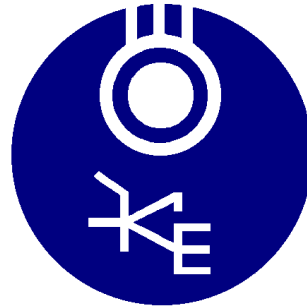


Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»



КАФЕДРА ЕЛЕКТРОПРИВОДА

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторної роботи Z-5
**«Програмування у оболонці ZelioSoft 2 мовою діаграм функціональних
блоків: інтерфейс, методи роботи»**
з дисципліни
«Засоби автоматизації»

Упорядник: к. т. н., доц. Яланський О.А.
(ДВНЗ «Національний гірничий університет», кафедра електропривода;
Авторизований навчальний центр «Schneider Electric»)

Дніпропетровськ
2007 – 2012

Лабораторна робота № 5

Програмування у оболонці ZelioSoft 2 мовою діаграм функціональних блоків: інтерфейс, методи роботи

1. Мета роботи

Ознайомлення з інтерфейсом програмної оболонки ZelioSoft 2 у режимі програмування мовою діаграм функціональних блоків (у подальшому – FBD, Functional Block Diagram), отримання навичок введення та редагування нескладних програм. Ознайомлення з переліком функціональних блоків, їх можливостями та функціонуванням. Отримання навичок програмування мовою FBD.

2. Програма роботи

Виконання лабораторної роботи передбачає попередню самостійну підготовку та роботу у обчислювальному класі чи лабораторії навчального закладу. Робота полягає у виконанні наступних пунктів:

1. Ознайомитися з теоретичними матеріалами, що наведені у методичних вказівках.
2. Апробувати основні функції і методи роботи з програмною оболонкою ZelioSoft 2 у режимі програмування мовою FBD на комп'ютері обчислювального центру або навчальної лабораторії. Для цього використати файли прикладів програм.
3. Ознайомитись з переліком функціональних блоків, їх можливостями та функціонуванням. Для цього використати файли прикладів програм.
4. Відповісти на запитання для самоконтролю.
5. Скласти програму користувача згідно з індивідуальним завданням. Промоделювати її роботу.
6. З дозволу викладача завантажити набрану програму до модуля інтелектуального реле та продемонструвати роботу програми.
7. Скласти та захистити підсумковий звіт.

3. Вказівки щодо складання звіту

Підсумковий звіт повинен містити:

1. Назву, мету та програму роботи.
2. Схему та лістинг програми користувача згідно з індивідуальним завданням.
3. Аналіз ефективності програмування мовою FBD, співставлення можливостей та ефективності мови діаграм функціональних блоків (FBD) та сходових діаграм (LD).

4. Теоретичні відомості

Програмна оболонка ZelioSoft 2 призначена для програмування модулів інтелектуальних реле Zelio другої та третьої серій (SR2, SR3), завантаження програми користувача до модуля або зчитування, встановлення параметрів функціональних блоків модуля, роботи з модулем у режимі моніторингу.

Лабораторна робота присвячена вивченню мови програмування «FBD, Functional Block Diagram» (діаграм функціональних блоків).

4.1. Інтерфейс оболонки ZelioSoft 2 у режимі програмування мовою FBD

Програмування мовою FBD полягає у використанні наперед визначених функціональних блоків. Передбачається візуальне програмування з використанням графічного редактора та графічних примітивів.

Якщо у вікні вибору мови програмування (рис. 1.4) натиснути піктограму FBD (Functional Block Diagram) і обрати мову програмування діаграм функціональних блоків, головне вікно програми буде мати наступний вигляд:

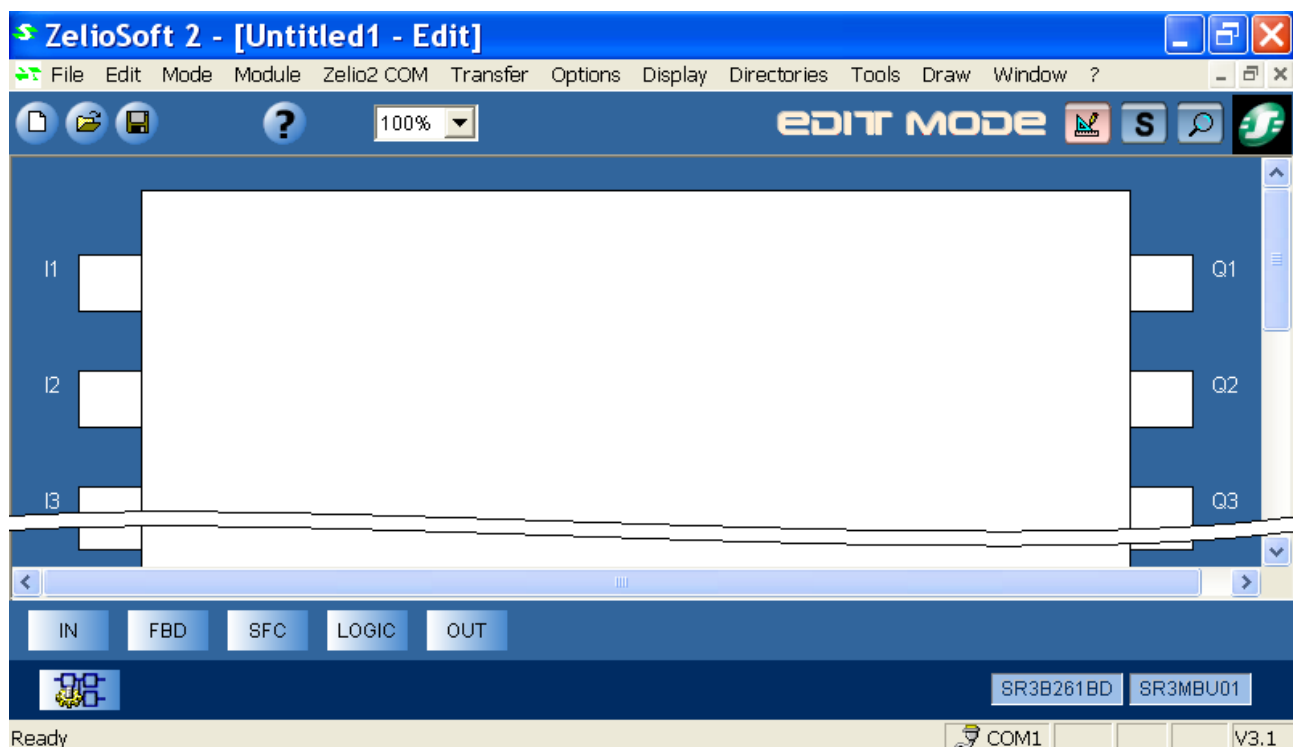


Рис. 5.1. Головне вікно програмної оболонки Zelio Soft 2 у режимі редагування програми мовою FBD

Графічний інтерфейс програмної оболонки містить такі елементи:

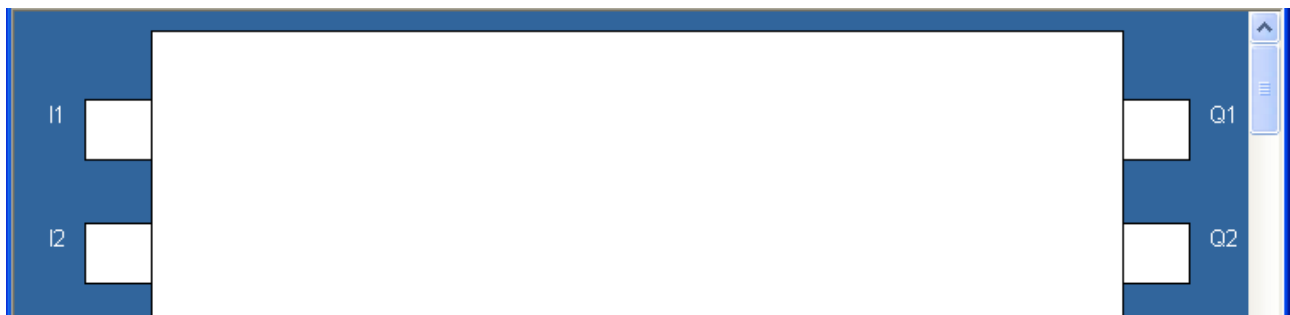
- Рядок меню:



- Панель інструментів:



- Поле для набору програми:



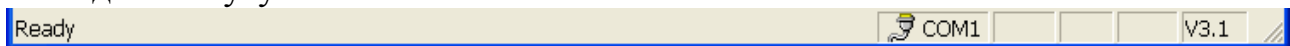
- Панель функцій (функціональних блоків):



- Поле індикаторів:



- Рядок статусу:



Рядок меню програмної оболонки Zelio Soft 2 у режимі програмування мовою FBD має майже такий самий перелік команд, що і у режимі програмування мовою сходових діаграм (LD). Відмінність складають команди, які призначені для набору та редагування програми:

- Display – група команд управління графічним інтерфейсом програмної оболонки ZelioSoft 2:
 - Block number – керування відображенням порядкових номерів блоків (All – показати порядкові номери усіх блоків програми, None – сховати порядкові номери усіх блоків);
 - Grid – керування графічною сіткою, що призначена для вирівнювання графічних об’єктів за вертикаллю та горизонталлю на полі набору програми (Display the grid – показати/сховати сітку, Spacing – крок сітки: 12 px, 24 px, 36 px, 48 px);
 - Zoom – масштаб відображення програми користувача на екрані (25 %, 50 %, ..., 150 %);
- Tools – інструменти:
 - Align – вирівняти виділені графічні об’єкти (Align left – за лівим краєм, Align right – за правим краєм, Align top – за верхньою межею, Align bottom – за нижньою межею, Center object vertically – центрувати уздовж вертикалі, Center object horizontally – центрувати уздовж горизонталі);
 - Distribute – розподілити рівномірно виділені графічні об’єкти (Distribute horizontally – розподілити уздовж горизонталі, Distribute vertically – розподілити уздовж вертикалі);
 - Order – змінити шар розташування графічного об’єкта (Bring to front – розмістити на передньому плані, Send to back – перенести на задній план);
 - Group – згрупувати виділені графічні об’єкти;

- Ungroup – розгрупувати графічний об’єкт;
- Renumber the functions – перенумерувати функціональні блоки;
- Renumber the links – перенумерувати зв’язки між блоками;
- Wiring mode – режим позначення зв’язків між блоками (Text – у вигляді текстових нумерованих позначок, Wiring – лініями);
- Type of wiring – тип виділеного зв’язку (Text – текстовою нумерованою позначкою, Wiring – лінією, Modify the text... – перейменувати позначений зв’язок);
- Draw – накреслити графічний об’єкт:
 - Line – відрізок прямої;
 - Rectangle – прямокутник або квадрат;
 - Ellipse – еліпс або коло;
 - Text – текстовий блок;
 - Image – завантажити зображення із bmp-файлу;
 - Border – показати/сховати бордюр (абрис) графічного об’єкта;
 - Width – товщина лінії бордюру (Single line – одинарна, Double line – подвійна, Triple line – потрійна товщина);
 - Shape color – колір бордюру графічного об’єкта (Assign – зробити однаковим для виділених об’єктів, Choose – вибрати з палітри);
 - Background color – колір заливки графічного об’єкта (Assign – зробити однаковим для виділених об’єктів, Choose – вибрати з палітри);

У режимі програмування мовою FBD користувачеві доступні такі види вікон:

- Edit window – вікно редагування програми;
- Supervision/Monitoring window – вікно супервізора/моніторингу.

Приклад програми, що написана мовою FBD, наведений на рис. 5.2.

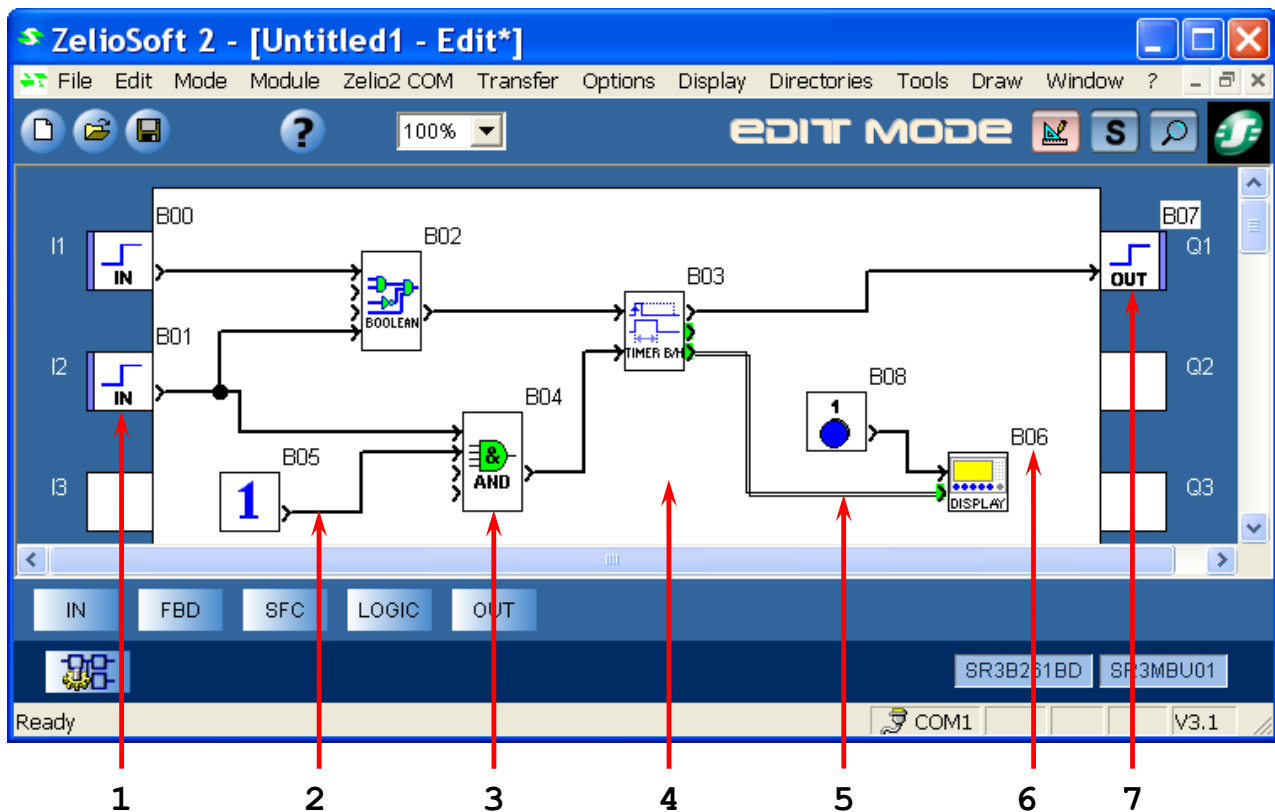


Рис. 5.2. Приклад програми, що написана мовою FBD

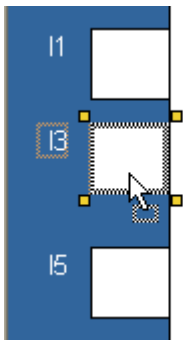
На [рис. 5.2](#) позначені:

1. Зона входів (дискретні чи аналогові входи);
2. З'єднання між двома функціональними блоками (однорозрядний провід);
3. Функціональний блок;
4. Аркуш з'єднань (поле для набору програми);
5. З'єднання між двома функціональними блоками (шина);
6. Номер функціонального блока;
7. Зона виходів (дискретні виходи).

Набір та редагування програми виконується з використанням стандартних методів (у тому числі «Drag/Drop» та «Copy/Paste»), також функціонують «гарячі» комбінації клавіш «Ctrl + c» – копіювати, «Ctrl + v» – вставити, «Ctrl + x» – вирізати, «Ctrl + a» – виділити (вибрати) все, тощо.

Окрім піктограм функціональних блоків на полі для набору програми можна розміщувати графічні примітиви та завантажені із bmp-файлу зображення.

Чашечки зони входів або зони виходів можна зміщувати та міняти місцями, рухаючи їх покажчиком миші:



Якщо підвести покажчик миші до чашечки зони входів або зони виходів та затримати його, можна побачити спливаючу підказку щодо призначення відповідного входу/виходу реле Zelio Logic або модуля розширення:



Призначення входів/виходів реле Zelio Logic та модулів розширення визначається апаратною побудовою та відображається такими спливаючими підказками:

- DISCR – дискретний вхід;
- 8I/6O DISCR – дискретний вхід модуля розширення;
- FAST COUNTING – інкрементний вхід швидкого лічильника;
- FAST DOWN COUNTING – декрементний вхід швидкого лічильника;
- ANA – аналоговий вхід;
- 2E/2S ANALOG 10 BITS – аналоговий 10-розрядний вхід або вихід;
- MODBUS INTEGER – вхід або вихід модуля зв'язку шиною Modbus;
- NORMALLY OPEN RELAY – релейний вихід;
- 8I/6O NORMALLY OPEN RELAY – релейний вихід модуля розширення.

4.2. Функціональні блоки (функції) мови FBD

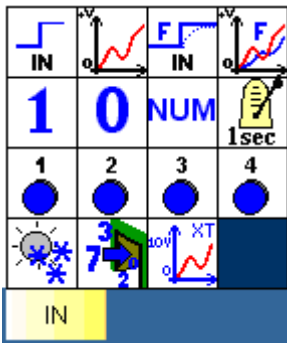
При складанні програми різноманітні функціональні блоки доступні користувачеві за допомогою панелі функцій (функціональних блоків) Function Bar:



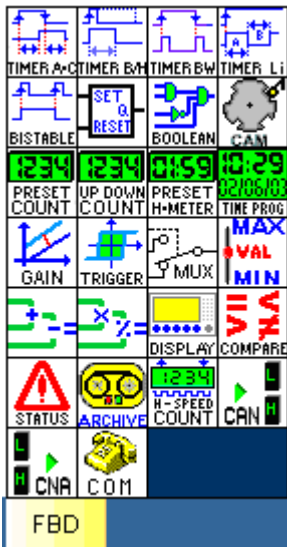
На цій панелі можна побачити такі кнопки:

-  – вставка дискретного, аналогового або спеціального входу (Input);
-  – вставка стандартного функціонального блока (Standard Function);
-  – вставка спеціального функціонального блока (SFC);
-  – вставка логічної функції (Logic Function);
-  – вставка виходу (Output).

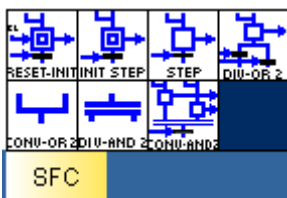
Коли курсор миші розміщується над кнопками панелі функцій, з'являються відповідні панелі піктограм доступних функціональних блоків:



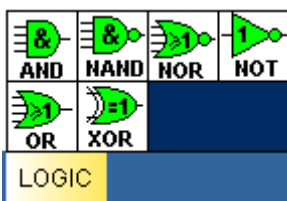
- **IN** – панель входів (Inputs Function Bar);



- **FBD** – панель стандартних функцій (Standard Functions Bar);



- **SFC** – панель спеціальних функцій (SFC Function Bar);

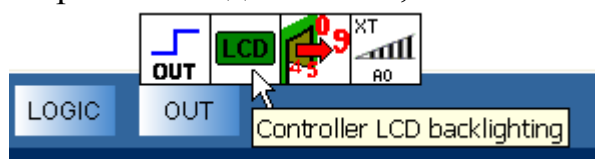


- **LOGIC** – панель логічних функцій (Logic Functions Bar);



- **OUT** – панель виходів (Outputs Function Bar).

Якщо навести покажчик миші на піктограму функціонального блока та затримати на деякий час, то з'явиться спливаюча підказка з назвою блока:



4.2.1. Функціональні блоки (функції) входів

Функціональні блоки входів (Input Blocks) доступні з відповідної панелі (Inputs function bar), якщо підвести курсор миші до кнопки «IN».



На панелі входів містяться такі елементи:

- – дискретні входи (Discrete-Type Inputs);
- – дискретний вхід з фільтром (Filtered Discrete-Type Input);
- – аналоговий вхід (Analog-Type Input);
- – аналоговий вхід з фільтром (Filtered Analog Input);
- – цілочисельний вхід (Integer Type Input);
- – спеціальні входи (Special Inputs).



Дискретні входи призначені для введення дискретних сигналів від кнопок, вимикачів, перемикачів, датчиків з релейним виходом, інших контролерів, тощо. Функціональний блок дискретного входу можна розмістити тільки у чашечку зони входів (див. [рис. 5.2](#)). Подвійним кліком лівою кнопкою миші на розміщеному елементі дискретного входу або за допомогою контекстного меню можна викликати вікно параметрів дискретного входу (рис. 5.3):

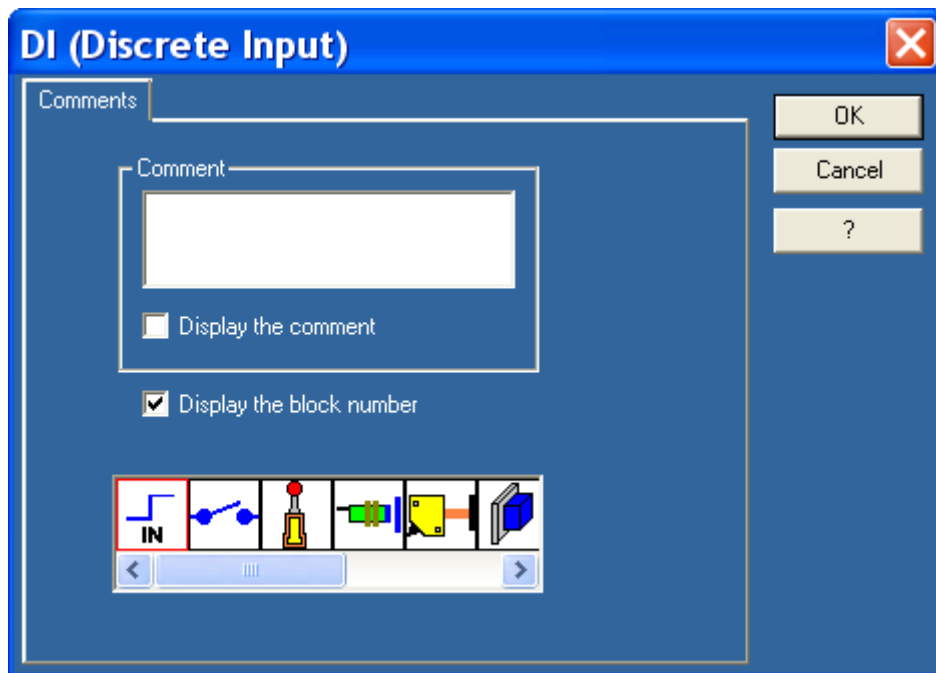


Рис. 5.3. Вікно параметрів дискретного входу

Це вікно має лише одну вкладку, у якій можна ввести текстовий коментар (Comment), дозволити/заборонити вивід коментарю на полі для набору програми (Display the comment), дозволити/заборонити вивід порядкового номеру блока (Display the block number) та вибрати бажану графічну піктограму дискретного входу.

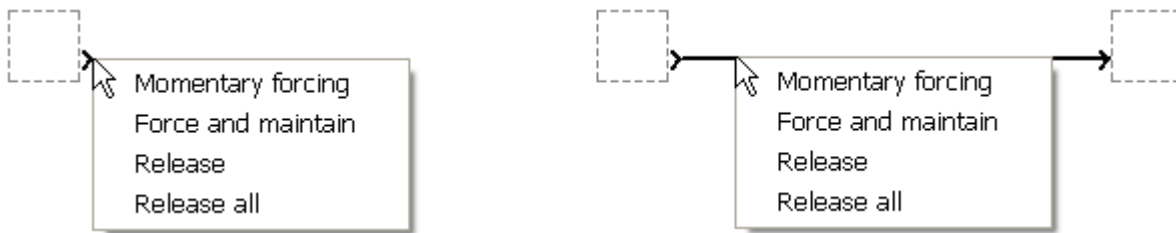
Обрана користувачем піктограма дискретного входу має однаковий вигляд як у вікні редагування програми, так і у вікні супервізора. Під час моделювання або у режимі моніторингу піктограма змінюється в залежності від стану входу (активний/пасивний). Користувач може вибрати піктограму дискретного входу з наступного переліку:

-  < неактивний / активний >  – дискретний вхід (Discrete input);
-   – контакт (Contact);
-   – кінцевий вимикач (Limit switch);
-   – датчик властивостей (Proximity sensor);
-   – датчик присутності або виявлення (Presence sensor);
-   – кнопка з підсвічуванням (Illuminated pushbutton);
-   – перемикач (Selector switch);
-   – кнопка (Pushbutton);
-   – замикаючий контакт реле (Normally open relay).

Обраний вигляд піктограми жодним чином не впливає на роботу дискретного входу і лише спрощує візуальний нагляд оператором за роботою системи у режимі моделювання або моніторингу.

Стан входу (активний/неактивний) можна довільно змінювати у режимах моделювання та моніторингу кліком миші на його піктограмі. Інші блоки функціонують при цьому як і раніше. Цей прийом, який може бути корисний для налагодження програми чи апаратної частини системи, називається форсуванням (Forcing). Його можна застосовувати для більшості функціональних блоків і навіть для ліній зв'язку між блоками.

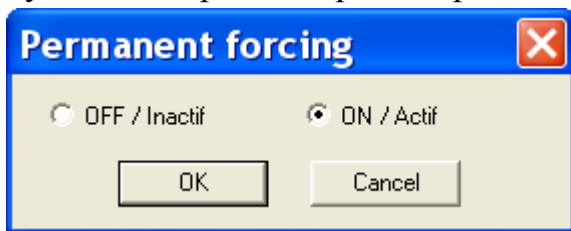
Застосувати або відмінити форсування до функціонального блока можна також за допомогою контекстного меню, що викликається кліком правої кнопки миші на вході чи виході блока, або на лінії зв'язку:



При цьому доступні такі команди:

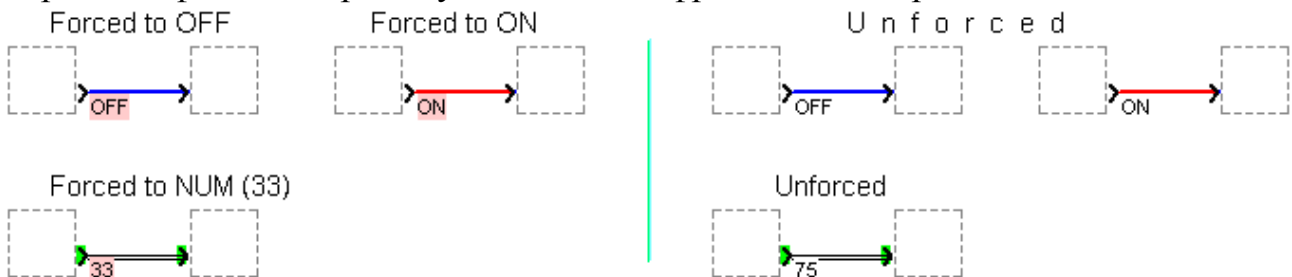
- **Momentary forcing** – тимчасове форсування на 1 цикл (Scan);
- **Force and maintain** –форсувати та утримувати (зафіксувати);
- **Release** – вивільнити указаний елемент;
- **Release all** – вивільнити усі дані.

Якщо у меню вибрати команду форсування, з'явиться діалогове вікно, у якому буде слід обрати потрібний рівень сигналу (стан функціонального блока):

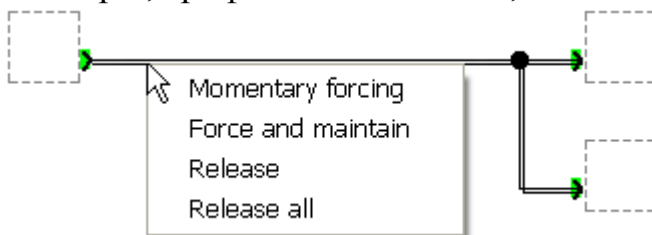


Для повернення контролю над форсованими блоками (Forced data) від оператора до апаратної частини Zelio Logic (тобто для скасування фіксації) необхідно використати команду контекстного меню «Вивільнити» (Release) або «Вивільнити усі дані» (Release all).

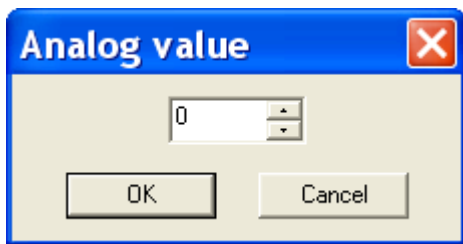
Зафіксований за допомогою форсування сигнал можна «впізнати» за блідо-червоним фоновим прямокутником на цифровій або літерній позначці:



Застосувати форсування можна також для багаторозрядних даних, наприклад цілочисельних входів/виходів, аналогових входів, лічильників, компараторів, таймерів, арифметичних блоків, тощо:



При цьому оболонка ZelioSoft 2 запропонує ввести потрібне значення:



При спробі ввести значення, що не відповідає розрядності даного функціонального блока, буде видане повідомлення про помилку: «Please enter an integer between 0 and 255», «...0 and 1023» або «...–32768 and 32767».



Дискретний вхід з фільтрацією дозволяє покращити завадозахищеність системи автоматизації за рахунок зменшення чутливості до імпульсних завад. Вікно параметрів такого входу має додаткову вкладку **Parameters**, у якій можна задати тривалість (умовну постійну часу) фільтрації (**Duration of filtering**) у межах $1...255 \times 10$ мс. Для спрацьовування входу на протязі встановленого часу вхідний сигнал має бути стабільним.

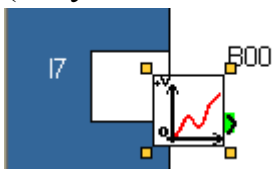


Увага! Кожного разу, коли ви змінюєте тривалість обчислювального циклу контролера, це може призвести до зміни тривалості інтервалу фільтрації. Тому слід перевіряти та виправляти тривалість фільтрації (**Duration of filtering**) усіх задіяних дискретних входів з фільтрацією. Невиконання цієї інструкції може призвести до непередбачуваної роботи або пошкодження технологічного устаткування.

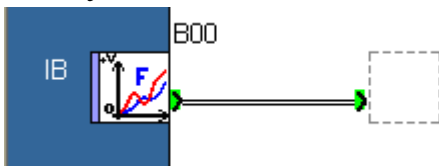


Аналогові входи доступні для всіх моделей реле Zelio Logic з постійною напругою живлення. Напруга на вході конвертується у числове значення за допомогою 8-бітного аналого-цифрового перетворювача (АЦП). Вихідний код АЦП знаходиться у межах $0...255$.

Функції введення аналогових сигналів можуть виконувати входи **IB...IG**. Якщо спробувати розмістити блок аналогового входу наприклад до чашечки з номером **I7**, оболонка ZelioSoft 2 виведе повідомлення «Incompatible types» (несумісність типів). При цьому блок не встановиться до чашечки:



Вихід аналогового блока на піктограмі зафарбований у зелений колір, що вказує на те, що вихідний сигнал не є однорозрядним і з подальшим блоком вихід буде з'єднуватися шиною:



За умовчанням вхідна напруга має змінюватись у межах 0...10 В. Проте за допомогою вкладки «Parameters» вікна параметрів аналогового входу можна задати діапазон зміни вхідної напруги від 0 В до напруги живлення реле (рис. 5.4). Для цього у рамці «Electrical connection at input» слід встановити опцію «Potentiometer» (потенціометричний вхід).

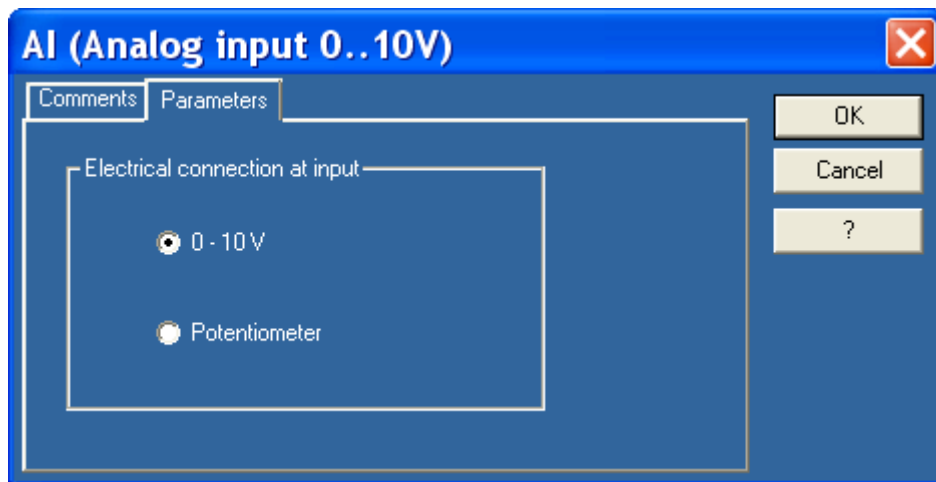


Рис. 5.4. Вікно параметрів аналогового входу

Заради зручності спостереження за роботою системи автоматизації користувач може використати одну з наступних піктограм для позначення аналогового входу:

-  – аналоговий вхід (Input);
-  – аналоговий вхід з діапазоном вхідної напруги 0...10 В (Input);
-  – температура (Temperature);
-  – потенціометр (Potentiometer).

Як і у випадку з дискретними входами, вигляд піктограми жодним чином не впливає на роботу завантаженої до реле Zelio Logic програми.



Аналогові входи з фільтрацією як і звичайні аналогові входи доступні для всіх моделей реле Zelio Logic з постійною напругою живлення і встановлюються у чашечки зони входів з номерами IB...IG. Проте додатково вони облаштовані пропусковим фільтром низьких частот (ФНЧ).

Вхідний фільтр змінює параметри сигналу, що надходить (а саме – амплітуду та фазовий зсув) у залежності від частоти сигналу. Фазовий зсув у даному випадку є фазовою затримкою. Користувач може змінювати лише один параметр ФНЧ – частоту зрізу. Слід мати на увазі: якщо частота вхідного гармонічного сигналу дорівнює частоті зрізу, його амплітуда зменшується приблизно на 30 %, а

фазовий зсув складатиме 45° . З подальшим зростанням частоти чутливість входу знижується а фазова затримка збільшується.

Частоту зрізу вхідного ФНЧ можна вибрати з ряду (0.06, 0.12, ..., 88.25) Hz у комбінованому списку «Cut-off frequency» на вкладці «Parameters» вікна налагодження аналогового входу з фільтрацією (рис. 5.5)

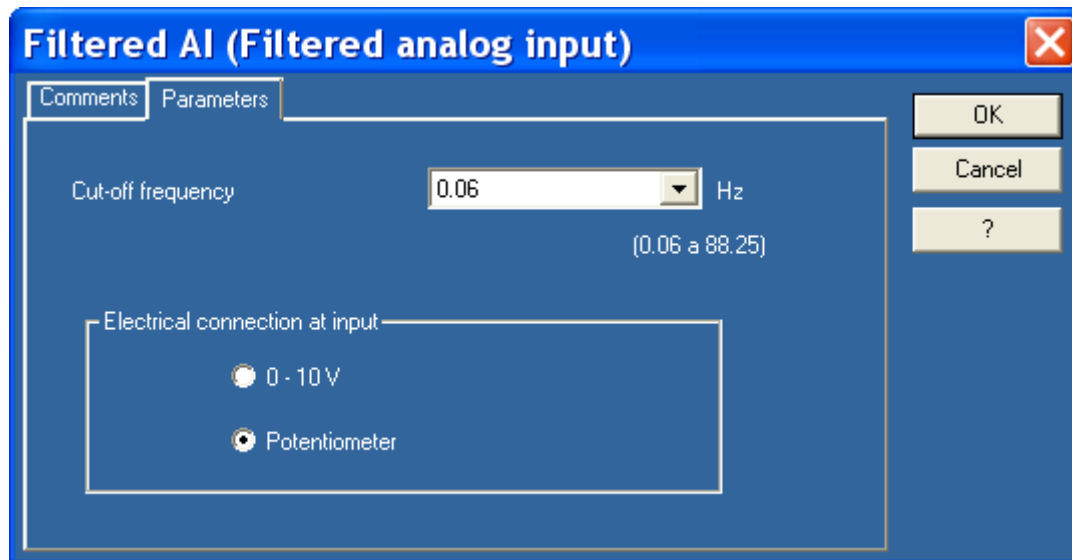


Рис. 5.5. Вікно параметрів аналогового входу з фільтрацією



Увага! Кожного разу, коли ви змінюєте тривалість обчислювального циклу контролера, слід перевіряти або встановлювати частоту зрізу. Невиконання цієї інструкції може призвести до непередбачуваної роботи або пошкодження технологічного устаткування.



Вхід цілочисельного типу доступний для деяких модулів розширення (наприклад, для модуля зв'язку шиною Modbus SR3MBU01 можна використовувати 4 таких входи). Цей функціональний блок призначений для введення 16-бітного двійкового слова (числа у діапазоні $-32768...+32767$).

Цілочисельний вхід можна розмістити у чашечки J1...J4 або J9...JB зони входів модулів розширення. Він не має параметрів для налагодження, окрім параметрів виводу коментарю та порядкового номера блока.

До спеціальних входів відносяться такі функціональні блоки:

-  – кнопки лицевої панелі (Button-Type Inputs);
-  – бітові константи (Discrete Constant-Type Inputs);
-  – цілочисельна константа (Numerical Constant-Type Input);
-  – вхід літнього часу (Summer Time Input);

-  – метроном (Blinking Input);
-  – 10-бітний аналоговий вхід (Analog Input 10 bits).



Кнопки лицевої панелі – це функціональні блоки, що генерують сигнали, які є логічним відображенням фізичного стану кнопок керування курсором «Zx keys» (Z1, Z2, Z3, Z4). Їх треба розміщати безпосередньо на полі для набору програми. Розмістити Такий блок у чашечку зони входів неможливо:



У режимі моделювання або моніторингу змінити стан (активний/неактивний) цього функціонального блока можна кліком лівої кнопки миші на його піктограмі:



Якщо відмінити форсування (зняти фіксацію) у режимі моніторингу буде відображатися дійсний стан відповідної кнопки.

Функція використання кнопок керування курсором на лицевій панелі як дискретних входів встановлюється автоматично при розміщенні блока на полі для набору програми. Тому вікно конфігурації програми (Edit > Program Configuration > Configuration) не має прапорця Zx keys inactive. Нагадуємо, що у режимі програмування мовою сходових діаграм така опція є: 



Бітові константи – це функціональні блоки що призначені для ініціалізації не під'єднаних логічних входів інших функціональних блоків. У режимах моделювання та моніторингу ці константи є взаємно оберненими. Змінити значення константи (активувати чи дезактивувати) можна простим кліком миші:



Контролер, на відміну від користувача, не може змінити значення бітової константи.



Цілочисельна константа – це функціональний блок що призначений для введення до програми 16-бітного цілого числа у діапазоні –32768...+32767. Його можна використовувати для ініціалізації не під'єднаних шин інших функціональних блоків або для задання коефіцієнтів підсилення, величин зміщення та уставок. Тобто цілочисельні константи частіше за все

використовуються спільно з такими блоками як підсилювачі (Gain), тригери (Trigger), суматори (ADD/SUBB), помножувачі (MUL/DIV), компаратори (Comparison), мультиплексори (MUX), перетворювачі кодів, тощо.

Значення константи можна попередньо встановити у вікні параметрів цього функціонального блока (рис. 5.6).

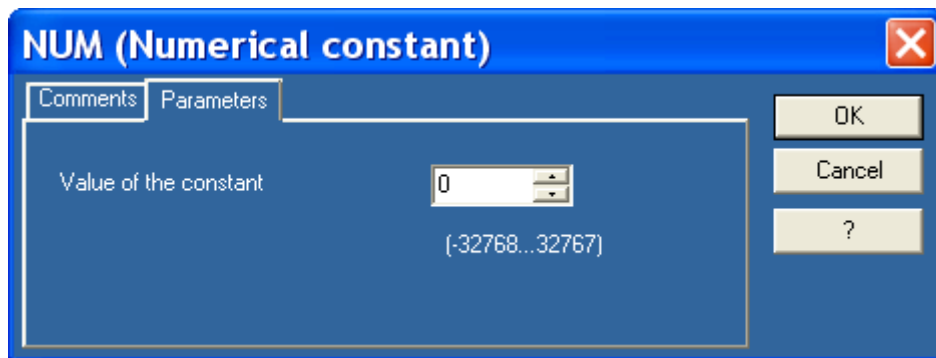


Рис. 5.6. Вікно параметрів функціонального блока цілочисельної константи

Також можна змінювати число що надходить до програми з виходу такого блока у режимах моделювання чи моніторингу:



Вихід функціонального блока цілочисельної константи є багаторозрядним (16-бітним), тому на піктограмі він позначений зеленим кольором і з іншими блоками з'єднується шиною.



Вхід літнього часу призначений для контролю переведення годинника на літній/зимовий час. Цей функціональний є активним і блок генерує сигнал «ON» у період дії літнього часу і навпаки – є пасивним у період дії зимового часу:



Для підтвердження цієї функції необхідно виконати наступні дії:

- За допомогою команди меню «Edit > Program configuration» або кнопки  відкрити вікно конфігурації програми та перейти на вкладку вибору формату календарної дати «Date format» (див. рис. 5.7).
- Встановити прапорець активізації переведення літнього/зимового часу (Activate the summer/winter time change);
- Визначити календарну дату, коли здійснюється перехід у залежності від місцезнаходження:
 - вибрати географічну зону («Europe» або «USA»),
 - або вручну задати дати переходу на літній та зимовий час.

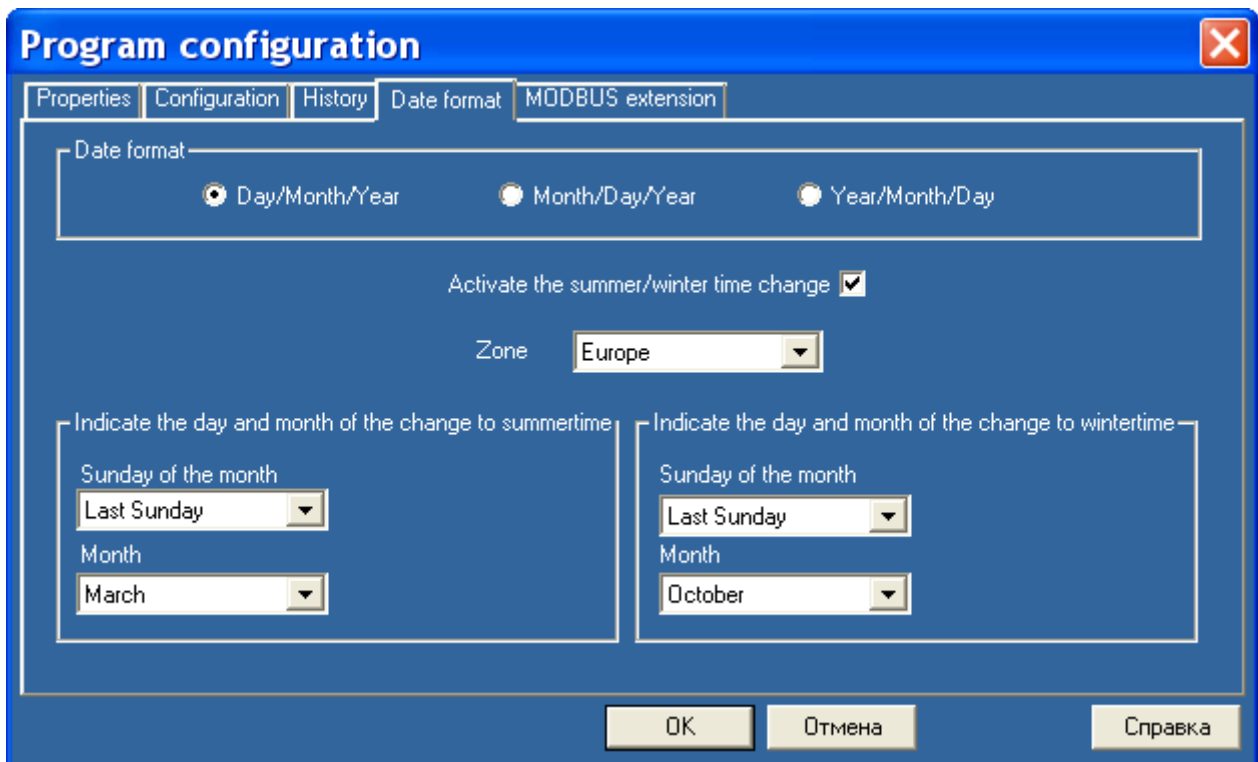


Рис. 5.7. Вкладка «Формат дати» вікна конфігурації програми



Метроном – це функціональний блок, що генерує меандр частотою 1 Гц шпаруватістю 2 (тобто з коефіцієнтом заповнення 0,5). Піктограма метронома змінюється у відповідності з поточним станом (Active/Inactive). До речі стан метронома можна змінювати також вручну кліком миші як у режимі моделювання так і моніторингу (наприклад, для налагодження програми):



Аналоговий вхід розрядністю 10 біт призначений для перетворення аналогового сигналу, що надходить на вхід ІН або ІJ модуля розширення SR3XT43BD, в ціле число у діапазоні 0...1023.

У програмі 10-розрядний аналоговий вхід блока розширення можна відрізнити від 10-вольтового входу базового блока (Smart relay) можна за буквами «XT» у піктограмі (Analog eXTension):



Для налагодження аналогових входів модуля розширення слід відкрити вікно конфігурації програми та перейти на вкладку «Analog extension» (рис. 5.8).

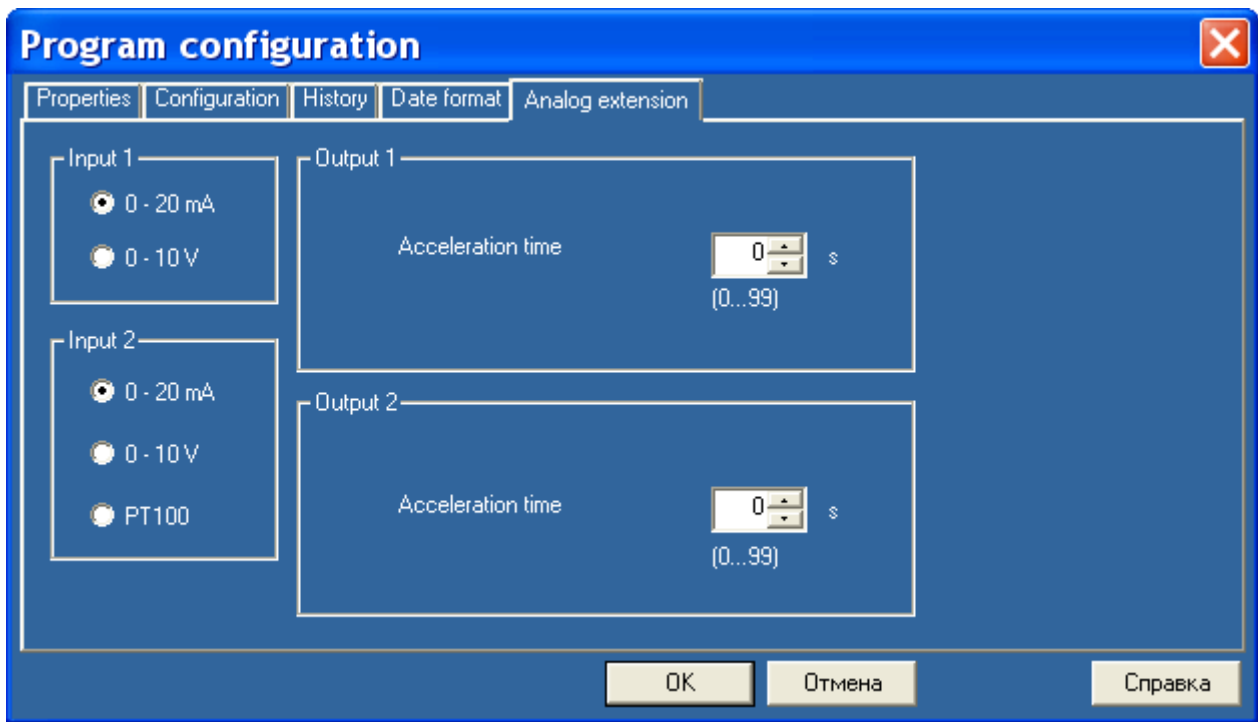


Рис. 5.8. Вкладка «Аналоговий модуль розширення» вікна конфігурації програми

Як можна побачити на рис. 5.8, кожен із входів можна використовувати як «струмовий» (Intensity input) з діапазоном струму 0...20 мА, або як «вольтовий» (Voltage input) з діапазоном вхідної напруги 0...10 В. Також вхід ІІ можна застосувати та налагодити для підключення терморезистора типу РТ100.

Про налагодження аналогового виходу читайте у розділі 4.2.2.

4.2.2. Функціональні блоки (функції) виходів

Функціональні блоки виходів (Output Blocks) доступні з відповідної панелі (Outputs Function Bar), якщо підвести курсор миші до кнопки «OUT».



На панелі виходів містяться такі елементи:

-  – дискретні виходи (Discrete-Type Outputs);
-  – підсвічування дисплею (LCD Screen Backlight Output);
-  – цілочисельний вихід (Integer Type Output);
-  – 10-бітний аналоговий вихід задавача інтенсивності (Analog-Type Output 10 bits).



Дискретні виходи призначені для виведення дискретних сигналів за допомогою транзисторних або релейних виходів реле Zelio Logic та керування таким технологічним обладнанням як пневматичні та гідравлічні клапани, лампи, мотори, нагрівачі, тощо. Функціональний блок дискретного виходу можна розмістити тільки у чашечку зони виходів (див. [рис. 5.2](#)).

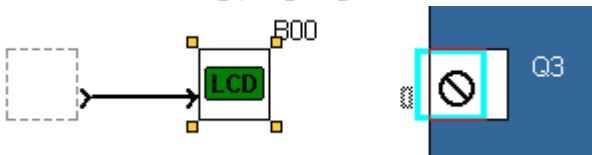
У вікні параметрів дискретного виходу можна обрати одну з таких піктограм:

-  < неактивний / активний >  – дискретний вихід (Discrete output);
-   – замикаючий контакт реле (Normally open relay);
-   – лампа (Lamp);
-   – твердотільне реле (Solid state relay);
-   – клапан (Valve);
-   – виконавчий механізм (Actuator);
-   – електродвигун (Motor);
-   – активне навантаження (Resistance);
-   – звуковий сигнал (Audible signal);
-   – зелена індикаторна лампа (Green indicator lamp);
-   – червона індикаторна лампа (Red indicator light);
-   – оранжева індикаторна лампа (Orange indicator light);
-   – світловий індикатор (Indicator light);
-   – нагрівач (Heating);
-   – вентилятор (Fan).

Обрана піктограма жодним чином не впливає на роботу дискретного виходу а лише дозволяє користувачу у режимах моделювання або моніторингу візуально асоціювати дискретний вихід з конкретною технологічною установкою.



Вихід підсвічування дисплею вмикає функцію підсвічування рідинно-кристалічного дисплею на лицевій панелі модуля у активному стані та вимикає – у пасивному. Цей функціональний блок слід розміщувати безпосередньо на полі для набору програми, а не до чашечки зони виходів:



У режимах моделювання та моніторингу можна спостерігати за станом цього блока (Active/Inactive):



На полі для набору програми можна розмістити кілька функціональних блоків підсвічування дисплею, при цьому якщо хоча б один з них знаходиться у активному стані дисплей буде підсвічуватись.



Цілочисельний вихід використовується для виводу 16-бітного двійкового слова (числа у діапазоні $-32768...+32767$). Він доступний для деяких модулів розширення, зокрема для модуля зв'язку шиною Modbus SR3MBU01.

Цілочисельний вихід можна розмістити у чашечки O1...O4 або O9...O_B зони виходів модулів розширення.



Аналоговий вихід розрядністю 10 біт призначений для виводу напруги, що змінюється лінійно, і функціонує як задавач інтенсивності або пускач чи прискорювач (Accelerator). Цей функціональний блок можна розміщувати до виходів QB та QC модуля розширення SR3XT43BD.

Час зміни вихідної напруги (Acceleration time), що відповідає максимальному перепаду вхідного коду, задається користувачем на вкладці «Analog extension» вікна конфігурації програми ([рис. 5.8](#)) у межах 0...99 с відповідно до динамічних властивостей керованої установки.

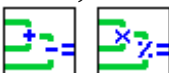
4.2.3. Стандартні функціональні блоки (стандартні функції)

Стандартні функціональні блоки (Standard Functions) доступні з відповідної панелі (Standard functions bar), якщо підвести курсор миші до кнопки «FBD».



На панелі стандартних функцій містяться такі елементи:

-    – таймери (Timers);
-  – таймер-формував імпульсів (Pulses on Edges Function)
-  – імпульсне реле (Bistable Impulse Relay Function);
-  – RS-тригер (RS-Switching, Function Set and Reset);
-  – булева функція (Boolean Function);
-  – барабанний (кулачковий) контролер (Cam Block, Cam Programmer);
-    – лічильники імпульсів (Counters);

-  – лічильник часу (Preset H-meter),
-  – годинник-програмувач (Time Programmer);
-  – підсилювач (Gain);
-  – тригер Шмітта (Schmitt Trigger);
-  – мультиплексор (Multiplexing Function);
-  – компаратори (Compare Functions);
-  – арифметичні функції (Arithmetic Functions);
-  – дисплей (Display Function);
-  – індикатор стану модуля (Module Status);
-  – архіватор даних (Archive Function);
-  – перетворювачі типів даних (Conversion Functions);
-  – вхід та вихід послідовного порту (Serial Port);
-  – функція повідомлення (Message Function).

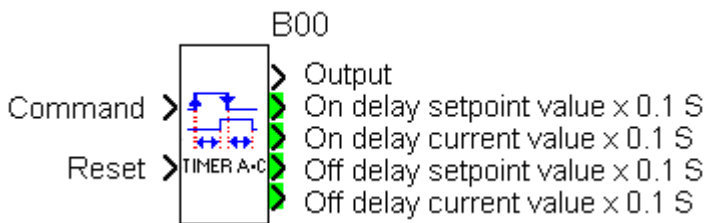
Розглянемо стандартні функціональні блоки докладно.

До таймерів можна віднести три таких функціональних блоки:

-  – таймер A/C (Timer A/C);
-  – таймер В/Н (Timer B/H Time Out);
-  – циклічний таймер або мултивібратор (Timer Li Cyclic Timing);

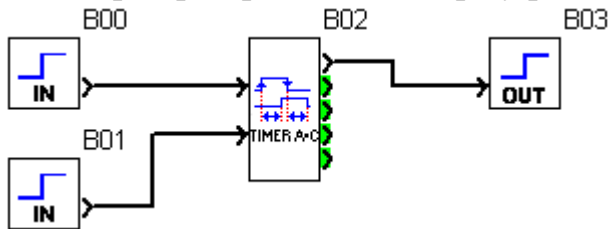
 Таймер A/C призначений для реалізації функцій затримки увімкнення (ON Delay, режим A), затримки вимкнення (OFF Delay, режим C) або затримок і увімкнення і ввімкнення (режим A/C).

Призначення входів/виходів таймера:

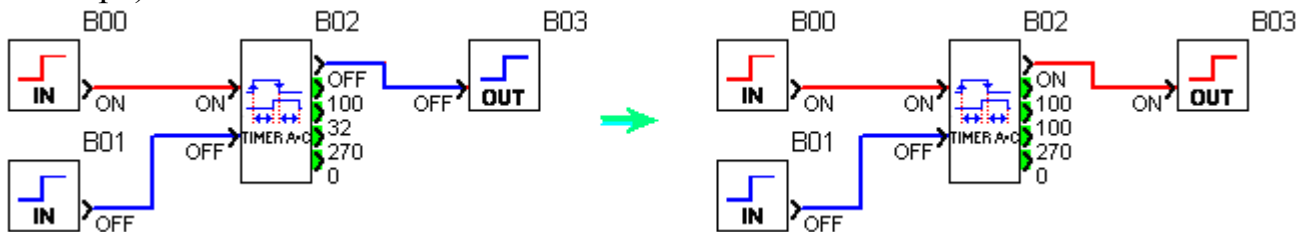


- Command – керуючий вхід сигналу активації/дезактивації;
- Reset – вхід скиду;
- Output – вихід функції затримки;
- On delay setpoint value x 0.1 S – значення уставки затримки увімкнення;
- On delay current value x 0.1 S – поточне значення затримки увімкнення;
- Off delay setpoint value x 0.1 S – значення уставки затримки вимкнення;
- Off delay current value x 0.1 S – поточне значення затримки вимкнення;

Для перевірки роботи таймера у різних режимах слід зібрати таку схему:



У режимах моделювання та моніторингу користувач має змогу спостерігати за зміною поточних значень параметрів таймера. Коли поточне значення параметру співпадає із значенням уставки, стається зміна вихідного сигналу (стану таймера):



Якщо бути точним, таймер має один режим роботи, а наявність та значення затримок часу на увімкнення та на вимкнення задається у вікні параметрів (рис. 5.9).

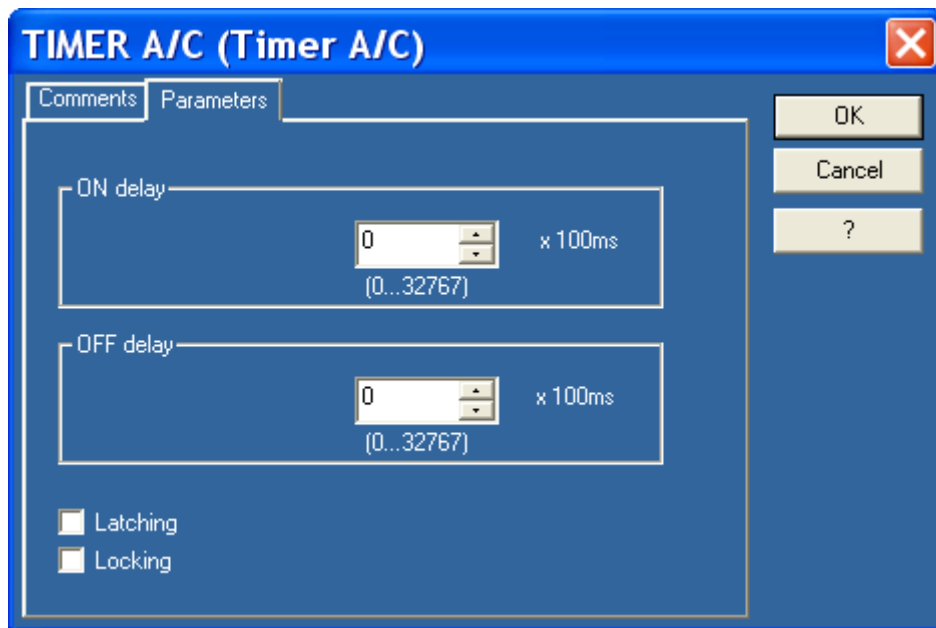
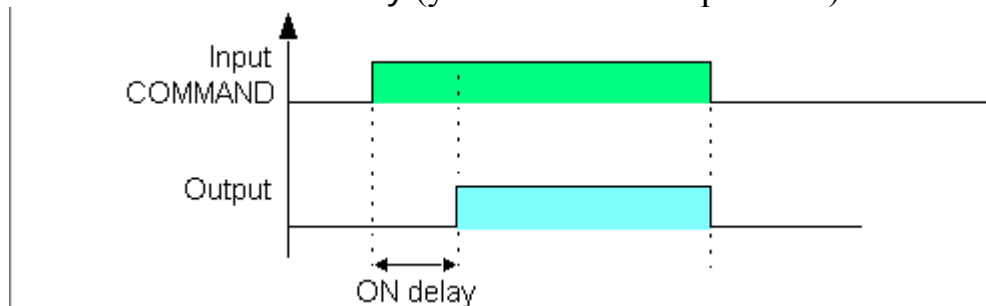


Рис. 5.9. Вікно параметрів таймера A/C

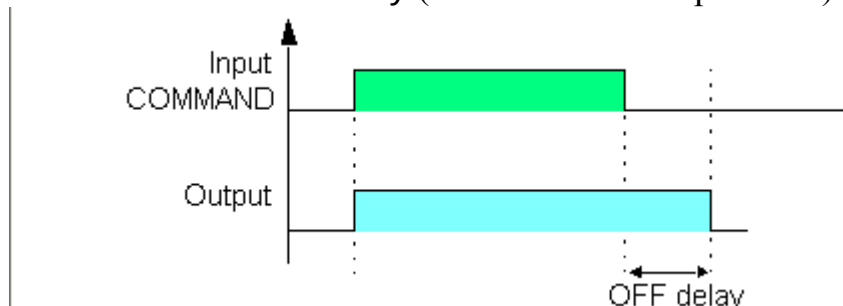
Як можна побачити, тривалість затримки не може перевищувати 32767×100 мс, що складає 3276.7 с, тобто 54 хв та 36.7 с.

Оскільки робота таймерів була докладно описана у частині, що стосується мови програмування сходових діаграм, далі наведемо лише часові діаграми роботи таймера A/C без пояснень.

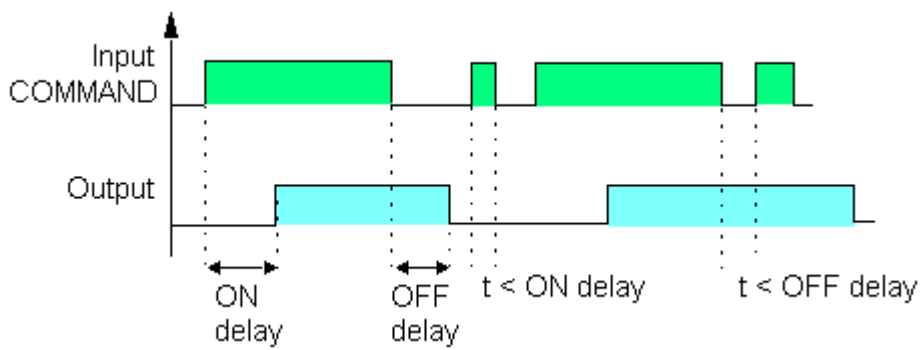
- Function A: On delay (увімкнення із затримкою):



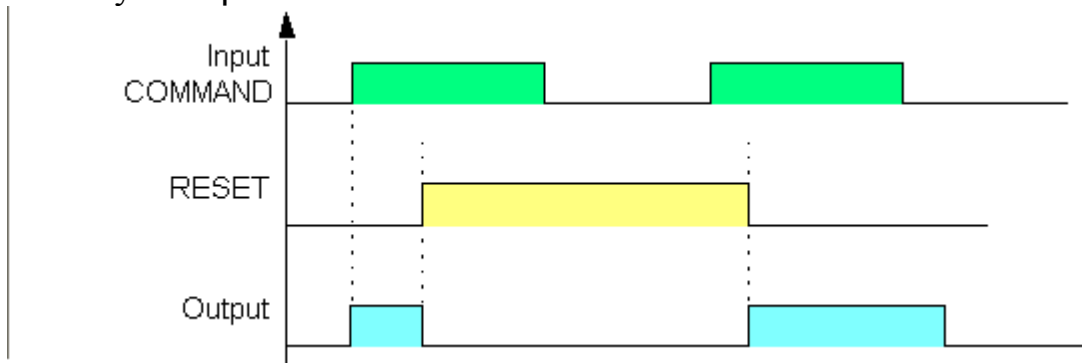
- Function C: Off delay (вимикання із затримкою):



- Function AC: Timing after closing and opening control (увімкнення із затримкою tON, вимкнення – із затримкою tOFF):

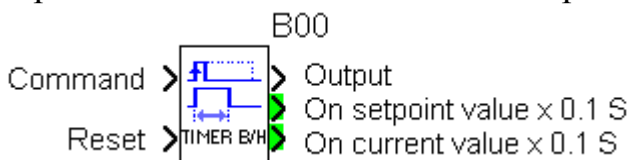


Активация входу «RESET» приводит до скиду поточних значень затримок (On/Off delay current values) і до дезактивації вихідного функціонального сигналу «Output»:



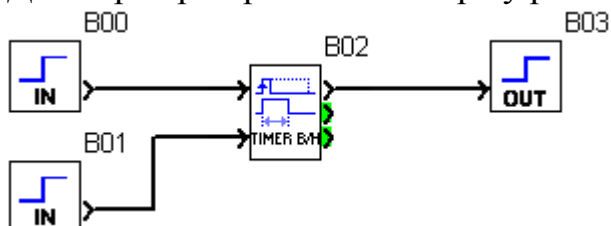
Таймер В/Н призначений для реалізації функцій генерування імпульсу заданої тривалості за фронтом вхідного імпульсу (режим одновібратора).

Призначення входів/виходів таймера:



- Command – керуючий вхід сигналу активації за фронтом (для режиму Н також дезактивації за спадом);
- Reset – вхід скиду;
- Output – вихід, на якому формується імпульс заданої тривалості;
- On setpoint value x 0.1 S – значення уставки тривалості імпульсу;
- On current value x 0.1 S – поточне значення тривалості імпульсу;

Для перевірки роботи таймера у різних режимах слід зібрати таку схему:



Вибрати режим функціонування та задати тривалість вихідного імпульсу можна у вікні параметрів таймера (рис. 5.10).

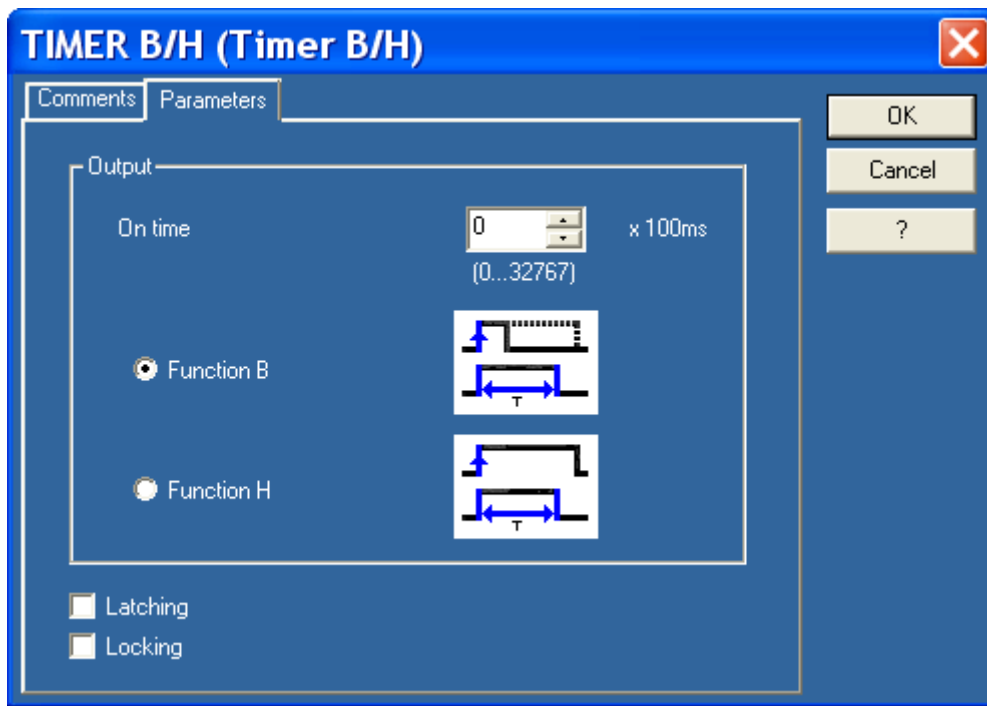
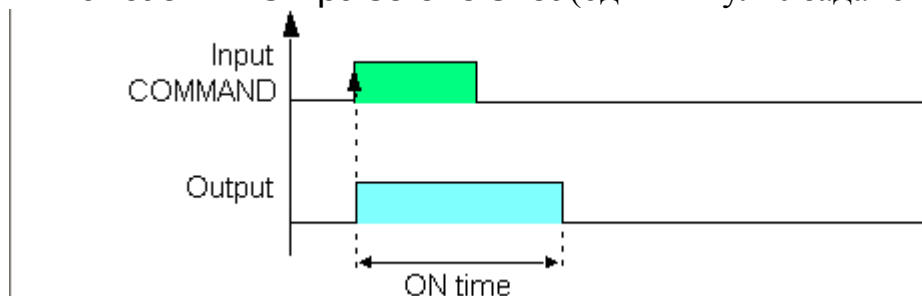


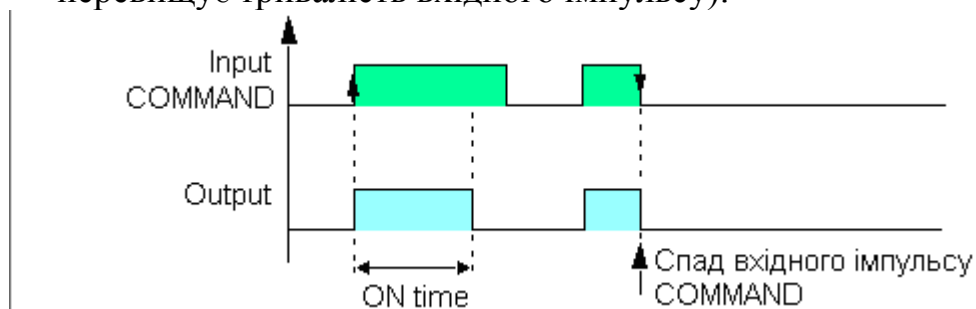
Рис. 5.10. Вікно параметрів таймера В/Н

Відмінність режимів В та Н полягає у тому, що у режимі Н, на відміну від В, вихідний імпульс не може бути довшим, ніж вхідний (див. діаграми).

- Function B: On pulse one shot (один імпульс заданої тривалості):

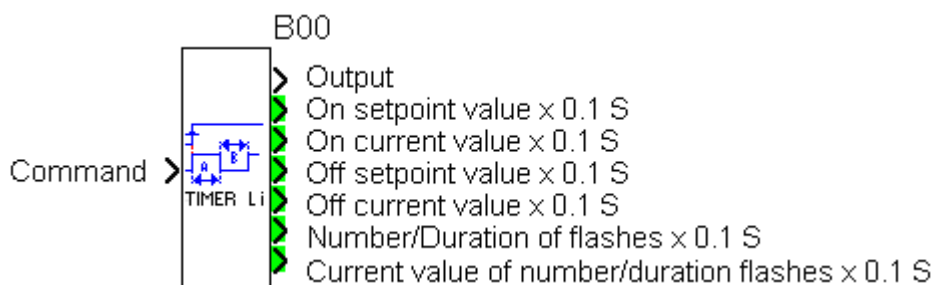


- Function H: On pulse one shot (один імпульс заданої тривалості, що не перевищує тривалість вхідного імпульсу):



Циклічний таймер Li призначений для генерування імпульсів заданої частоти та шпаруватості (режим мультивібратора або генератора).

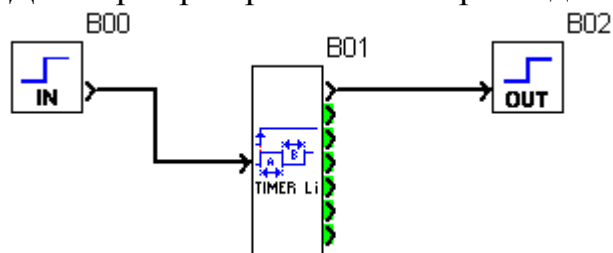
Призначення входів/виходів таймера:



Output
 On setpoint value $\times 0.1 \text{ S}$
 On current value $\times 0.1 \text{ S}$
 Off setpoint value $\times 0.1 \text{ S}$
 Off current value $\times 0.1 \text{ S}$
 Number/Duration of flashes $\times 0.1 \text{ S}$
 Current value of number/duration flashes $\times 0.1 \text{ S}$

- Command – керуючий вхід сигналу активації/дезактивації за рівнем;
- Output – вихід функції затримки;
- On setpoint value $\times 0.1 \text{ S}$ – значення уставки тривалості імпульсу;
- On current value $\times 0.1 \text{ S}$ – поточне значення тривалості імпульсу;
- Off setpoint value $\times 0.1 \text{ S}$ – значення уставки тривалості паузи;
- Off current value $\times 0.1 \text{ S}$ – поточне значення тривалості паузи;
- Number/Duration of flashes $\times 0.1 \text{ S}$ – кількість імпульсів або час їх генерування;
- Current value number/duration of flashes $\times 0.1 \text{ S}$ – поточна кількість згенерованих імпульсів або поточний час роботи таймера.

Для перевірки роботи таймера слід зібрати таку схему:



У вікні параметрів таймера (рис. 5.11) можна не тільки задати тривалості імпульсу та паузи, а й визначити час роботи генератора.

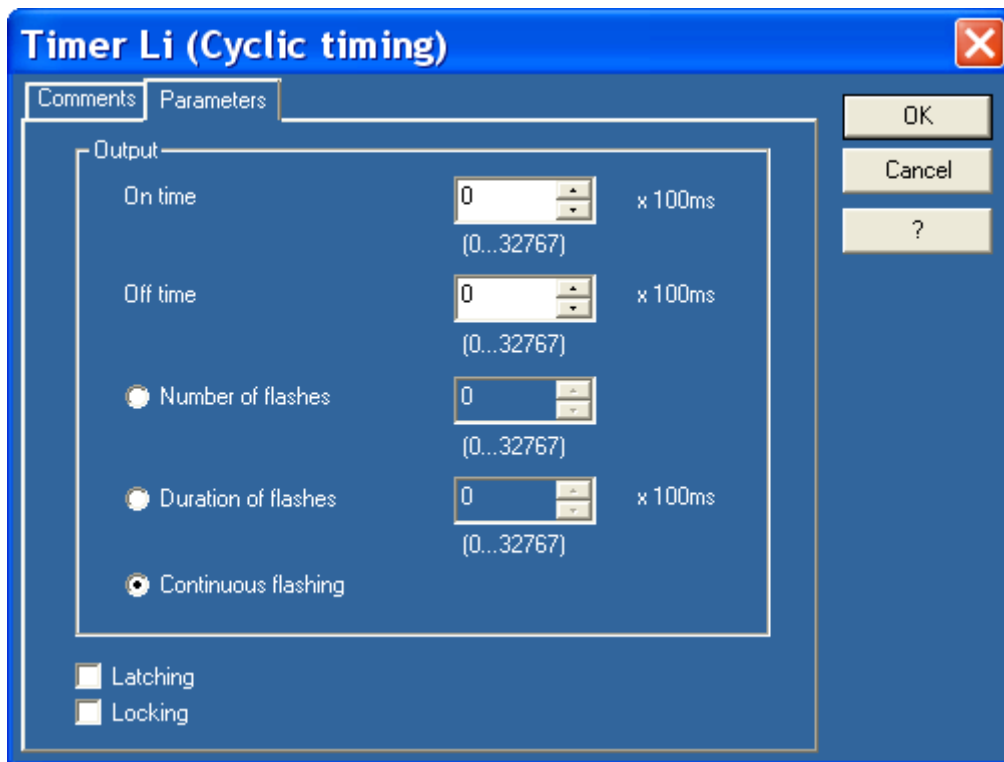


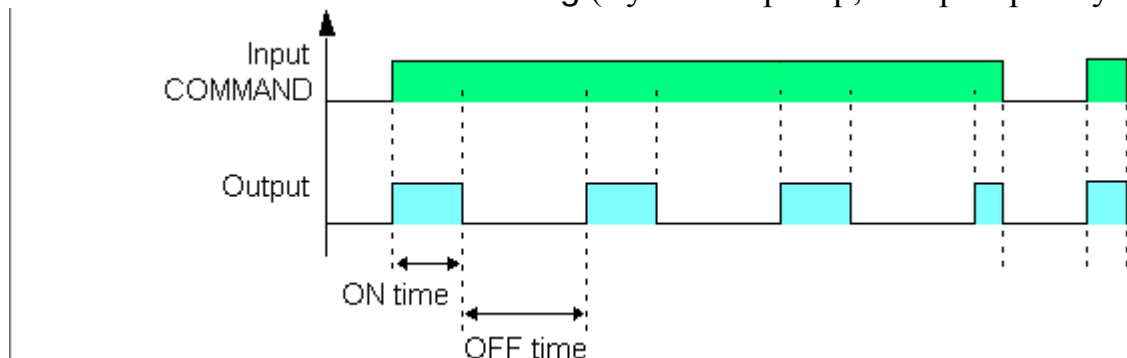
Рис. 5.11. Вікно параметрів таймера Li

На рис. 5.11 можна побачити такі опції:

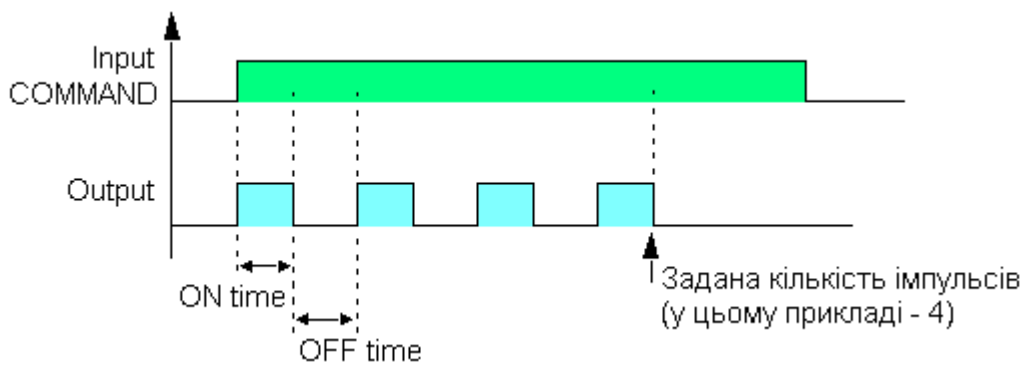
- On time – тривалість імпульсу;
- Off time – тривалість паузи між імпульсами;
- Number of flashes – задати кількість імпульсів, що будуть згенеровані при активації вхідного сигналу «Command»;
- Duration of flashes – задати час, на протязі якого будуть генеруватися імпульси за умови активного рівня вхідного сигналу «Command»;
- Continuous flashing – генерувати імпульси постійно у активному стані.

Далі наведені діаграми роботи таймера Li.

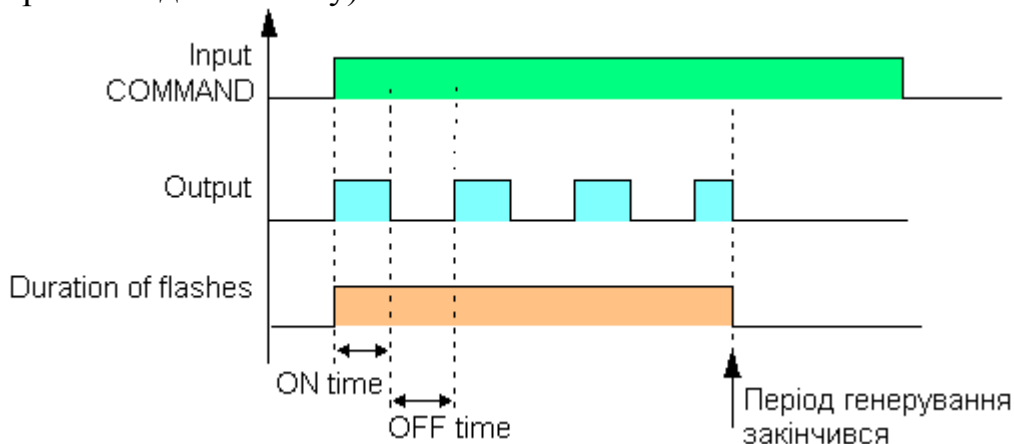
- Function L: Continuous flashing (мультивібратор, генератор імпульсів):



- Function L: Number of flashes (генератор заданої кількості імпульсів):



- **Function L: Duration of flashes** (генератор імпульсів, що генеруються на протязі заданого часу):



Таймер BW – це функціональний блок, який працює як детектор фронтів вхідного сигналу (переднього та/або заднього). Він призначений для формування імпульсів тривалістю 1 цикл (Scan), які запускаються фронтами вхідного сигналу. Оскільки тривалість вихідних імпульсів не регулюється, то цей блок лише умовно можна назвати таймером.

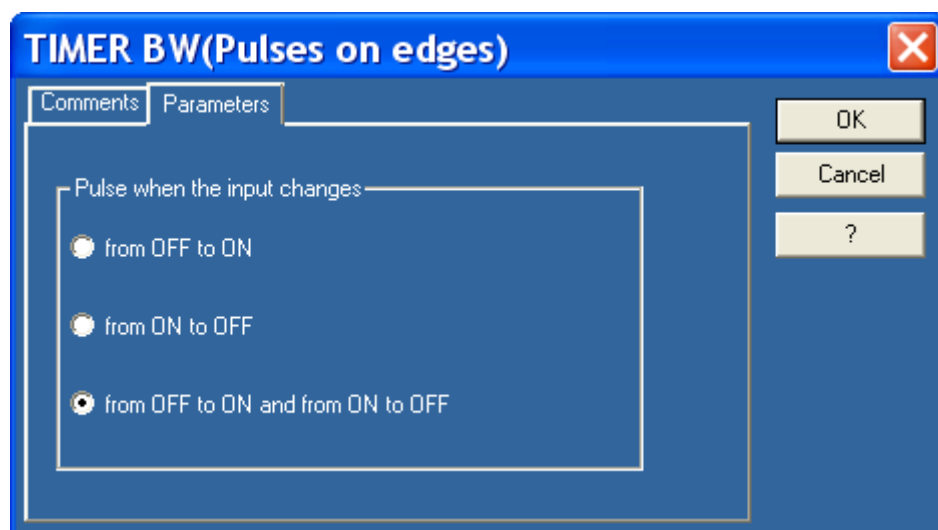


Рис. 5.11. Вікно параметрів таймера BW

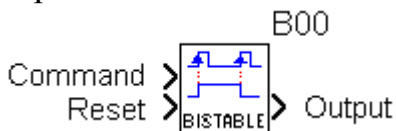
У вікні параметрів таймера-формуваця імпульсів (рис. 5.11) можна встановити такі опції:

- Pulse when the input changes – формувати імпульс на виході, якщо:
 - from OFF to ON – зафіксований передній фронт вхідного сигналу;
 - from ON to OFF – зафіксований задній фронт (спад) вхідного сигналу;
 - from OFF to ON and from ON to OFF – і за фронтом і за спадом вхідного дискретного сигналу.



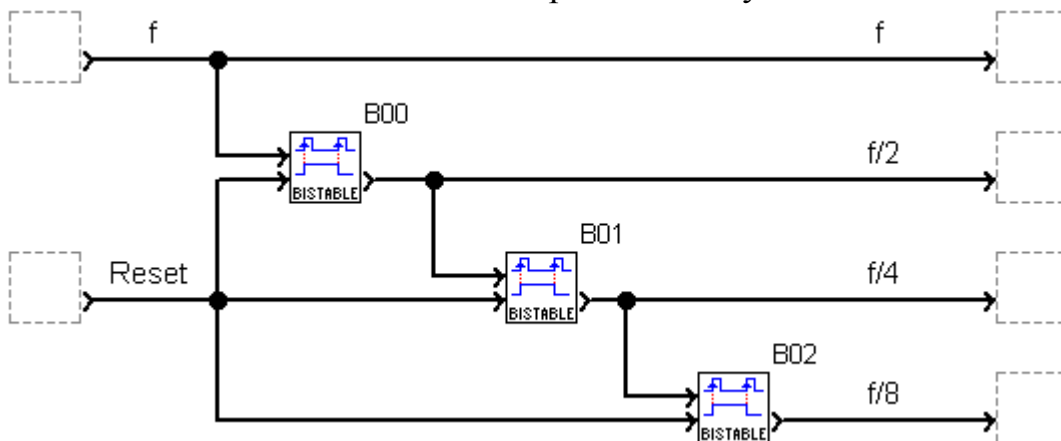
Імпульсне реле – це функціональний блок, який за кожним переднім фронтом імпульсів вхідного командного сигналу змінює рівень вихідного сигналу на протилежний.

Призначення входів/виходів імпульсного реле:



- Command – керуючий вхід перемикавання за фронтом;
- Reset – вхід скиду (активний рівень на цьому вході перемикає та утримує вихід у стані «OFF»);
- Output – вихід.

Функціональний блок імпульсного реле можна застосовувати для побудови подільників частоти з метою використання їх у комбінаційних схемах:



RS-тригер – це функціональний блок, який функціонує у повній відповідності до своєї назви і не потребує докладних пояснень. У вікні параметрів цього блока можна встановити пріоритет одного з дискретних входів (рис. 5.12).

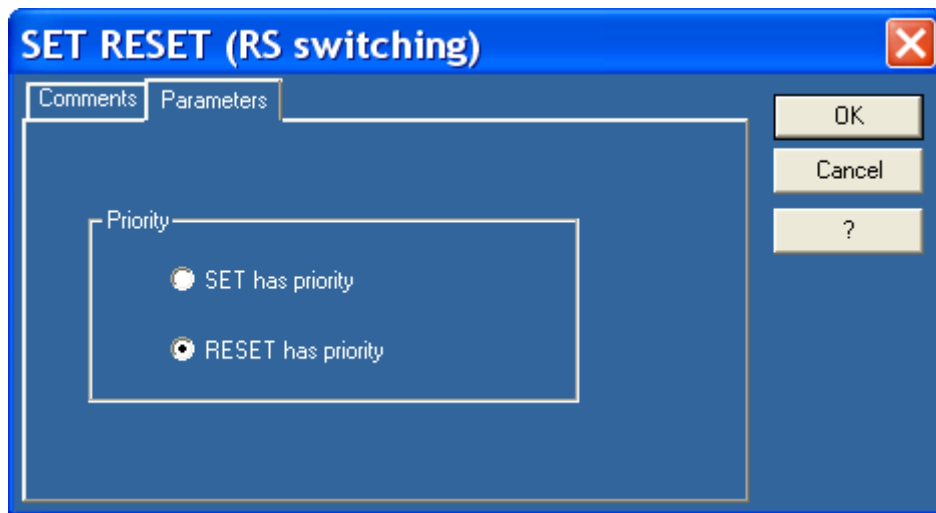
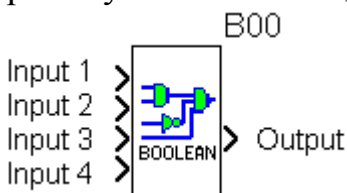


Рис. 5.12. Вікно параметрів RS-тригера

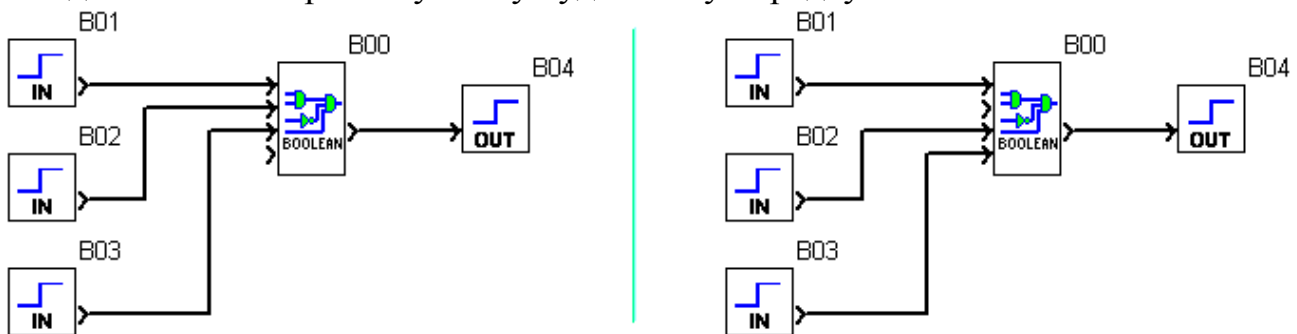
У випадку, коли сигнал високого рівня надходить на обидва входи, вихід встановлюється у відповідності до пріоритетного входу. За умовчанням пріоритет закріплений за входом «Reset».



Булева функція – це функціональний блок, за допомогою якого можна реалізувати комбінаційну схему з одним виходом та 2...4 входами:



Кількість комбінацій визначається кількістю задіяних входів виразом 2^n . Наприклад, якщо підключити 3 входи з чотирьох, то можна задати 8 комбінацій. Входи можна використовувати у будь-якому порядку:



Логіка роботи функціонального блока визначається таблицею істинності, у якій для кожної можливої комбінації вхідних сигналів задається значення вихідного сигналу:

Input 1	Input 2	Input 3	Input 4	Output	
0	0	0	0	1	Активна частина таблиці
1	0	0	0	0	
0	1	0	0	1	
1	1	0	0	0	
0	0	1	0	0	Пасивна частина таблиці
1	0	1	0	0	
0	1	1	0	0	
1	1	1	0	0	

Комбінації вхідних сигналів Вихідне значення

Якщо задіяні не усі входи булевої функції, і до того ж непослідовно (наприклад з номерами 1, 3 та 4), то у таблиці істинності світлі рядки (активна частина таблиці) та темні рядки (пасивна частина) будуть чергуватися:

Input 1	Input 2	Input 3	Input 4	Output
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

Заповнити таблицю істинності булевої функції можна у вікні параметрів цього функціонального блока (рис. 5.13).

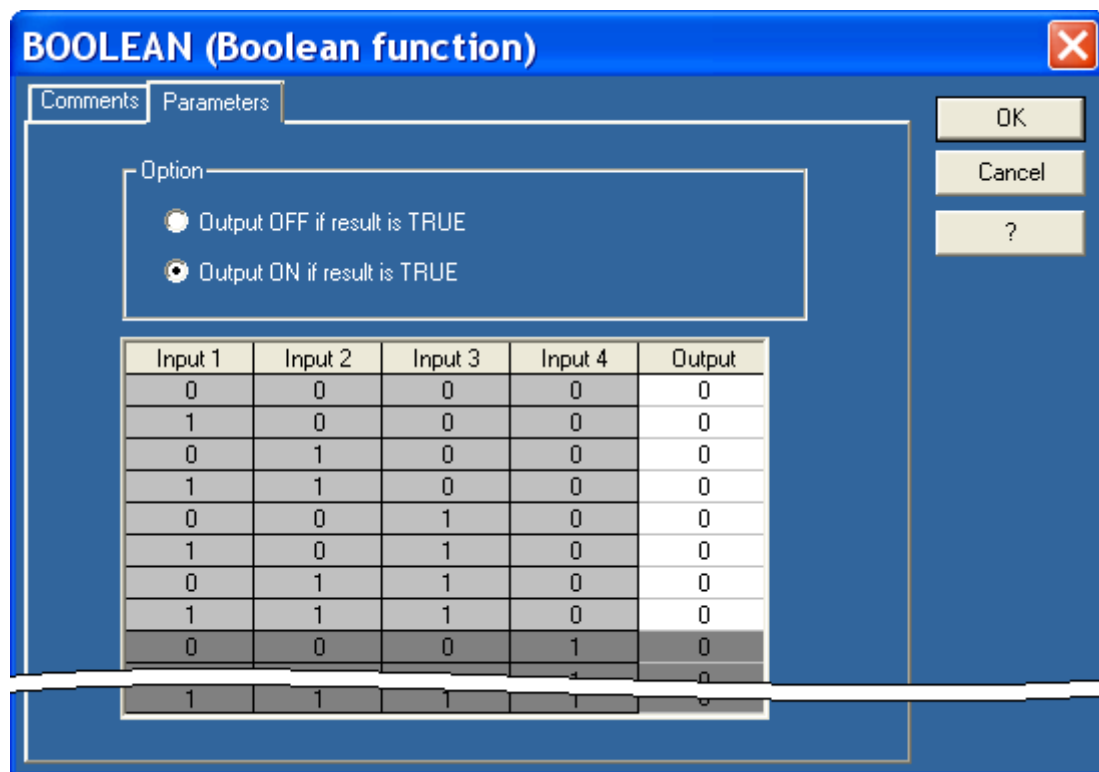


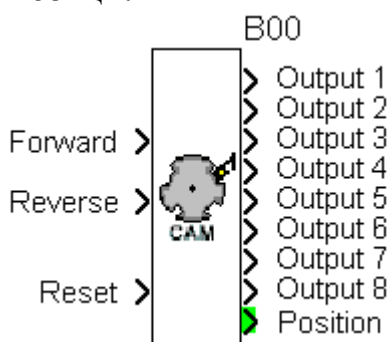
Рис. 5.13. Вікно параметрів булевої функції

У таблиці можна змінити лише значення вихідного сигналу «Output» кліком покажчика миші на відповідних комірках. Також для усієї таблиці можна застосувати пряму або інверсну логіку:

- Output OFF if result is TRUE – інверсна логіка;
- Output ON if result is TRUE – пряма логіка (за умовчанням).

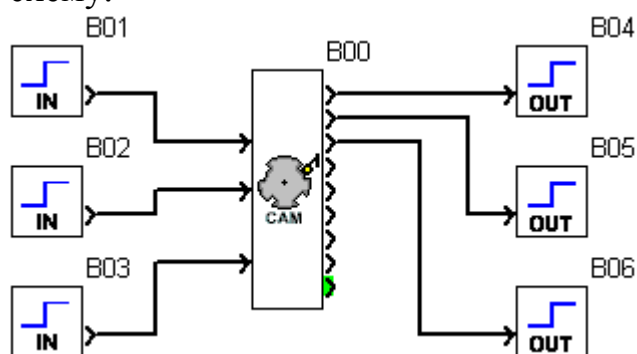


Барабанний (кулачковий) контролер має 8 дискретних виходів, стан яких визначається позицією барабанного колеса. Барабан можна обертати у обох напрямках. Конфігурація блока може бути встановлена для кожної з 50-ти можливих позицій окремо для кожного дискретного виходу. Барабанний контролер має також цифровий вихід, на який виводиться номер поточної позиції:



- Forward – обертання на 1 позицію уперед за переднім фронтом імпульсу;
- Reverse – обертання на 1 позицію назад за фронтом імпульсу;
- Reset – вхід скиду позиції (встановлення колеса у позицію 1);
- Output 1...Output 8 – дискретні виходи;
- Position – номер поточної позиції барабана.

Для перевірки роботи барабанного контролера можна скласти таку найпростішу схему:



Логіка роботи барабанного контролера визначається конфігураційною таблицею (таблицею істинності):

Position	S1	S2	S3	S4	S5	S6	S7	S8
1	0	0	0	0	0	1	0	0
2	1	0	1	0	0	0	1	1
3	0	0	1	0	1	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0

↑
Позиція колеса

Комбінації вихідних сигналів для кожної позиції

Активна частина таблиці

Пасивна частина таблиці

У кожний момент часу актуальною є лише одна поточна позиція барабанного колеса. Якщо у рядку таблиці, що відповідає поточній позиції, є одиниці, то відповідні дискретні виходи встановлюються у активний стан «ON». Інші виходи скидаються у стан «OFF». Вікно параметрів функціонального блока барабанного контролера наведено на рис. 5.14.

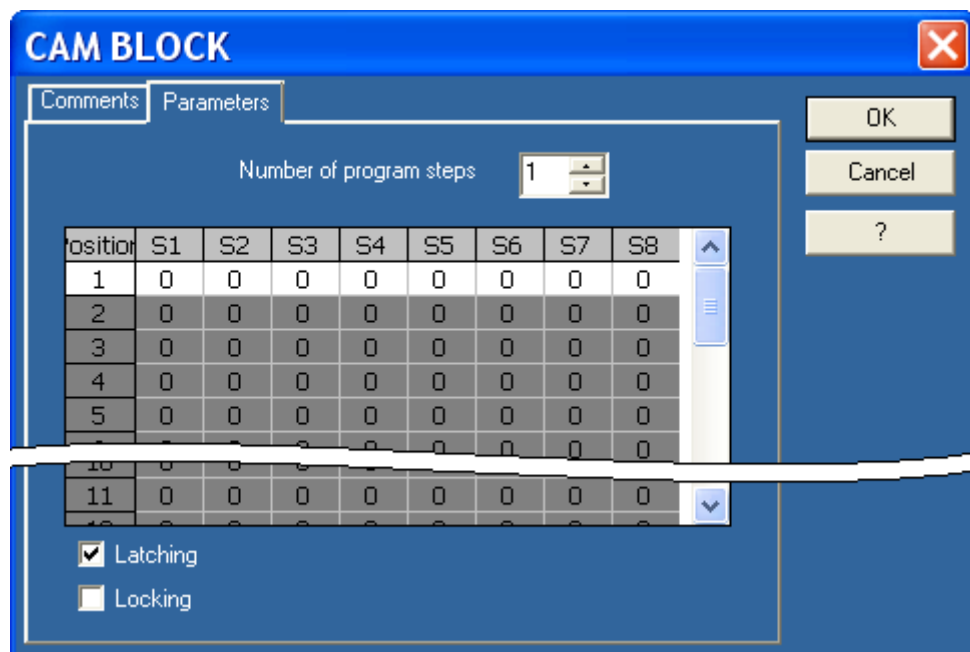


Рис. 5.14. Вікно параметрів барабанного контролера

У вікні можна побачити такі позначки:

- Number of program steps – кількість активних позицій контролера;
- Position – номер позиції барабанного колеса;
- S1, S2, ..., S8 – колонки таблиці, що відповідають дискретним виходам.

Із двох дискретних входів «Forward» та «Reverse» більш пріоритетним є вхід обертання уперед (Forward). А найбільший пріоритет має вхід скиду «Reset».

До лічильників імпульсів можна віднести три таких функціональних блоки:

-  – лічильник з уставкою (PRESET COUNT, Preset up/down counter);
-  – реверсивний лічильник (Up/Down counter);



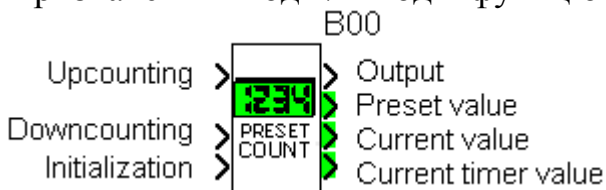
- — швидкий лічильник (Fast Counter).

Лічильники призначені для підрахунку зовнішніх імпульсів, що надходять на дискретні входи реле Zelio Logic або імпульсів, що формуються функціональними блоками програми користувача. Швидкий лічильник апаратно з'єднаний з входами I1 та I2 модуля і може рахувати тільки зовнішні імпульси. Усі три функціональні блоки лічильника є реверсивними.



Лічильник з уставкою може підраховувати як зовнішні імпульси, так і імпульси, що генеруються іншими функціональними блоками усередині програми користувача.

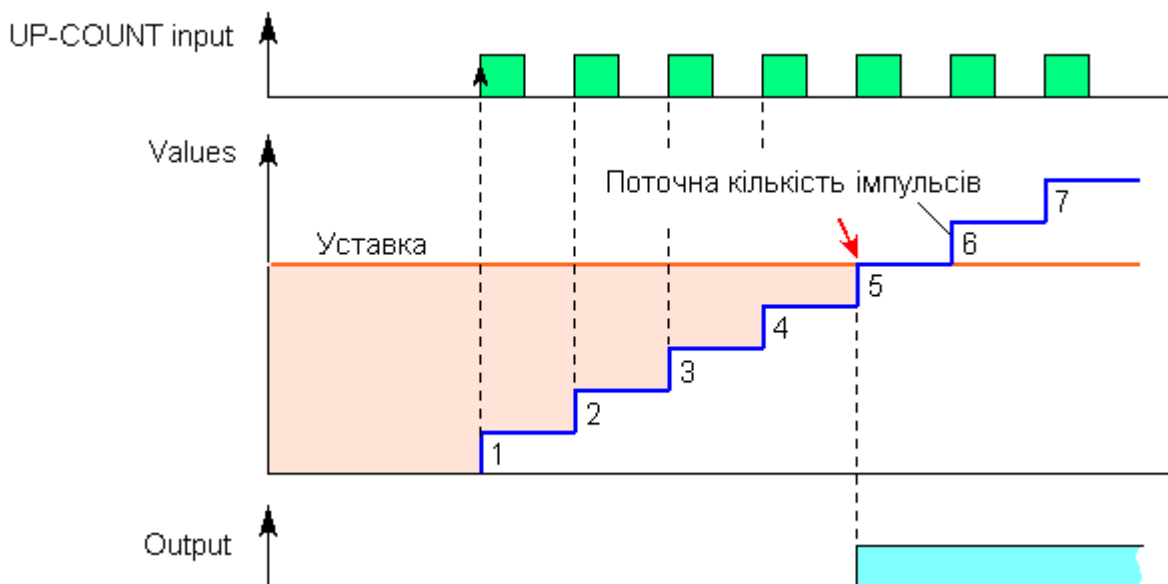
Призначення входів/виходів функціонального блока лічильника з уставкою:



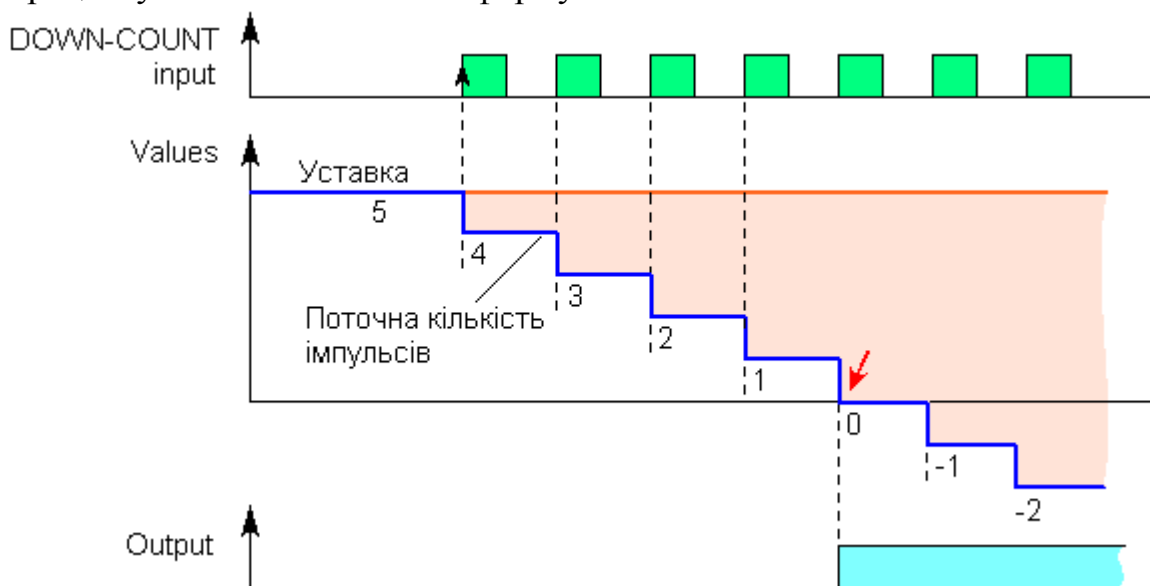
- Upcounting – лічильний вхід рахування уверх (за фронтом імпульсу);
- Downcounting – лічильний вхід рахування униз (за фронтом імпульсу);
- Initialization – ініціалізація лічильника (за рівнем сигналу);
- Output – вихід;
- Preset value – значення уставки кількості імпульсів;
- Current value – поточна кількість підрахованих імпульсів;
- Current timer value – поточна тривалість вихідного імпульсу у режимі «Repetitive».

У будь-якому режимі лічильник є реверсивним, тобто може рахувати і уверх і униз. Проте для зручності два з чотирьох режимів роботи названі «рахуванням уверх» (Up-counting), а інші два – «рахуванням униз» (Down-counting). Це пов'язано з традиційним описом роботи лічильників та таймерів.

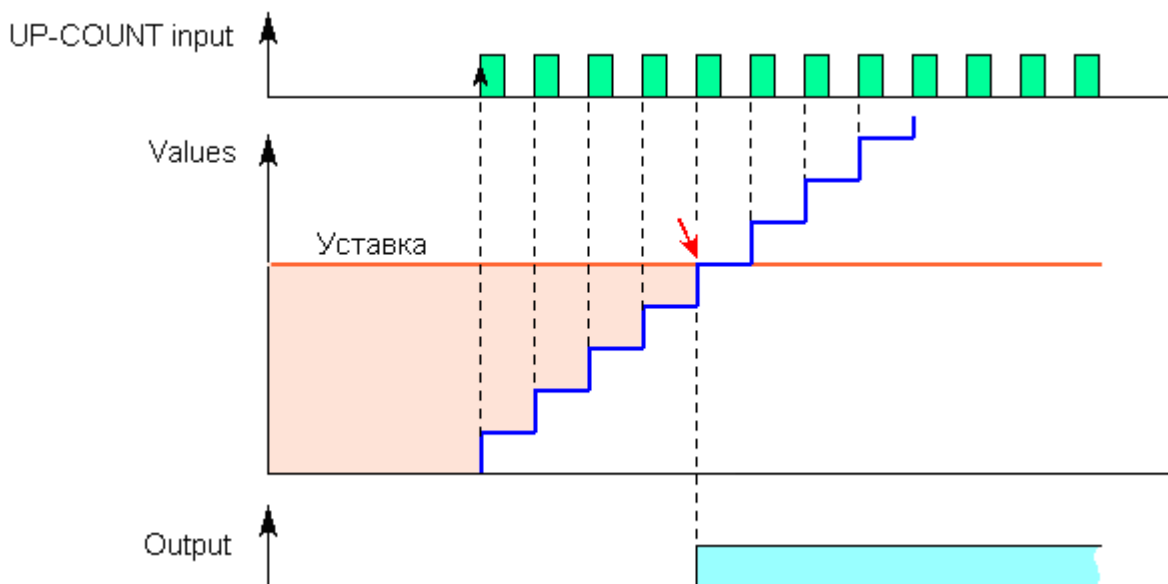
При «рахуванні уверх» зазвичай рахування починається з нуля до значення уставки. Коли поточна кількість імпульсів зрівняється (а далі й перевищить) значення уставки, вихідний контакт спрацює (замкнеться). Тобто умова спрацьовування визначається формулою «Current value $\geq$ Preset value»:



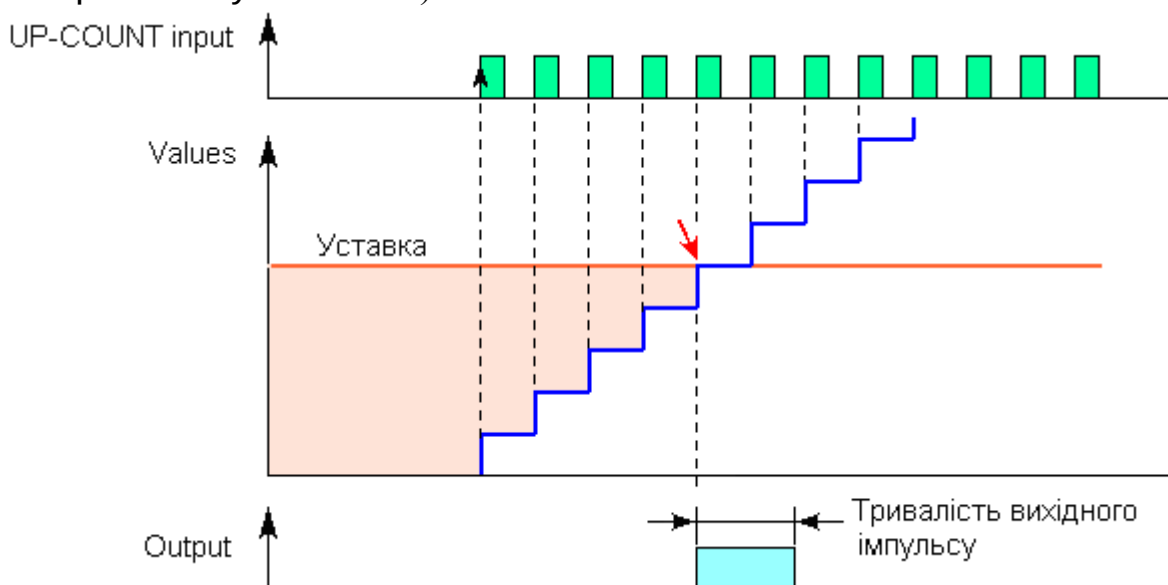
При «рахуванні униз» зазвичай рахування починається зі значення уставки до нуля. Коли поточна кількість імпульсів дійде до нуля, вихідний контакт спрацює (замкнеться). Іноді це називають «зворотний відлік». Тобто умова спрацювання визначається формулою «Current value ≤ 0 »:



Ще два варіанти функціонування лічильника з уставкою визначаються способом формування вихідного сигналу. У першому випадку вихідний сигнал активний на протязі усього часу виконання умови порівняння поточної кількості імпульсів з уставкою або з нулем (режим одного циклу, або «Single Cycle Mode»):

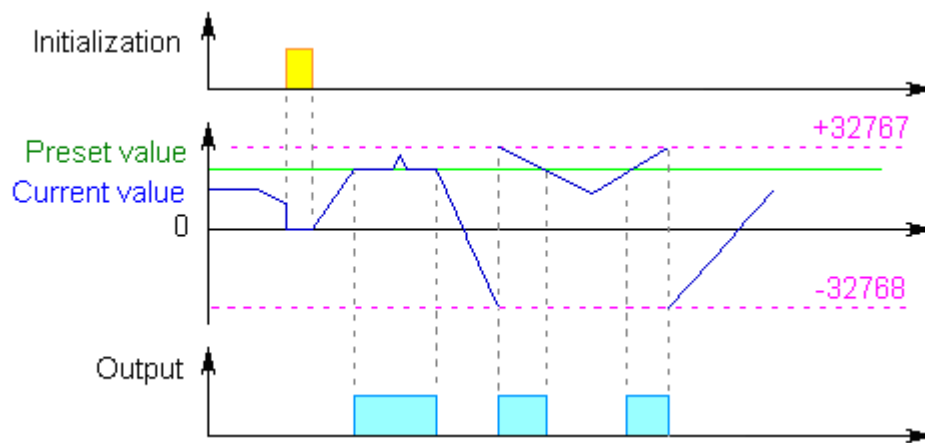


У другому випадку у момент «спрацьовування» умови порівняння на виході формується імпульс заданої тривалості (режим повторного циклу, або «Repetitive Cycle Mode»):

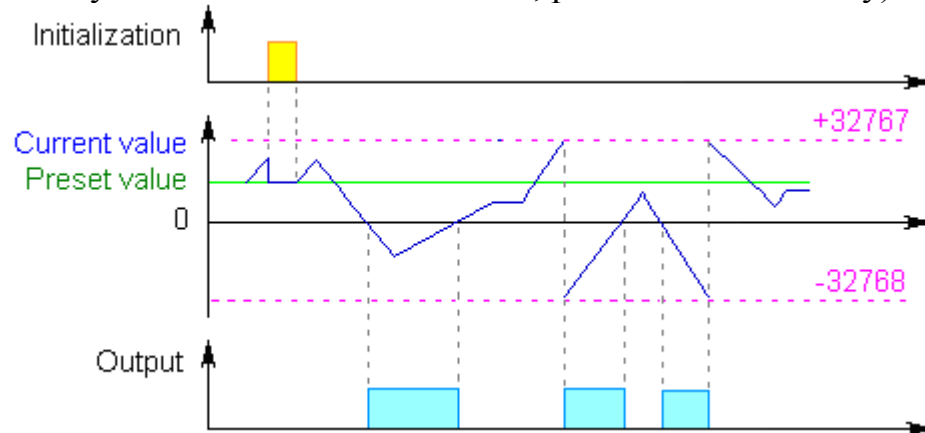


Таким чином, можливі чотири режими роботи функціонального блока лічильника з уставкою, які ілюструються подальшими діаграмами (вхідні імпульси не показані):

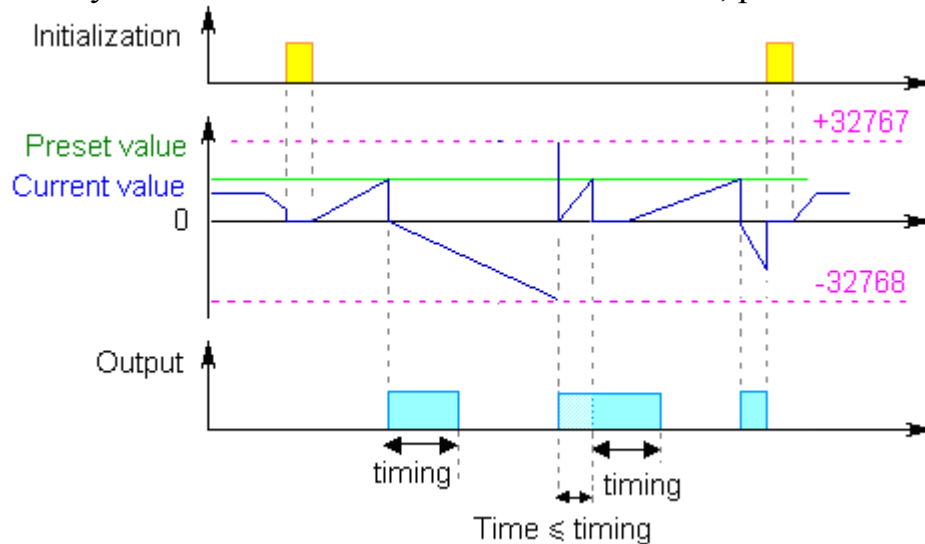
- Up-Counting Function in Single Cycle Mode (рахунок уверх, умова спрацьовування «Current value $\geq$ Preset value», режим одного циклу):



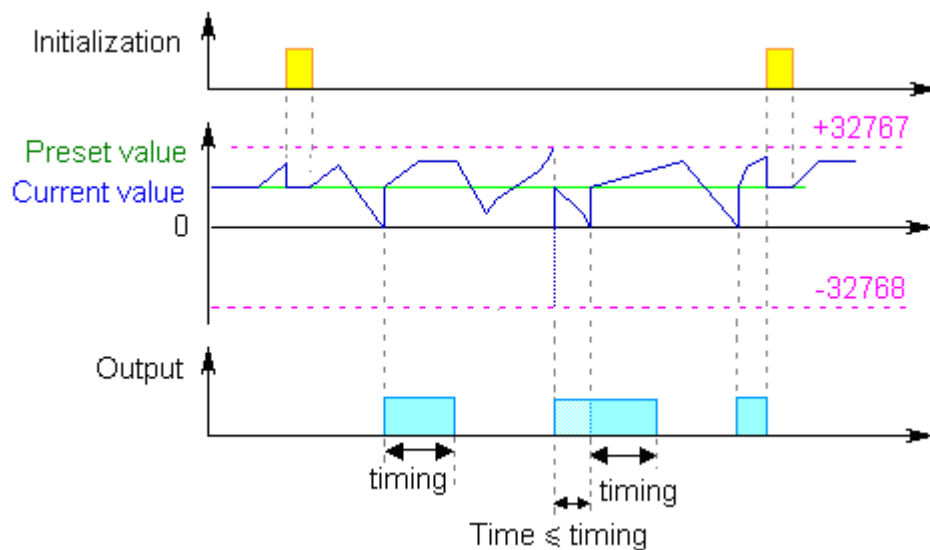
- Down-Counting Function in Single Cycle Mode (зворотний відлік, умова спрацьовування «Current value ≤ 0 », режим одного циклу):



- Up-Counting Function in Repetitive Cycle Mode (рахунок уверх, умова спрацьовування «Current value $\geq$ Preset value», режим повторного циклу):



- Down-Counting Function in Repetitive Cycle Mode (зворотний відлік, умова спрацьовування «Current value ≤ 0 », режим повторного циклу):



Як можна побачити з графіків, переповнення лічильника призводить до зміни поточного коду на максимальне протилежне за знаком значення.

Слід зауважити, що у режимі «Up-counting» сигнал ініціалізації скидає поточне значення лічильника до 0, а у режимі «Down-counting» сигнал ініціалізації записує у лічильник значення уставки замість поточного значення.

Вигляд вікна параметрів лічильника з уставкою показаний на рис. 5.15.

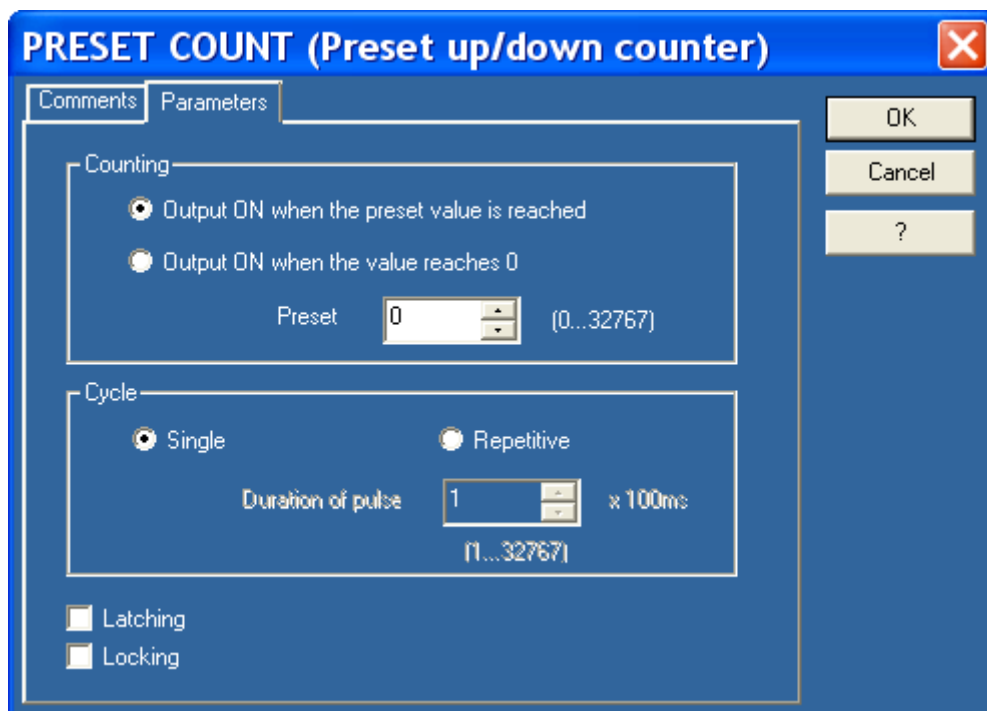


Рис. 5.15. Вікно параметрів лічильника з уставкою

У цьому вікні користувачеві доступні такі опції:

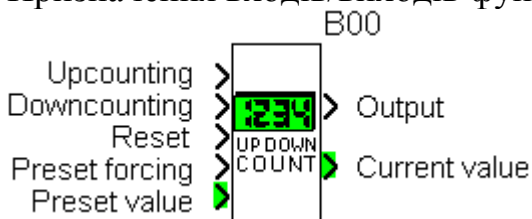
- Output ON when the preset value is reached – встановити вихідний сигнал, коли поточне значення $\geq$ значенню уставки;
- Output ON when the value is reaches 0 – встановити вихідний сигнал, коли поточне значення ≤ 0 ;

- Preset – значення уставки (0...32767 імпульсів);
- Single – режим одного циклу;
- Repetitive – режим повторного циклу із заданою тривалістю вихідного імпульсу;
 - Duration of pulse – тривалість вихідного імпульсу $(1...32767) \times 100$ мс.



Реверсивний лічильник може підраховувати як зовнішні імпульси, так і імпульси, що генеруються іншими функціональними блоками усередині програми користувача.

Призначення входів/виходів функціонального блока реверсивного лічильника:



- Upcounting – лічильний вхід рахування уверх (за фронтом імпульсу);
- Downcounting – лічильний вхід рахування униз (за фронтом імпульсу);
- Reset – скид лічильника ($\text{Current value} \leftarrow 0$);
- Preset forcing – ініціалізація лічильника ($\text{Current value} \leftarrow \text{Preset value}$);
- Preset value – значення уставки кількості імпульсів;
- Output – вихід;
- Current value – поточна кількість підрахованих імпульсів.

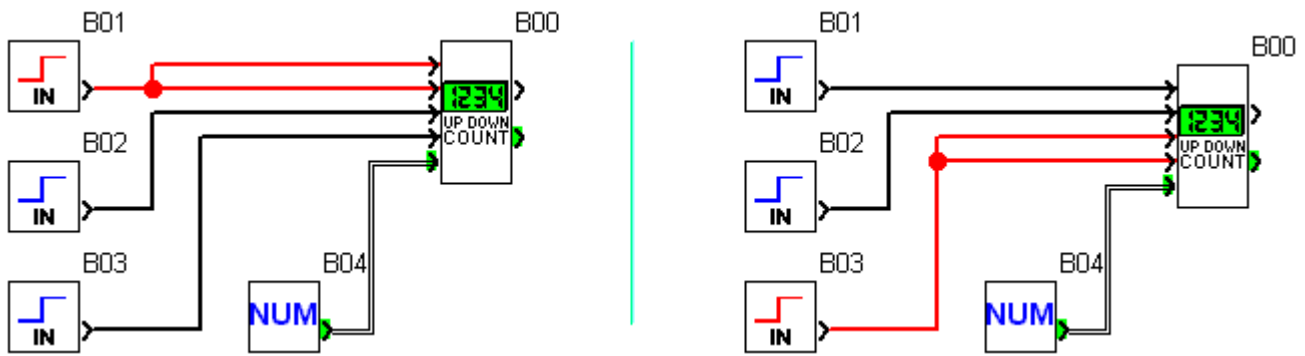
Одна з відмінностей цього реверсивного лічильника від лічильника з уставкою, який розглядався вище, полягає у тому, що значення уставки зчитується із входу «Preset value», а не встановлюється у вікні параметрів при налагодженні. Вхід «Preset value» може бути з'єднаний з блоком «NUM», аналоговим входом або з виходом цілочисельного типу якого-небудь іншого функціонального блока. При цьому значення уставки може змінюватись в процесі роботи системи.

Активний рівень на вході «Reset» утримує лічильник в нульовому стані ($\text{Output} = \text{OFF}$, $\text{Current value} = 0$).

Активний рівень на вході «Preset forcing» утримує лічильник в стані ініціалізації ($\text{Output} = \text{ON}$, $\text{Current value} = \text{Preset value}$).

Задній фронт Імпульсу на вході «Reset» або «Preset forcing» призводить до перезапуску лічильника відповідно з 0 або з числа, що подається на вхід «Preset value».

Із двох лічильних входів («Upcounting» та «Downcounting») жоден не має пріоритету. Тобто, якщо імпульси подаються на обидва входи одночасно, поточне значення лічильника не змінюється. Що стосується керуючих входів «Reset» та «Preset forcing», то пріоритетним є вхід скиду «Reset». В цьому можна упевнитися, якщо зібрати та промоделювати роботу таких схем:



Вкладка «Parameters» вікна параметрів реверсивного лічильника не має жоних опцій, окрім прапорця «Latching», значення якого відоме із попередніх матеріалів: запам'ятовування поточного коду у випадку збоїв напруги живлення.



Швидкий лічильник призначений для підрахунку імпульсів, що надходять до дискретних входів реле Zelio Logic або модулів розширення від таких пристроїв як індуктивні датчики швидкості обертання, оптичні датчики для підрахунку комплектуючих або готової продукції. У зв'язку з можливим ефектом деренчання контактів швидкий лічильник неможна використовувати для підрахунку кількості натискань кнопок з механічними контактами. також не можуть бути опрацьовані сигнали від механічних контактів реле, герконів, кінцевих вимикачів, тощо. Швидкий лічильник не має схеми визначення напрямку обертання за фазовим зсувом двох імпульсних послідовностей і тому не може бути використаний для опрацювання сигналів енкодерів.

Тож для коректної роботи швидкого лічильника необхідне обов'язкове виконання таких умов:

- відсутність деренчання контактів;
- частота входних імпульсів не перевищує 1 кГц.

У програмі можна використовувати лише один функціональний блок швидкого лічильника, оскільки він реалізується апаратними засобами, які користувач не може продублювати. Після того, як цей функціональний блок внесений до програми, його піктограма на панелі стандартних функцій стає недоступною:

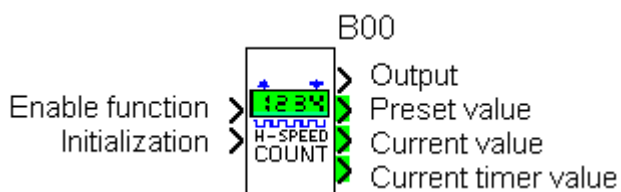


Швидкий лічильник має два лічильних входи, які апаратно з'єднані з дискретними входами I1 та I2 реле Zelio Logic:

- I1 – Fast counting – інкрементний вхід (за фронтом імпульсу);
- I2 – Fast down-counting – декрементний вхід (за фронтом імпульсу).

Ці входи є програмно-недоступними, тому роботу швидкого лічильника неможливо промоделювати. Проте за роботою цього функціонального блока можна спостерігати у режимі моніторингу.

Окрім лічильних входів функціональний блок швидкого лічильника має такі входи/виходи:



- Enable function – дозвіл рахування (за рівнем сигналу);
- Initialization – ініціалізація лічильника (за рівнем сигналу);
- Output – вихід;
- Preset value – значення уставки кількості імпульсів;
- Current value – поточна кількість підрахованих імпульсів;
- Current timer value – поточна тривалість вихідного імпульсу у режимі «Repetitive».

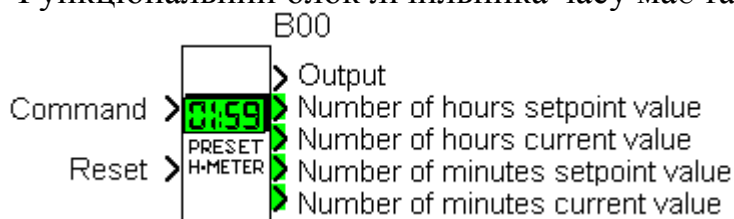
Для того, щоб швидкий лічильник міг підраховувати імпульси, на вхід дозволу рахування має подаватись сигнал високого рівня (Enable function = ON). В іншому випадку апаратні входи I1 та I2 «відсічені» від лічильника.

В усьому іншому робота швидкого лічильника збігається з роботою лічильника з уставкою (часові діаграми, конфігурація, налагодження, вигляд та опції вікна параметрів, тощо).



Лічильник часу вимірює тривалість інтервалу, на протязі якого на його вхід подається активний сигнал. Тому на базі цього функціонального блока можна створити, наприклад, лічильник часу напрацювання деякого зовнішнього пристрою (двигуна, лампи освітлення, тощо). Також цей функціональний блок використовується для створення тривалих затримок часу (до 32767 годин та 59 хвилин).

Функціональний блок лічильника часу має такі входи/виходи:



- Command – дозвіл рахування (за рівнем «ON» сигналу);
- Reset – вхід скиду;
- Output – вихід;
- Number of hours setpoint value – значення уставки (показчик годин);
- Number of hours current value – поточний час напрацювання (показчик годин);
- Number of minutes setpoint value – значення уставки (показчик хвилин);
- Number of minutes current value – поточний час напрацювання (показчик хвилин).

Дискретний вихід лічильника є активним, якщо нарахований час дорівнює або перевищує уставку ($\text{Current value} \geq \text{Setpoint value}$).

Активний рівень на вході «Reset» утримує лічильник часу в нульовому стані (Output = OFF, Current values = 0). Вхід «Reset» має пріоритет над входом «Command».

Бажаний час уставки можна задати у вікні параметрів лічильника часу (рис. 5.16).

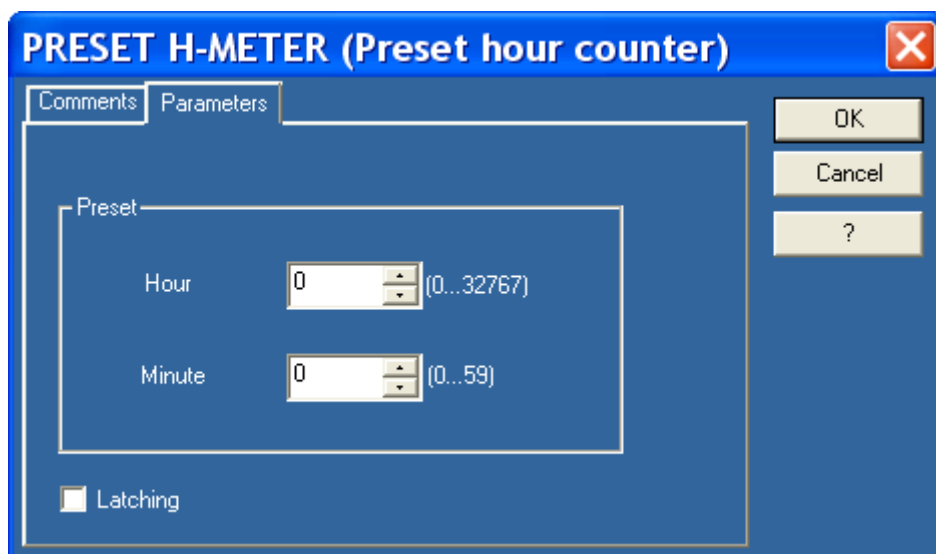


Рис. 5.16. Вікно параметрів лічильника часу



Годинник-програмувач дозволяє програмувати виконання певних дій у довільно задані моменти часу з точністю до однієї хвилини (у відповідності з системним часом).

Події можуть програмуватися за наступними схемами:

- Annual – щорічно (один раз на рік), задається час, дата та місяць року;
- Monthly – щомісячно (один раз на місяць), задається час та дата;
- Periodic – періодично:
 - Weekly – щотижнево, задається час та дні тижня;
 - Daily – щоденно, задається час та номер тижня;
 - довільно у заданий час заданих днів тижня визначених тижнів;
- Date – одноразово, задається час, дата, місяць та рік події.

Задати схему, час та акцію (ON/OFF) події можна у вікні параметрів годинника-програмувача (рис. 5.17).

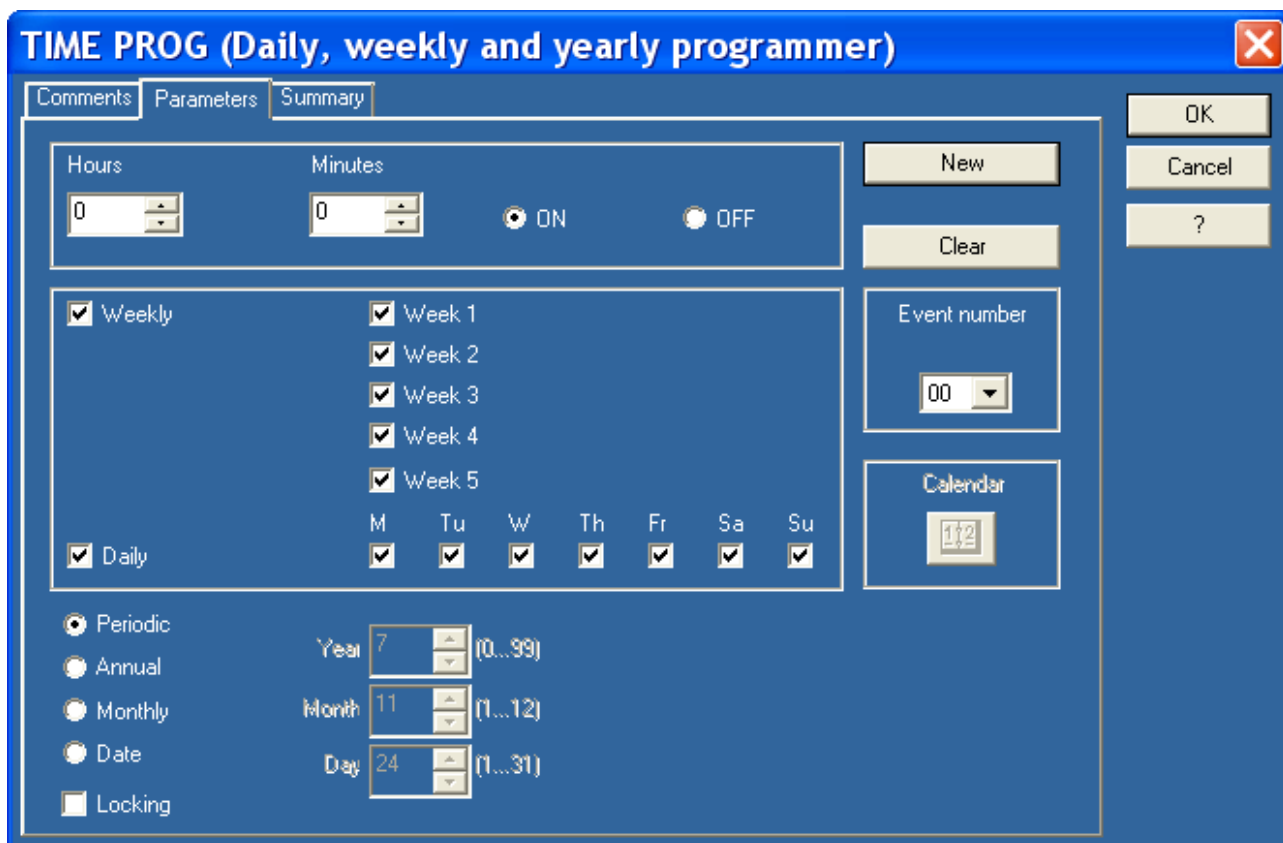


Рис. 5.17. Вікно параметрів годинника-програмувача

Окрім вище зазначених параметрів, що задають схему програмування, у вікні параметрів є такі опції:

- New – створити подію;
- New – стерти подію;
- Event number – номер події;
- Hours – уставка часу спрацьовування (показчик годин);
- Minutes – уставка часу спрацьовування (показчик хвилин);
- ON – активація виходу при спрацьовуванні;
- OFF – дезактивація виходу при спрацьовуванні;
- Week 1, Week 2, ..., Week 5 – номер тижня;
- M(onday), Tu(esday), ..., Su(nday) – день тижня;
- Calendar – вибрати дату події у календарі;
- Year, Month, Day – рік, місяць та дата настання події.

Програмування даного функціонального блока полягає у послідовному виконанні таких дій:

- Створити нову подію (натиснути кнопку «New»);
- Вибрати акцію події (натиснути радіокнопку «ON» або «OFF»);
- Задати схему роботи (щорічно, щомісячно, щотижнево, ..., одноразово);
- Встановити час настання події;
- На вкладці «Зведення» (Summary) вікна параметрів перевірити послідовність подій (рис. 5.18).

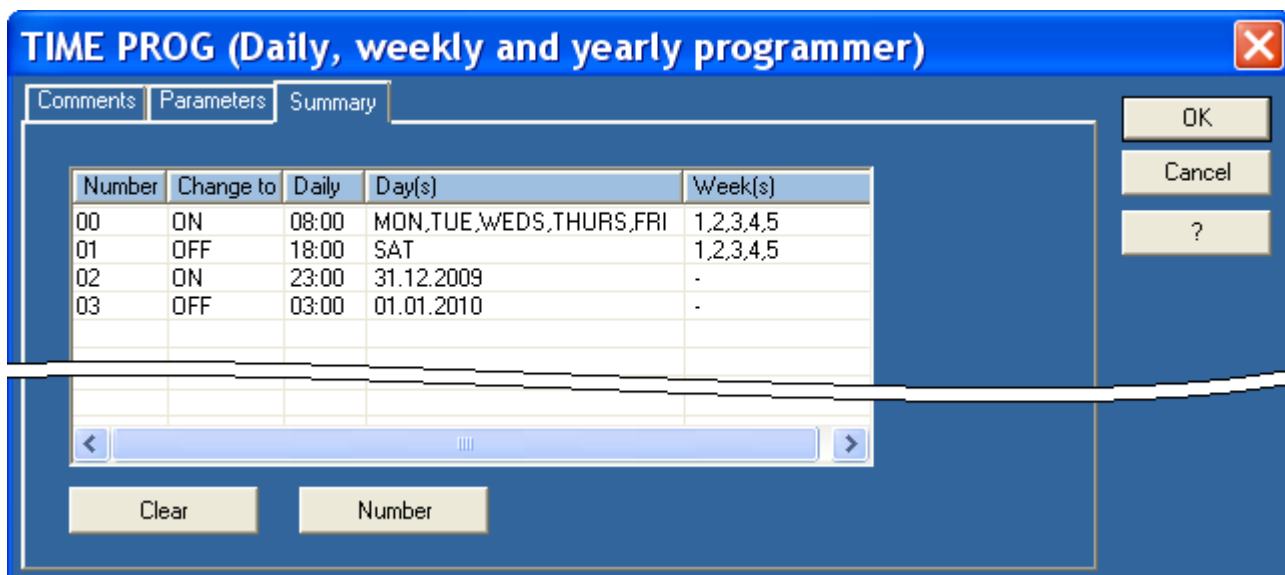


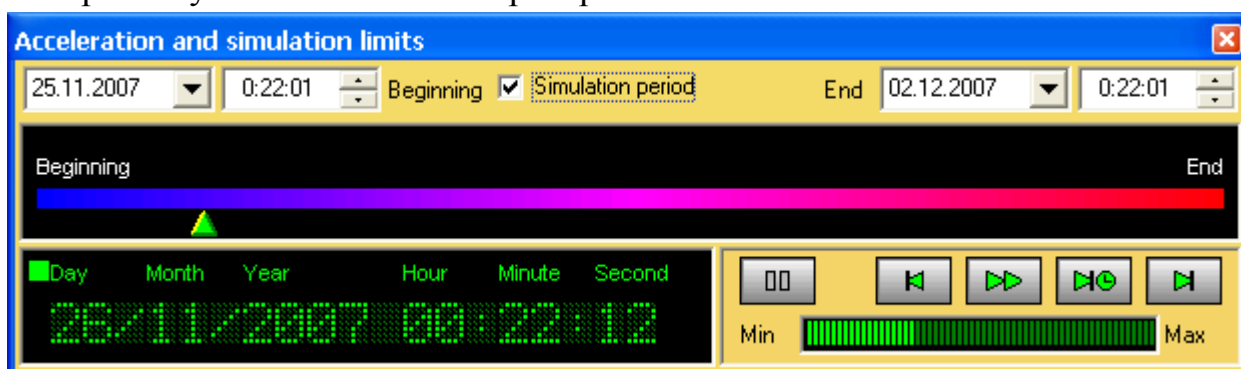
Рис. 5.18. Вкладка зведень у вікні параметрів годинника-програмувача

Якщо параметри якої-небудь події необхідно змінити, слід подвійним кліком покажчика миші на відповідному рядку таблиці (рис. 5.18) перейти до редагування цієї події (при цьому Ви автоматично перейдете на вкладку «Parameters»). Якщо ж Ви хочете змінити порядковий номер події, слід виділити відповідний рядок таблиці, натиснути кнопку «Number» та ввести бажаний номер.



Увага! Порядок настання подій визначається схемою та заданим часом кожної з них, а не їх порядковими номерами.

Нагадуємо, що для більшої зручності моделювання довготривалих процесів слід використовувати панель акселератора:



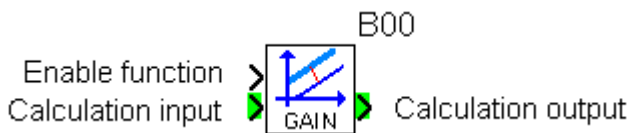
Слід зазначити, що параметри годинника-програмувача не можуть бути змінені у режимах моделювання та моніторингу, а лише у режимі редагування програми.



Підсилювач – це функціональний блок, який виконує масштабування та зміщення аналогового значення (з обмеженням результату) за формулою:

$$\text{Calculation output} = A / B \times \text{Calculation input} + C$$

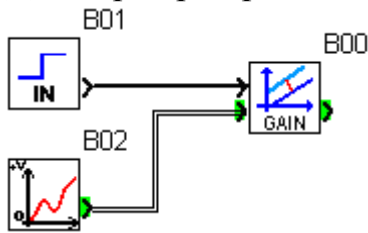
Призначення входів/виходів підсилювача:



- Enable function – дозвіл роботи (за рівнем сигналу);
- Calculation input – обчислювальний вхід (цілочисельний операнд);
- Calculation output – обчислювальний вихід (результат).

На обчислювальний вхід підсилювача можна подавати сигнал, що змінюється у межах $-32768 \dots +32767$. Результат обчислень також ні в якому разі не вийде за межі змінної цілочисельного типу ($-32768 \dots +32767$).

Для перевірки роботи підсилювача слід зібрати таку схему:



Якщо сигнал дозволу роботи не надходить (рівень «OFF»), вихідне значення підсилювача не змінюється. Якщо ж сигнал дозволу активний, постійно (один раз кожного обчислювального циклу) здійснюється перерахунок вихідного значення.

Задати коефіцієнти підсилення, величину зміщення та обмеження вихідного сигналу можна у вікні параметрів підсилювача (рис. 5.19)

Рис. 5.19. Вікно параметрів підсилювача

У цьому вікні користувачеві доступні такі опції:

- Gain numerator (A) – чисельник коефіцієнта підсилення;
- Gain denominator (B) – знаменник коефіцієнта підсилення ($B \neq 0$);

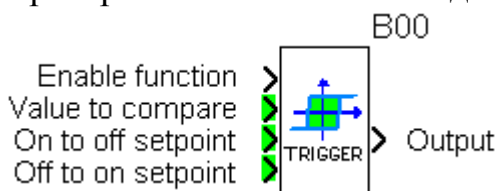
- Offset (C) – величина зміщення;
- Range – діапазон вихідного сигналу:
 - Upper limit – верхнє обмеження;
 - Lower limit – нижнє обмеження.

За умовчанням у вікні параметрів встановлюються значення, які передають вхідний сигнал на вихід без змін (див. рис. 5.19).



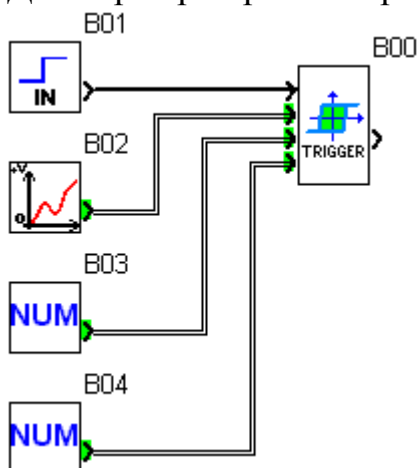
Тригер Шмітта – це функціональний блок, який реалізує функцію компаратора з петлею гістерезиса. Його можна застосовувати, наприклад, для виключення ефекту деренчання при увімкненні нагрівачів або ламп освітлення у релейних системах із зворотнім зв'язком за температурою або рівнем освітленості (коли величина зворотного зв'язку коливається навколо рівня завдання).

Тригер Шмітта має такі входи/виходи:



- Enable function – дозвіл роботи (за рівнем сигналу);
- Value to compare – значення для порівняння (вхідний сигнал);
- On to off setpoint – нижня уставка вхідного сигналу (точка вимикання);
- Off to on setpoint – верхня уставка вхідного сигналу (точка увімкнення);
- Output – вихід.

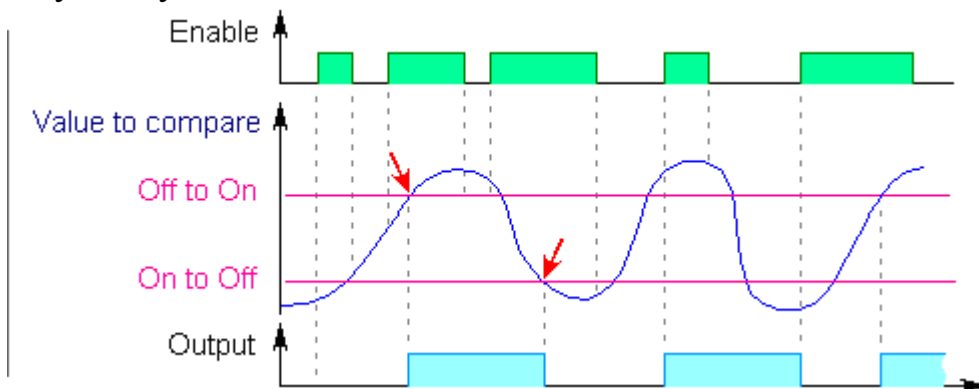
Для перевірки роботи тригера Шмітта можна зібрати таку схему:



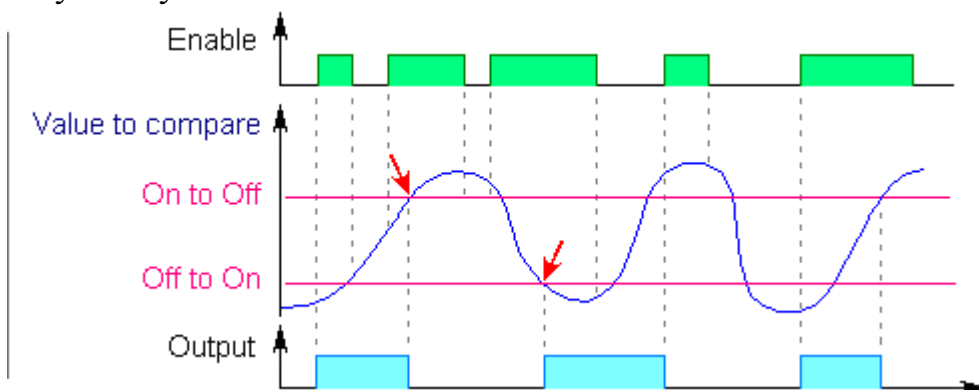
Якщо сигнал дозволу роботи не надходить (рівень «OFF»), вихідне значення тригера не змінюється. Якщо ж сигнал дозволу активний, постійно (один раз кожного обчислювального циклу) здійснюється модифікація вихідного сигналу.

Можливі два протилежні випадки співвідношення нижньої та верхньої уставок тригера, що призводить до інверсної осцилограми вихідного сигналу (зверніть увагу на графіки сигналу «Output»):

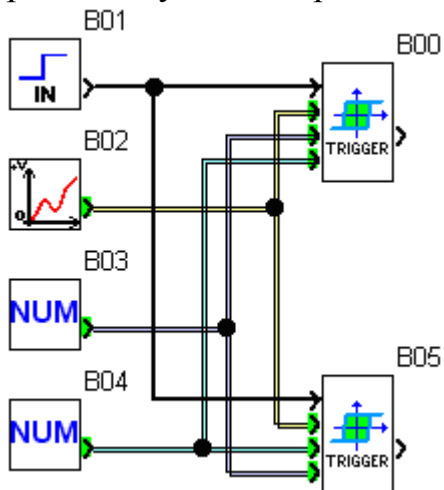
- Off to On setpoint > On to Off setpoint – уставка увімкнення більша за уставку вимикання:



- Off to On setpoint < On to Off setpoint – уставка увімкнення менша за уставку вимикання:



Для перевірки роботи тригера при протилежних значеннях уставок перемикавання рекомендується зібрати та дослідити роботу такої схеми:

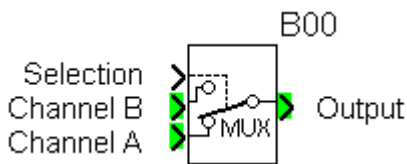


При моделюванні роботи схеми зверніть увагу на те, що вихідні сигнали двох тригерів постійно протилежні.



Мультиплексор передає на вихід сигнал з одного із двох вхідних каналів. Вибір вхідного каналу здійснюється сигналом вибірки.

Призначення входів/виходів мультиплексора:



- Selection – сигнал вибірки (вибір вхідного каналу);
- Channel B – вхідний канал B;
- Channel A – вхідний канал A;
- Output – вихід.

Комутація вхідних каналів виконується за рівнем сигналу вибірки. Якщо:

- Selection = OFF – канал A підключається до виходу;
- Selection = ON – канал B підключається до виходу;

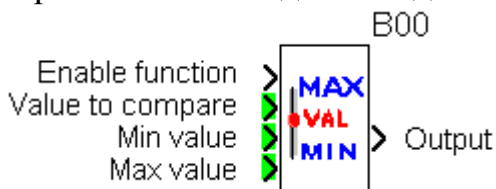
Якщо жоден з вхідних каналних входів не підключений, або не підключений вхід вибраного каналу, на вихід надходить число 0.

До компараторів відносяться такі функціональні блоки:

-  – зонний компаратор (COMP IN ZONE Comparison);
-  – аналоговий компаратор для порівняння двох значень (COMPARE Function for Comparing Two Analog Values).

 Зонний компаратор виявляє, чи належить значення вхідного сигналу до певного діапазону.

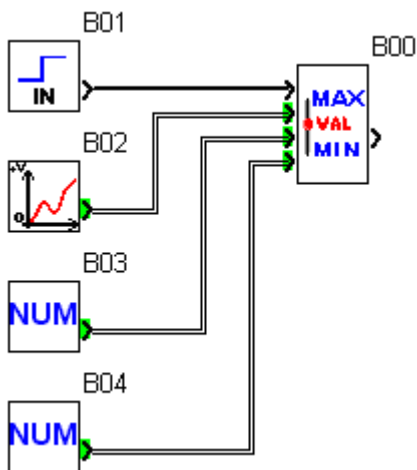
Призначення входів/виходів зонного компаратора:



- Enable function – дозвіл роботи (за рівнем сигналу);
- Value to compare – значення для порівняння (вхідний сигнал);
- Min value – нижня межа діапазону (включно);
- Max value – верхня межа діапазону (включно);
- Output – вихід.

Якщо сигнал дозволу роботи неактивний (Enable function = OFF), стан виходу компаратора не змінюється. В іншому випадку значення вихідного сигналу визначається належністю рівня вхідного значення до заданого діапазону та конфігурацією компаратора.

Для перевірки роботи зонного компаратора можна зібрати таку схему:



Логіку роботи зонного компаратора (пряму чи інверсну) можна задати у вікні параметрів (рис. 5.20):

- On in the zone – вихід активний, якщо вхідне значення належить діапазону (пряма логіка);
- Off in the zone – вихід пасивний, якщо вхідне значення належить діапазону (інверсна логіка).

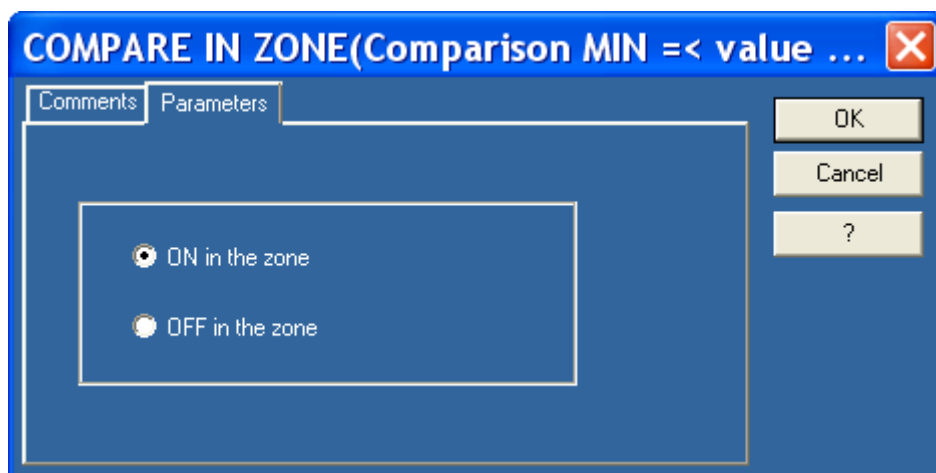
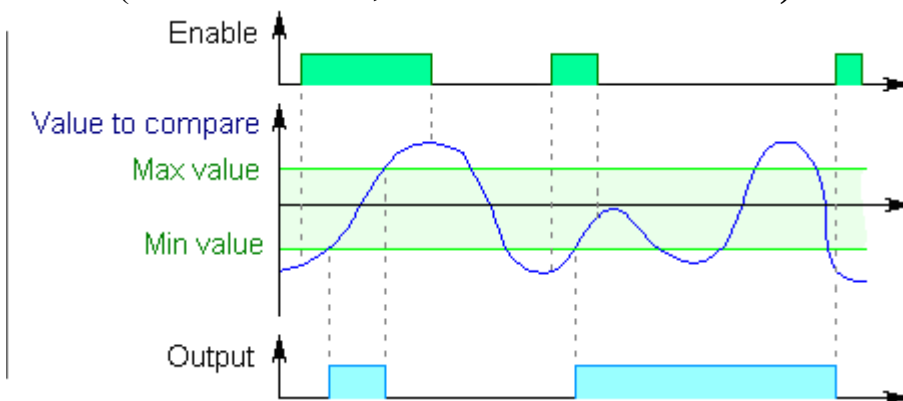


Рис. 5.20. Вікно параметрів зонного компаратора

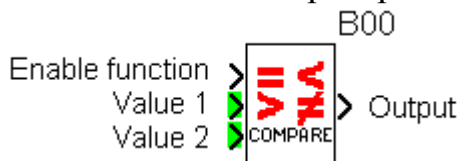
Подальша діаграма ілюструє роботу зонного компаратора у режимі прямої логіки (On in the zone; Min value < Max value):





Аналоговий компаратор здійснює порівняння двох цілочисельних значень за такими операторами: $>$, $\geq$, $=$, $\neq$, $\leq$ та $<$.

Аналоговий компаратор має такі входи/виходи:



- Enable function – дозвіл роботи (за рівнем сигналу);
- Value 1, Value 2 – вхідні значення для порівняння.

Вихідний сигнал компаратора встановлюється в активний стан, якщо логічний результат порівняння дорівнює «True» (істина). Інакше вихід компаратора знаходиться у пасивному стані.

Якщо сигнал дозволу роботи не надходить (рівень «OFF»), вихідне значення компаратора не змінюється.

Вибрати оператор порівняння для логічного виразу можна у вікні параметрів аналогового компаратора (рис. 5.21).

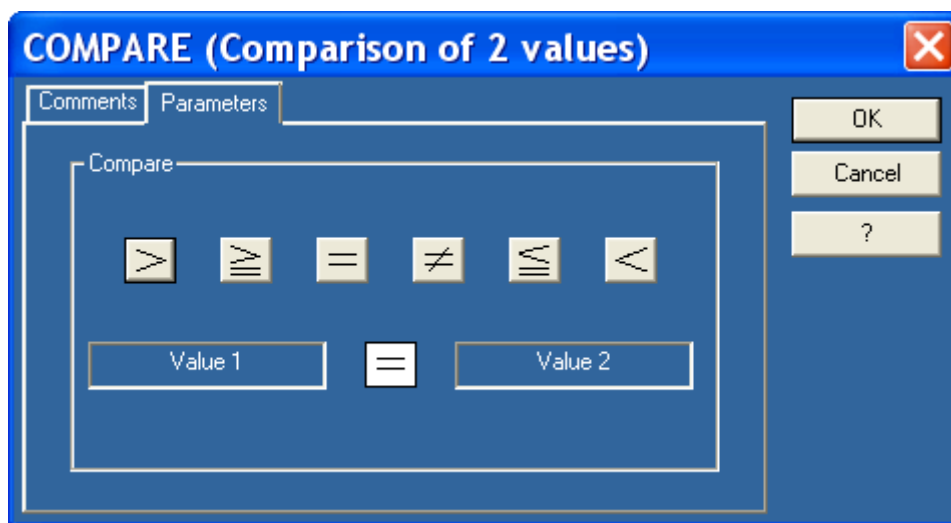


Рис. 5.21. Вікно параметрів аналогового компаратора

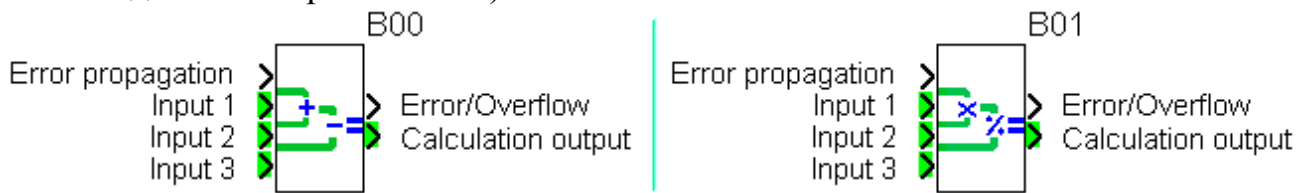
Якщо вхід «Value 1» або «Value 2» не підключений, його значення дорівнюватиме 0.

Арифметичні функції виконують такі функціональні блоки:

- – суматор (ADD/SUB Arithmetic Function);
- – помножувач (MUL/DIV Arithmetic Function).

Перший з цих блоків виконує додавання та (або) віднімання, а другий – множення та (або) ділення. Обидва блоки виконують обчислення над трьома

цілочисельними операндами, що надходять на входи (входи/виходи блоків мають ідентичне призначення):



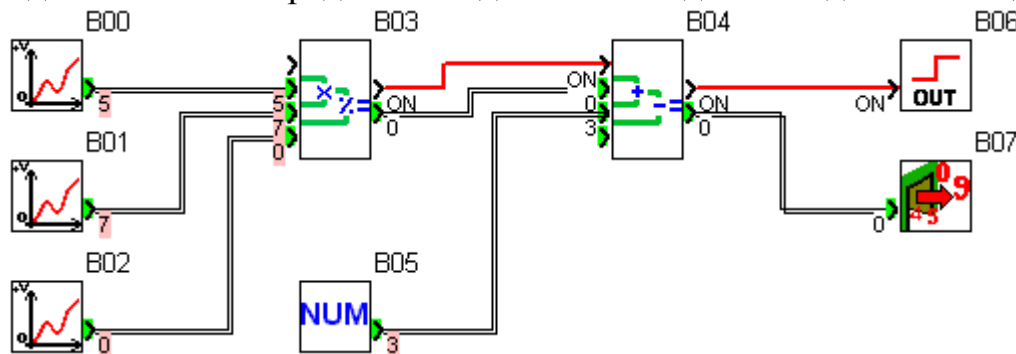
- Error propagation – вхід сигналу передачі помилки попереднього етапу обчислень;
- Input 1, Input 2, Input 3 – вхідні значення операндів;
- Error/Overflow – вихід помилки або переповнення;
- Calculation output – вихід результату обчислень.

Як можна побачити з піктограм функціональних блоків, обчислення виконуються за такими формулами:

$$\text{Calculation output} = \text{Input 1} + \text{Input 2} - \text{Input 3}$$

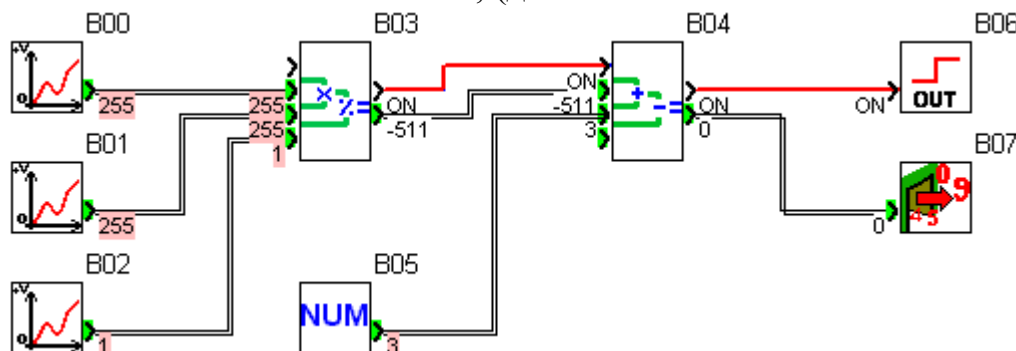
$$\text{Calculation output} = \text{Input 1} \times \text{Input 2} / \text{Input 3}$$

Вхід «Error propagation» використовується при каскадному з'єднанні арифметичних блоків для передачі на вихід сигналу помилки від попередніх каскадів. Якщо на цей вхід блока подати активний сигнал, він продублюється на виході «Error/Overflow», а обчислювальний вихід «Calculation output» скинеться у 0. На подальшому рисунку можна побачити, як сигнал помилки «ділення на 0» передається від блока B03 до B04 і далі на вихід:



На подальшому рисунку показано, як у першому арифметичному блоці на виході «Error/Overflow» формується сигнал переповнення, оскільки результат множення виходить за межі діапазону цілочисельної змінної:

$$255 \times 255 / 1 = 65025 > 32767; \text{ (для кмітливих: } 32768 \times 2 - 511 = 65025 \text{):}$$



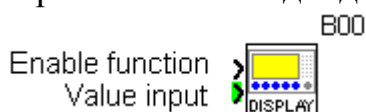
Таким чином, вихід помилки/переповнення (Error/Overflow) встановлюється в активний стан у таких випадках:

- результат обчислень вийшов за межі інтервалу $-32767...32768$ змінної цілочисельного типу (переповнення);
- подільник блока MUL/DIV дорівнює 0 (ділення на 0);
- на вхід «Error propagation» поданий активний сигнал (передача помилки від попереднього блока).



Дисплей – це функціональний блок, який використовується для виводу різноманітних текстових повідомлень на екран LCD-дисплея (індикатора) фронтальної панелі модуля.

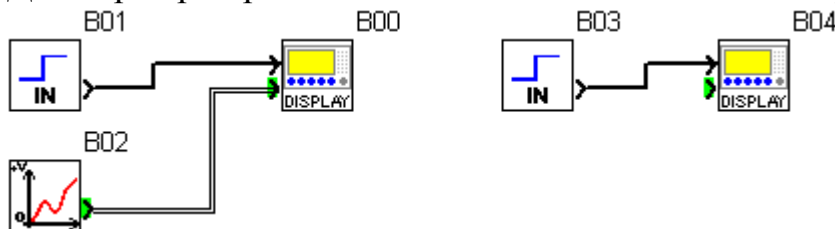
Призначення входів дисплейного функціонального блока:



- **Enable function** – дозвіл роботи (за рівнем сигналу);
- **Value input** – вхідне значення для індикації.

У програмі можна використовувати кілька дисплейних блоків (до 32-х), кожен з яких здатен виводити тільки одне повідомлення. Кілька повідомлень можна висвітлити одночасно у різних позиціях дисплея. Числові параметри повідомлень можуть змінювати свої значення лише за активного рівня сигналу дозволу роботи (Enable function = On).

Для перевірки роботи дисплейного блока можна зібрати таку схему:



Якщо вхід «Value input» підключений, дисплейний блок використовується для індикації числового значення, що надходить до цього входу. Вікно параметрів функціонального блока має при цьому такий вигляд (рис. 5.22):

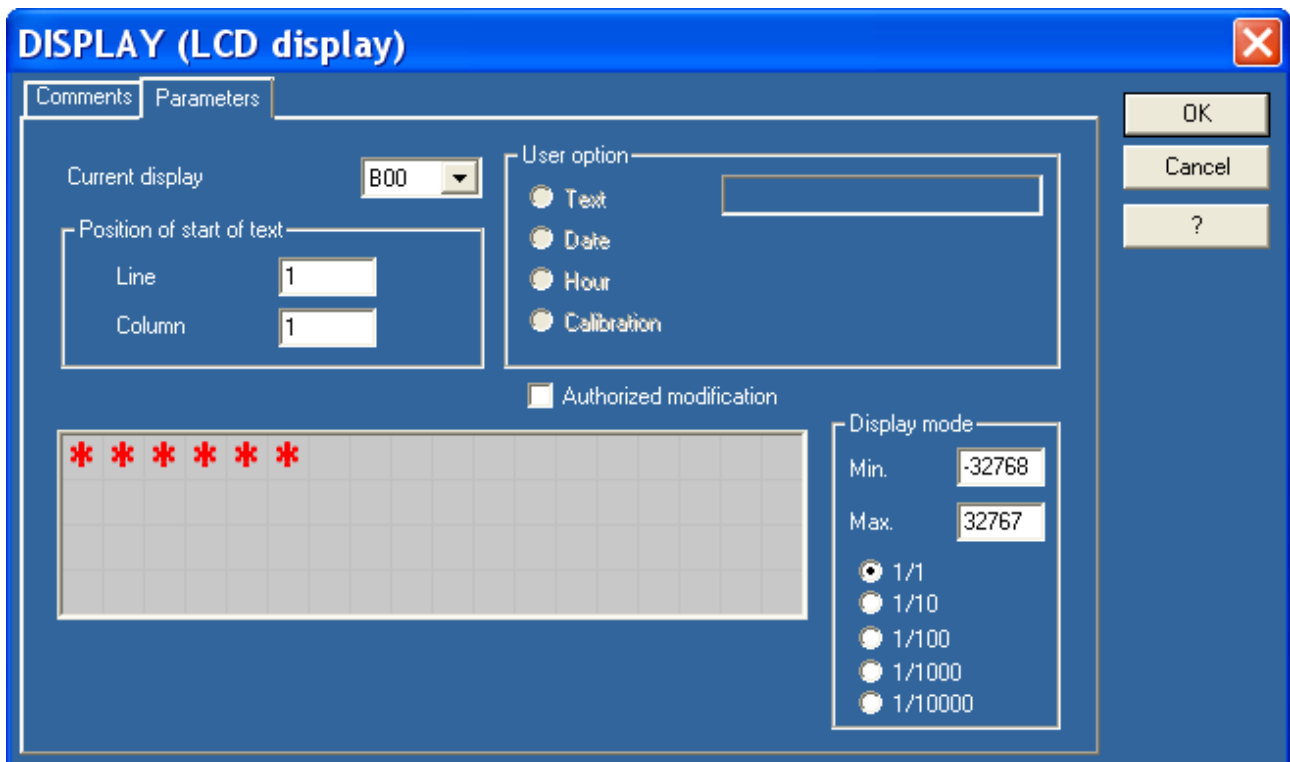


Рис. 5.22. Вікно параметрів дисплейного блоку у режимі виводу значення, що надходить до входу «Value input»

Вікно параметрів дисплейного блоку має такі активні опції:

- **Current display** – номер функціонального блоку;
- **Position of start of text** – позиція початкового символу повідомлення:
 - **Line** – номер рядка дисплея (1...4);
 - **Column** – позиція символу в рядку або номер колонки (1...18);
- **Authorized modification** – дозвіл зміни значення за допомогою кнопок Zx фронтальної панелі модуля;
- **Display mode** – режим відображення:
 - **Min.** – мінімальне значення;
 - **Max.** – максимальне значення;
 - 1/10, 1/10, ..., 1/10000 – формат подання числа, що визначає кількість знаків після десяткової коми:

1/1 –	* * * * *	;
1/10 –	* * * * * . *	;
1/100 –	* * * * . * *	;
...		
1/10000 –	* * . * * * *	.

Якщо вхід «Value input» не підключений, то за допомогою такого дисплейного блоку можна вивести будь-яке текстове повідомлення довжиною не більше 72 символів, або ж поточну дату, час чи калібровочну константу. При цьому вікно параметрів має вигляд, що наведений на рис. 5.23.

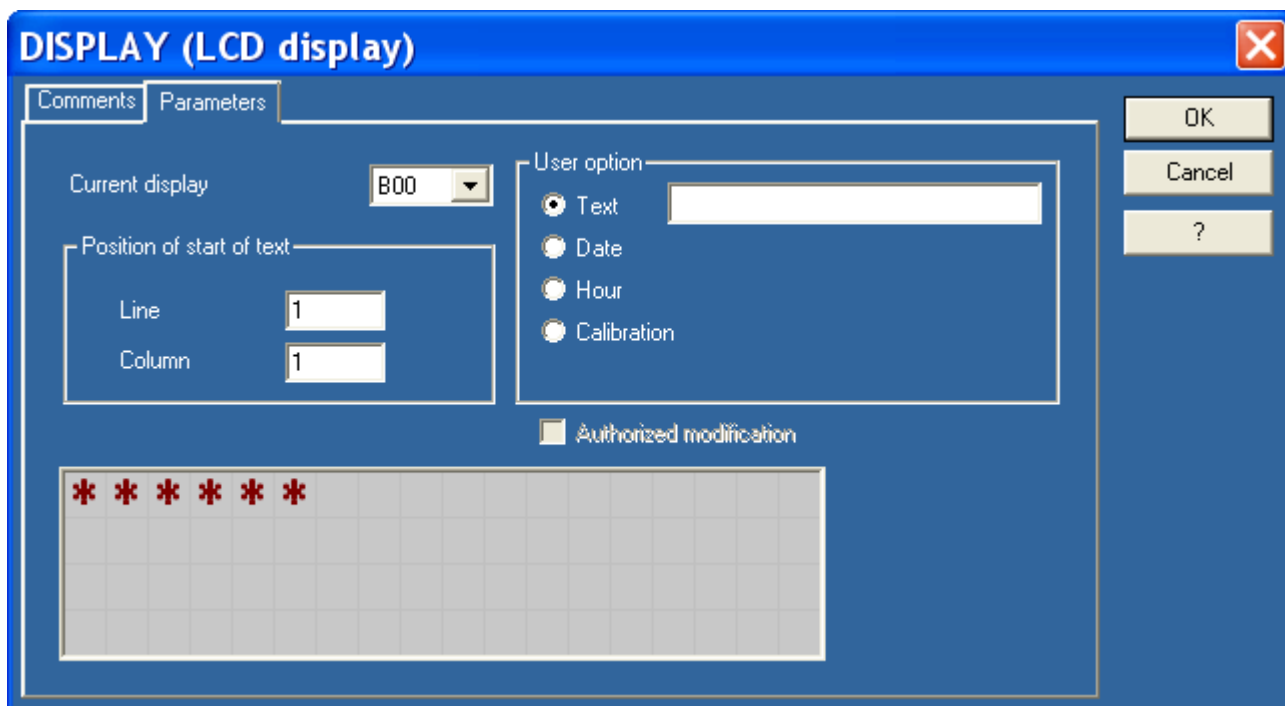
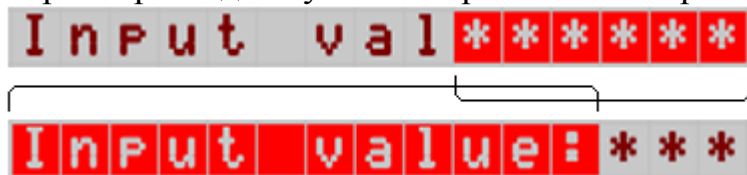


Рис. 5.23. Вікно параметрів дисплейного блока

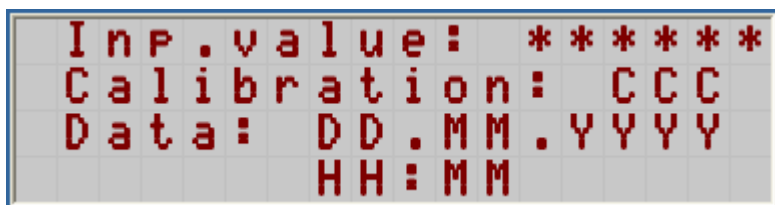
Окрім опцій, що розглянуті вище, на рис. 5.23 також позначені:

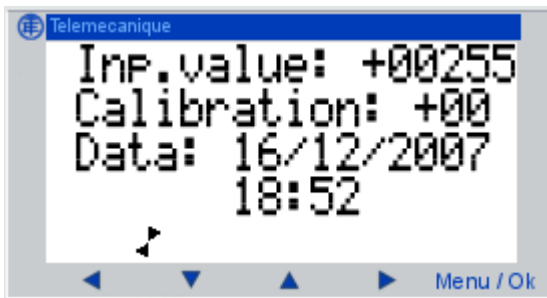
- User option – вибір повідомлення для індикації:
 - Text – текстовий блок **Hello, world!** ;
 - Date – поточна календарна дата **DD.MM.YYYY** ;
 - Hour – поточний час **HH:MM** ;
 - Calibration – калібровочна константа **CCC** .

Якщо поля дисплейних блоків перетинаються, відповідні надписи у вікні параметрів підсвічуються червоним кольором:



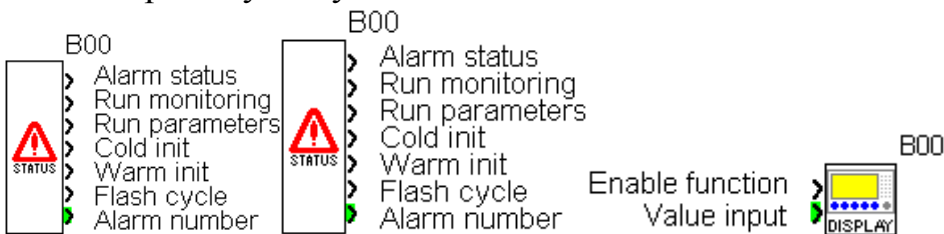
Можливий варіант оформлення дисплея, та вигляд фронтальної панелі у режимі роботи показані на рисунках:





Індикатор стану модуля генерує сигнали, що пов'язані з апаратним станом модуля Zelio Logic. Блок дозволяє отримати інформацію про стан модуля та змінювати у разі необхідності поведінку програми користувача відповідно до цього стану.

Індикатор стану модуля має тільки виходи:



- Alarm status – сигнал тревоги;
- Run monitoring – режим моніторингу;
- Run parameters – зміна параметрів;
- Cold init – холодний старт;
- Warm init – теплий старт;
- Flash cycle – новий цикл (Scan);
- Alarm number – номер помилки.

Сигнал тревоги (Alarm) сповіщає про виникнення некритичної помилки (Warning – попередження) або критичної помилки чи несправності (Error – помилка). У разі виникнення некритичної помилки модуль продовжує функціонувати (виконувати програму користувача). При цьому номер попередження може бути зчитаний з цілочисельного виходу «Alarm number» і програмно оброблений програмою користувача. У разі виникнення помилки або несправності модуль зупиняє виконання програми користувача, при цьому функціональний блок стану модуля також перестає функціонувати. При цьому номер помилки можна дізнатися з лицевої панелі модуля за допомогою команди меню «FAULT».

Єдиний спосіб скинути вихід «Alarm status» в неактивний стан та скинути в нуль номер помилки на виході «Alarm number» – це використати команди меню «FAULT > CLEAR > YES» за допомогою лицевої панелі модуля.

Сигнал «Run monitoring» активний тоді, коли програма користувача коректно виконується і контролер знаходиться у режимі моніторингу (тобто у оболонці Zelio Soft відображується внутрішній стан контролера).

На виході «Run parameters» генерується імпульс тривалістю один машинний цикл тоді, коли при виконанні програми користувача вступають в дію зміни параметрів, що внесені з лицевої панелі модуля або з приєднаного персонального комп'ютера.

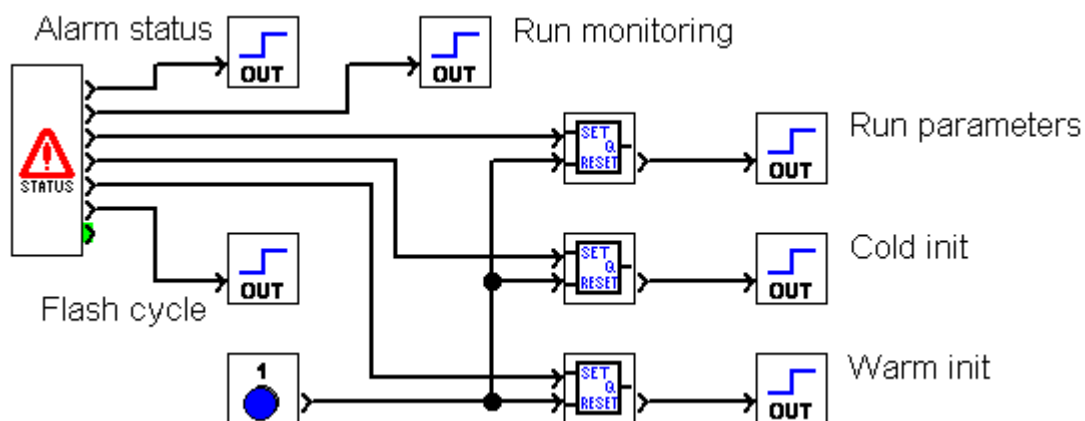
На виході «Cold init» генерується імпульс впродовж першого машинного циклу (Scan) виконання програми користувача при переході з режиму «STOP» у режим «RUN».

На виході «Warm init» генерується імпульс впродовж першого машинного циклу (Scan) виконання програми користувача при запуску після відновлення живлення, якщо до припинення живлення контролер знаходився у стані «RUN».

На виході «Flash cycle» при виконанні програми користувача генерується періодичний сигнал, який змінює свій стан у кожному машинному циклі (Scan). Період цього сигналу дорівнює подвійній тривалості машинного циклу.

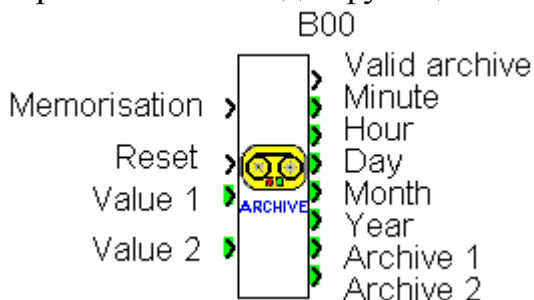
Перелік помилок та попереджень наведений у додатку 1.

Для перевірки роботи блока стану модуля можна зібрати таку схему:



Архів дає можливість у довільний момент часу записати два цілочисельних значення а також виконати датування цього запису (зберегти дату та час здійснення запису до архіву)

Призначення входів функціонального блока архіву:



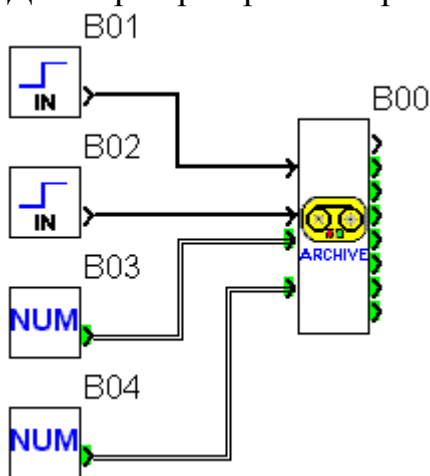
- Memorisation – запис до архіву;

- Reset – скид;
- Value 1, Value 2 – значення, які записуються до архіву;
- Valid archive – актуальність даних архіву;
- Minute ... Year – час запису до архіву:
 - Minute – хвилини,
 - Hour – години,
 - Day – дата,
 - Month – місяць,
 - Year – рік;
- Value 1, Value 2 – архівні дані (значення, що були записані до архіву).

Запис двох цілочисельних значень (Value 1 та Value 2) до архіву виконується за фронтом імпульсу на вході «Memorisation». При цьому: активується вихід актуальності даних «Valid archive», записані дані з'являються на виходах «Value 1» та «Value 2», а також фіксуються поточний час та дата архівації.

Якщо подати сигнал на вхід «RESET», вихід актуальності даних «Valid archive» буде скинутий в нульовий стан, при цьому з архіву не зникнуть ані записані дані, ані інформація датування запису. Дані в архіві та інформація датування оновлюються лише процедурою нового запису до архіву.

Для перевірки роботи архіву можна зібрати таку схему:



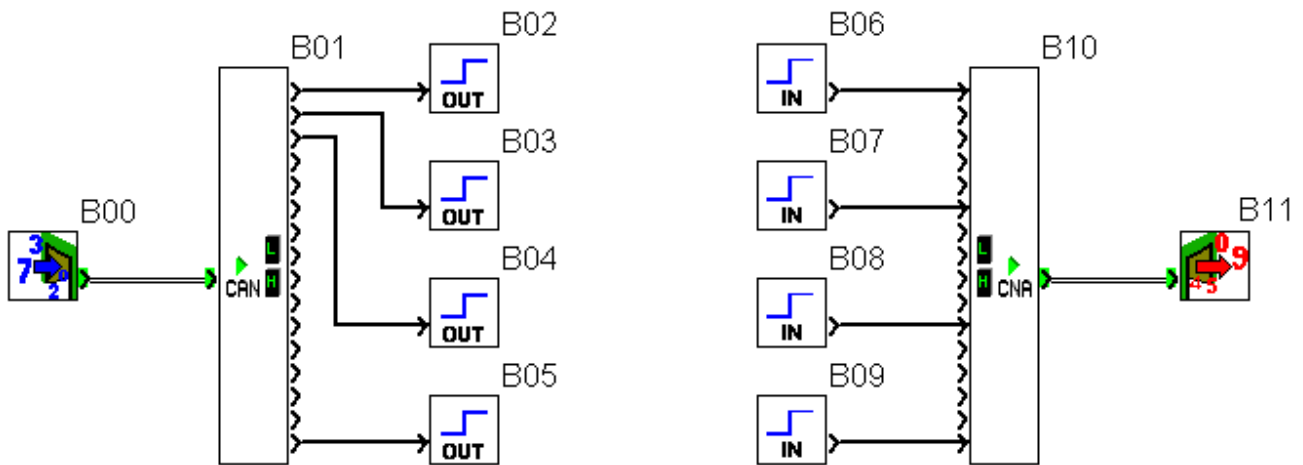
CAN-перетворювач виконує «розщеплення» шини на окремі сигнальні лінії, тобто перетворює цілочисельне значення на дискретні сигнали.



CNA-перетворювач «зв'язує» окремі сигнальні лінії у шину, тобто перетворює дискретні сигнали на цілочисельне значення.

Таким чином, маємо 16-бітні перетворювачі типів даних «Word to Bits» (CAN) та «Bits to Word» (CNA).

Як приклад використання перетворювачів CAN та CNA наведемо таку схему:



Допускається використовувати не всі входи або виходи блоків перетворювачів. Незадіяні входи будуть сприйматися як неактивні (нульові).

Запитання для самоперевірки:

1. Вхід I2 – це лічильний декрементний вхід швидкого лічильника, чи керуючий вхід, що забезпечує зворотний рахунок імпульсів, які надходять до входу I1?
2. Чи можна вставити до програми 2 або більше виходів керування підсвічуванням дисплею? Якщо так, то який з них буде діючим у випадку, коли вони знаходяться у різних станах (активний/пасивний)?
3. Чи можна у режимі моніторингу вручну зупиняти роботу функціонального блока «Метроном» та довільно змінювати його стан (активний/неактивний)?
4. Чи можна змінювати шпаруватість сигналу, що генерує функціональний блок «Метроном»? Як саме і у яких межах?
5. У чому полягає відмінність роботи таймера у режимах В та Н?
6. Скільки швидких лічильників може бути використано у програмі? Чому саме?
7. Які значення параметрів функції підсилювача (Gain) встановлюються за умовчанням і чому?
8. Яким буде вихідний сигнал зонного компаратора, якщо Min value > Max value?
9. Для чого використовується вхід «Error propagation» арифметичних блоків?
10. У яких випадках необхідно використовувати вхід «Input 3» блока MUL/DIV?
11. Чи відрізняються функціональні блоки «LCD Display» та «LCD screen backlighting»? Якщо так, то чим саме?
12. Якщо у вікні параметрів дисплейного функціонального блока надпис виводиться негативним шрифтом і підсвічується червоним кольором, про що це свідчить?

Додаток 1

Список помилок:

Number	Type of error
00	No faults
01	Fault in writing to EEPROM This fault defines transfer problems between the memory cartridge and the smart relay. If the fault occurs frequently, contact the after-sales service.
02	Fault in writing to the clock If the fault occurs frequently, contact the after-sales service.
04	Overload on transistor outputs Once a transistor output reaches a temperature of 170°C, the group of 4 outputs to which it belongs is deactivated. To make this group of outputs operational, the cause of the over current (short-circuit, etc.) must first be deleted, and then the fault cleared from the FAULT menu .
50	The smart relay firmware is damaged Reload the firmware on the module and the user application. If this problem persists, contact the after-sales service.
51	Watchdog overflow Warning or error according to the selection made in the configuration menu (module display) or in the configuration window (Zelio Soft 2 Programming Software). The cycle time in the smart relay is too short compared with the application program execution time programmed in the smart relay. If the application requires a strict sampling of the module inputs/outputs, lengthen the cycle time in the module. To do this, configure the information either in the CONFIGURATION menu (module display) or in the configuration window (Zelio Soft 2 Programming Software). If the application does not require the cycle time, in CONFIGURATION select: No Action for the WATCHDOG.
52	The smart relay has executed an unknown operation If the fault is permanent, reload the firmware on the smart relay and the user application. If this problem persists, contact the after-sales service.
53	Link between module and bus extension faulty Check operation of the extension (connection, power supply, fault).
54	Link between module and input/output extension faulty Check operation of the extension (connection, power supply and fault).
58	A fault is present in the firmware (software specific to the smart relay) or on a part of the smart relay hardware. If the fault is permanent, reload the firmware on the module and the user program. If this problem persists, contact the after-sales service.
59	At the beginning of RUN on the module application: the application cannot switch to RUN as it is incompatible with the module physically connected to the supply. If this problem occurs, contact the after-sales service.
60	At the beginning of RUN on the module application: program incompatible with the bus extension physically connected to the power supply. If this problem occurs, contact the after-sales service.

61	At the beginning of RUN on the module application: program incompatible with the Input/Output extension physically connected to the power supply. If this problem occurs, contact the after-sales service.
62	Version (or build number) incompatibility when loading a program from the backup memory If this problem occurs, contact the after-sales service.
63	Hardware configuration incompatibility when loading a program from the backup memory If this problem occurs, contact the after-sales service.

Список помилок:

Number	Type of warning
05	Recipient problem The type of recipient is incorrect. Modify the recipient. Note: This fault is non-blocking.
42	Smart Relay Application Checksum Fault The application in the smart relay is not correct. Transfer the application in the programming software to the module.
43	Power supply return A loss of power has been detected. Note: This fault is non-blocking.
44	PIN Error Codes 2 incorrect PIN codes have been entered. Check the PIN code on the SIM card of the GSM modem connected to the communication interface.
45	IS463 Unknown Status An unknown operation was executed. Reload the firmware and the communication interface.
46	Program fault The application in the smart relay is absent. Transfer the application in the programming software to the module.
47	IS498 Unknown Status An unknown operation was executed. Reload the firmware and the communication interface.
48	Modem missing The modem cannot be detected. Check: the link between the SR2COM01 communication interface and the modem Modem power supply
49	Smart Relay Absent The smart relay cannot be detected. Check: the link between the SR2COM01 communication interface and the smart relay The state of the smart relay Note: This fault is non-blocking.
50	Binary fault

	The firmware of the communication interface is damaged. Reload the firmware and the communication interface.
51	Modem parameters incorrect Check the modem configuration using the programming software. See Remote Stations Directory .
58	Watchdog Fault Internal watchdog overflow. Note: This fault is non-blocking.
60	OFF Alarms Alarm message send is disabled. See Message OFF
61	SIM Erase Error A SIM card erase error has been detected. Check the SIM card.