17
Функція AND.....	18
Функція OR.....	18
Функція NAND.....	19
Функція NOR.....	19
Функція XOR (eXclusive OR).....	20
Функція A/C TIMER (Timer).....	20
Функція ADD/SUB.....	21
Функція BOOLEAN.....	22
Функція CAN (Word to Bit Conversion).....	23
Функція CNA (Bit to Word Conversion).....	23
Функція COMPARE (Порівняння двох значень).....	23
Функція EDGE.....	24

Функція GAIN.....	25
Арифметична функція MUL/DIV.....	25
Функція MUX (мультиплексування).....	26
Функція BIT READ.....	27
Функція READ Ana Param.....	27
Функція READ Reg Param.....	28
Функція RS Memory.....	29
Функція SHIFT/ROLL.....	30
Функція TRIGGER (Schmitt Trigger).....	31
Функція лічильника COUNT Up/Down.....	32
Функція BIT WRITE (WriteBitParam).....	33
Функція WRITE Ana Param.....	34
Функція WRITE Reg Param.....	35
Функція STU (16-bit analog TO 16-bit register).....	36
Функція UTS (16-Bit Register Input to 16-Bit Analog Output).....	36
5. Опис параметрів ATV320.....	37
5.1. Керування програмою FB.....	38
5.2. Моніторинг стану програми FB.....	39
Статус програми FB.....	39
Помилки програми FB.....	40
5.3. Комунікаційні параметри програми FB.....	41
Параметри FB.....	41
Контейнери ADL.....	41
5.4. Системні слова FB.....	42
Системне слово FB 01.....	42
Системне слово FB 02.....	43
Системне слово FB 06.....	43
6. Приклади застосування.....	44
Використання тригера.....	44
Тимчасовий старт/стоп.....	44
Індикація обмеження моменту двигуна.....	46
Регулювання температури в теплиці.....	48
Індикація стану «Сон».....	51
ДЖЕРЕЛА.....	52

ВСТУП

Метою даного посібника є допомога у вивченні принципів програмування ATV Logic – вбудованого ПЛК в перетворювач частоти Altivar Machine ATV320 виробництва компанії Schneider Electric. Посібник створений переважно на основі «ATV Logic Manual» [1], а також за допомогою інших інструкцій, відео тощо.

Порядок викладу відрізняється від порядку, прийнятого в [1]. Часом текст супроводжується коментарями і власною думкою автора на поточну ситуацію із інформування користувачів, на які треба звернути увагу читача, оскільки інструкції містять помилки, неточності, наслідки переписування інструкцій від попередніх версій ПЧ та програмного забезпечення, а деякі питання «спливають» лишень на форумах користувачів FAQ.

Посібник доповнений прикладами програм, які оригінальна інструкція [1] не містить. Приклади відтворювались на ПЧ ATV320U04N4C в SoMove v.2.7.6 та v.2.9.8

1. Загальний огляд

За допомогою ATV Logic (вбудованого ПЛК в перетворювач частоти Altivar Machine ATV320) користувач може створити власну програму роботи ПЧ; доповнити, розширити чи змінити вбудовані функції; програму технологічного процесу.

Програмування здійснюється засобами стандартизованої мови Функціональна Блокова Діаграма (FBD) і відбувається лише в середовищі програмного забезпечення SoMove зі встановленою бібліотекою Altivar Machine 320 DTM (ATV Logic доступний лише всередині цієї бібліотеки).

Щоб встановити Altivar Machine 320 DTM, спочатку треба завантажити та встановити FDT: SoMove з www.schneider-electric.com, а потім вже Altivar Machine 320 DTM.

В перетворювачі завжди працює Основна програма керування приводом, яку користувач не має змоги змінити, і фонові програми (здійснення комунікації, «спілкування» з графічним терміналом тощо). Програма користувача, створена засобами ATV Logic, може використовувати ті ж самі ресурси (входи, виходи, регістри, біти) і виконується у решту програмного часу від основної програми.

Програмовані логічні функції взаємодіють з

- Основною програмою (ядром) керування приводом
- Функціями вводу/виводу, комунікаційними функціями
- Людино-машинним інтерфейсом через M0xx (локальна або зовнішня графічна панель).

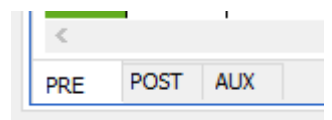
В основі лежить набір інструкцій FBD: логічні операції та операції передачі, арифметичні операції, таймери, лічильники тощо.

Для створення програми FB, потрібно вставити в схему входи, виходи, функціональні блоки (ФБ, FB) і зв'язати їх між собою, налаштувати блоки за необхідності.

На аркуші можуть бути розміщені всі види функціональних блоків, крім входів і виходів, для яких призначені відповідні місця з лівого та правого боків поля схеми.

2. Організація ATV Logic

Вкладка ATV Logic в SoMove DTM містить робочу область із трьома шарами (завданнями), які можна вибрати за допомогою вкладок (PRE, POST, AUX). Завдання PRE і POST є пріоритетними перед завданням AUX.



Групи параметрів

Правило використання параметрів в завданнях таке:

- У завданнях PRE/POST: підключення лише до параметрів швидкого доступу.
- У завданні AUX: підключення до параметрів повільного та швидкого доступу.

Визначальними є 4 групи параметрів:

- **FastReadFunctionBlocks**

ETA, RFR, FRH, LCR, OTR, ETI, ULN, UOP, THD, OPR
THR1, THR2, THR3
LRS1, LRS2, LRS3, LRS4, LRS5, LRS6, LRS7, LRS8
IL1I, IL1R, OL1R, AI1C, AI2C, AI3C, AO1R, AO1C
RFRD, FRHD, SPD

M001, M002, M003, M004, M005, M006, M007, M008
S001, SLSS, STOS, SS1S

- **FastWriteFunctionBlocks**

M001, M002, M003, M004, M005, M006, M007, M008, S006

- **SlowReadFunctionBlocks**

HSP, LSP, BSP, ETA, RFR, FRH, LCR, OTR, ETI, ULN, UOP, THD, OPR

THR1, THR2, THR3, APH, RTH, PTH, UNT

LRS1, LRS2, LRS3, LRS4, LRS5, LRS6, LRS7, LRS8

IL1I, IL1R, OL1R, AI1C, AI2C, AI3C, AO1R, AO1C

RFRD, FRHD, ACC, DEC, AC2, DE2, INR, FRO

JPF, JF2, JF3, JFH

SP2, SP3, SP4, SP5, SP6, SP7, SP8, SP9, SP10, SP11, SP12, SP13, SP14, SP15, SP16

RPI, RP2, RP3, RP4, RPG, RIG, RDG, RPE, RPF, RPC, RPO, SPD

M001, M002, M003, M004, M005, M006, M007, M008

S001, SLSS, STOS, SS1S

- **SlowWriteFunctionBlocks**

HSP, LSP, BSP, RPR, ACC, DEC, AC2, DE2, INR, JPF, JF2, JF3, JFH

SP2, SP3, SP4, SP5, SP6, SP7, SP8, SP9, SP10, SP11, SP12, SP13, SP14, SP15, SP16,

RPI, RP2, RP3, RP4, RPG, RIG, RDG,

M001, M002, M003, M004, M005, M006, M007, M008, S006

ПРИМІТКА: При створенні програми FB в завданнях «PRE» та «POST» використання параметрів із груп **SlowRead** та **SlowWrite** може призвести до помилки виконання програми, але це з'ясується вже після завантаження й спроби запуску програми на виконання в ПЧ.

Параметри, яких немає у списку вище, не можуть бути записані в ATVLogic.

Типи завдань

Під «завданням» мається на увазі окрема програма в ПЧ, що реалізує певну задачу. Завдання викликаються циклічно: пріоритетні – з визначеним періодом, не пріоритетні – в фоновому режимі у решту процесорного часу.

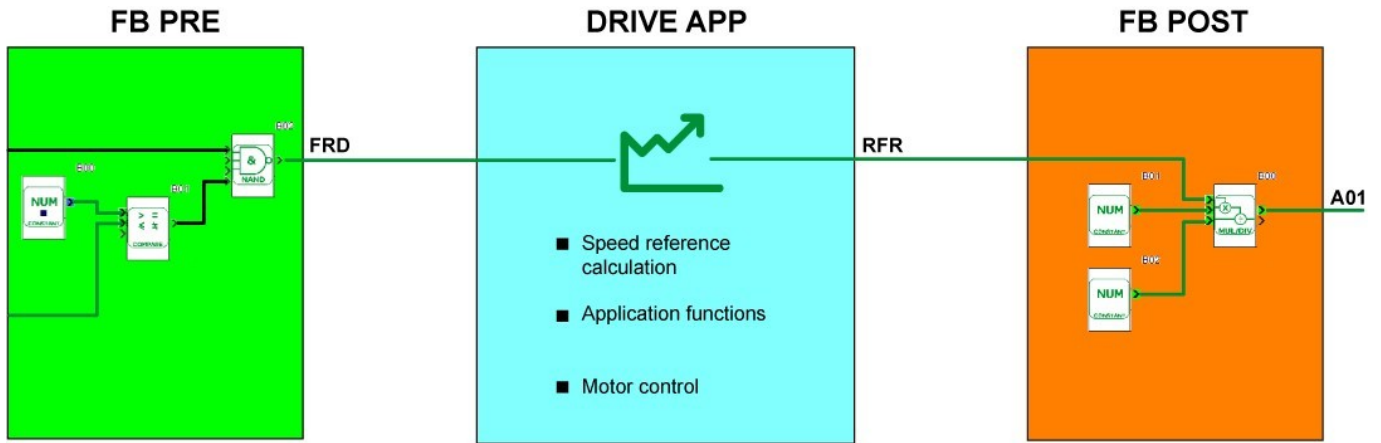
Завдання програми привода (**Drive Application Task**) виконуються в **режимі реального часу** в такій послідовності: **завдання перед основною програмою (FB PRE)** – **основна програма керування приводом (DRIVE APP)** – **завдання після основної програми (FB POST)**. Ця «**трийця**» повторюється кожні 2 мс (З), а її тривалість менше 2 мс.

Інші завдання виконуються в **фоновому режимі**: **фонові завдання привода (DRIVE BACKGROUND)** – **допоміжна програма (FB AUX)**.

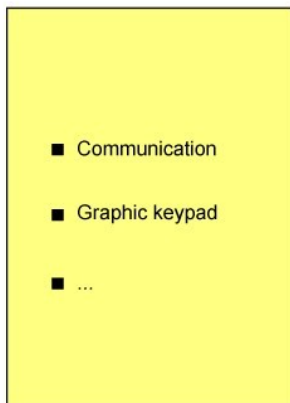
PRE Завдання PRE виконується в пріоритеті перед основним завданням привода, використовується для опитування входів і формування завдання керування двигуном. Цикл = 2 мс з синхронізацією.

POST Завдання POST виконується в пріоритеті після основного завдання привода, використовується для формування вихідних сигналів для зворотного зв'язку про стан параметрів двигуна. Цикл = 2 мс з синхронізацією.

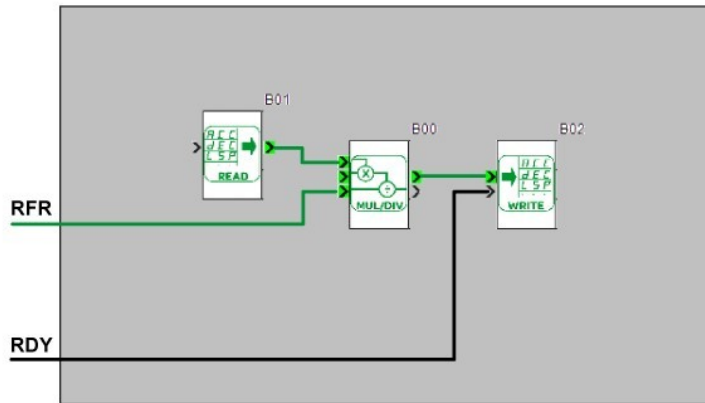
AUX Завдання AUX, використовується для виконання довготривалого фонового завдання без пріоритету. Цикл залежить від довжини програми та пріоритетних завдань привода.



DRIVE BACKGROUND

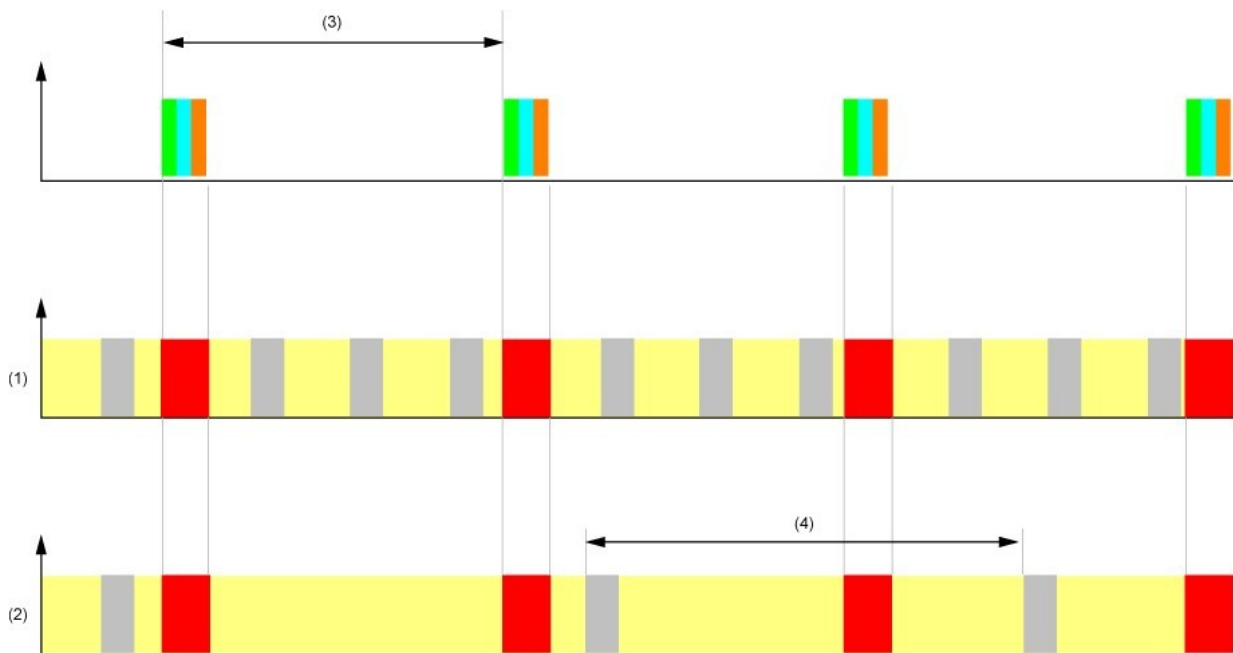


FB AUX



Фонове завдання привода може завантажуватися і, отже, виконуватися кілька разів між двома викликами завдання основної програми привода (1).

З іншого боку, якщо фонове завдання привода має велике навантаження, воно часто перериватиметься завданням програми привода (2).



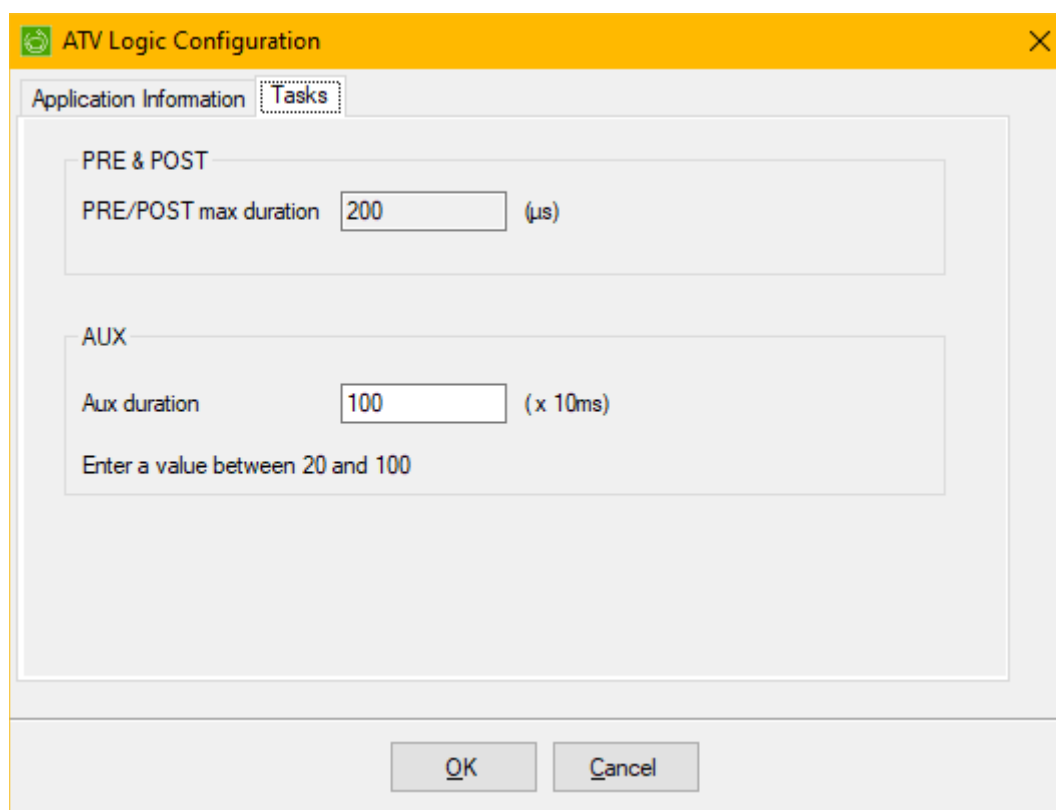
(1) Фонове завдання привода < 1 мс

(2) Фонове завдання привода > 1 мс

(3) 2 мс

(4) $t >$ тривалість завдання AUX = помилка несправності FBE TOAU

Оскільки час виконання завдання AUX може бути перервано завданням програми привода, неможливо визначити, чи є його тривалість фіксованою або меншою за заздалегідь визначений рівень. Однак тривалість періоду для AUX (меню **Device** → **ATVLogic** → **Configure** вкладка **Tasks** параметр «Aux duration») допомагає гарантувати, що завдання AUX оновлюється достатньо часто.



ПРИМІТКА: Якщо команди «RUN» та «STOP» змінюються в завданні AUX, їх активація може бути відкладена через пріоритети завдань PRE та POST.

Завдання FB PRE і POST виконуються в **Drive Application Task**, яка сама виконується кожні 2 мс. Тривалість **Drive Application Task** становить менше 2 мс. Це залишає від 500 мкс до 1 мс для виконання завдання програми привода. Однак фонове завдання привода може перериватися щоразу, коли потрібно виконати завдання програми привода.

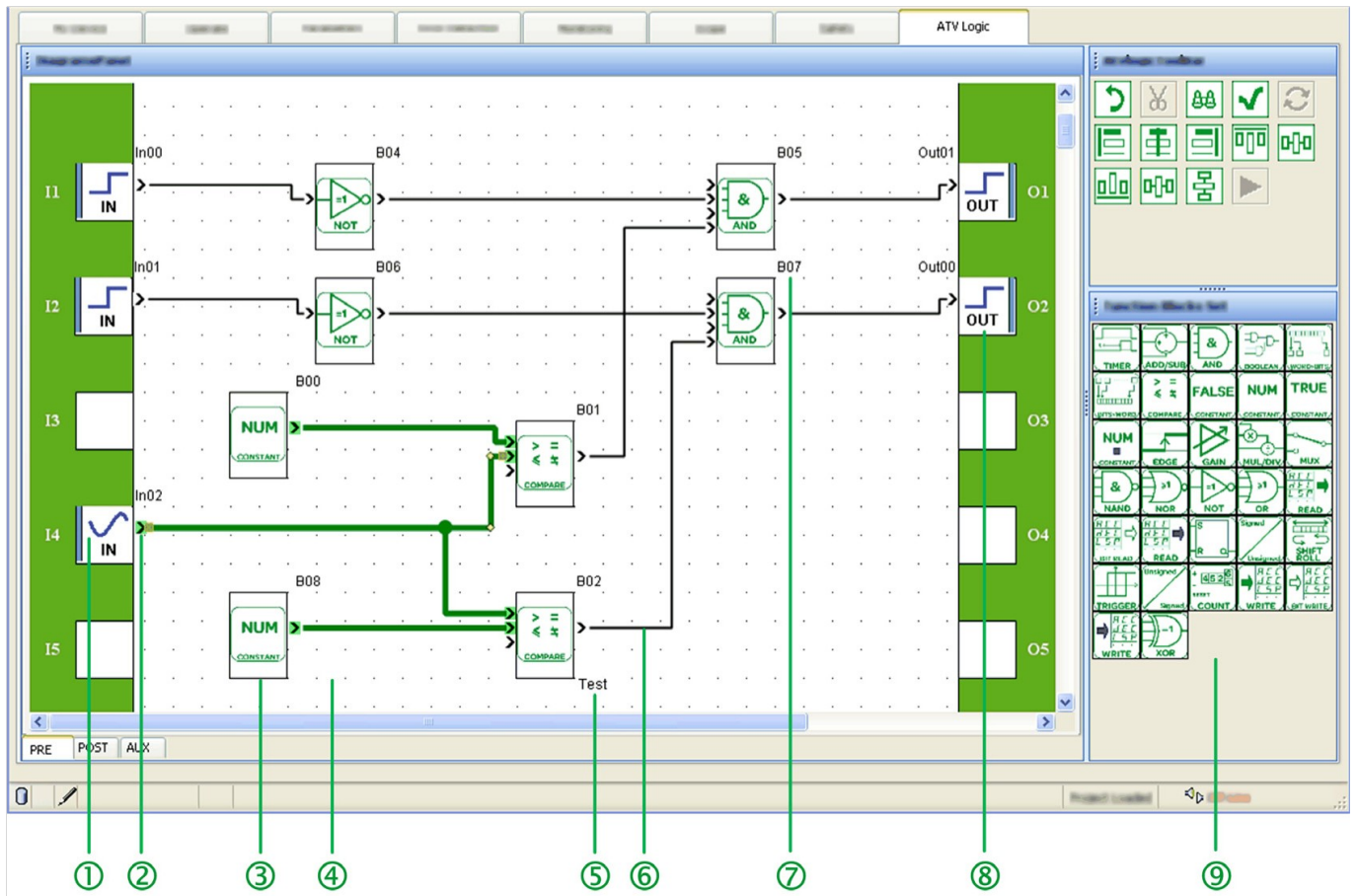
Програми в завданнях PRE, POST, AUX можуть існувати одночасно, проте використовуються загальні ресурси. Це означає, наприклад, при призначенні віртуального виходу R2 в завданні AUX його не можна використати в інших завданнях.

3. Створення програми FB

Створення й будь-яке редагування програми FB в середовищі програмного забезпечення SoMove зі встановленою бібліотекою Altivar Machine 320 DTM і доступно лише в офлайн режимі, тобто без підключення до ПЧ.

Одразу треба визначитись в якому завданні (PRE, POST чи AUX) створюватиметься схема, оскільки перенесення за допомогою «Вирізати - Вставити» не завжди працює коректно.

Інтерфейс



①: Контакт віртуального входу, ②: З'єднувач, ③: Функціональний блок, ④: Поле діаграми, ⑤: Коментарі, ⑥: З'єднання між блоками, ⑦: Номер функціонального блоку, ⑧: Контакт віртуального виходу, ⑨: Панелі інструментів: Засоби редагування, Набір функціональних блоків, Масштабування, Інструменти.

Панель інструментів

Для роботи з функціональними блоками та програмою FV на панелі **AtvlogicToolBar** є декілька інструментів. В залежності від обраного елемента чи стану виконання програми, деякі інструменти (кнопки) можуть бути неактивними.



Призначення кнопок

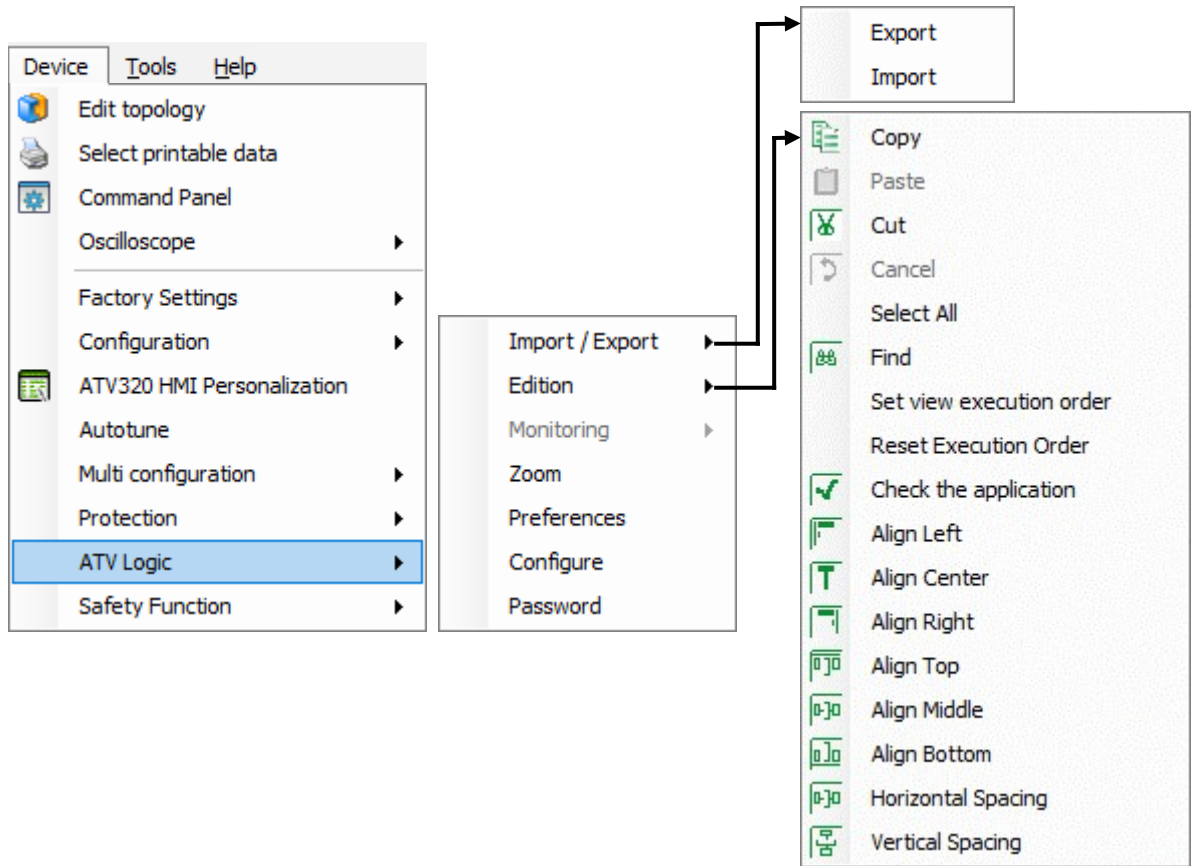
	Скасувати останню дію		Вирівняти обрані елементи праворуч
	Вирізати блок/блоки		Вирівняти обрані елементи за верхнім краєм
	Знайти		Вирівняти обрані елементи по середині горизонтально
	Перевірка програми (компіляція)		Вирівняти обрані елементи за нижнім краєм

	Оновлення відображення під час роботи програми		Розподілити горизонтально
	Вирівняти обрані елементи ліворуч		Розподілити вертикально
	Вирівняти обрані елементи по середині вертикально	 	Запуск програми FB на виконання Зупинка програми FB

Більше інструментів в меню **Device** → **ATVLogic** → **Edition**

Структура меню

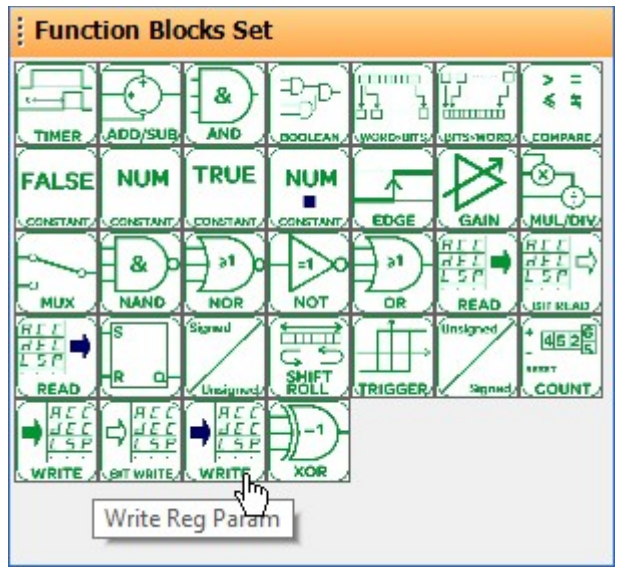
Більшість налаштувань програми FB, інструменти для роботи з блоками розташовані в меню **Device** → **ATVLogic**



В залежності від стану програми та обраних блоків деякі елементи в меню можуть бути не активні.

Набір функціональних блоків

Для створення FB програми в наборі функціональних блоків (**Function Block Set**) доступні різні функції. Кожне зображення в наборі функціональних блоків представляє окремий тип функції. При наведенні вказівника миші на один з типів функцій з'являється її назва. Кожен блок детально описаний у Елементи мови FBD.



Вставка функціональних блоків

Щоб додати функціональний блок, клацніть лівою кнопкою миші на потрібний блок в **Function Block Set** і перетягніть вибраний блок в порожнє місце на діаграмі.

Блоки неможливо розмістити доволі щільно, тому якщо перетягувати блок із набору і відпустити його близько до вже наявних на схемі елементів, новий блок може не вставитись – просто зникне. Спробуйте перетягнути повторно у більш вільне місце.

Створення віртуального порту

Щоб додати віртуальний вхід (вихід) потрібно двічі клацнути лівою кнопкою миші на порожній контактній площадці входу (I1, I2,...I20) чи виходу (O1,O2,...O20)



З'єднання блоків

У наведеному нижче процесі описано, як зв'язати функціональні блоки разом:

Крок	Дія	Коментар
1	Клацніть лівою кнопкою миші на виході функціонального блоку.	Результат: Курсор миші відображається у вигляді хрестика.
2	Затисніть ліву кнопку.	
3	Утримуючи натиснутою кнопку, наведіть курсор на вихід блоку.	
4	Відпустіть кнопку миші.	Результат: Лінія відображається між двома зв'язаними блоками
5	Повторіть кроки 1 і 2, щоб зв'язати всі блоки.	

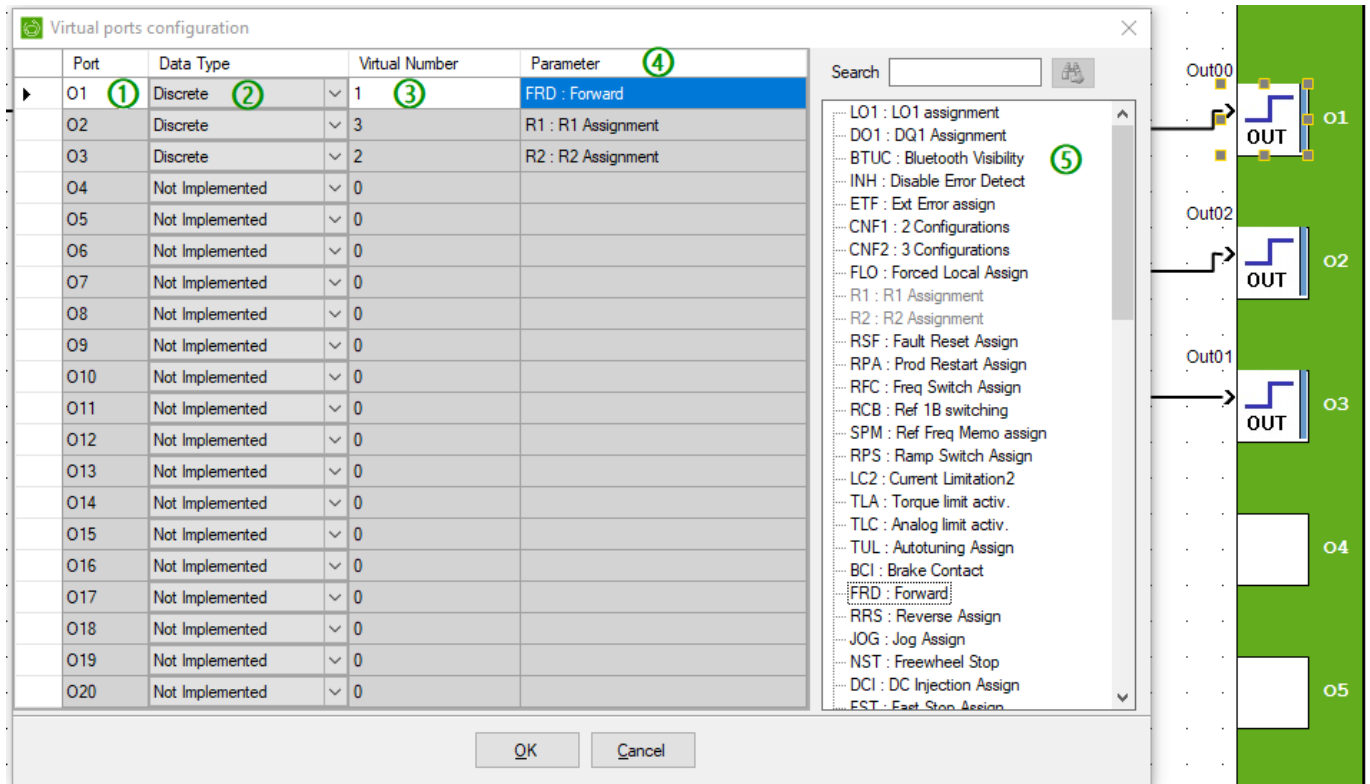
Типи з'єднання

В залежності від типу даних, лінія з'єднання (а також відповідні входи й виходи блоків) позначаються різним кольором

З'єднання	Опис	Можливе значення
	Логічне з'єднання – чорна лінія.	1 або 0
	Аналогове з'єднання – зелена лінія.	від -32768 до 32767
	Регістрове з'єднання – синя лінія.	від 0 до 65535

3.1. Конфігурація віртуальних входів та виходів

Кожен вхід або вихід діаграми має вікно конфігурації, яке викликається подвійним клацанням на контактній площадці входу/виходу

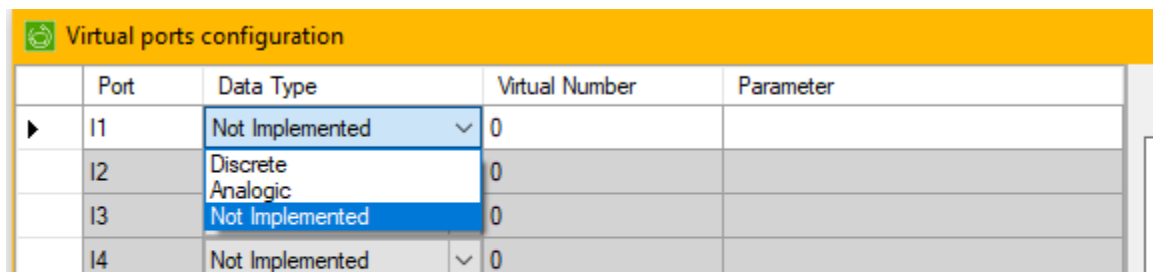


①: Назва віртуального порту (той, що редагується — білий), ②: Тип даних, ③: Номер віртуального порту, ④: Обраний параметр, ⑤: Перелік можливих параметрів, в залежності від типу даних.

В полі ⑤ з'являється перелік можливих функцій тільки після вибору типу даних в ② та при клацанні курсором в поточне поле редагування стовбця ④.

ПРИМІТКА. Вже використані параметри в переліку ⑤ відображаються світло-сірим і не доступні для поточного вибору.

При подвійному клацанні на вході/виході можна змінити тільки цей порт; решта – сірі. Конфігурація віртуальних входів аналогічна зі своїм переліком параметрів



Типи контактів входів/виходів, які можна налаштувати подвійним клацанням по ним:

- Призначення логічного входу 1 (IL01) - 10 (IL10): Логічні входи
- Призначення аналогового входу 1 (IA01) - 10 (IA10): Аналогові входи
- Призначення логічного виходу 1 (OL01) - 10 (OL10): Логічні виходи
- Призначення аналогового виходу 1 (OA01) - 10 (OA10): Аналогові виходи

ПОПЕРЕДЖЕННЯ

ОТРИМАННЯ НЕОЧІКУВАНИХ ДАНИХ

Аналогові входи **IAXX** та виходи **OAXX** мають цифрове значення даних лише в діапазоні **[0...8192]**, замість заявлених в інструкції **[- 32767...32767]**. Дані, що перевищують 8192 – обмежуються цим значенням.

Цифрові значення аналогових та реєстрових сигналів часом важко порівнювати із параметрами, що можна бачити при безпосередньому налаштуванні ПЧ.

Наприклад, значення моменту двигуна $\alpha \text{ } \tau \text{ } \gamma$ оцінюється у відсотках і в файлі параметрів [3] зазначено максимальне значення реєстра 3276,7. Але якому саме значенню відсотків відповідає це максимальне число? Згідно інструкції з програмування [5] максимальне значення моменту згадується в параметрі [Motoring torque lim] ($\tau \text{ } L \text{ } \gamma$) з можливим діапазоном 0...300%, а діапазон значень реєстра 0...3000. Можна сподіватися, що максимальне значення $\alpha \text{ } \tau \text{ } \gamma$ також 3000 (а зовсім не 3276,7) і відповідає воно 300%, але це не точно.

Попередня перевірка отриманих даних обов'язкова!

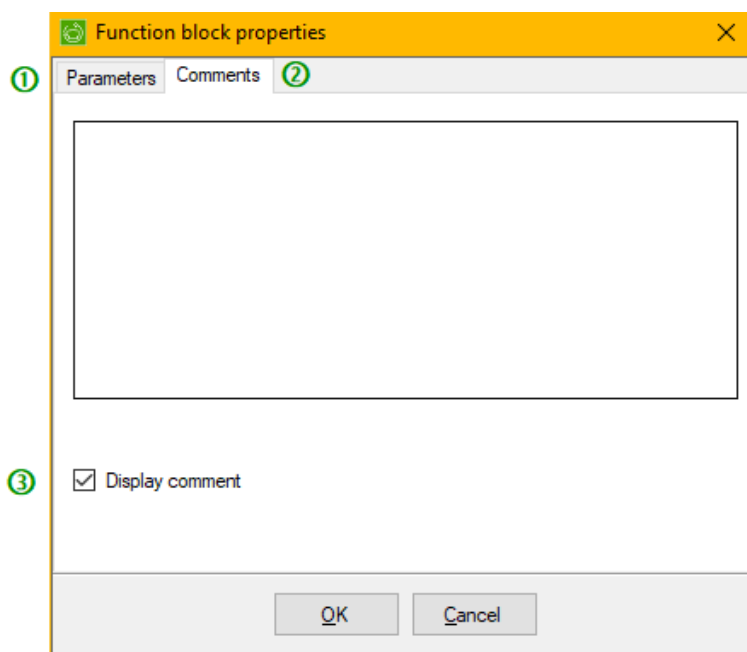
3.2. Властивості функціональних блоків

Властивості блоку

Кожен функціональний блок має вікно властивостей, яке може містити:

- Коментарі до функціонального блоку.
- Параметри, що залежать від типу функціонального блоку.

При подвійному натисканні на блок з'являється вікно властивостей блоку.



①: Вкладка **Parameters**: Більшість функціональних блоків мають цю вкладку, але не всі. У цій вкладці потрібно задати конкретні параметри функціонального блоку. Параметри докладно описані в п.4.

②: Вкладка **Comments**: Усі блоки можуть мати пов'язаний коментар. Ці коментарі відображаються поруч із блоком на схемі.

③: **Display comment**: Коли коментар додано до ФБ, то він відобразатиметься, якщо встановлено цей прапорець. Ця опція активна за замовчуванням **Device** → **ATVLogic** → **Preferences**.

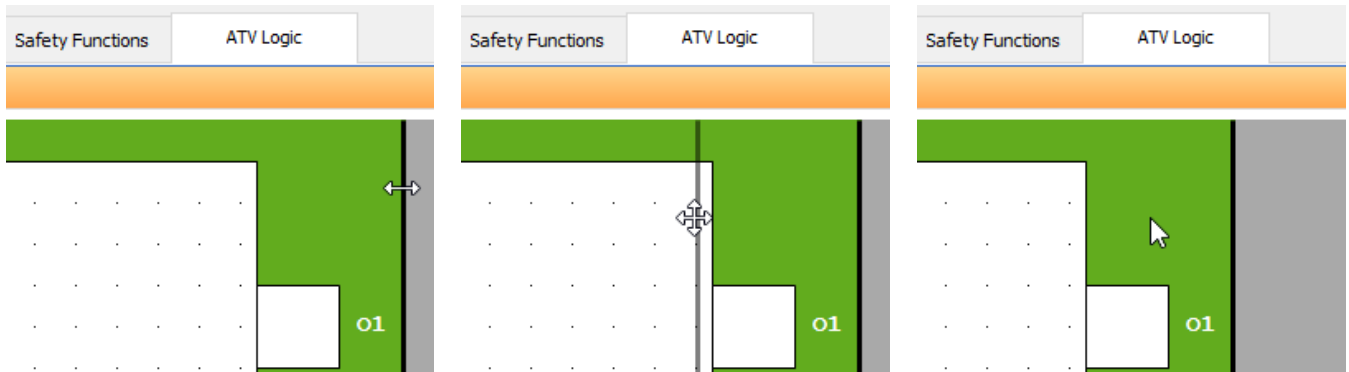
Властивості віртуального порту

До віртуального входу чи виходу можна додати коментар, як і до будь-якого функціонального блоку, для цього потрібно двічі натиснути безпосередньо на контакт підключення > входу чи виходу.

3.3. Параметри відображення

Розміри поля діаграми

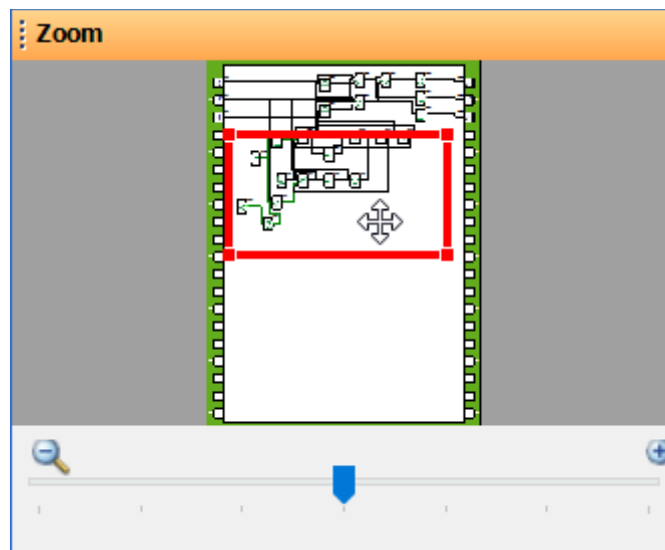
За необхідності можна змінити розміри поля діаграми за шириною, для цього треба захопити правий край поля перетягнути його ліворуч або вправоруч.



При наявності блоків в полі діаграми, спроба зменшити розміри може бути невдалою.


Масштабування області відображення діаграми **Device** → **ATV Logic** → **Zoom**.

Праворуч на панелі з'являється вікно **Zoom**, повзунком знизу можна змінювати масштаб. Червоний прямокутник показує, що відображається на **DiagramsPanel**, його можна переміщувати для швидкої навігації по схемі та змінювати розмір.

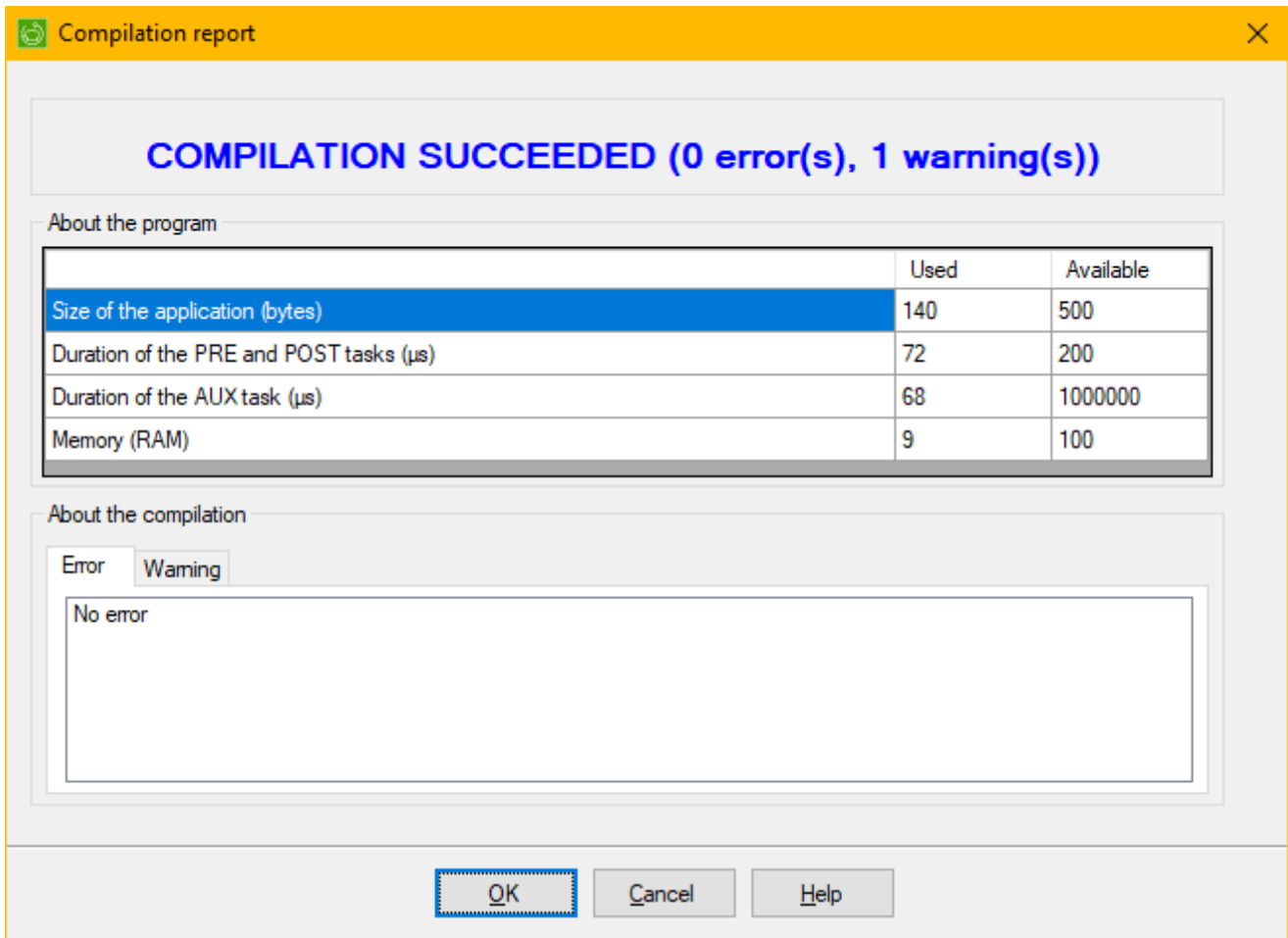


Якщо в налаштуваннях (п.3.6) обрано **Set Zoom Auto-Hide**, то вікно **Zoom** буде розташовано ліворуч у вигляді вкладки, при наведенні на яку вікно розкриватиметься.

3.4. Перевірка програми

Після створення програми FB, можна її перевірити на наявність критичних помилок, для цього на панелі **AtvlogicToolBar** натискаємо на кнопку  перевірки програми, в результаті відкриється вікно **Compilation report**, в якому наведені

- відомості про програму (розмір в байтах; тривалість виконання завдань PRE + POST в мкс; тривалість виконання завдання AUX в мкс; обсяг зайнятої пам'яті RAM);
- результат компіляції: критичні помилки Error та попередження Warning.



3.5. Збереження проєкту / програми FB

При створенні проєкту є можливість зберігти як весь проєкт загалом, так й окремо створену програму FB.

Проект

Проект SoMove прив'язаний до конкретного референсу ПЧ, містить всі налаштування параметрів двигуна, ПЧ, програму FB.

- Зберігається в меню **File** → **Save As...** із розширенням *.psx, або F12.
- Відкривається із файлового менеджера windows чи кнопкою «Open a Project» зі стартового вікна SoMove.

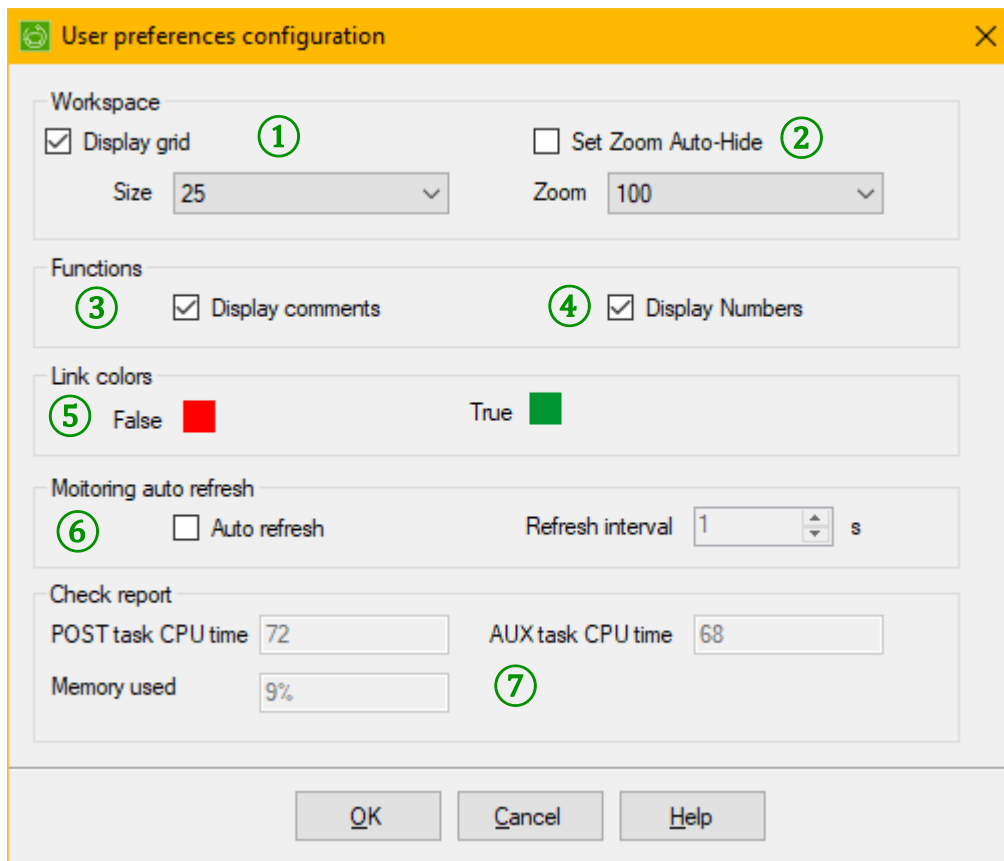
Програма FB

Програма FB не прив'язана до конкретного ПЧ і може бути збережена окремим файлом, а потім завантажена в будь-який проєкт ПЧ ATV320. При цьому цей файл містить лише програму FB і параметри її запуску та керування, тобто значення параметрів ПЧ в меню Function Blocks, і не торкається параметрів керування ПЧ й двигуна.

- Зберігається в меню **Device** → **ATV Logic** → **Edition** → **Import / Export** → **Export** із розширенням *.atvlogic.
- Відкривається в меню **Device** → **ATV Logic** → **Edition** → **Import / Export** → **Import**.

3.6. Налаштування

Користувач може змінити налаштування відображення програми в меню **Device** → **ATVLogic** → **Preferences**

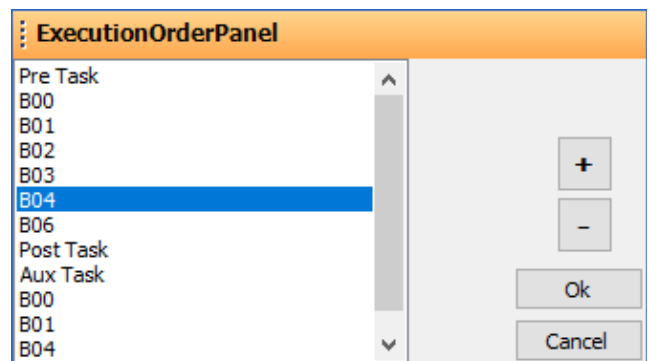


①: Відображення сітки на полі діаграми (12,5; 25; 50; 100); ②: Відображення вікна масштабування ліворуч від діаграми, яке ховається та вибір поточного масштабування; ③: Відображення коментарів функціональних блоків за замовчуванням; ④: Відображення номерів функціональних блоків за замовчуванням; ⑤: Вибір кольору відображення стану сигналів булевих ліній з'єднання; ⑥: Автоматичне оновлення запущеної програми (1...300 с з кроком 1 с); ⑦: Відомості про програму (тривалість виконання завдань PRE + POST в мкс; тривалість виконання завдання AUX в мкс; відсоток зайнятої пам'яті RAM).

3.7. Порядок виконання

В меню **Device** → **ATV Logic** → **Edition** → **Set view execution order** можна змінити порядок виконання.

Після вибору цього пункту, праворуч на панелі з'являється вікно. За допомогою кнопок «+» та «-» можна змінювати позицію вибраного блоку в переліку, але тільки в межах завдання, в якому він знаходиться. Після зміни порядку в переліку треба підтвердити вибір кнопкою «Ok» або скасувати зміни «Cancel».

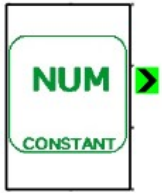


Повернути початковий порядок виконання **Device** → **ATVLogic** → **Edition** → **Reset Execution Order**.

4. Елементи мови FBD

4.1. Спеціальні функції

NUM (analog)

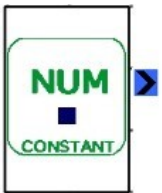


Числова константа NUM (analog) є аналоговим цілим числом зі значенням від -32768 до +32767. Значення константи можна встановити у вкладці Parameters.

Parameters Comments

Value of the constant 0 (-32768...32767)

NUM (register)

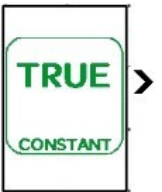


Числова константа NUM (register) є цілим регістром зі значенням від 0 до 65535. Значення константи можна встановити у вкладці Parameters.

Parameters Comments

Value of the constant 0 (0...65535)

TRUE



Ця константа може бути використана для встановлення входів функції в 1.

FALSE



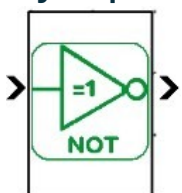
Ця константа може бути використана для встановлення входів функцій в 0.

4.2. Логічні функції

У мові FBD можна використовувати логічні функції при створенні програми. Доступ до цих функцій можна отримати за допомогою **Function Block Set**.

ПРИМІТКА: Якщо один або кілька входів не підключені, їх стан не враховується при розрахунку вихідних значень функцій. Однак входи **ENABLE** завжди активовані.

Функція NOT

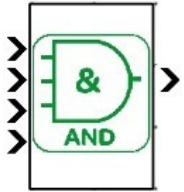


Якщо вхід неактивний або не підключений, вихід активний.
Якщо вхід активний, вихід неактивний.

- [IN]: 1 логічний вхід
- [Q]: 1 логічний вихід

IN	Q
0	1
1	0

Функція AND

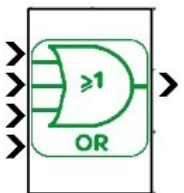


Якщо всі входи активні або не підключені, вихід активний. Якщо підключено хоча б один вхід, вихід неактивний.

- Від [IN1] до [IN4]: 4 логічні входи
- [Q]: 1 логічний вихід

IN1	IN2	IN3	IN4	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Функція OR

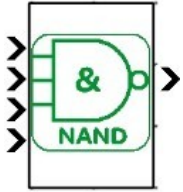


Якщо хоча б один вхід активний, вихід активний. Якщо всі входи неактивні або не підключені, вихід неактивний.

- Від [IN1] до [IN4]: 4 логічні входи
- [Q]: 1 логічний вихід

IN1	IN2	IN3	IN4	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Функція NAND

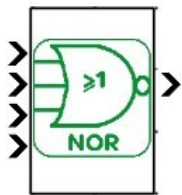


Якщо хоча б один вхід неактивний, вихід активний. Якщо всі входи активні або не підключені, вихід неактивний.

- Від [IN1] до [IN4]: 4 логічні входи
- [Q]: 1 логічний вихід

IN1	IN2	IN3	IN4	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Функція NOR

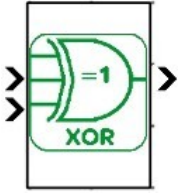


Якщо всі входи неактивні або не підключені, вихід активний. Якщо хоча б один вхід активний, вихід неактивний.

- Від [IN1] до [IN4]: 4 логічні входи
- [Q]: 1 логічний вихід

IN1	IN2	IN3	IN4	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Функція XOR (eXclusive OR)

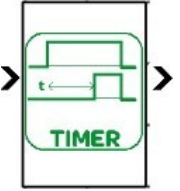


Якщо один вхід неактивний, а інший вхід активний або не підключений, вихід активний. Якщо обидва входи активні або неактивні, або не підключені, неактивний.

IN1	IN2	Q
0	0	0
0	1	1
1	0	1
1	1	0

- [IN1] та [IN2]: 2 логічні входи
- [Q]: 1 логічний вихід

Функція A/C TIMER (Timer)



Функція «Таймер» використовується для затримки, продовження та керування діями протягом заданого часу.

Таймер має три типи:

- Тип А: затримка увімкнена-таймера або активний таймер
- Тип С: затримка вимкнення таймера або таймер у режимі очікування
- Тип А/С: комбінація типів А і С

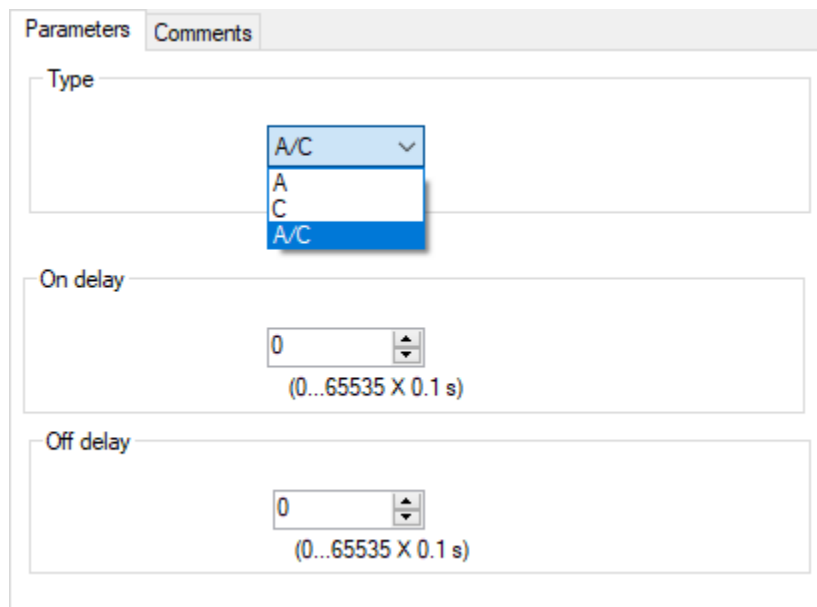
Входи/виходи

- [IN]: 1 логічний вхід
- [Q]: 1 логічний вихід

Конфігурація

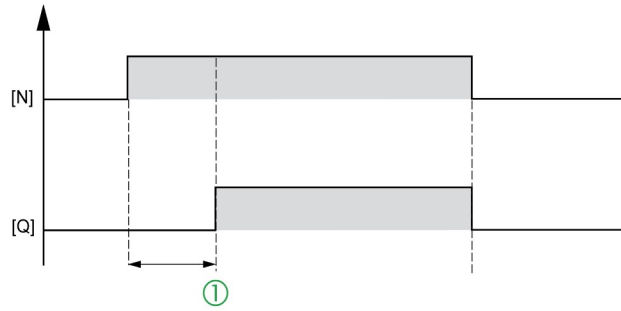
Ви можете налаштувати параметри у вікні властивостей. У вкладці Parameters можна задати значення затримок для кожного типу (А та С):

- Затримка на увімкнення для типу А
- Затримка на вимкнення для типу С
- Комбінація затримок увімкнення та вимкнення може бути використана для налаштування типу А/С.



Тип А

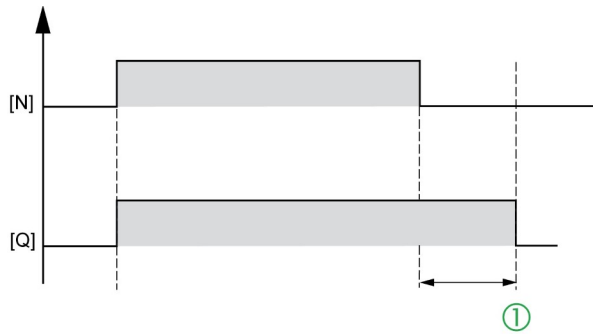
На наступній схемі показана робота таймера типу А.



①: Затримка на увімкнення

Tun C

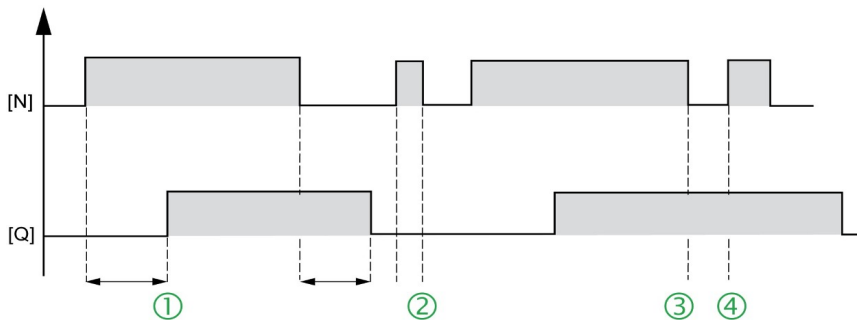
На наступній схемі показана робота таймера типу C.



①: Затримка на вимкнення

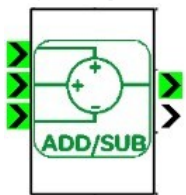
Tun A/C

На наступній схемі показана робота таймера типу A/C.



①: Затримка на увімкнення, ②: Затримка на вимкнення, ③: $t <$ Затримки на увімкнення, ④: $t <$ Затримки на вимкнення

Функція ADD/SUB



Арифметична функція ADD-SUB використовується для виконання простих операцій над цілими числами:

- Додавання
- Віднімання

Формула розрахунку

$$\text{РОЗРАХУНОК ВИХОДУ } [Q] = [A] + [B] - [C]$$

Входи/виходи

Опис входів:

- [A]: Аналоговий вхід

- [B]: Аналоговий вхід
- [C]: Аналоговий вхід

ПРИМІТКА: Якщо входи не підключені, вони встановлюються в 0. Математичні операції з регістрами виконати неможливо.

Опис виходів:

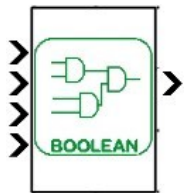
- [Q]: Аналоговий вихід: це значення вихідних даних формули розрахунку.
- [OVERFLOW]: Цей вивід, який має булевий тип, вказує на наявність помилок переповнення. Цей вихід активується в наступних випадках.
- Результат не входить в інтервал $[-32768, +32767]$

Особливості

Додавання: Не використовуйте вхід [C].

Віднімання: Не використовуйте [A] або [B].

Функція BOOLEAN



Функція BOOLEAN дає значення виводу відповідно до комбінації входів.

Функція має чотири входи, а значить і 16 комбінацій. Ці комбінації можна знайти в таблиці істинності; Для кожного з них можна регулювати вихідне значення. Кількість конфігурованих комбінацій залежить від кількості входів, підключених до функції. Непідключені входи встановлюються в 0.

Входи/виходи

Опис входів:

- [IN1], [IN2], [IN3] та [IN4]: 4 логічні входи
- [OUT]: 1 логічний вихід

Конфігурація

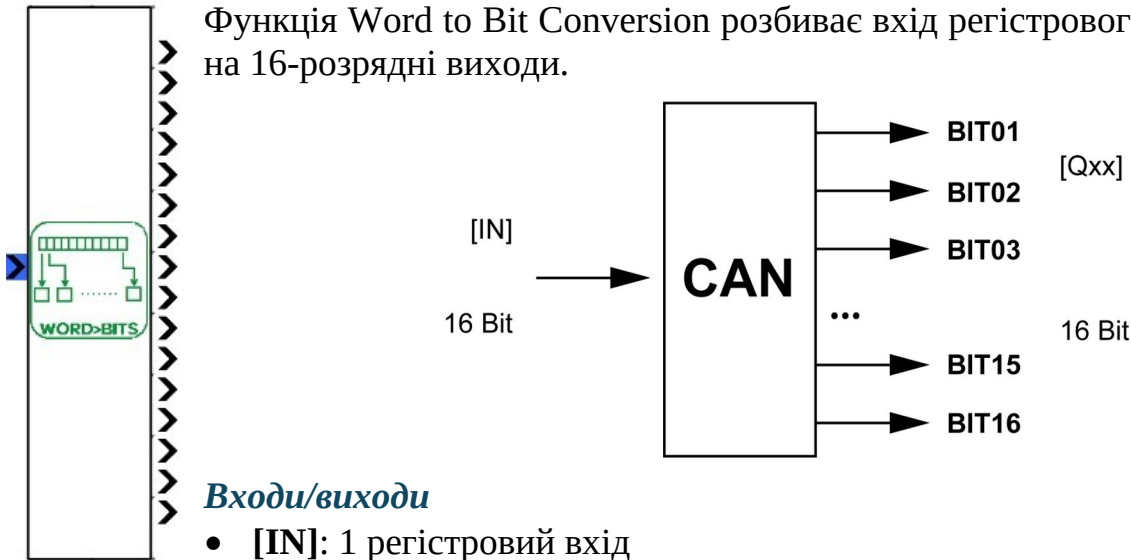
У вікні властивостей ви можете налаштувати параметри. Підключивши хоча б один вхід, налаштуйте значення виводу в таблиці істинності, у вкладці Parameters.

Вихідні значення можуть бути 0 для неактивного стану та 1 для активного стану.

IN1	IN2	IN3	IN4	OUT
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0

Функція CAN (Word to Bit Conversion)

Функція Word to Bit Conversion розбиває вхід регістрового типу (16-біт) на 16-розрядні виходи.

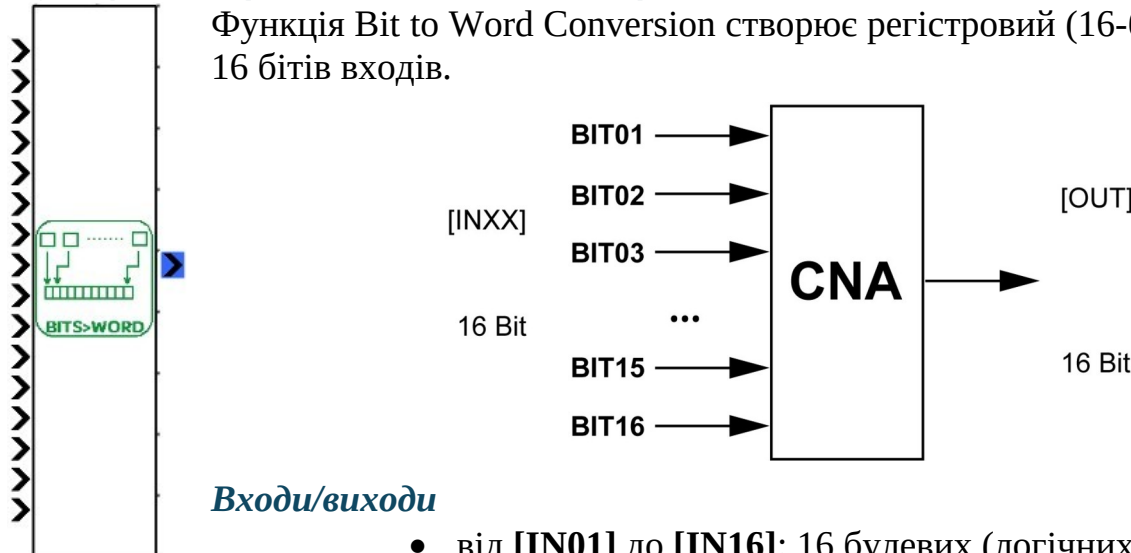


Входи/виходи

- **[IN]**: 1 регістровий вхід
- **[Q01 ... Q16]**: 16 бітів виходів: Bit01 (молодший біт) ... Bit16 (старший біт).

Функція CNA (Bit to Word Conversion)

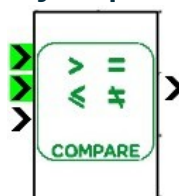
Функція Bit to Word Conversion створює регістровий (16-бітний) вихід з 16 бітів входів.



Входи/виходи

- від **[IN01]** до **[IN16]**: 16 булевих (логічних) входів: Bit01 (молодший біт) ... Bit16 (старший біт)
- **[OUT]**: 1 регістровий вихід

Функція COMPARE (Порівняння двох значень)



Ця функція використовується для порівняння двох аналогових значень. Вихід **Q** є активним, якщо результат порівняння між **IN1** та **IN2** істина, а також якщо вхід функції **ENABLE** активний або не підключений. Вихід не змінює стан, якщо вхід для функції **ENABLE** змінюється з Активного на Неактивний.

Входи/Виходи

- **[IN1]**: 1 аналоговий вхід
- **[IN2]**: 1 аналоговий вхід
- **[ENABLE]**: 1 логічний вхід

Якщо вхід **IN1** або **IN2** не підключений, значення встановлюється в 0.

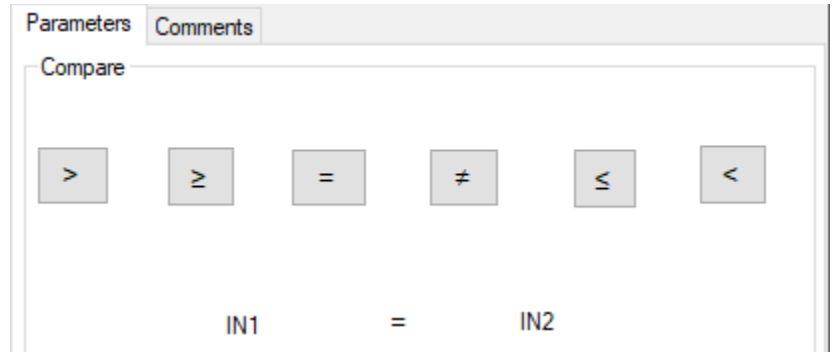
ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід **ENABLE** перемикається в 0, вихід залишиться на рівні 1.

- [Q]: 1 логічний вихід

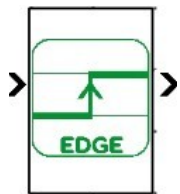
Конфігурація

У вікні властивостей у вкладці Parameters треба вибрати один з можливих операторів порівняння:

- > Більше
- ≥ Більше або дорівнює
- = Дорівнює
- ≠ Не дорівнює
- ≤ Менше або дорівнює
- < Менше



Функція EDGE



Ця функція виявляє фронт сигналу, що спадає або зростає.

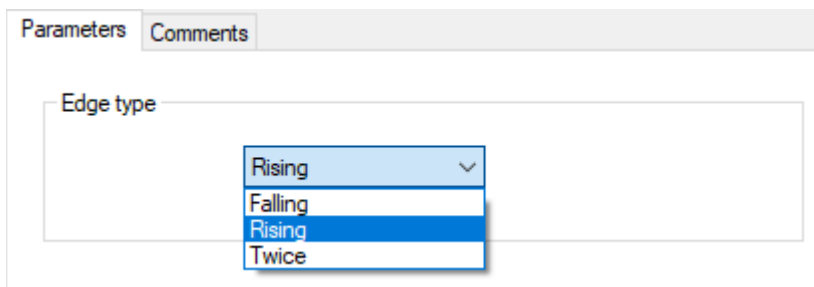
Входи/виходи

- [Q]: 1 логічний вихід. Тестуємий сигнал.
- [IN]: 1 логічний вхід. Пульс при виявленні фронту вхідного сигналу.

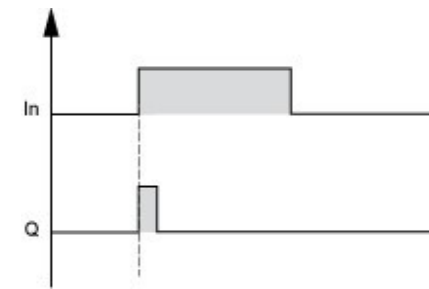
Конфігурація

У вікні властивостей вкладка Parameters налаштуйте тип фронту, за яким буде генеруватись пульс вихідного сигналу функції:

- Falling (задній фронт)
- Rising (передній фронт)
- Twice (передній і задній фронт)



Тип фронту = Rising



Тип фронту		In (t-1)	In (t)	Q (t)
Twice (передній і задній фронт)	Falling (задній фронт)	1	0	1
		0	0	0
	Rising (передній фронт)	1	1	0
		0	1	0

Функція GAIN



Функція підсилення дозволяє перетворювати аналогові значення шляхом зміни шкали та зміщення.

Формула розрахунку коефіцієнта підсилення:

$$[Q] = (A / B) \times [IN] + C.$$

Опис входів

- **[IN]**: 1 аналоговий вхід. Це ціле число від -32768 до 32767 .
- **[ENABLE]**: 1 логічний вхід. Це команда входу функції підсилення.

Стан цього входу визначає роботу блоку: якщо вхід **ENABLE** неактивний, вихід **Q** зберігає останнє обчислене значення.

Опис виходів

- **[Q]**: 1 аналоговий вихід. Це вихідне значення функції підсилення, воно залежить від стану входу функції **ENABLE**.
- **[OVERFLOW]**: 1 логічний вихід. Якщо він:
 - Активний: вихід **Q** дорівнює результату формули розрахунку підсилення.
 - Неактивний: вихідні дані обчислення дорівнюють нулю.

ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід **FB** встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

Конфігурація

У вікні властивостей у вкладці Parameters встановлюємо:

A/B, що відповідає коефіцієнту підсилення, який застосовується функцією:

- **A**: від -32768 до 32767
- **B**: від -32768 до -1 і від 1 до 32767

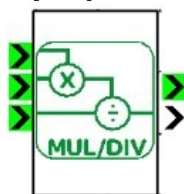
C – це зміщення, застосоване функцією, і є цілим числом

- **C**: від -32768 до 32767



A, B, C – тільки цілі значення.

Арифметична функція MUL/DIV



Арифметична функція **MUL/DIV** використовується для виконання операцій над цілими числами:

- Множення
- Ділення

Формула розрахунку **MUL/DIV**: $[Q] = [A] \times [B] / [C]$

Опис входів

- [A]: 1 аналоговий вхід
- [B]: 1 аналоговий вхід
- [C]: 1 аналоговий вхід

ПРИМІТКА: Якщо ВХОДИ не підключені, вони встановлюються в 1.
Математичні операції з регістрами виконати неможливо.

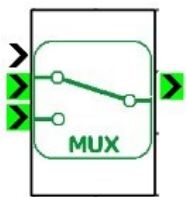
Опис виходів

- [Q]: 1 аналоговий вихід. Це і є значення вихідної формули розрахунку.
- [OVERFLOW]: Цей вихід, який є булевим типом, вказує на наявність будь-яких помилок насичення. Цей вихід активується в наступних випадках:
 - Наслідком операцій є результат, який виходить за межі діапазону [-32768, +32767]
 - Активний вхід поширення помилки
 - Вхід 3 дорівнює 0

Особливості

- Множення: Не використовуйте вхід С
- Ділення: Не використовуйте входи А або В
- А, В, С – тільки цілі значення.

Функція MUX (мультиплексування)



Функція Multiplexing здійснює мультиплексування двох вхідних каналів на виході.

Опис входів

- [SEL]: 1 логічний вхід. Цей вхід використовується для вибору вхідного каналу, який буде застосовано до виходу.
- [IN1]: 1 аналоговий вхід. Це вхід мультиплексора IN1.
- [IN2]: 1 аналоговий вхід. Це вхід мультиплексора IN2.

ПРИМІТКА: Якщо канали [IN1] або [IN2] не підключені, вони встановлюються в 0.

Опис виходів

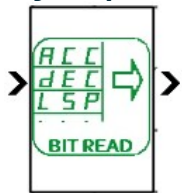
- [Q]: 1 аналоговий вихід. Це і є вихід мультиплексора Q.

Це значення залежить від стану входу керування SEL.

Якщо...	То...
вхід [SEL] неактивний	вихід відповідає [IN1].
вхід [SEL] активний	вихід відповідає [IN2].

ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід SEL перемикається на 0, вихід залишиться на рівні 1.

Функція BIT READ



Функціональний блок BIT READ використовується для зчитування одного біта параметрів.

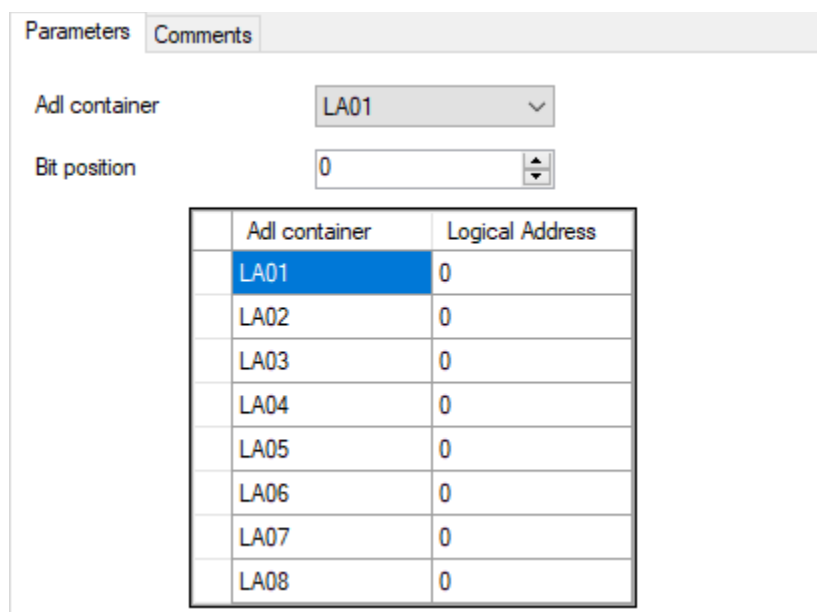
Для...	Використовуються...
зчитування одного біта параметрів привода	контейнери ADL для зв'язування параметрів привода. [ADL CONTAINERS] [F P d -]
зчитування біта параметра протоколу зв'язку	параметри M00x для зберігання даних. [FB PARAMETERS] [F b P -]

Опис входів / виходів

- [ENABLE]: 1 логічний вхід
- [Q]: 1 логічний вихід, значення зчитаного біту.

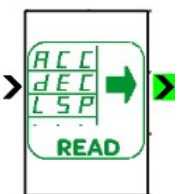
Конфігурація

У вікні властивостей ви можете налаштувати параметри. У вкладці Parameters ви можете вибрати контейнер ADL (LA0x), який буде використовуватися та номер біту.



Для зчитування значення біта, в **Logical Address** необхідно вписати адресу регістру в десятковому форматі (перелік відповідних адрес наведений в [3]) та в **Bit position** – номер біта.

Функція READ Ana Param



Для зчитування значення використовується блок функцій Read Ana Param.

Для...	Використовуються...
зчитування параметрів привода	контейнери ADL для зв'язування параметрів привода. [ADL CONTAINERS] [F P d -]
зчитування параметрів протоколу зв'язку	параметри M00x для зберігання даних. Зберігайте значення для повторного використання функціональними блоками та протоколом зв'язку. [FB PARAMETERS] [F b P -]

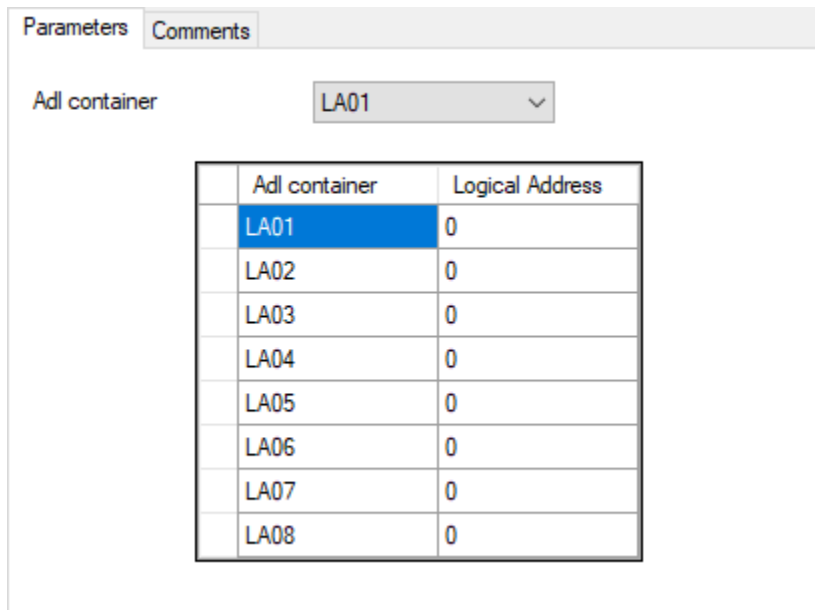
Входи / Виходи

- [ENABLE]: 1 логічний вхід
- [Q]: 1 аналоговий вихід

ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

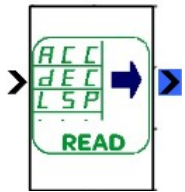
Конфігурація

У вікні властивостей ви можете налаштувати параметри. У вкладці Parameters ви можете вибрати контейнер ADL (LA0x), який буде використовуватися.



Для зчитування значення, в **Logical Address** необхідно вписати адресу регістру в десятковому форматі (перелік відповідних адрес наведений в [3]).

Функція READ Reg Param



Для зчитування значення використовується функціональний блок Read Reg Param.

Для...	Використовуються...
зчитування параметрів привода	контейнери ADL для зв'язування параметрів привода. [ADL CONTAINERS] [F A d -]
зчитування параметрів протоколу зв'язку	параметри M00x для зберігання даних. Зберігайте значення для повторного використання функціональними блоками та протоколом зв'язку. [FB PARAMETERS] [F b P -]

Входи / Виходи

- **[ENABLE]:** 1 Логічний вхід
- **[Q]:** 1 Регістровий вихід

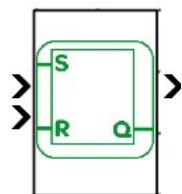
Конфігурація

У вікні властивостей ви можете налаштувати параметри. У вкладці Parameters ви можете вибрати контейнер ADL (LA0x), який буде використовуватися.

Parameters		Comments
Adl container LA01		
	Adl container	Logical Address
	LA01	0
	LA02	0
	LA03	0
	LA04	0
	LA05	0
	LA06	0
	LA07	0
	LA08	0

Для зчитування значення, в **Logical Address** необхідно вписати адресу регістру в десятковому форматі (перелік відповідних адрес наведений в [3]).

Функція RS Memory



Функція SET/RESET працює наступним чином:

- Активація входу **SET** активує вихід, який залишається в цьому стані, навіть якщо вхід **SET** потім деактивовано.
- Активація входу **RESET** деактивує вихід.

Непідключені входи встановлюються в положення Неактивні.

Опис входів / виходів

- **[SET]**: 1 логічний вхід
- **[RESET]**: 1 логічний вхід
- **[Q]**: 1 логічний вихід

Конфігурація

У вікні властивостей ви можете змінити налаштування на вкладці Parameters.

Якщо обидва входи активні, то стан виходу залежить від того, як налаштована функція:

- Вихід **Q** активний, якщо налаштована опція пріоритету **SET**
- Вихід **Q** неактивний, якщо налаштована опція пріоритету **RESET**

Parameters	Comments
Priority	
<input type="radio"/> SET has priority <input checked="" type="radio"/> RESET has priority	

Функція SHIFT/ROLL



ROLL переміщує біти в обраному напрямку і замінює нові порожні біти бітами, що викидаються з регістра. Він являє собою круговий регістр. Вихід **CARRY** містить значення (0/1) останнього переміщеного біта. Функція використовується для зсуву або перекочування значення **[IN]** фіксованої кількості бітів вліво/вправо.

На кожному такті, якщо встановлено сигнал **ENABLE**, логічний блок виконуватиме зсув/перекат вліво/вправо до запиту Number of bits, що стосується конфігурації параметру.

SHIFT переміщує біти в обраному напрямку і замінює порожні біти на 0.

ROLL переміщує біти в обраному напрямку і замінює нові порожні біти бітами, що викидаються з регістра. Він являє собою круговий регістр. Вихід **CARRY** містить рівень (0/1) останнього переміщеного біта.

ПРИМІТКА: Зсув записів з аналоговими значеннями неможливий.

Опис входів

- **[IN]:** 1 регістровий вхід
- **[ENABLE]:** 1 вхід булевої функції

Опис виходів

- **[Q]:** 1 регістровий вихід
- **[CARRY]:** 1 логічний вихід. Вихід перенесення повідомляє вам значення останнього зсунутого біта.

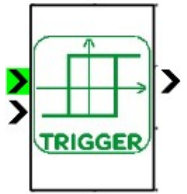
ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

Конфігурація

У вікні властивостей ви можете змінити налаштування у вкладці Parameters

- Режим: Зсув (SHIFT) або Перекочування (ROLL).
- Напрямок: ліворуч або праворуч.
- Кількість бітів.

Функція TRIGGER (Schmitt Trigger)



Функція Schmitt Trigger дозволяє контролювати аналогове значення щодо двох порогових значень.

Вихід змінює стан, якщо:

- Значення **IN** менше мінімального значення (Stop threshold)
- Значення **IN** більше за максимальне значення (Start threshold)

Якщо вхід **IN** знаходиться між ними, вихід **Q** не змінює стан.

Два задані значення – «Start threshold» і «Stop threshold» – можуть бути встановлені як мінімальне або максимальне значення. Це передбачає зворотню роботу функції. Ці дві операції показані на схемах.

Якщо вхід **ENABLE** неактивний, вихід **Q** залишається неактивним. Вихід **Q** не змінює стан, якщо вхід **ENABLE** змінюється з активного на неактивний.

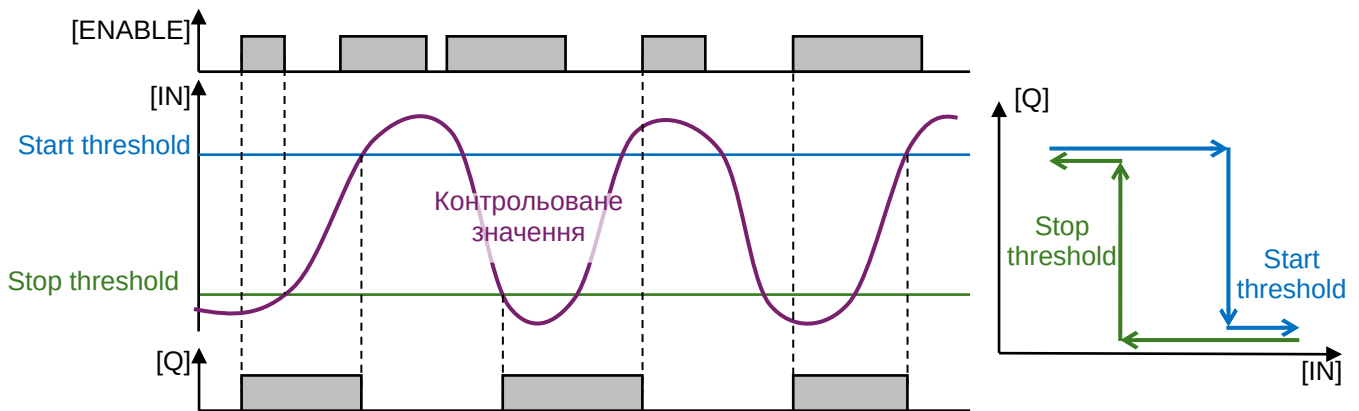
Входи/виходи

- **[IN]**: 1 аналоговий вхід заданого значення
- **[ENABLE]**: 1 логічний вхід
- **[Q]**: 1 логічний вихід

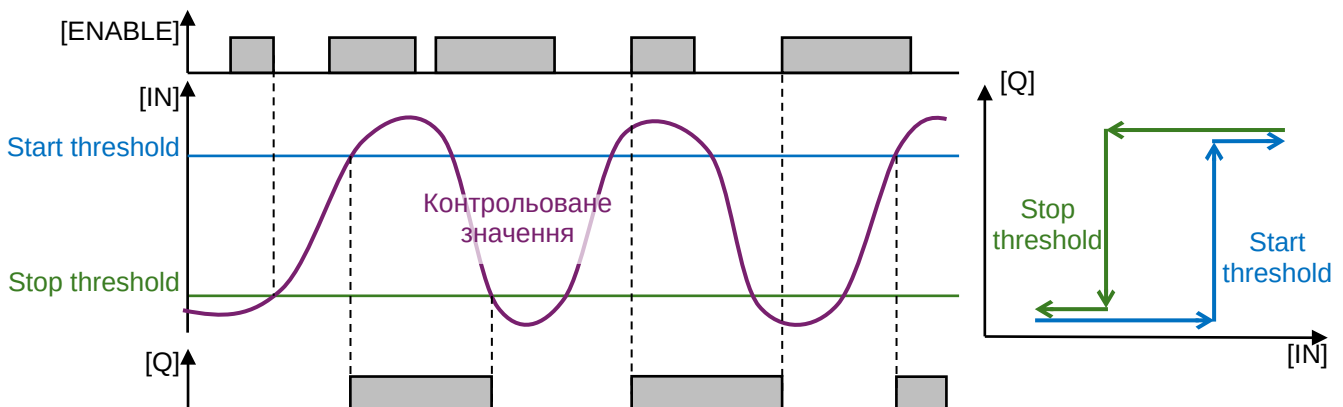
ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід **FB** встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

Схеми роботи

На рисунку нижче показані можливі стани виходу **Q**, коли уставка «Stop threshold» вища за уставку «Start threshold» уставки увімкнення.

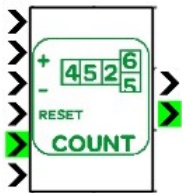


На рисунку нижче показано можливі стани виходу **Q**, коли уставка «Start threshold» вища за уставку «Stop threshold».



Parameters	Comments
Start threshold	
0	
	(-32768...32767)
Stop threshold	
0	
	(-32768...32767)

Функція лічильника COUNT Up/Down



Функція лічильника вгору/вниз використовується для відліку вгору/вниз від заданого значення, отриманого в результаті обчислення за межами функції.

Опис входів

- **[UP]**: 1 логічний вхід для збільшення лічильника
- **[DOWN]**: 1 логічний вхід зменшення лічильника
- **[RESET]**: 1 логічний вхід
- **[LOAD]**: 1 логічний вхід
- **[PV]**: 1 аналоговий вхід попередньо встановленого значення
- **[ENABLE]**: 1 вхід

Активація входу **LOAD** в 1 використовується для зміни значення лічильника на те, що подано на вхід **PV**.

До входу **PV** можна підключити вихід NUM (analog), чи до аналогового виходу будь-якого іншого функціонального блоку, який передає значення аналогового типу.

Зростаючий фронт на:

- Вхід збільшення: збільшує лічильник на 1
- Вхід зменшення: зменшує лічильник на 1

Опис виходів

- **[Q]**: 1 логічний вихід
- **[CURRENT]**: поточне значення лічильника

ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

Доступні функції

Доступно кілька функцій:

- Збільшення рахунку і примусове приведення лічильника до 0 при ініціалізації
- Збільшення рахунку і примусове приведення лічильника до 0 при ініціалізації і при досягненні значення рахунку
- Зменшення рахунку і примусове приведення лічильника до заданого значення при ініціалізації
- Зменшення рахунку і примусове приведення лічильника до заданого значення при ініціалізації і при досягненні 0

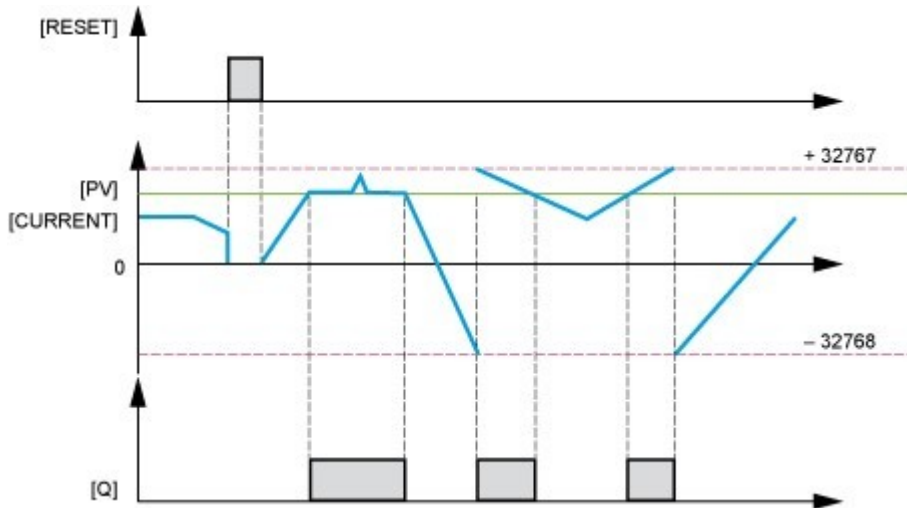
Стан вихідних даних

1: Коли досягається значення лічильника, вихід **Q** змінюється на 1 і залишається на рівні 1 доти, доки значення лічильника перевищує або дорівнює попередньо встановленому значенню **PV**.

0: Якщо переходи на вході зменшення лічильника змінюють значення лічильника назад на значення, менше встановленого значення

Активація входів **RESET** або **LOAD** скидає лічильник.

Коли вхід **RESET** встановлено в 1, вихід **Q** залишається в стані 0. Коли вхід **RESET** змінюється на 0, операція відліку вгору/вниз починається з нуля.



Функція BIT WRITE (WriteBitParam)



Функціональний блок WriteBitParam використовується для запису одного біта параметрів.

Для зв'язування програми FB з параметрами привода необхідно використовувати контейнери ADL (див. [ADL CONTAINERS] [F A D -]).

Параметри M00x легко використовувати для зберігання даних. Ви можете зберігати значення для повторного використання за допомогою функціональних блоків і протоколу зв'язку (див. [FB PARAMETERS] [F B P -]).

Входи/виходи

Функція WriteBitParam використовує:

- **[IN]**: 1 логічний вхід WriteBit
- **[ENABLE]**: 1 логічний вхід

ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

Конфігурація

Ви можете налаштувати параметри у вікні властивостей. У вкладці Parameters ви можете вибрати контейнер ADL (LA0x), який буде використовуватися.

Parameters Comments

Adl container LA01

Bit position 0

Adl container	Logical Address
LA01	0
LA02	0
LA03	0
LA04	0
LA05	0
LA06	0
LA07	0
LA08	0

Для запису значення біта, в **Logical Address** необхідно вписати адресу регістру в десятковому форматі (перелік відповідних адрес наведений в [3]) та в **Bit position** – номер біта.



ПОПЕРЕДЖЕННЯ

ЗМІНИ ПАРАМЕТРІВ БЕЗ ЕФЕКТУ

Хоча параметри зберігаються в EEPROM, жодні інші модифікації параметрів не можуть бути збережені, оскільки доступ на запис до EEPROM недоступний під час цього процесу.

Переконайтеся, що доступ до запису до EEPROM можливий, перш ніж змінювати подальші параметри (біт 0 внутрішнього регістра стану ETI має дорівнювати 0).

Недотримання цих інструкцій може призвести до смерті, серйозних травм або пошкодження обладнання.

Функція WRITE Ana Param



Блок функцій WRITE Ana Param використовується для запису одного біта параметрів.

Для зв'язування програми FB з параметрами привода необхідно використовувати контейнери ADL (див. [ADL CONTAINERS] [F H A -]). Параметри M00x легко використовувати для зберігання даних. Ви можете зберігати значення для повторного використання за допомогою функціональних блоків і протоколу зв'язку (див. [FB PARAMETERS] [F B P -]).

Входи/виходи

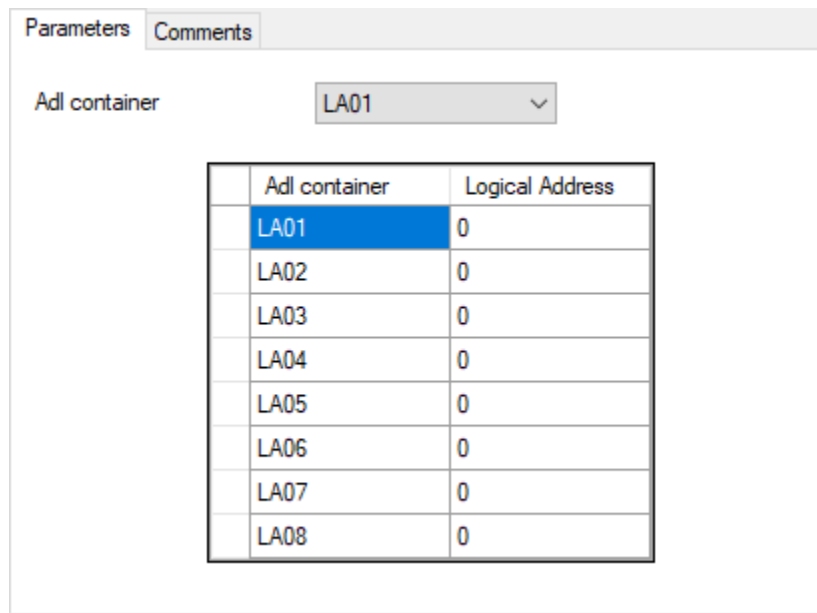
Функція WriteBitParam використовує:

- [IN]: 1 Аналоговий вхід
- [ENABLE]: 1 Логічний вхід

ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

Конфігурація

Ви можете налаштувати параметри у вікні властивостей. У вкладці Parameters ви можете вибрати контейнер ADL (LA0x), який буде використовуватися.



Для запису значення, в **Logical Address** необхідно вписати адресу регістру в десятковому форматі (перелік відповідних адрес наведений в [3]).

Функція WRITE Reg Param



Блок функцій WRITE Reg Param використовується для запису одного біта параметрів.

Для зв'язування програми FB з параметрами привода необхідно використовувати контейнери ADL (див. [ADL CONTAINERS] [F H D -]). Параметри M00x легко використовувати для зберігання даних. Ви можете зберігати значення для повторного використання за допомогою функціональних блоків і протоколу зв'язку (див. [FB PARAMETERS] [F B P -]).

Входи/виходи

Функція WriteBitParam використовує:

- **[IN]**: 1 Регістровий вхід
- **[ENABLE]**: 1 Логічний вхід

ПРИМІТКА: Вихід функціонального блоку не оновлюється, тому якщо вихід FB встановлено в 1, а вхід **ENABLE** перемикається на 0, вихід залишиться на рівні 1.

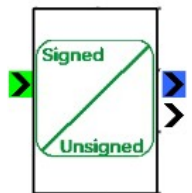
Конфігурація

Ви можете налаштувати параметри у вікні властивостей. У вкладці Parameters ви можете вибрати контейнер ADL (LA0x), який буде використовуватися.

Parameters		Comments
Adl container LA01		
Adl container	Logical Address	
LA01	0	
LA02	0	
LA03	0	
LA04	0	
LA05	0	
LA06	0	
LA07	0	
LA08	0	

Для запису значення, в **Logical Address** необхідно вписати адресу регістру в десятковому форматі (перелік відповідних адрес наведений в [3]).

Функція STU (16-bit analog TO 16-bit register)



Функціональний блок STU використовується для перетворення аналогового сигналу в регістровий шляхом обмеження.

Діапазон аналогового входу: від -32768 до $+32767$.

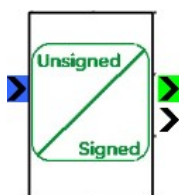
Діапазон виходу регістра: від 0 до 65535 .

Входи/виходи

- [IN]: 1 аналоговий вхід
- [Q]: 1 регістровий вихід
- [OVERFLOW]: 1 вихід переповнення

OVERFLOW є булевим виводом. Якщо значення аналогового входу **IN** негативне, **OVERFLOW** = 1. Якщо значення аналогового входу **IN** позитивне, **OVERFLOW** = 0.

Функція UTS (16-Bit Register Input to 16-Bit Analog Output)



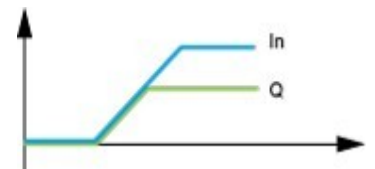
Функціональний блок UTS використовується для перетворення 16-розрядного регістрового входу в 16-розрядний аналоговий вихід (з обмеження).

Діапазон вхідного сигналу регістра: від 0 до 65535 .

Діапазон виходу регістра: від -32768 до $+32767$.

Входи/виходи

- [IN]: 1 регістровий вхід
- [Q]: 1 аналоговий вихід
- [OVERFLOW]: 1 вихід переповнення



Вихід **OVERFLOW** має логічний тип. Цей вихід активний, якщо вхідне значення регістра перевищує 32767 . Значення аналогового виходу буде встановлено на рівні 32767 .

5. Опис параметрів ATV320

Стан програми FB, керування режимом запуску/зупинки та зміну параметрів можна здійснювати в меню [1 Drive menu] [d r i -] > [1.3 CONFIGURATION] [C o n F] > [FULL] [F u L L] > [FUNCTION BLOCKS] [F b П -].

Структура меню [FUNCTION BLOCKS] [F b П -]

[FUNCTION BLOCKS] [F b П -]		
[Monit. Fun Dlocks] [П F б -]		
	[FB Status] [F b S t]	
	[FB Error] [F b F t]	
	[FB Identification] [F b i -]	додатковий ГТ
	[Program version] [b u E r]	додатковий ГТ
	[Program size] [b n S]	додатковий ГТ
	[Prg. format version] [b n u]	додатковий ГТ
	[Catalog version] [C t u]	додатковий ГТ
[FB Activation] [FbCd]	[Stop] (S t o P) [Start] (S t r t)	
[FB start mode] [FbrM]	[no] (n o) [Yes] (Y E S) [DI1]-[DI5] (L i 1)-(L i 5) [DA11], [DA12] (d A 1 1)(d A 1 2) [Cd1x] (C d 1x), [C11x] (C 1 1x), [C21x] (C 2 1x)	
[FB Motor Stop Type] [FbSM]	[Ignore] (n o) [Freewheel] (Y E S) [Ramp stop] (r П P) [Fast stop] (F S t) [DC injection] (d C i)	
[FB Drive Error Resp] [FbdF]	[Stop] (S t o P) [Ignore] (i G n)	
[INPUTS ASSIGNMENTS] [FbA-]		
	[IL01] - [IL10] [i L 0 1] - [i L 1 0] – Логічні входи	
	[IA01] - [IA10] [i A 0 1] - [i A 1 0] – Аналогові входи	
[ADL CONTAINERS] [FAd-]		
	[LA01] - [LA08] [L A 0 1] - [L A 0 8] – Контейнери ADL	
[FB Parameters] [FbP-]		
	[M001] - [M008] [П 0 0 1] - [П 0 0 8] – параметри FB	

Значення контейнерів ADL [LA01] - [LA08] на вбудованому графічному терміналі відображаються як код параметра/регістра, а при спробі змінити значення змінюється на адресу регістра у шістнадцятковому форматі; на додатковому графічному терміналі при перегляді меню [ADL CONTAINERS] [F A d -] відображається код параметра/регістра, а при налаштуванні окремого контейнера – адреса регістру у десятковому форматі. Наприклад

Вбудований ГТ		Додатковий ГТ	
Відображення	Налаштування	Відображення	Налаштування
S 0 0 1	Э А В Ч	S001	14980
П 0 0 1	Э А 7 А	M001	14970
o t r	О С В 5	OTR	3205

5.1. Керування програмою FB

З'являється в меню НМІ для керування функціональним блоком для автоматичного перемикання з режиму «RUN» в режим «STOP».



ПОПЕРЕДЖЕННЯ

ВТРАТА КЕРУВАННЯ

Якщо параметри [FB start mode] (*F b r Π*) або [FB Command] (*F b C d*) змінюються мережею зв'язку, стан зв'язку необхідно контролювати за допомогою бітів 11, 12 і 13 [FB SYSTEM WORD 06] (*5 Π Π ρ*).

Недотримання цих інструкцій може призвести до смерті, серйозних травм або пошкодження обладнання.

	[FB Command] (<i>F b C d</i>)	[FB start mode] (<i>F b r Π</i>)	FB Behaviour	New [FB stop mode] (<i>F b S t</i>)
On PowerOn	STOP	No	STOP	STOP
	STOP	Yes	AutoRun	RUN
	STOP	Llx	STOPRUN на передньому фронті LI	STOPRUN на передньому фронті LI
On change of <i>F b r u</i>	STOP	XX → No	STOP	STOP
	STOP	XX → Yes	Не впливає, враховується лише при наступному ввімкненні живлення	STOP
	STOP	XX → Llx	STOPRUN на передньому фронті LI	STOPRUN на передньому фронті LI
On Start with <i>F b C d</i>	STOP → START	No	RUN	START
	STOP → START	Yes	RUN	START
	STOP → START	Llx	STOPRUN на передньому фронті LI	STOPRUN на передньому фронті LI
On Stop with <i>F b C d</i>	START → STOP	XX	STOP	STOP
End of download/ Factory setting	STOP	No	STOP	STOP
	STOP	Yes	STOP	STOP
	STOP	Llx	STOPRUN на передньому фронті LI	STOPRUN на передньому фронті LI
On LI fall edge	START	Llx	STOP	STOP

Виконання програми FB має той самий ефект, що і запуск двигуна, тобто ПЧ переходить в стан RUN, при цьому налаштування параметрів блокується.

5.2. Моніторинг стану програми FB

Моніторинг стану програми FB можна здійснити в

- меню [1 Drive menu] [drive] > [1.3 CONFIGURATION] [CONF] > [FULL] [FULL] > [FUNCTION BLOCKS] [FB] > [MONIT. FUN. BLOCKS] [PF] з додаткового графічного терміналу;
- вкладці Parameters програми SoMove

Function Blocks			
FBCD	FunctionBlock activation	Stop	Stop
FBDP	FunctionBlock response to drive error	Stop	Stop
FBRM	FunctionBlock start mode	Digital input 5	Not assigned
FBSM	Motor stop type on FunctionBlock stop	Freewheel stop	Freewheel stop
Monit. Fun. Blocks			
FBFT	FunctionBlock error	No	No
FBST	FunctionBlock status	Stop	Not active
FB Identification			

- панелі інструментів вкладки ATV Logic програми SoMove

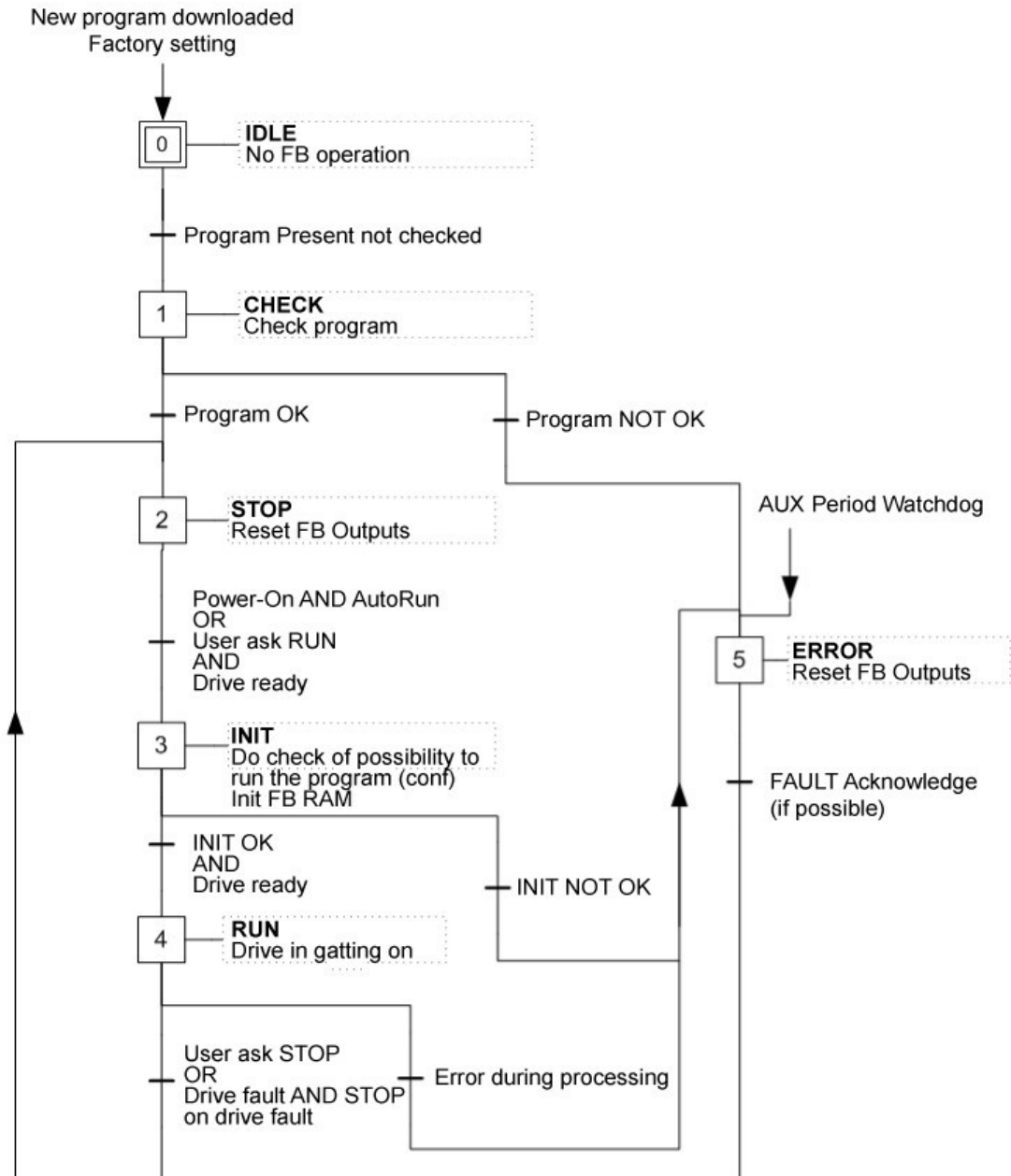
Monitoring Parameters			
Code	Label	Value	Comment
M001		5	
M002		0	
M003		0	
M004		0	
M005		0	
M006		0	
M007		0	
M008		0	
FBST	FunctionBlock status	2	

Статус програми FB

Статус вказує на стан програми FB в людино-машинному інтерфейсі.

Можливі стани параметра [FB Status] (FBSt)

Код вбудованого ГТ	Значення в панелі інструментів	Назва/Опис
Idle	0	[Idle] Немає двійкового файлу в цільовому файлі, FB очікує на завантаження
Check	1	[Check prog.] Перевірте, чи завантажено програму
Stop	2	[Stop] Виконання програми FB зупинено
Init	3	[Init] Ініціалізація програми FB у функції конфігурації, виконує також скидання оперативної пам'яті FB
Run	4	[Run] Програма FB запущена
Error	5	[Error] Виникла внутрішня помилка. FB працює в режимі за замовчуванням



Помилки програми FB

Якщо привод знаходиться в режимі помилки *F b E*, див. [FB fault] (*F b F t*).

Код	Назва/Опис
<i>F b F t</i>	[FB fault]
<i>n o</i>	[No]: несправність не виявлена
<i>i n t</i>	[Internal]: виявлено внутрішню несправність
<i>b i n</i>	[Binary file]: виявлено двійкову помилку
<i>i n P</i>	[Intern para.]: виявлено несправність внутрішнього параметра
<i>P a r</i>	[Para. RW]: виявлено помилку доступу до параметрів
<i>C a l</i>	[Calculation]: виявлено помилку в розрахунках
<i>t o A u x</i>	[TO AUX]: Тайм-аут у завданні AUX
<i>t o P P</i>	[TO synch]: Тайм-аут у завданні PRE/POST
<i>B a d A D L C</i>	[Bad ADLC]: Контейнер ADL з хибним параметром
<i>i n</i>	[Input assign.]: Вхід не налаштовано

Помилки не зберігаються, увімкнення і вимкнення скидає несправність FB.

Під час роботи, а точніше, при спробі зупинки програми FBD коли двигун працює, може виникати помилка $F b E 5$ («Функціональні блоки були зупинені під час роботи двигуна»). Така ситуація може виникати коли пуск/зупинка програми FB здійснюється сигналом на дискретному вході ([FB start mode] ($F b r 1$) = [DI1] ($L 1$)...), запуск двигуна задається цією програмою ($F r d$ чи $r r 5 = 1/0$), а уставка [FB Stop motor] ($F b 5 r$) \neq [Ignore] ($r o$).

Одним із рішень¹ є призначення параметру [Stop FB Stop motor] ($F b 5 r$) = [Ignore] ($r o$), якщо це дозволяє технологічний механізм.

5.3. Комунікаційні параметри програми FB

За допомогою наступних параметрів програма FB «спілкується» із ПЧ, тобто зчитує данні з нього і записує в нього ніби по комунікаційній шині, не маючи безпосередньо доступу до ресурсів перетворювача.

Параметри FB

Програма FB має 8 комірок пам'яті (M001 - M008), які можуть зберігати значення змінних через параметри FB ([FB PARAMETERS] ($F b P -$) [5]). Ці комірки можна використовувати для налаштування, роботи або моніторингу програми, оскільки M00x доступні для читання і запису. Звертання до параметрів здійснюється через їх адреси 14790 - 14797, відповідно.

ПРИМІТКА: M001 - M004 зберігаються в EEPROM (енергонезалежна пам'ять), а M005 - M008 записуються в оперативну пам'ять.

Контейнери ADL

Контейнери ADL пов'язані з 6 функціональними блоками: Read Ana Param, Read Reg Param, Write Ana Param, Write Reg Param, ReadBitParam, WriteBitParam.

Привод має 8 контейнерів ADL (від LA01 до LA08). Ці контейнери LA0x використовуються для підключення %MW, %SW або інших параметрів ПЧ ([ADL CONTAINERS] [$F A d -$] [5]). Тобто під час редагування програми FB LA0x присвоюють адреси цільових параметрів.

Залежно від того, в якому завданні застосовано блок Read/Write Ana/Reg Param, це дозволить підключитися до відповідної Групи параметрів Fast або Slow.

Правило таке:

- У завданнях PRE/POST: Підключення лише до параметрів швидкого доступу
- У завданні AUX: Підключення до параметрів повільного та швидкого доступу завдання PRE та POST мають пріоритет над завданням AUX.

Програмні засоби, і компілятор (п.3.4) зокрема, не перевіряє чи були правильно використані параметри груп Fast або Slow у відповідному завданні, проте застосування параметрів із групи Slow у завданнях PRE чи POST призведе до помилки під час спроби запуску програми в ПЧ.

¹ [<https://www.se.com/in/en/faqs/FA354143/>]

Code	Long Label	Conf0	Default Value	Min Value	Max Value	Logical address
ADL Containers						
LA01		3205	0	0	65535	14940
LA02		14970	0	0	65535	14941
LA03		0	0	0	65535	14942
LA04		0	0	0	65535	14943
LA05		0	0	0	65535	14944
LA06		0	0	0	65535	14945
LA07		0	0	0	65535	14946
LA08		0	0	0	65535	14947
FB Parameters						
M001		10	0	0	65535	14970
M002		0	0	0	65535	14971
M003		0	0	0	65535	14972
M004		0	0	0	65535	14973
M005		0	0	0	65535	14974
M006		0	0	0	65535	14975

5.4. Системні слова FB

Системне слово FB 01

[FB SYSTEM WORD 01] (5 0 0 1)							
Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
ТВ1MIN: Годинник	ТВ1CEK: Годинник	ТВ100MS: Годинник	Захищені	Захищені	Захищені	Захищені	Захищені
біт: період 1 хв	біт: період 1 с	біт: період 100 мс,					
Робочий цикл50%	Робочий цикл50%	Робочий цикл 50%					
Біт 15	Біт 14	Біт 13	Біт 12	Біт 11	Біт 10	Біт 9	Біт 8
Захищені	Захищені	1RSTSCANRUN: FB виконує свій 1-й цикл з моменту останнього RUN. Використовується тільки в PRE і POST	Захищені	Захищені	Захищені	Захищені	Захищені

Внутрішній таймер керує зміною статусу бітів 5, 6 і 7.

ПРИМІТКА: Біти 5, 6 і 7 повинні використовуватися в завданнях PRE і POST. Як і в завданні AUX, повторюваність не гарантується.

Перехід з режиму STOP у режим RUN (навіть після холодного запуску) відкладається встановленням системного біта %S13. Цей біт скидається в 0 наприкінці першого циклу виконання завдання POST у режимі RUN.

Системне слово FB 02

[FB SYSTEM WORD 02] (5 0 0 2)							
Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
Захищені	Захищені	Захищені	Захищені	Захищені	Захищені	Захищені	Захищені
Біт 15	Біт 14	Біт 13	Біт 12	Біт 11	Біт 10	Біт 9	Біт 8
Захищені	Захищені	Втрата зв'язку Modbus	Втрата зв'язку CANopen	Втрата зв'язку з опціональною картою	Захищені	Захищені	Захищені

Інформація про біти 13, 12 та 11

Виявлення базується на втраті зв'язку, а не на відсутності зв'язку. Тому для виявлення має бути початковий зв'язок.

Системне слово FB 06

[FB SYSTEM WORD 06] (5 0 0 6)							
Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
Захищені	Захищені	Захищені	Захищені	Захищені	Захищені	Захищені	Захищені
Біт 15	Біт 14	Біт 13	Біт 12	Біт 11	Біт 10	Біт 9	Біт 8
Захищені	USRMWSAVE: Запит на збереження змінної користувача	Захищені	Захищені	Захищені	Захищені	Захищені	Захищені

Інформація про біт 14

Біт %S006.14 — це біт %S94 на моделях Modicon M340 та Premium.



ПОПЕРЕДЖЕННЯ

ЗМІНИ ПАРАМЕТРІВ БЕЗ ЕФЕКТУ

Встановлення біта 14 параметра [FB SYSTEM WORD 06] в 1 запуску збереження параметрів у EEPROM. Хоча параметри зберігаються в EEPROM, жодні інші модифікації параметрів не можуть бути збережені, оскільки доступ на запис до EEPROM недоступний під час цього процесу.

Переконайтеся, що доступ до запису до EEPROM можливий, перш ніж змінювати подальші параметри (біт 0 внутрішнього регістра стану ETI має дорівнювати 0).

Недотримання цих інструкцій може призвести до смерті, серйозних травм або пошкодження обладнання.

6. Приклади застосування

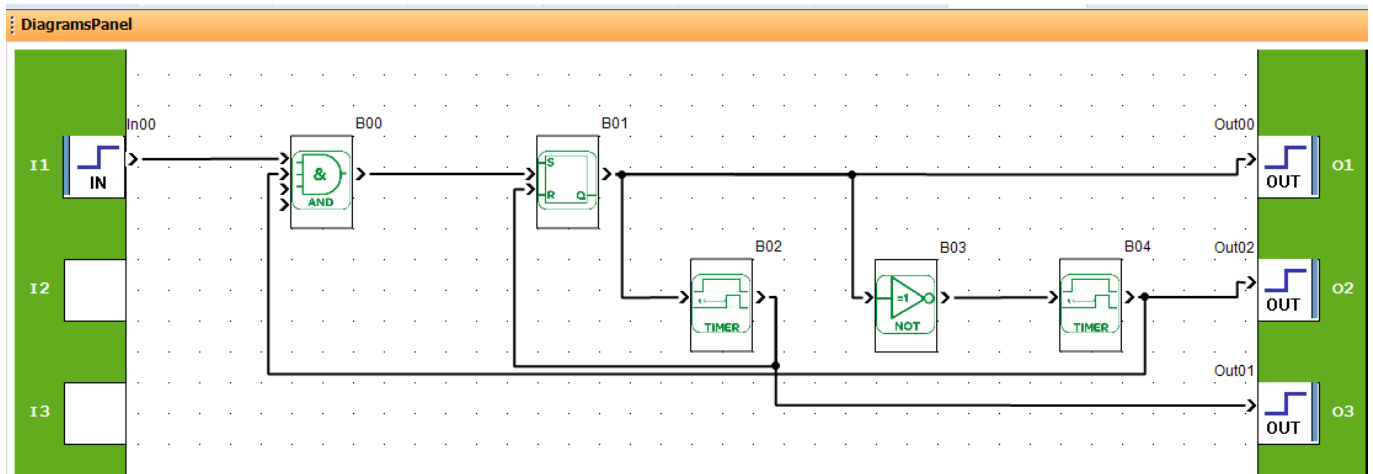
Використання тригера

Приклад програми ATV Logic: використання тригера
[\[https://www.se.com/ua/uk/faqs/FA348565/\]](https://www.se.com/ua/uk/faqs/FA348565/)

Релейні виходи використані для індикації станів таймерів.

Фізичні входи	Віртуальні входи	Призначення
DI1	IL01	Перехід в режим RUN
Фізичні виходи	Віртуальні виходи	Призначення
FRD	OL01	Пуск вперед
R1	OL02	Індикація ввімкнення таймера B04
R2	OL02	Індикація ввімкнення таймера B02

Застосовані таймери типу А/С для фіксації стану релейних виходів.



Програма реалізована в завданні «AUX».

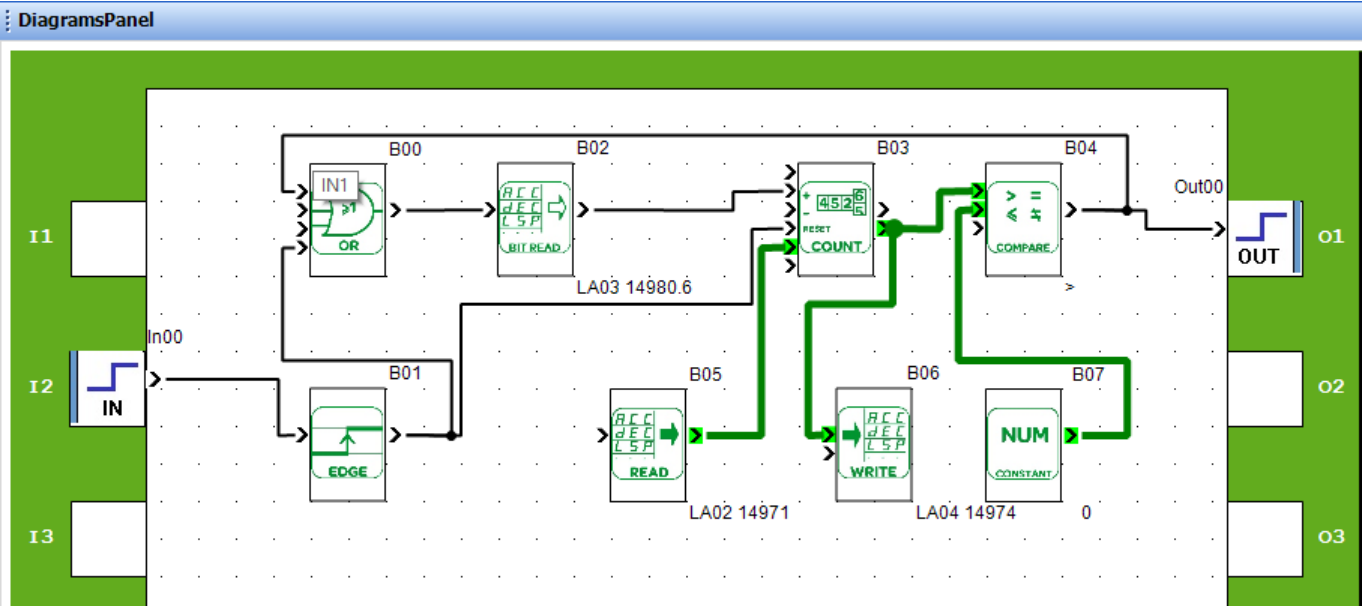
Тимчасовий старт/стоп

Приклад з How to use temporized start/stop with ATV Logic on ATV320 ? | Schneider Electric
[\[https://www.youtube.com/watch?v=leD6keJd5PM\]](https://www.youtube.com/watch?v=leD6keJd5PM)

Програма реалізована в завданні «PRE»

Фізичні входи	Віртуальні входи	Призначення
DI1	IL01	Перехід в режим RUN
Фізичні виходи	Віртуальні виходи	Призначення
FRD	OL01	Пуск вперед

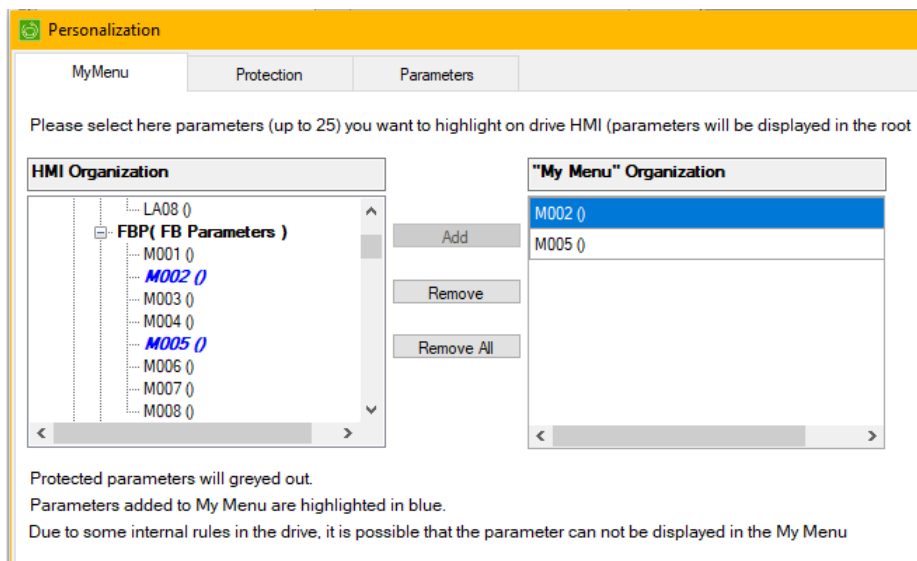
В прикладі використовується лічильник з відліком «вниз», в якості імпульсів для зменшення використані секунди (14980.6) внутрішнього таймера [FB SYSTEM WORD 01].

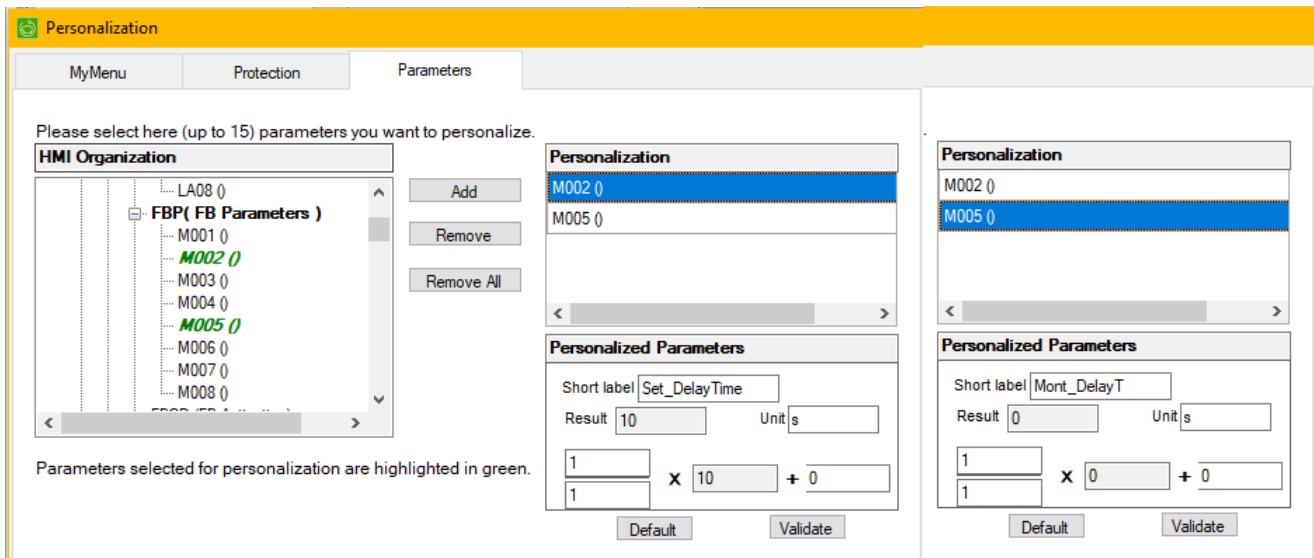


Базове значення для відліку встановлюється в M002 (через LA02), а поточне значення записується в M005 (через LA05). Оскільки M002 знаходиться в енергонезалежній пам'яті, то значення зберігається і після вимкнення ПЧ, поточне значення відліку скидається.

ADL Containers					
LA01		0	0	65535	14940
LA02		14971	0	65535	14941
LA03		14980	0	65535	14942
LA04		0	0	65535	14943
LA05		14974	0	65535	14944
LA06		0	0	65535	14945
LA07		0	0	65535	14946
LA08		0	0	65535	14947
FB Parameters					
M001		0	0	65535	14970
M002		10	0	65535	14971
M003		0	0	65535	14972
M004		0	0	65535	14973
M005		0	0	65535	14974
M006		0	0	65535	14975

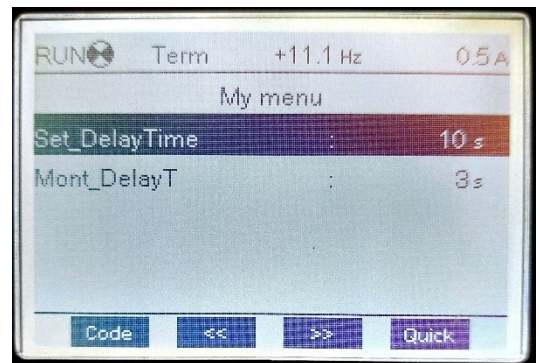
Запуск програми ФБ здійснюється сигналом на DI5. Запуск ПЧ сигналом на DI1. Після налаштування власного меню задане й поточне значення лічильника можна спостерігати на виносному графічному терміналі, а задане ще й змінювати





На екрані додаткового графічного терміналу VW3A1111, при його підключенні до ПЧ, відображаються задані змінні стану Set_DelayTime та Mont_DelayT.

Файл конфігурації (.psx) реалізований для ATV320U04N4C та програма ФБ (.atvlogic) можна завантажити



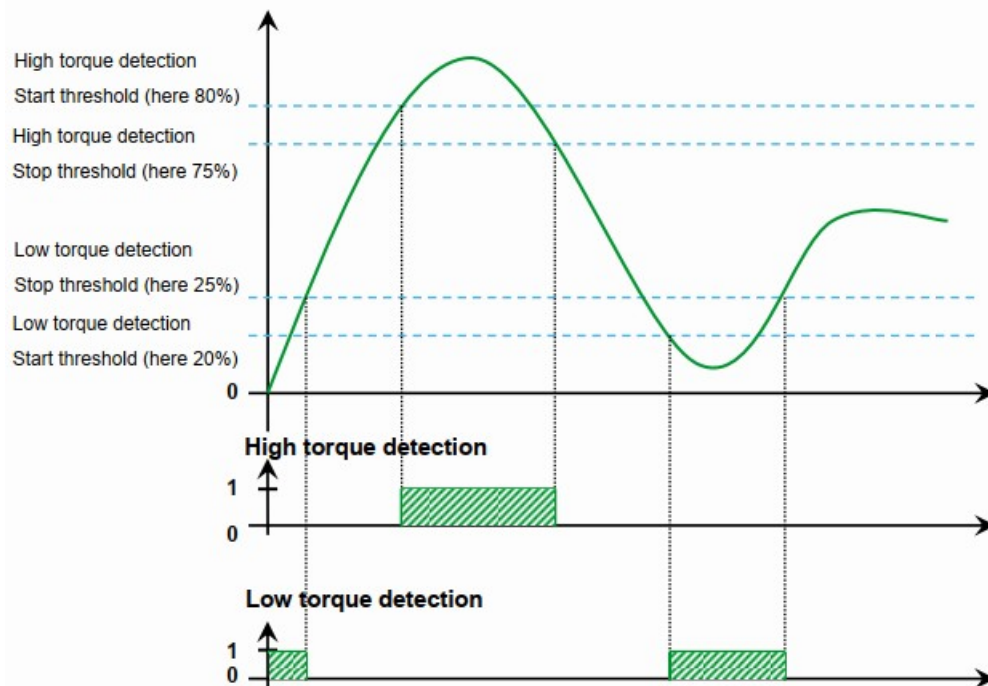
<https://drive.google.com/drive/folders/1DGMT0Od5B2E74xXXAdqWewrY-DCBGayD?usp=sharing>

Індикація обмеження моменту двигуна

Програма визначає високий та низький поріг моменту двигуна.

Досягнення високого порогу активує реле R1, а досягнення низького порогу активує реле R2. Коли значення поза зазначеними межами — реле деактивовані.

Принцип роботи показаний на діаграмі



Момент двигуна відображається в параметрі [Motor torque] ($\square \text{ t r}$) = 3205, оцінюється у відсотках і в файлі параметрів зазначено діапазон -3276.7 % ... 3276.7 % з одиницею виміру 0.1 %. Але якому саме значенню відсотків відповідає максимальне число, адже можливе перевантаження? Згідно інструкції з програмування [5] максимальне значення моменту згадується в параметрі [Motoring torque lim] (t L , П) 0...300% з діапазоном значень в регістрі 0...3000 [3]. Можна сподіватися, що максимальне значення $\square \text{ t r}$ також 3000 (а зовсім не 3276,7) і відповідає воно 300%, **але це не точно** 😞

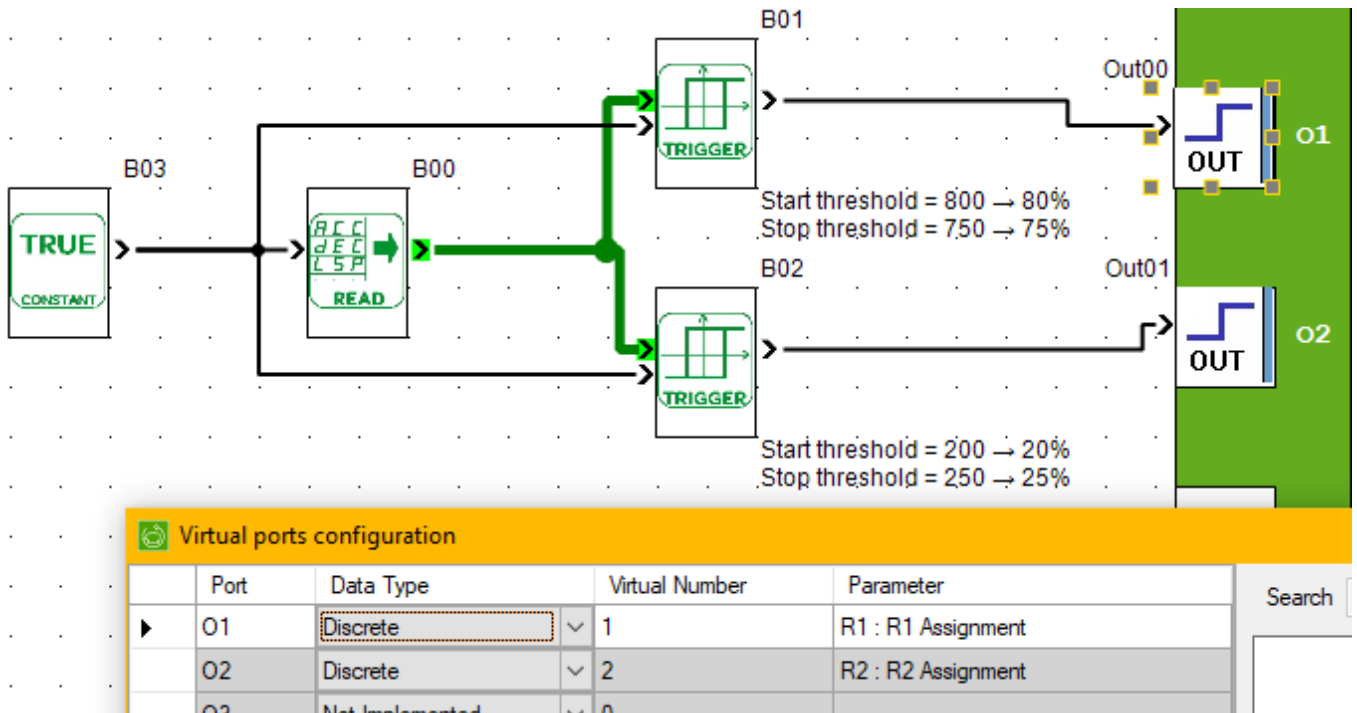
Виходячи з такого припущення для визначення заданих уставок порогів спрацювання

Для високого:

- Start threshold = 800 → 80%
- Stop threshold = 750 → 75%

Для низького:

- Start threshold = 200 → 20%
- Stop threshold = 250 → 25%

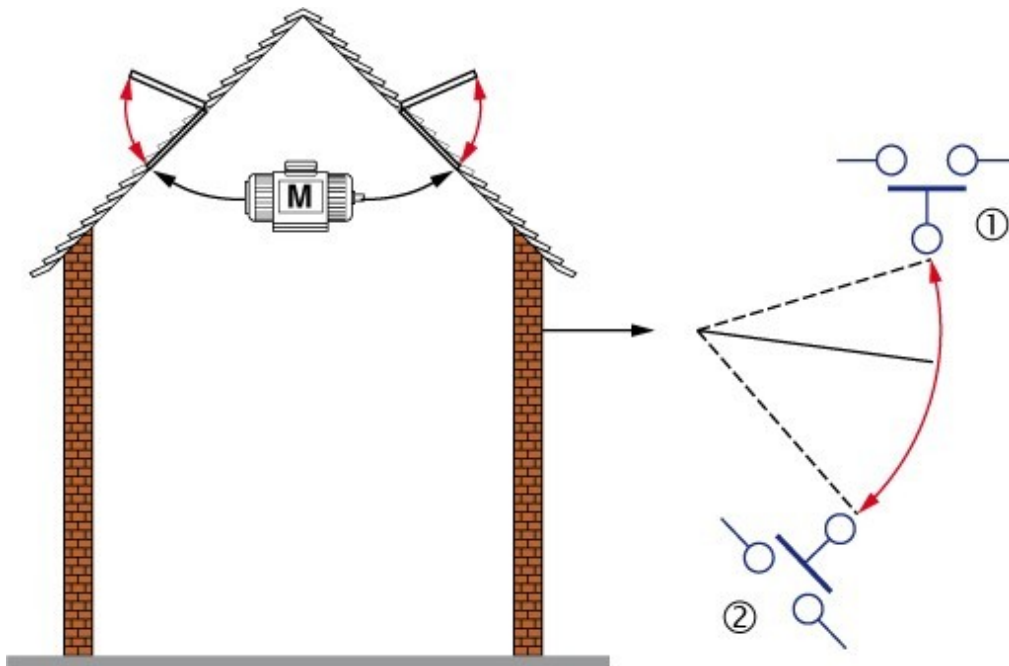


Регулювання температури в теплиці

У цьому прикладі описано, як можна автоматично керувати скляними шибками в теплиці.

Власник теплиці бажає керувати відкриттям і закриттям вентиляційних шибок, розташованих на даху теплиці. Теплиця має два шибки для забезпечення вентиляції. Їх відкриття здійснюється двигуном, що живиться від ПЧ, а положення шибок контролюється двома кінцевими датчиками, які вказують: відкриті або закриті.

Вдень, коли температура досягне 25°C, вікна відкриваються для провітрювання. Якщо температура опускається нижче 20°C, вікна повинні знову закритися.



Таблиці введення-виведення

Відповідно до наданої схеми маємо призначення входів/виходів

Фізичні входи	Віртуальні входи	Призначення
DI3	IL01	Кінцевий датчик 1: Вікно відкрите
DI4	IL02	Кінцевий датчик 2: Вікно закрите
AI3	IL03	Струмний сигнал 4-20мА датчика температури
Фізичні виходи	Віртуальні виходи	Призначення
FRD	OL01	Вперед: Відкривання вікна
RRS	OL02	Назад: Закривання вікна

Визначення параметрів

Застосуємо для контролю температури датчик TM1SHTCC4 з діапазоном вимірювання $-40...+60^{\circ}\text{C}$ та струмним виходом 4...20 мА.

Зважаючи на технічні характеристики датчика температури та необхідні значення температури, потрібно з'ясувати відповідні значення констант.

На практиці, визначення конкретних значень для регулювання є суто аналітичною задачею, проте у випадку Schneider electric — це квест!



ПОПЕРЕДЖЕННЯ

ОТРИМАННЯ НЕОЧІКУВАНИХ ДАНИХ

З якихось незрозумілих причин (пояснень цьому ніхто дати не може, лише констатація наявного факту) всі регістри, що відповідають аналоговим входам (AI1R, AI2R, AI3R), видають значення розмірністю **13 біт**, тобто коди в діапазоні **0...8192**, хоча заявлено INT (Signed16) $-32767...32767$.

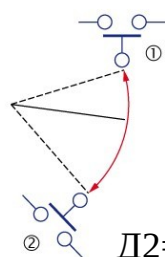
<https://www.se.com/myschneider/faq/FA324292/uk>

Попередня перевірка отриманих даних обов'язкова!

Для діапазону вихідного сигналу датчика 4...20 мА, й означеного діапазону кодів регістру аналогового входу 0...8192

t, °C	-40	+20	+25	60
I, mA	4	13.6	14.4	20
R, code	0	4915	5324	8192

Умови переходу



Д1=1, $t \leq 20^{\circ}\text{C} \Rightarrow$ Рух назад

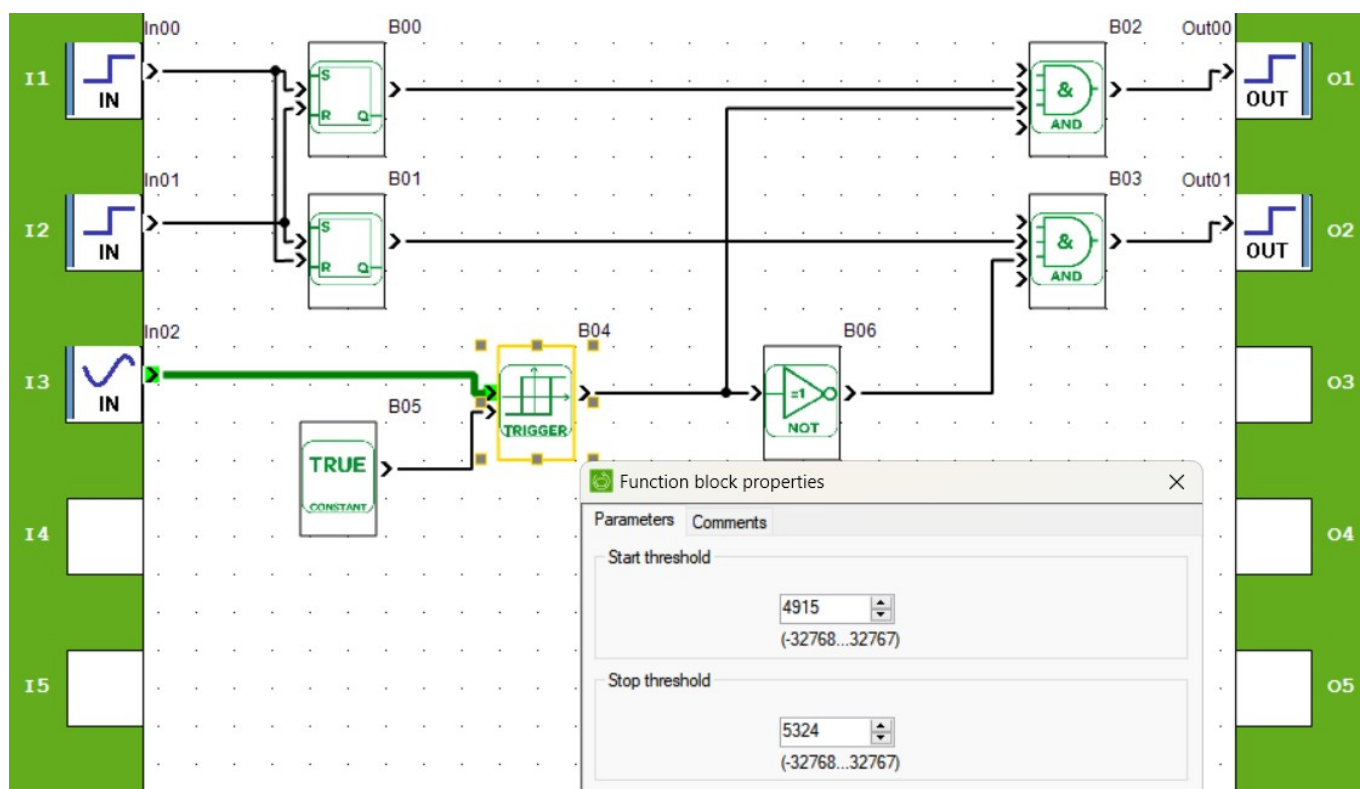
Д2=1, $t \geq 25^{\circ}\text{C} \Rightarrow$ Рух вперед

При розв'язанні завдання необхідно враховувати, що коли вікно знаходиться в одному з крайніх положень (спрацював відповідний датчик) подальший рух в цьому напрямку неможливий.

Програми FB

Умовно програма має дві частини:


1. Визначення напрямку обертання
2. Визначення меж температури для запуску у відповідному напрямку

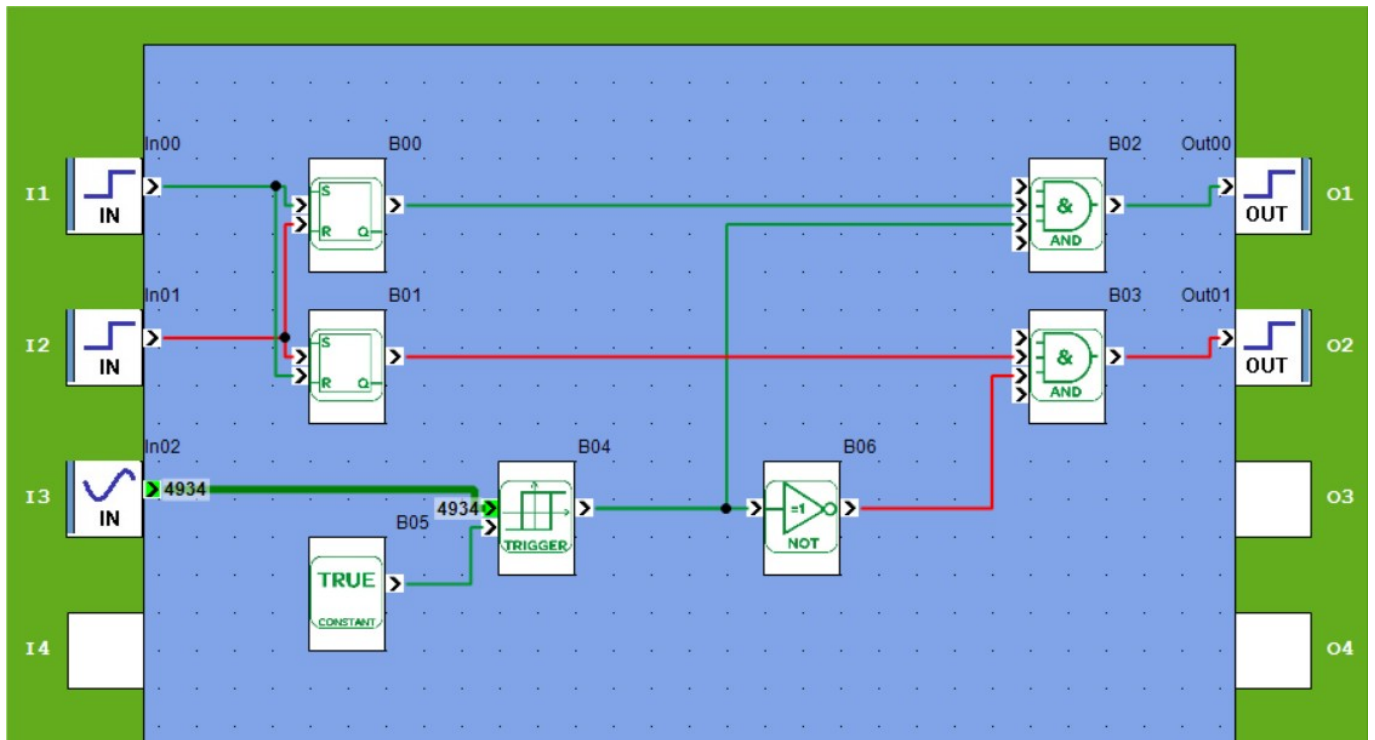


Virtual ports configuration			
Port	Data Type	Virtual Number	Parameter
I1	Discrete	1	Digital input 3
I2	Discrete	2	Digital input 4
I3	Analogic	1	A13 Analog input
I4	Not Implemented	0	

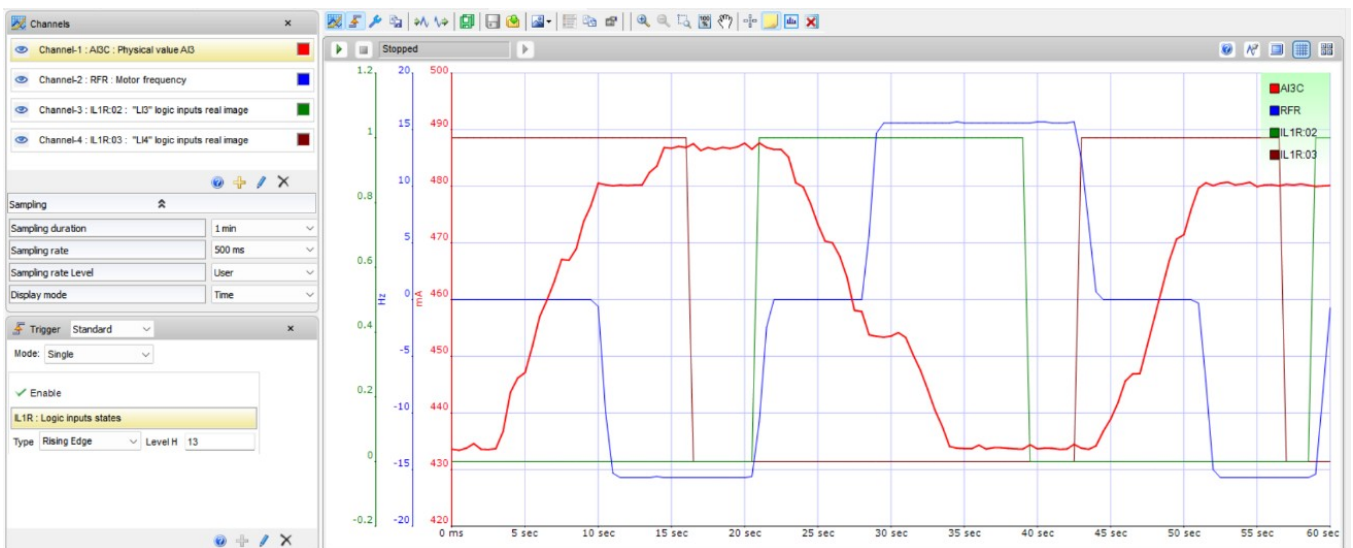
Virtual ports configuration			
Port	Data Type	Virtual Number	Parameter
O1	Discrete	1	RRS : Reverse Assign
O2	Discrete	2	FRD : Forward
O3	Not Implemented	0	

RS-тригери забезпечать наявність сигналу напрямку руху доки не буде досягнуто кінцевого вимикача. Вихід RS-тригера буде скинуто в 0, коли спрацює протилежний кінцевий вимикач.

При підключенні до ПЧ й запусненій програмі FB можна спостерігати стан булевих сигналів (тонка зелена лінія 1, тонка червона лінія 0) та аналогові значення (товста зелена лінія й значення на виході й вході блоків з аналоговими сигналами), які зчитуються зі входу. Для актуальності даних треба натискати кнопку Update  на панелі AtvlogicToolBar, якщо не вибране автоматичне оновлення даних Auto Refresh (п.3.6).



На рисунку наведені осцилограми змінних привода при роботі програми, зняті внутрішнім осцилографом Score програми SoMove в режимі Slow.



Червона лінія мусила б відображати значення аналогового входу АІЗС в діапазоні 4...20 мА (значення змінювались в діапазоні 13...15 мА), проте спостерігаємо на шкалі криву в діапазоні значень 420...500, що не відповідає нічому.

Індикація стану «Сон»

Перетворювач частоти використаний для підтримання заданого тиску в водонапірній магістралі поливу з використанням ПД-регулятора ПЧ. Для індикації стану роботи ПЧ на лицьовій панелі шафи розташовані лампочки: HL1 - «Робота», HL2 - «Сон».

В стані «Сон» ПЧ одночасно знаходиться і в стані «Робота», оскільки він відпрацьовує задану швидкість. Таким чином, треба окремо показувати ці стани.

Стан «Робота» призначається на релейний вихід $R1 = \text{r u n}$, через який і заживлено лампочку HL1. Відповідно, на релейний вихід R2 варто призначити індикацію стану «Сон», проте напряму в параметрах призначення виходу такого

варіанту нема, отже можемо вирішити це питання за допомогою ATVLogic, оскільки в слові статусу LRS3 є біт, який показує активний стан «Сон»

Code	Name	Logic address
LRS3	Logic result status 3 (PSL 032-047)	16#0CB4 = 3252

LRS3	_(032_RFC) Bit0 = 0 : Reference channel 1, 1 : Reference channel 2
	_(033_CCS) Bit1 = 0 : Command channel 1, 1 : Command channel 2
	_(034_RP2) Bit2 = 1 : Ramp switching active
	_(035_CL2) Bit3 = 1 : Current limitation switching active
	_(036) Bit4 : Reserved
	_(037_TS2) Bit5 = 1 : Motor 2 thermal threshold (TTD2) reached
	_(038_TS3) Bit6 = 1 : Motor 3 thermal threshold (TTD3) reached
	_(039) Bit7 : Reserved
	_(040_EPR) Bit8 = 1 : External Power Present
	_(041_SLM) Bit9 = 1 : Sleeping Mode active
	_(042) Bit10 : Reserved
	_(043) Bit11 : Reserved
	_(044) Bit12 : Reserved
	_(045) Bit13 : Reserved
	_(046) Bit14 : Reserved
_(047_ATS) Bit15 = 1 : Actual torque sign	

Отже, скориставшись блоком READ BIT PARAM можемо призначити релейному виходу стан «Сон».

Port	Data Type	Virtual Number	Parameter
O1	Discrete	1	R2 : R2 Assignment
O2	Not Implemented	0	

ДЖЕРЕЛА

1. Altivar Machine 320. Variable Speed Drives for Synchronous and Asynchronous Motors: ATV Logic Manual NVE71954.01. Schneider Electric. 04/2016. – 66 p.
2. Altivar 32. Variable speed drives for synchronous and asynchronous motors: ATV Logic Manual S1B90747.01. Schneider Electric. 10/2012. – 64 p.
3. ATV320 - Communication Parameters NVE41316. V3.5. 03/01/2022.
4. Altivar 32 Function Blocks
5. Altivar Machine ATV320. Variable Speed Drives for Asynchronous and Synchronous Motors: Programming Manual NVE41295. 09/2021. Schneider Electric. 2021. – 358 p.