



Національний
технічний університет

**ДНІПРОВСЬКА
ПОЛІТЕХНІКА**
1899

Основи програмування у програмному середовищі ТХТ-контролера

Основні питання



1. Створення підпрограми
2. Інформаційні канали. Глобальні та локальні змінні. Константи. Оператори
3. Команди. Входи і виходи контролера. Розгалуження, очікування.



1. Створення підпрограми



Subprogram I/O (Підпрограма) (Level 2)

The screenshot displays the ROBO Pro software interface. On the left, the 'Element groups' panel shows 'Subprogram I/O' highlighted with a red circle. Below it, the 'Program elements' panel shows a list of elements: 'Entry En', 'Exit Ex', 'In In', and 'Out Ou'. A red rectangle encloses these four elements. On the right, the main workspace shows a 'Main program' area with a grid. Three icons in the top toolbar are circled in red: a blue icon with a plus sign, a blue icon with a square, and a red icon with an 'X'. Red arrows point from these icons to the text 'Дублювання підпрограми' (Duplicate subprogram), 'Створення підпрограми' (Create subprogram), and 'Видалення підпрограми' (Delete subprogram) respectively. Another red arrow points from the 'Subprogram I/O' menu item to the text 'Підпрограма' (Subprogram). A final red arrow points from the 'Program elements' list to the text 'Програмні елементи' (Program elements).

ROBO Pro - [unnamed1]

File Edit Draw View Level Environment Bluetooth Window Help

Element groups

- Program elements
 - Basic elements
 - Subprogram I/O**
 - Send, receive
 - Variables, timers...
 - Commands
 - Branch, wait...
 - Inputs, outputs
 - Operators
 - Operating elements

Program elements

- Entry En
- Exit Ex
- In In
- Out Ou

Main program

Function Symbol Panel TXT/TX Display Camera Properties Description

Дублювання підпрограми

Створення підпрограми

Видалення підпрограми

Підпрограма

Програмні елементи

Створення підпрограми (Level 2)

Базовий елемент

Місце знаходження

Закладка підпрограми

Вхід у підпрограму

Вихід з підпрограми

```
graph TD; Entry[Entry] --> M1[M1 V=8]; M1 --> Exit[Exit];
```

```
graph TD; Main[Main program] --> Pyx[Pyx вліво];
```



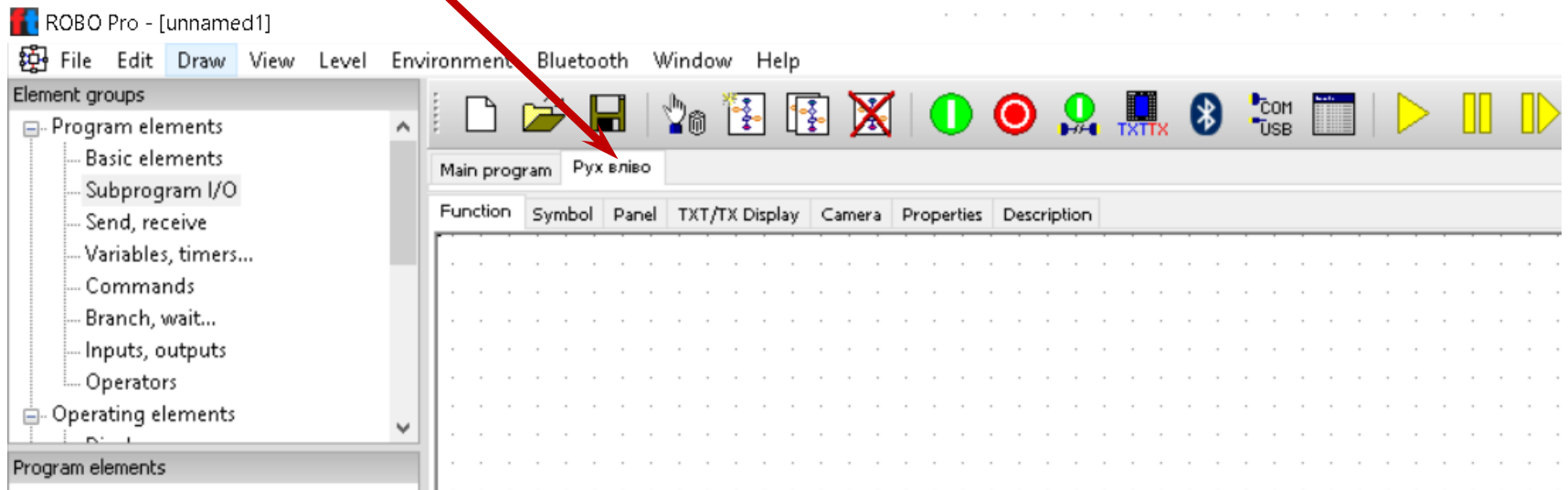
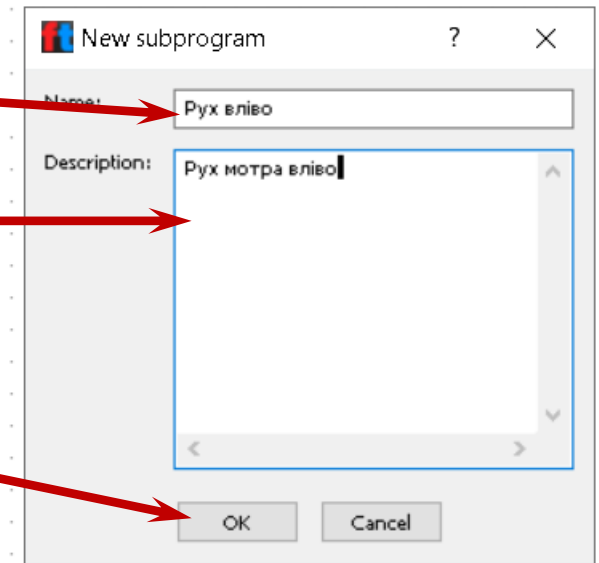
Створення підпрограми (Level 2)

Назва
підпрограми

Опис
підпрограми

Натиснути

З'являється закладка
підпрограми



Створення підпрограми (Level 2)

Базовий елемент

Місце знаходження

Закладка підпрограми

Вхід у підпрограму

Вихід з підпрограми

```
graph TD; Entry[Entry] --> M1[M1 V=8]; M1 --> Exit[Exit]
```

```
graph TD; Main[Main program] --> Pyx[Pyx вліво]
```

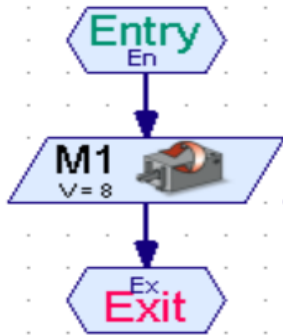


Використання підпрограми

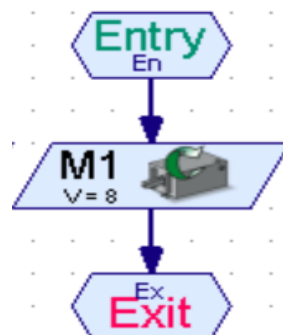
(Приклад 1)

Запустити двигун з обертанням проти годинникової стрілки, через 1 секунду змінити напрям обертання і через 1 секунду зупинити двигун.

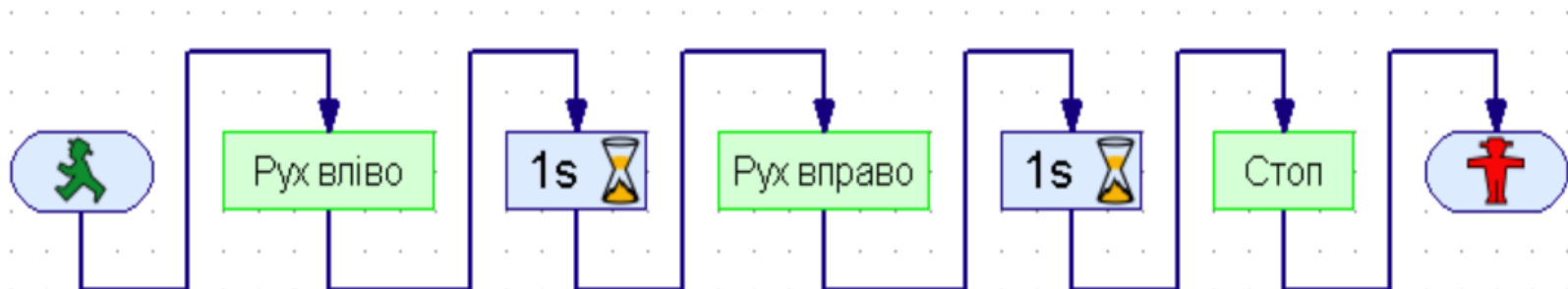
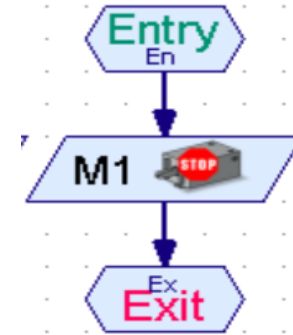
Підпрограма
«Рух вліво»



Підпрограма
«Рух вправо»



Підпрограма
«Стоп»



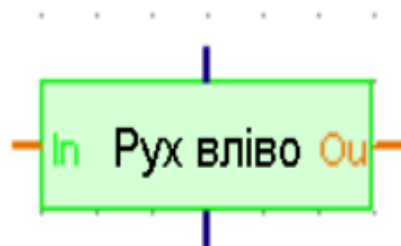
Елементи підпрограми Input та Output (Level 2)



Інформаційний вхід підпрограми можливо пов'язати з вхідними елементами (наприклад, перемикачами) основної програми або передати в підпрограму значення змінних.



Інформаційний вихід підпрограми. За допомогою цього елемента можливо відправляти в основну програму команди і дані.

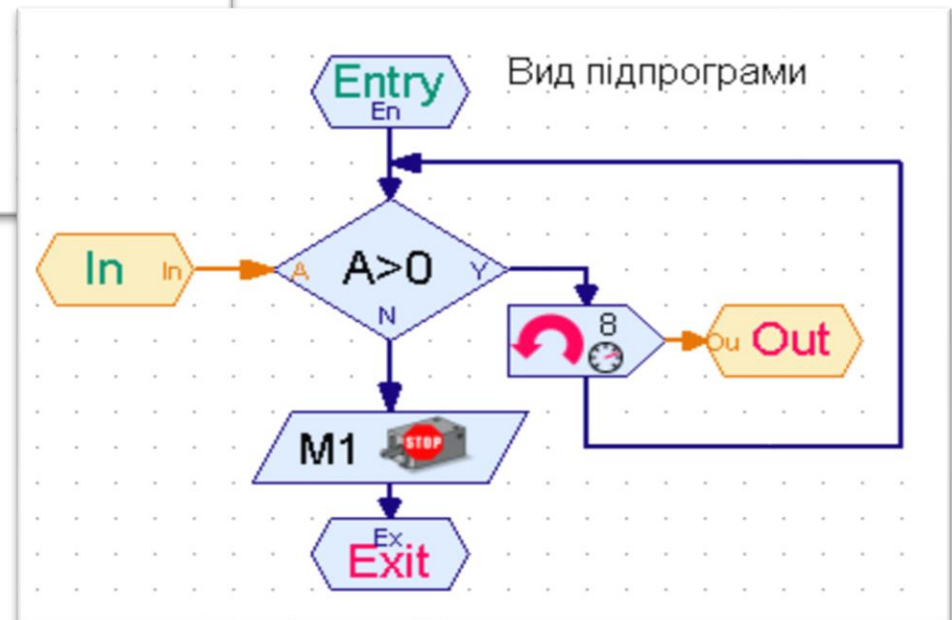
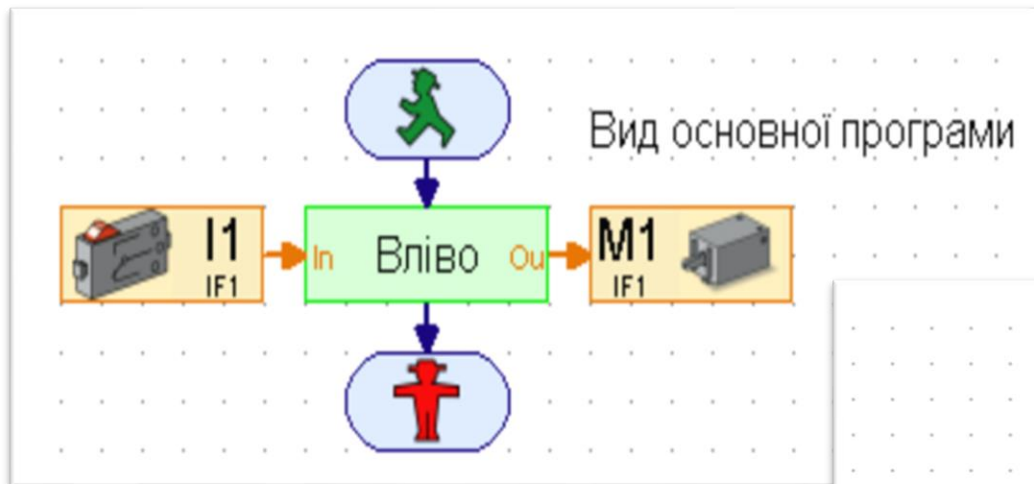


Вигляд підпрограми в основній програмі



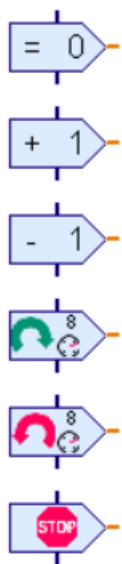
Використання підпрограми (Приклад 2)

При замиканні кінцевого вимикача двигун має обертатись вліво. При розмиканні кінцевого вимикача двигун має зупинитись.



2. Команди. Входи і виходи контролера. Розгалуження, очікування.

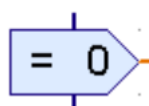




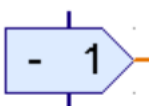
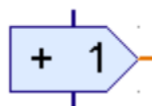
Команди

Коли виконується **командний елемент** (наприклад, коли потік програми приходить до входу (зверху) цього елемента), то він відправляє значення або повідомлення в елемент, підключений до виходу справа.

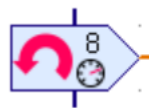
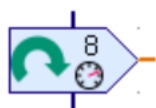
Команди



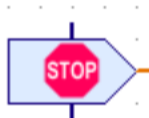
= присвоює значення
приймаючому елементу



+/- відправляє змінним або таймерам для збільшення/зменшення їх значення на вказану величину



Вправо/вліво відправляє на вхід мотору команду на включення і обертання за/проти годинникової стрілкою/ки



Стоп відправляє на вихід мотору команду для зупинки або виключення



I1

IF1



I1

IF1



M1

IF1



O1

IF1



Panel Input



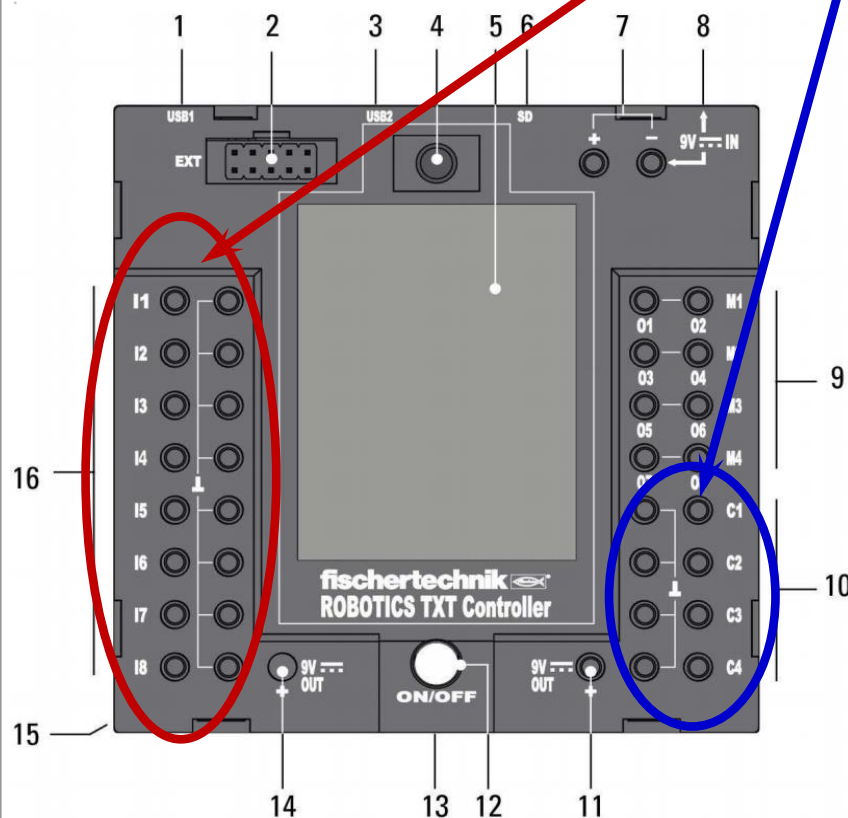
Panel Display



Camera Input

Входи і виходи контролера (Inputs/Outputs)

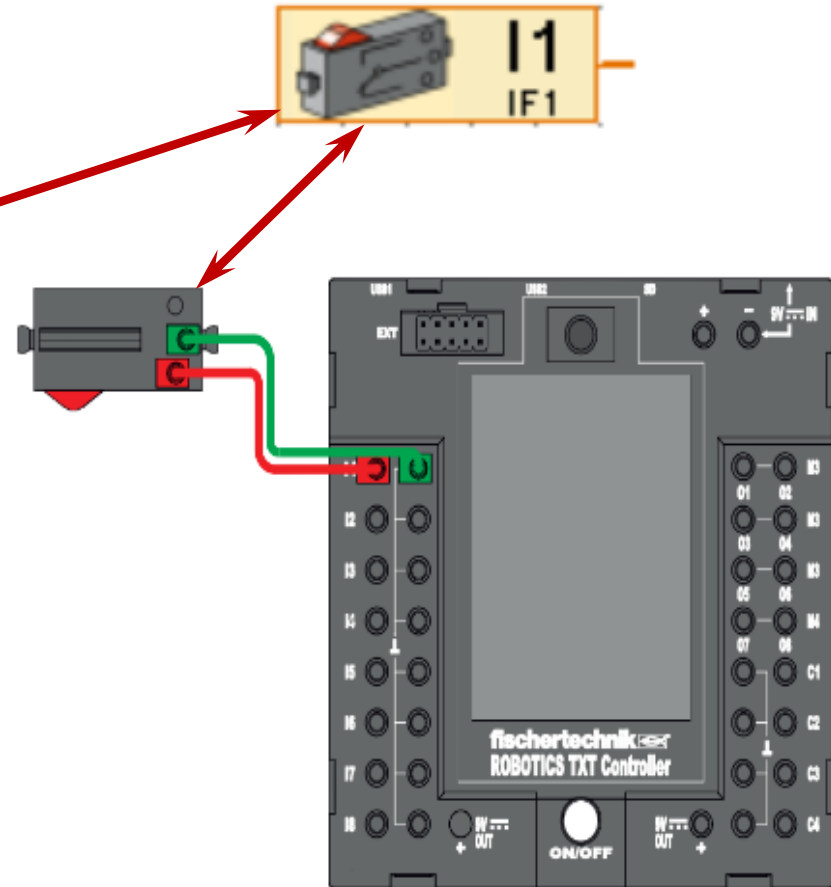
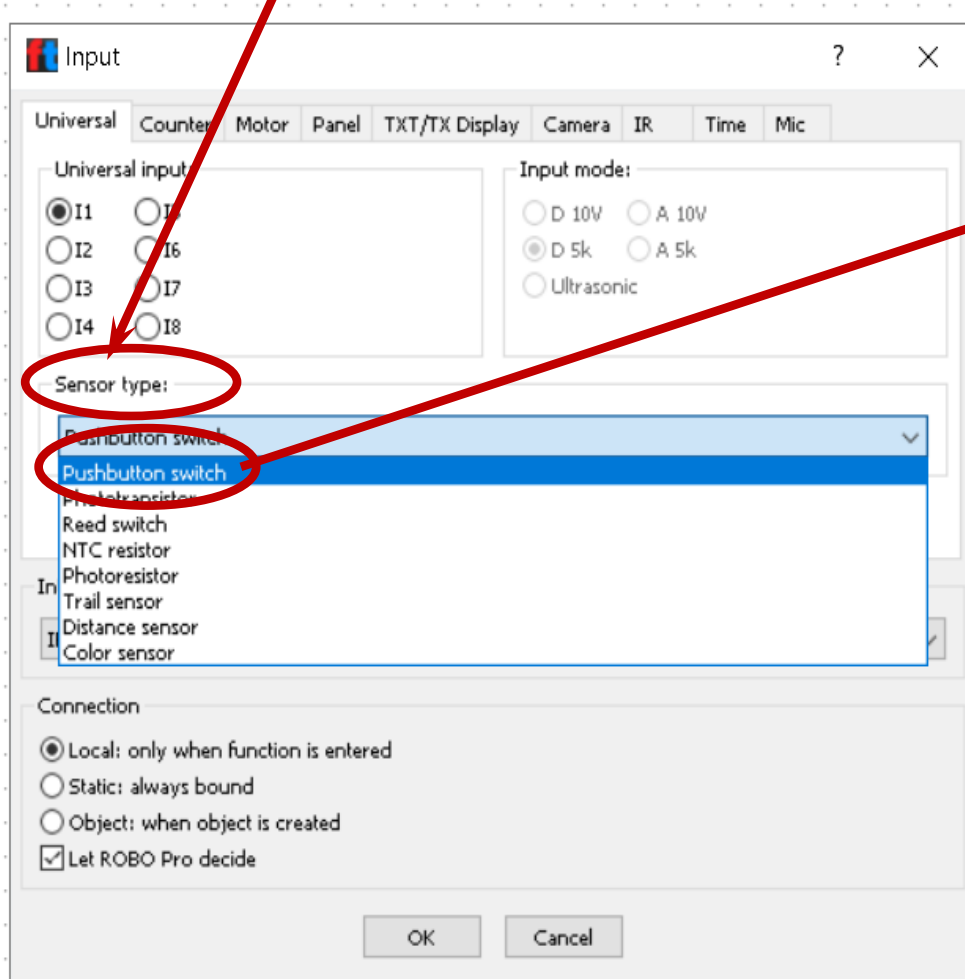
Контролер ROBOTICS TXT має **8 універсальних входів I1-I8**, які можуть використовуватися в якості цифрових або аналогових входів і **4 входи з функцією лічильника C1-C4**



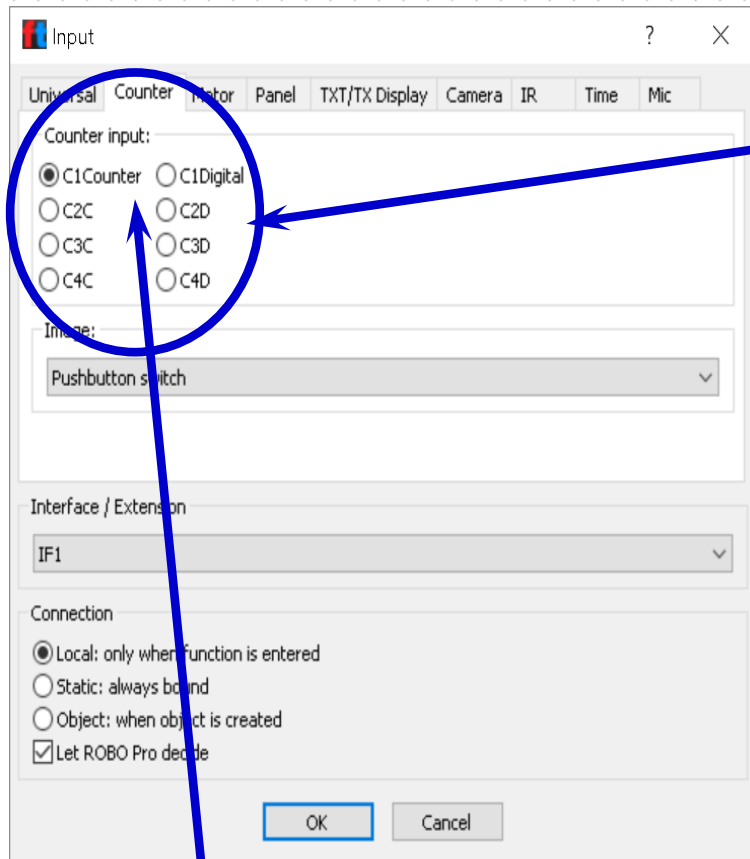
1. USB-A порт (USB-1)
2. EXT з'єднання для розширень
3. Mini USB порт (USB-2)
4. IR діод приймача
5. Сенсорний дисплей
6. Micro SD слот для картки
7. 9V вхід, підключення акумуляторної батареї
8. 9 V вхід, DC роз'єм живлення
9. Виходи M1...M4 (O1...O8)
10. Входи C1...C4
11. 9V вихід
12. ON/OFF перемикач
13. Динамік
14. 9 V вихід
15. Відсік для батареї
16. Універсальні входи від I1 до I8

Входи контролера (Inputs)

В меню **Sensor type** можна вибрати тип датчика, який підключений до входу.

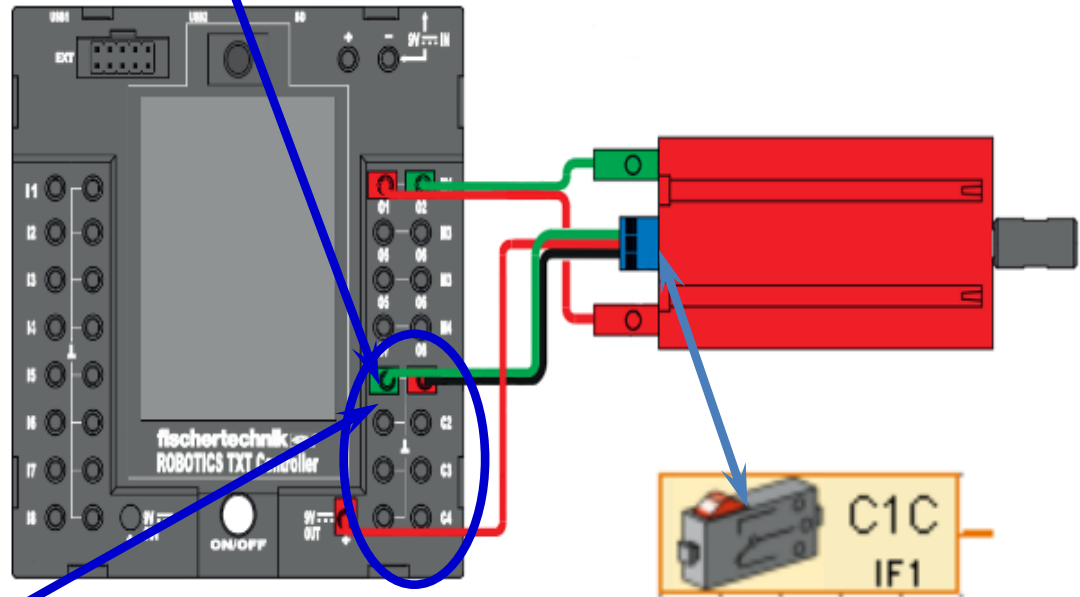


Входи контролера (Inputs)



Контролер містить

4 входи з функцією лічильника C1-C4



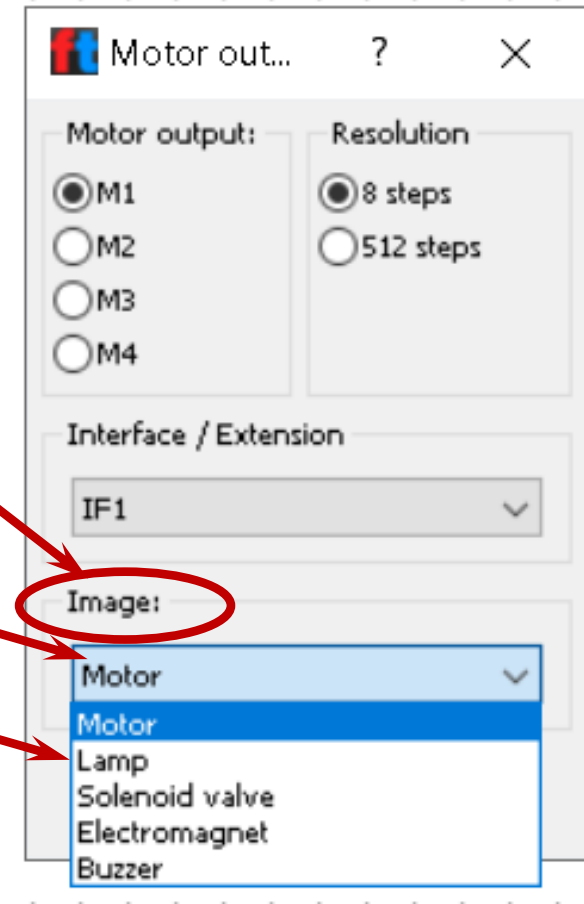
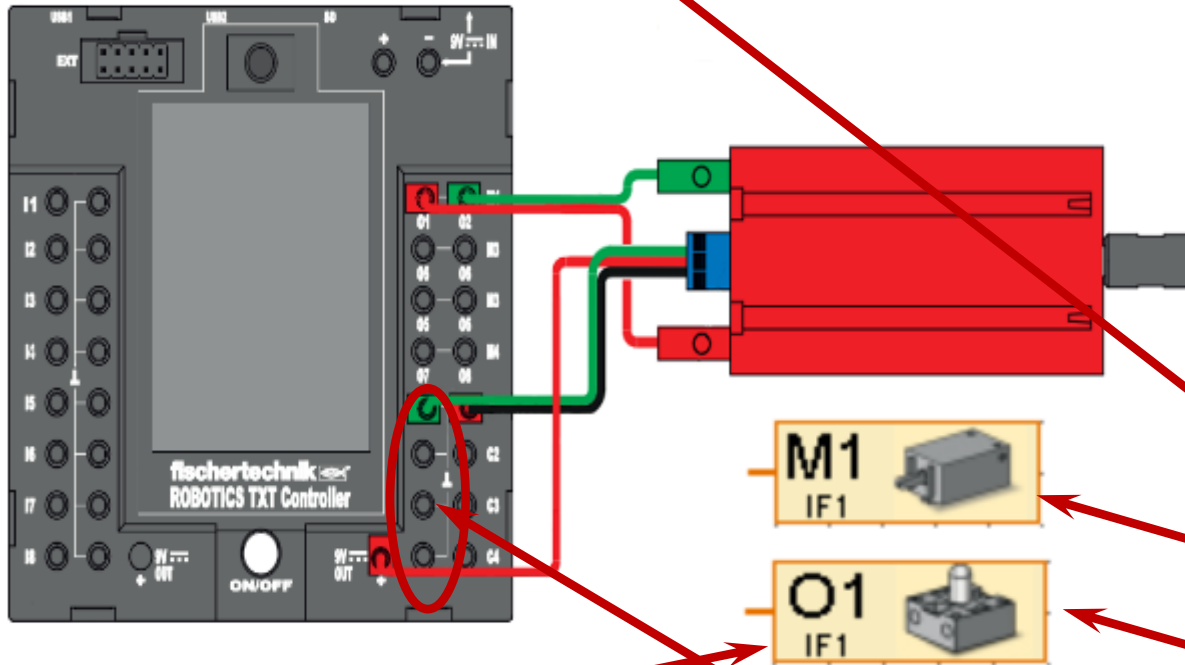
Кожен рахунковий вхід представлений двома «віртуальними» входами

Наприклад, входу C1, який розташований на контролері, відповідають входи C1C і C1D в ROBO Pro. **Вхід C1D** функціонує як звичайний цифровий вхід. **Вхід C1C** рахує кількість імпульсів, що надійшли на вхід C1.



Виходи контролера (Outputs)

В меню **Image** є можливість вибрати тип пристрою. Команди надсилаються в цей елемент за допомогою командних елементів в групі Commands.

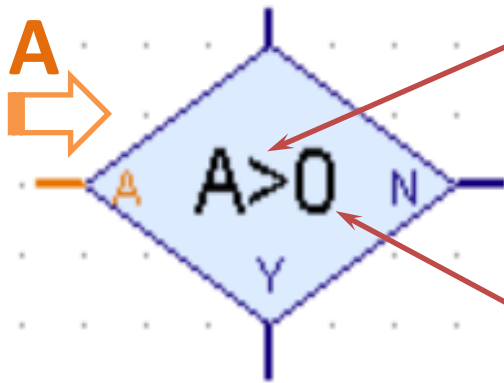


Елемент Лампа використовує тільки один з O1...O8 вихід на контролері.

Другий контакт лампи підключається до **загального потенціалу (\perp)**. Лампу можна тільки включити або виключити.



Елемент Branch (Розгалуження) (Level 3)



Елемент
порівнює
отримане
значення з
заданим
у властивостях

ft Branch ? X

Condition:

Input value: > 0

Data type:

☒ Integer -32767..32767

☐ Floating point 48bit

Swap Y/N branches

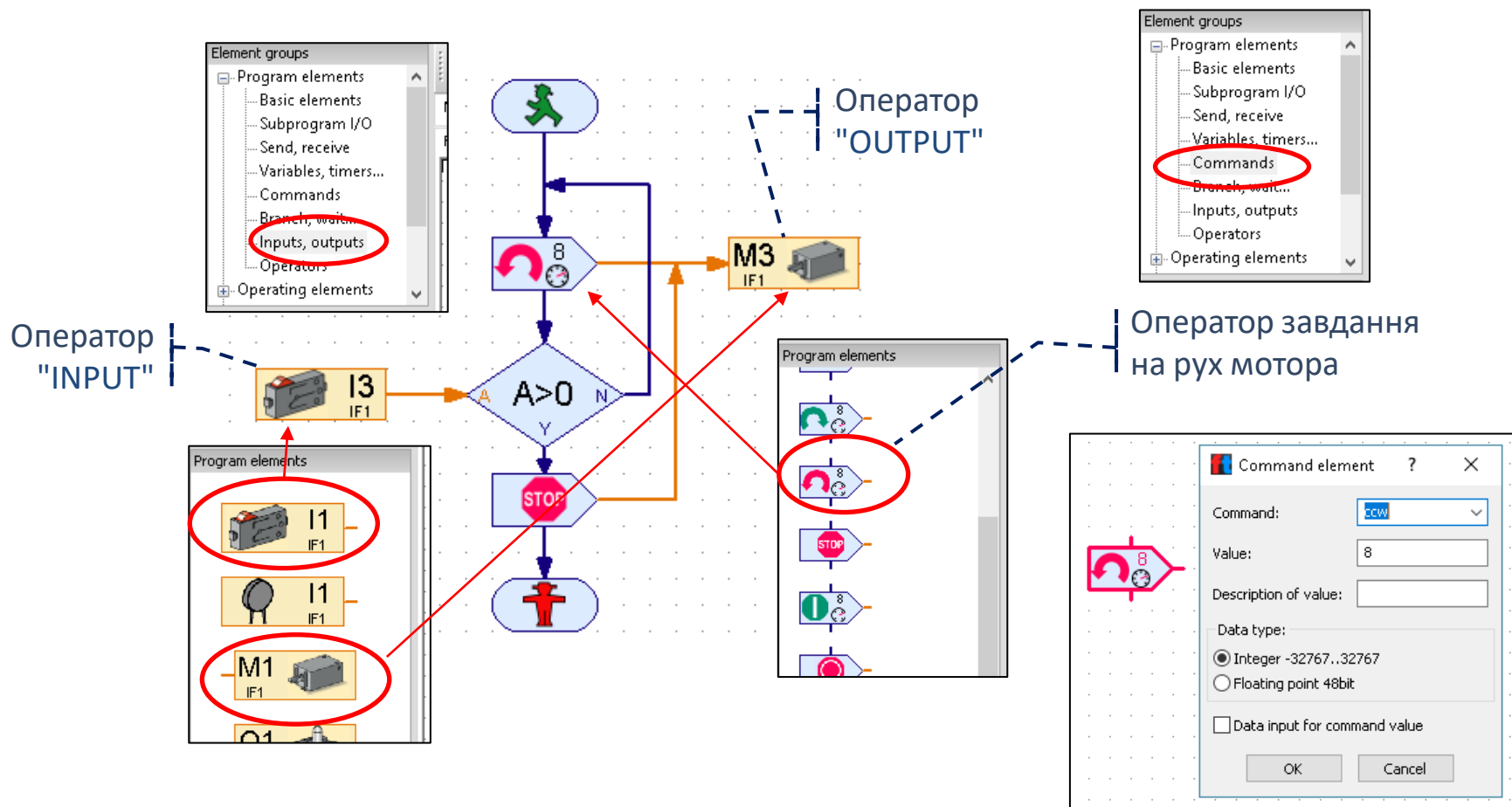
☒ Leave Y/N branches as they are

☐ Swap Y/N branches

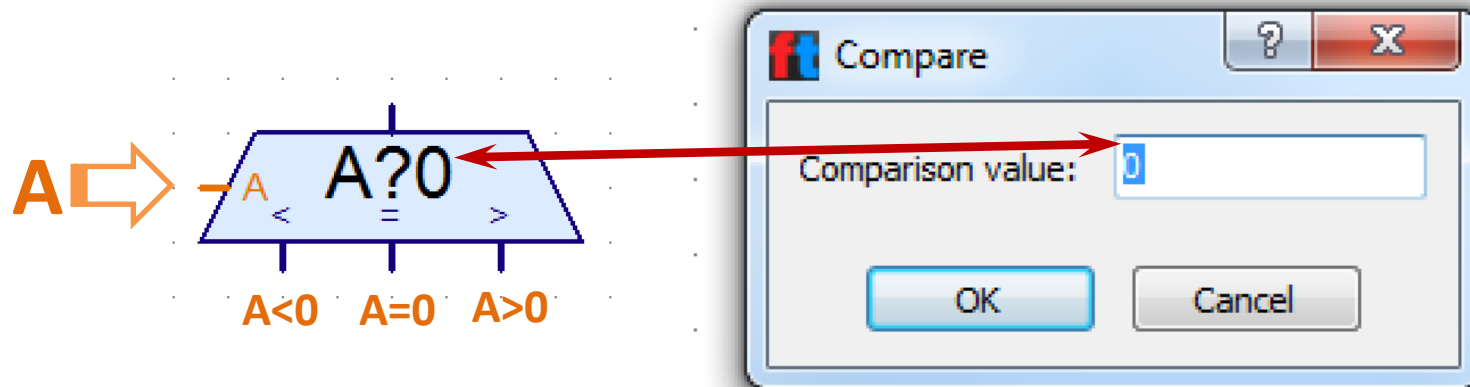
OK Cancel



Програма руху до упору з використанням операторів "Input" і "Output" та елемента "Branch" (Приклад 3)



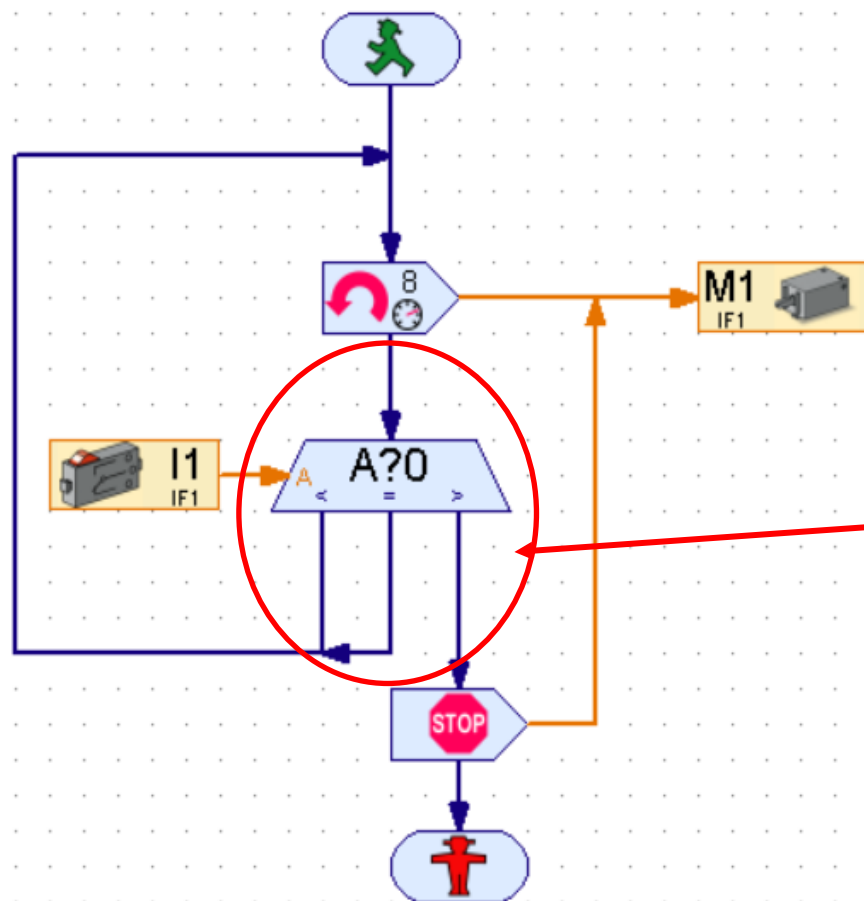
Елемент Comparison with fixed value (Порівняння з фіксованим значенням) (Категорія: Branch – Level 3)



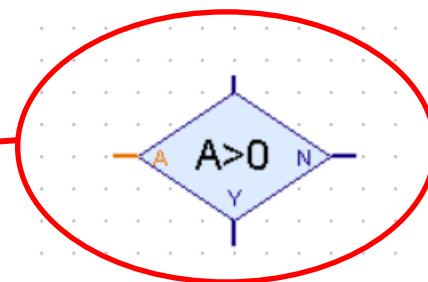
Елемент “Comparison with fixed value” організує інваріантне розгалуження у залежності від співвідношення значення змінної до фіксованого числового значення.

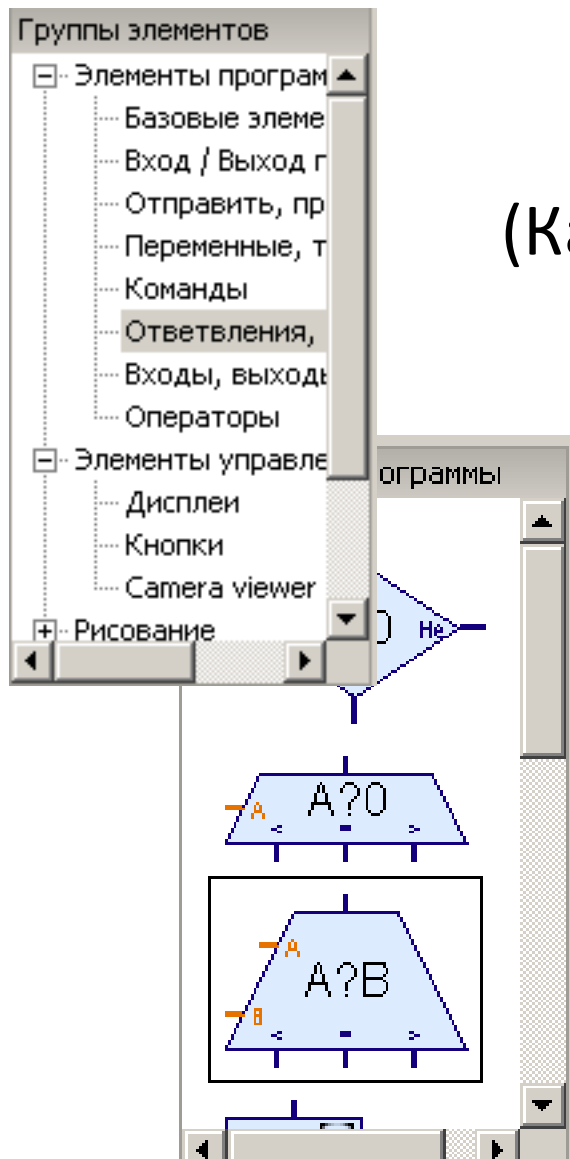


Програма руху до упору з використанням операторів “Input” і “Output” та елемента “Branch” (Приклад 4)



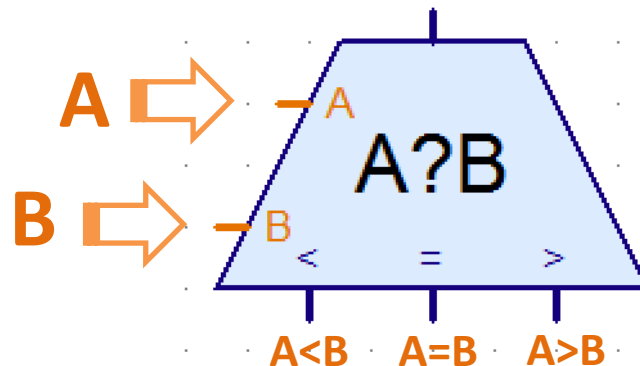
Еквівалентне
використання
елементів





Елемент Compare (Порівняння)

(Категорія: Branch – Level 3)



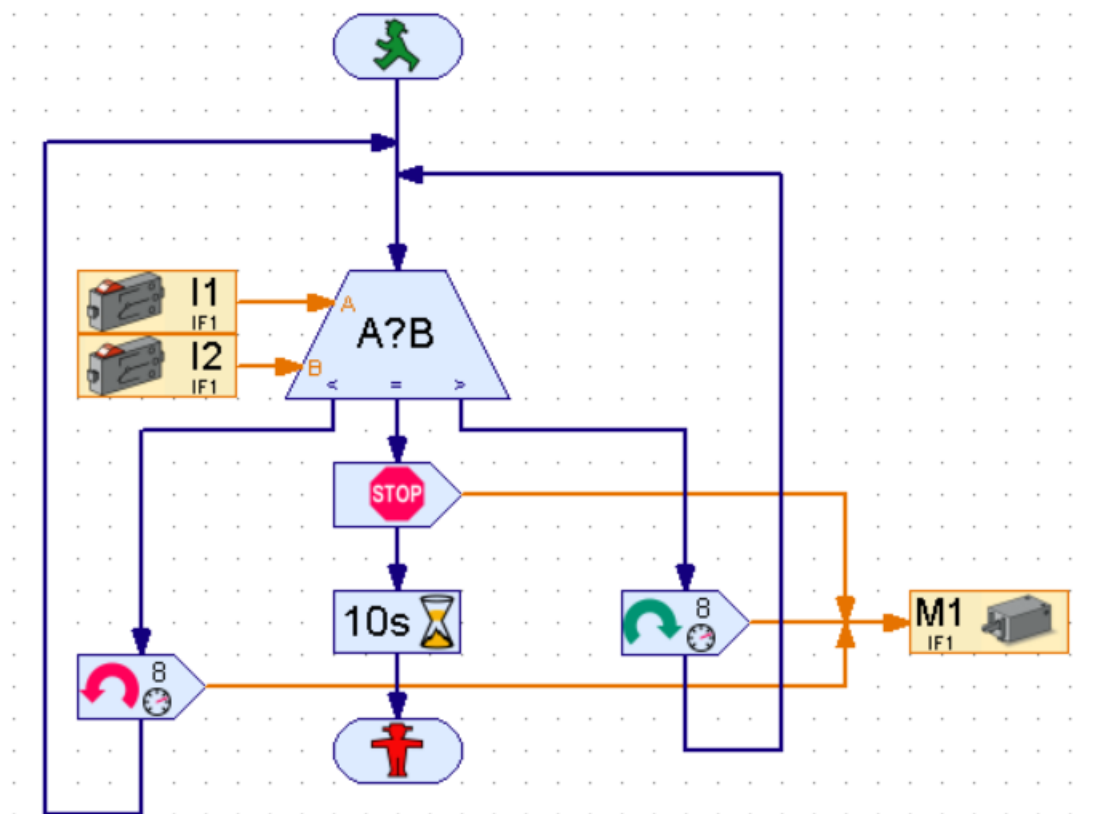
Елемент “Comparison” організує інваріантне розгалуження у залежності від співвідношення значень двох змінних.

Елемент не має опцій налаштування.



Програма руху до упору з використанням операторів “Input” і “Output” та елемента “Branch” (Приклад 5)

Програма порівнює стан двох кінцевих вимикачів і в залежності від цього змінює напрям обертання мотору. Якщо стан вимикачів однаковий, то мотор зупиняється і через 10 секунд програма закінчується



Завдання 1

(на основі Прикладу 5)

Забезпечити роботу двох моторів за логікою, що представлена у таблиці

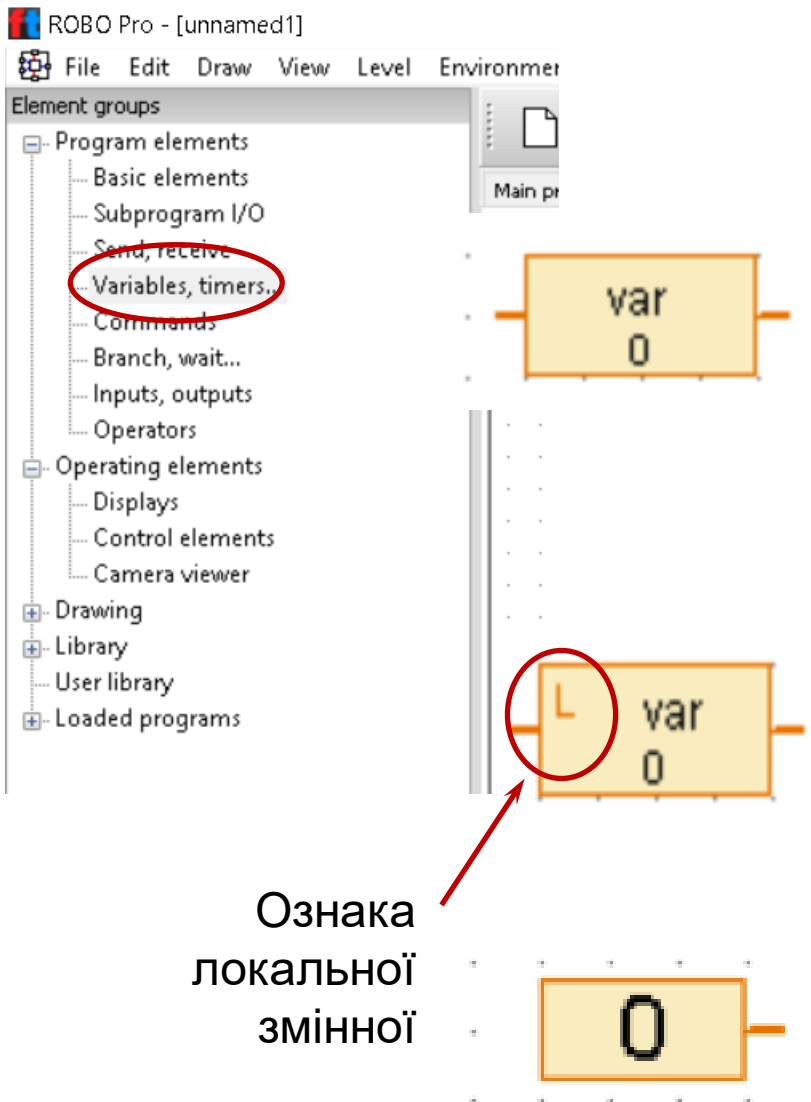
I1	I2	M1	M2
0	0	очікування	очікування
1	0	→	←
0	1	←	→
1	1	stop	stop



3. Інформаційні канали. Глобальні та локальні змінні. Константи. Оператори



Глобальні та локальні змінні. Константи



Ознака
локальної
змінної

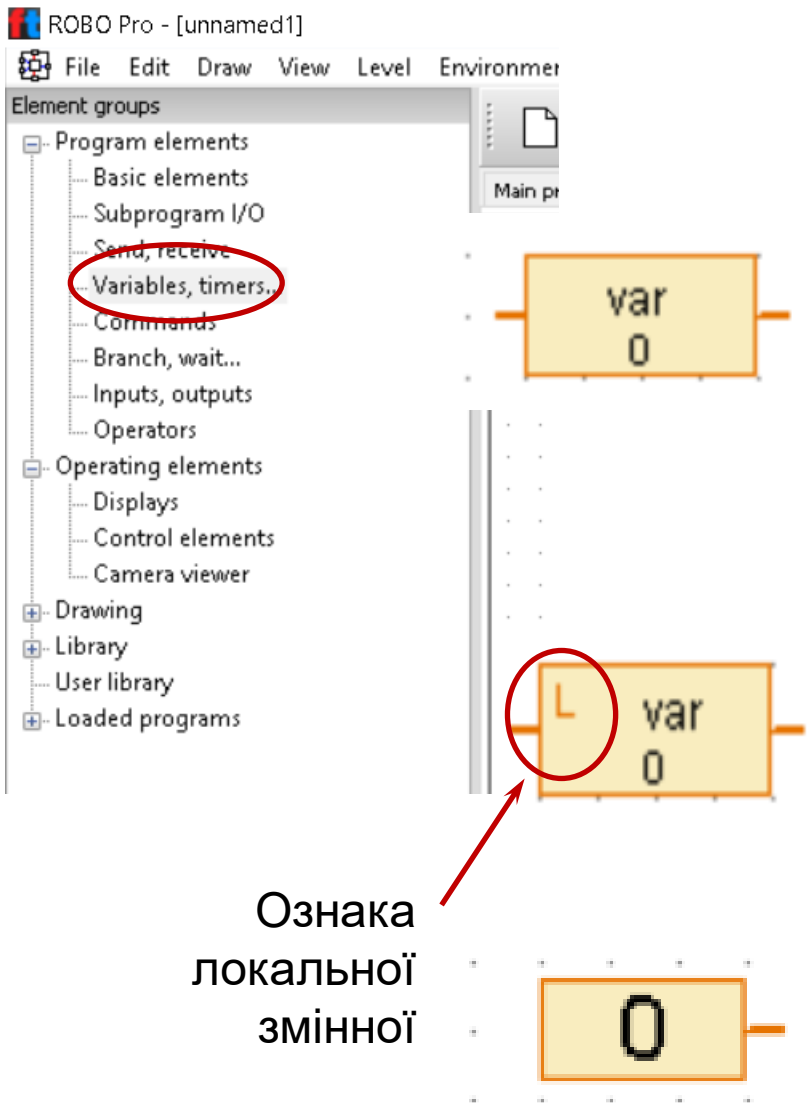
Глобальна змінна визначається у основній програмі і усіх підпрограмах. може зберігати числове значення в діапазоні від -32767 до 32767. Значення можна задати за допомогою елементів «=,+» , які треба підключити до входу змінної (зліва).

Локальна змінна діє тільки в тій підпрограмі, в якій вона знаходиться

Константа. Як і змінна, константа зберігає значення, але це значення не може бути змінено програмою.



Глобальні та локальні змінні. Константи

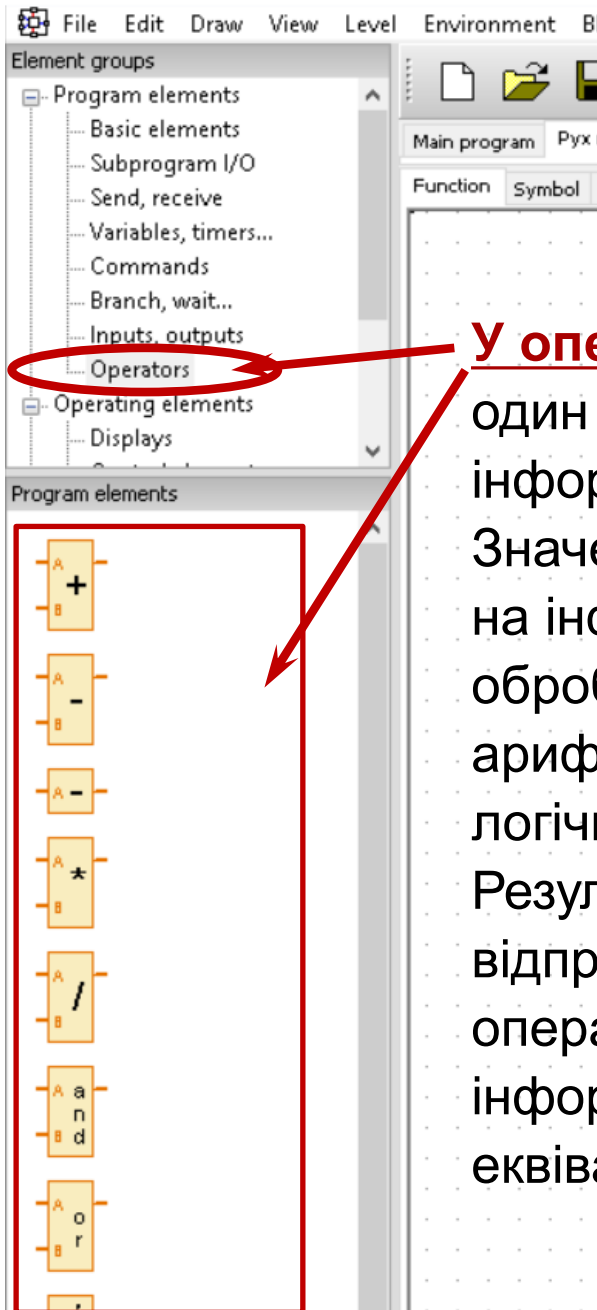


Глобальна змінна визначається у основній програмі і усіх підпрограмах. може зберігати числове значення в діапазоні від -32767 до 32767. Значення можна задати за допомогою елементів «=,+» , які треба підключити до входу змінної (зліва).

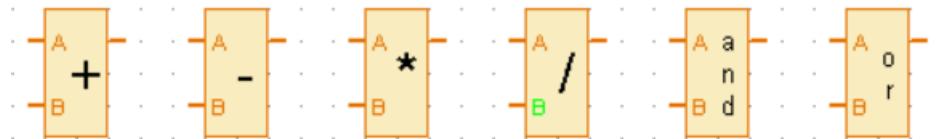
Локальна змінна діє тільки в тій підпрограмі, в якій вона знаходиться

Константа. Як і змінна, константа зберігає значення, але це значення не може бути змінено програмою.





Оператори (Operators)

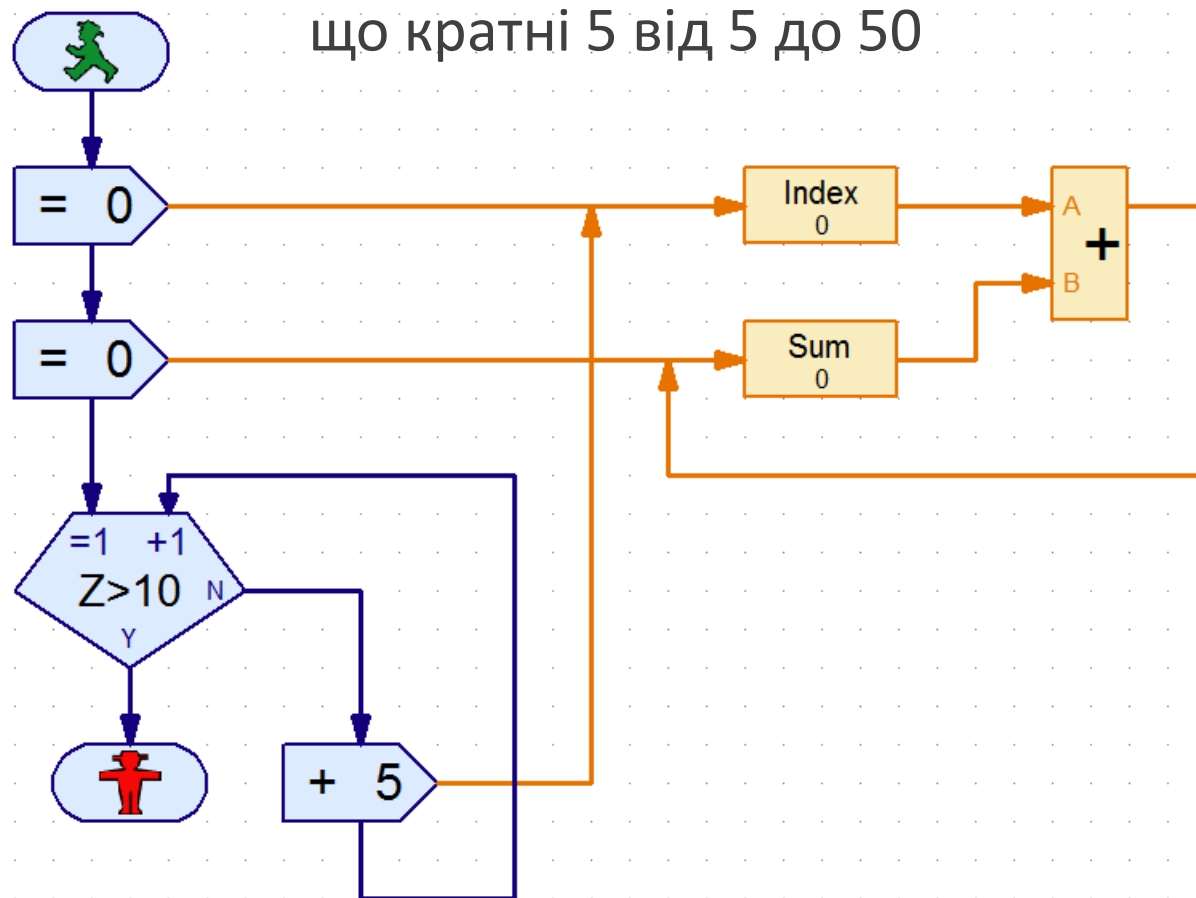


У операторів завжди є один або кілька жовтих інформаційних входів. Значення, які надходять на інформаційні входи, обробляються арифметичними або логічними операторами. Результат обробки відправляється оператором на інформаційний вихід, який еквівалентний команді =.

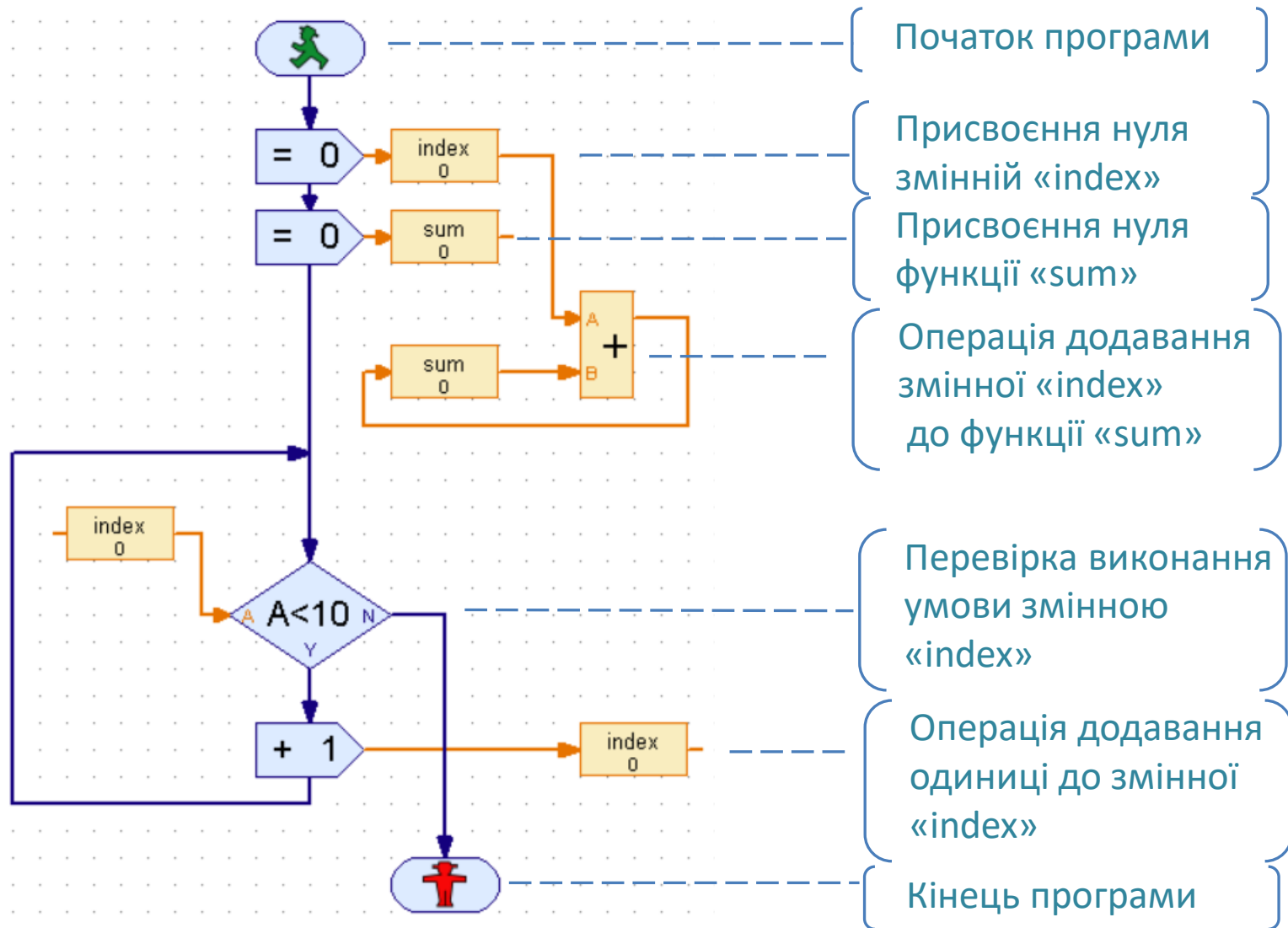


Програма визначення суми ряду (Приклад 6)

Визначення суми ряду натуральних чисел,
що кратні 5 від 5 до 50

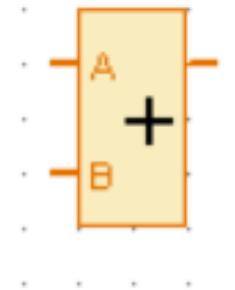
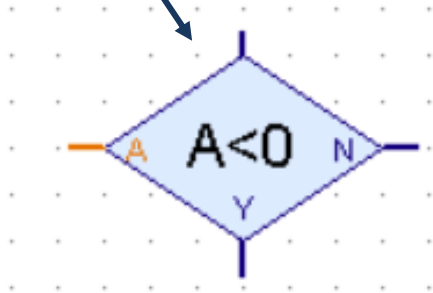


Програма визначення суми чисел від 1 до 10 (Приклад 7)



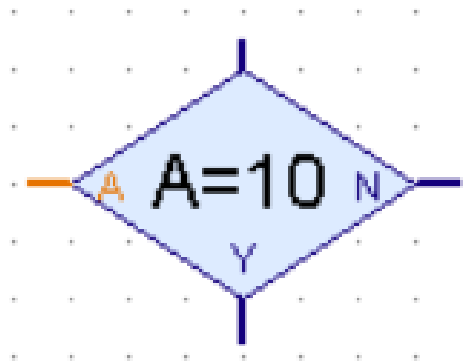
Завдання 3

Створити на основі Прикладу 7 програму визначення суми чисел від 1 до 10 з використанням елемента виду $A < 0$ без використання оператора додавання



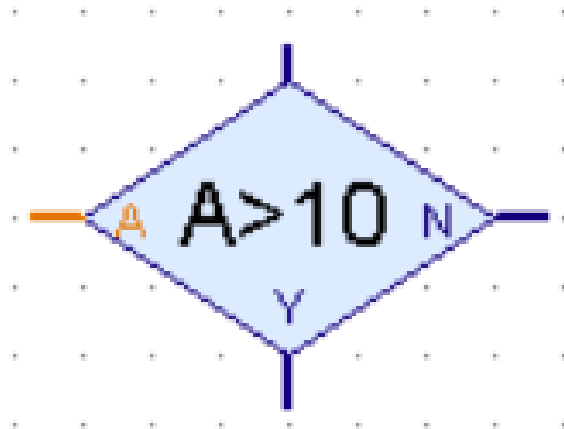
Завдання 4

Створити на основі Прикладу 7 програму визначення суми чисел від 1 до 10 з використанням оператора виду




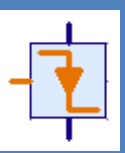













Завдання 5

Створити на основі Прикладу 7 програму визначення суми чисел від 1 до 10 з використанням оператора виду

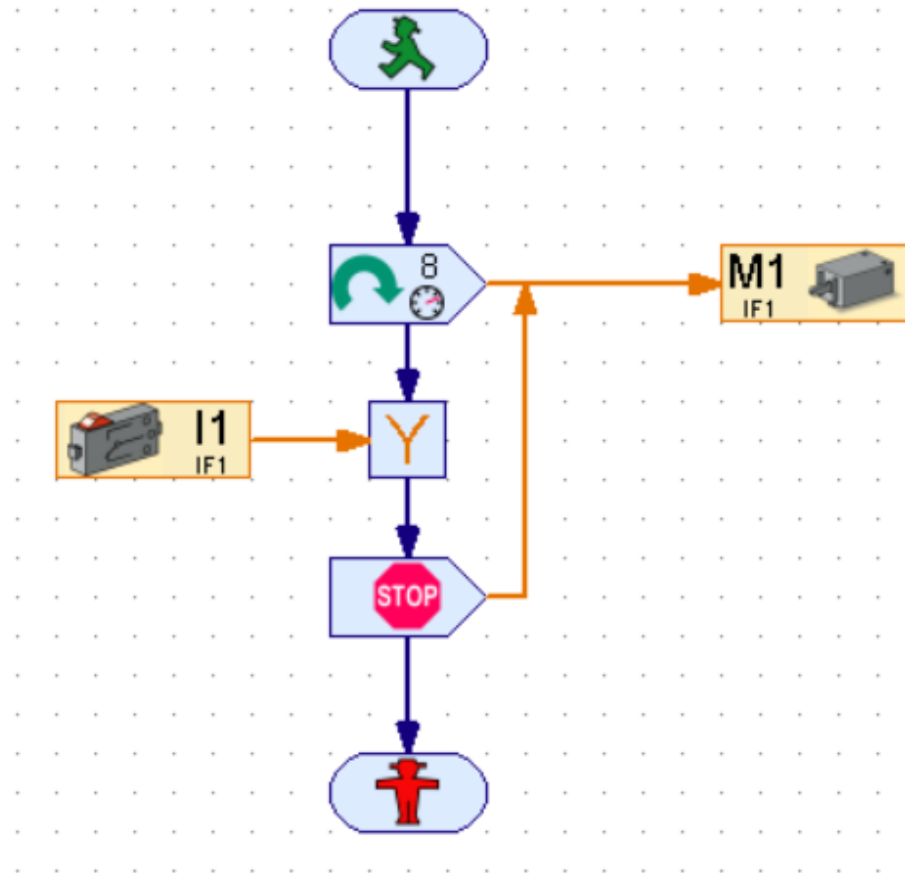


Логіка використання операторів очікування

				
1 	0 	0 → 1 	1 → 0 	0 → 1 → 0 
0 	1 	0 	1 	0 

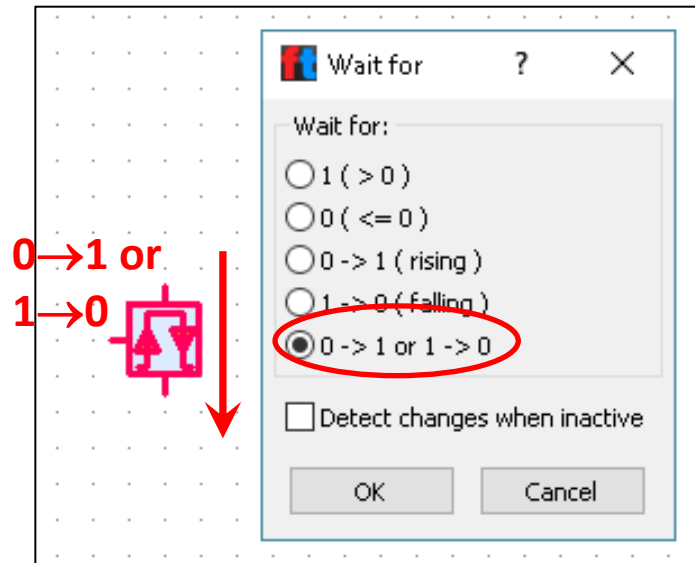


Програма руху до упору з використанням операторів “Wait for” (Приклад 8)

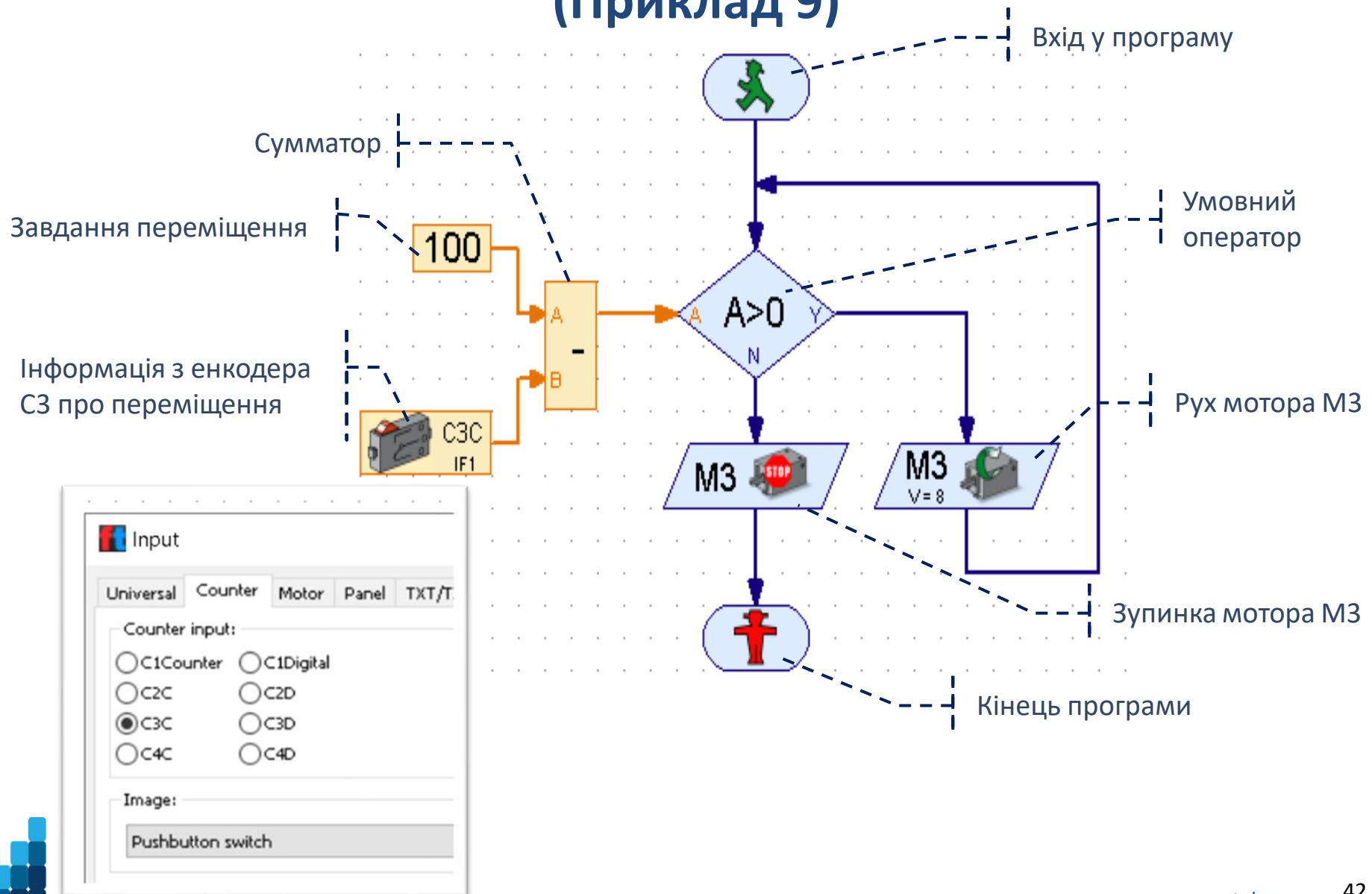


Завдання 6

Створити на основі Прикладу 8 програму руху мотора з використанням оператора очікування
 $0 \rightarrow 1$ or $1 \rightarrow 0$

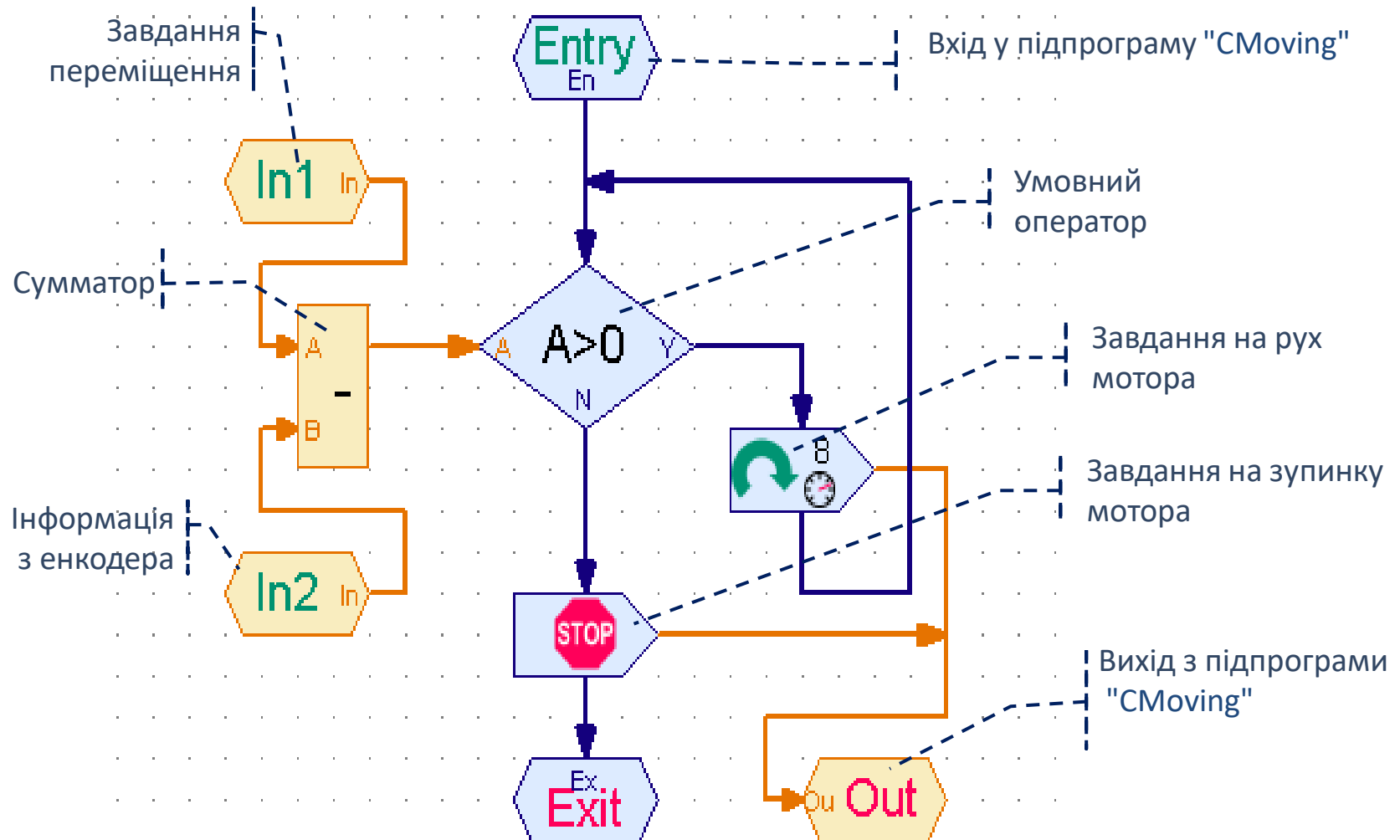


Рух з використанням лічильника імпульсів (Приклад 9)

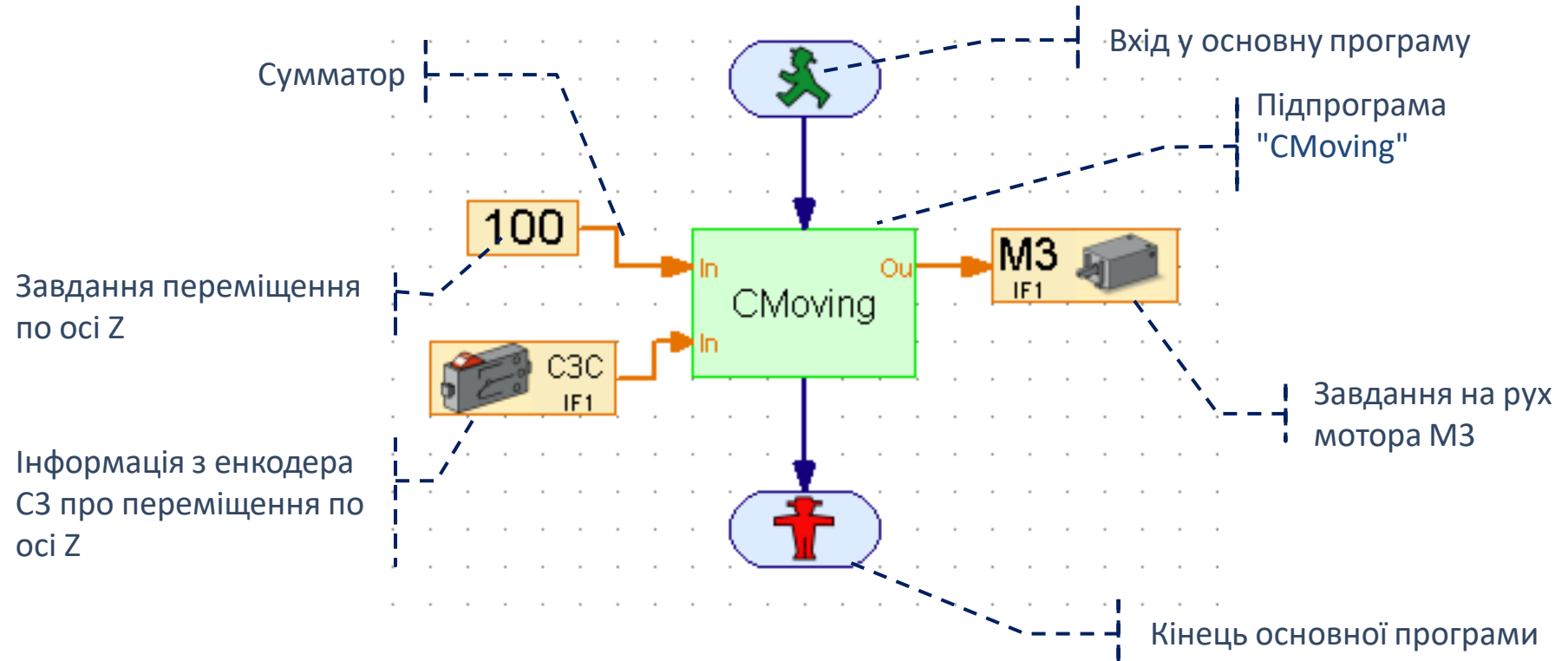


Створення підпрограми руху "Сmoving"

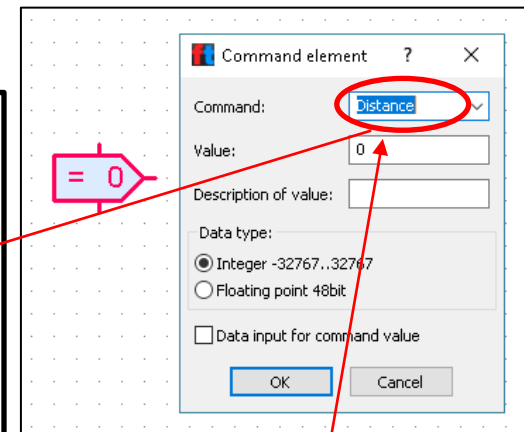
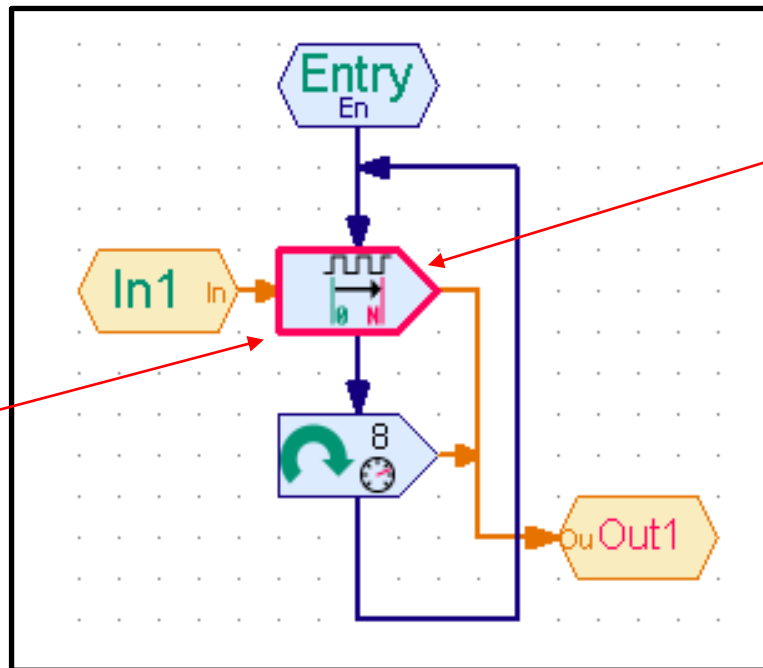
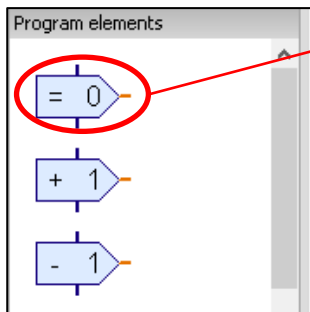
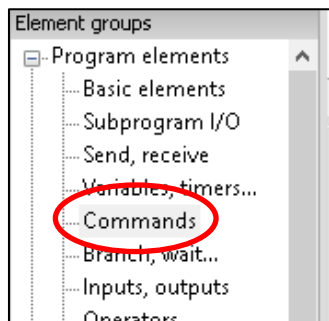
(Приклад 10)



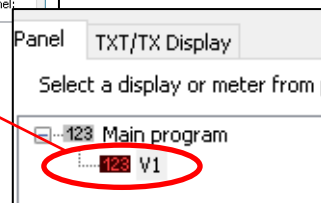
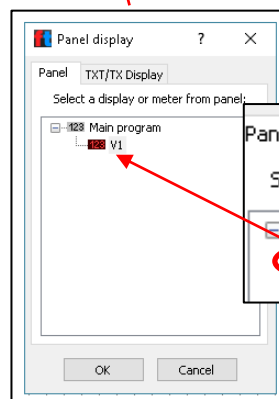
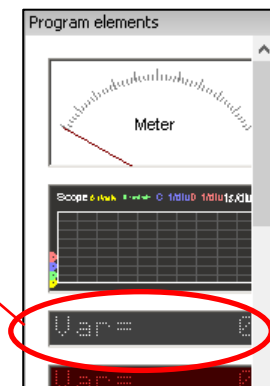
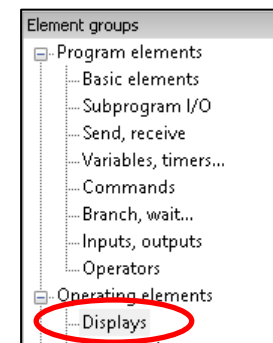
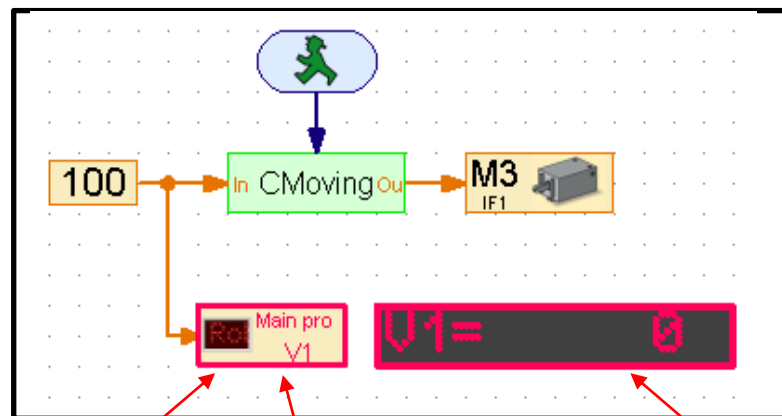
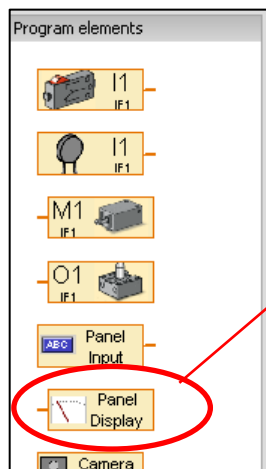
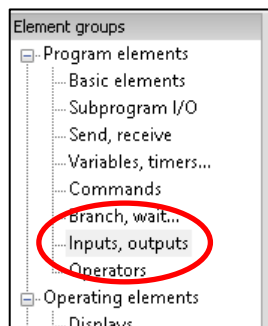
Створення основної програми руху (Приклад 11)



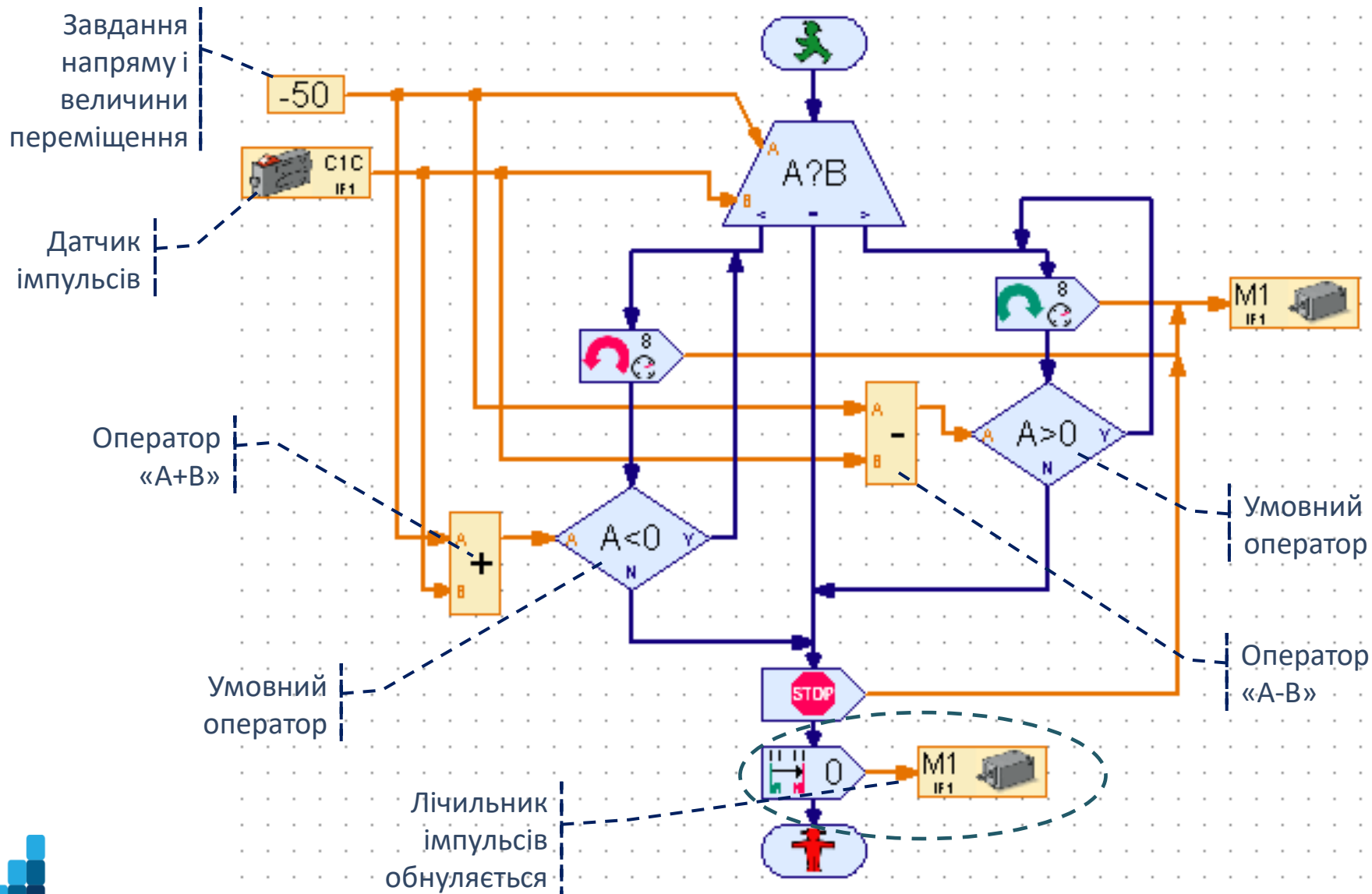
Створення підпрограми руху за допомогою лічильника імпульсів (Приклад 12)



Створення основної програми руху за допомогою лічильника імпульсів (Приклад 13)



Створення підпрограми руху “ \pm Cmoving” з визначенням напрямку руху (Приклад 14)



Дякую за увагу!

