Programowanie Sterowników Przemysłowych

Bloki Funkcyjne. Funkcje. Bloki danych

Paweł Strączyński

p.straczynski@tu.kielce.pl

Katedra Urządzeń Elektrycznych i Automatyki

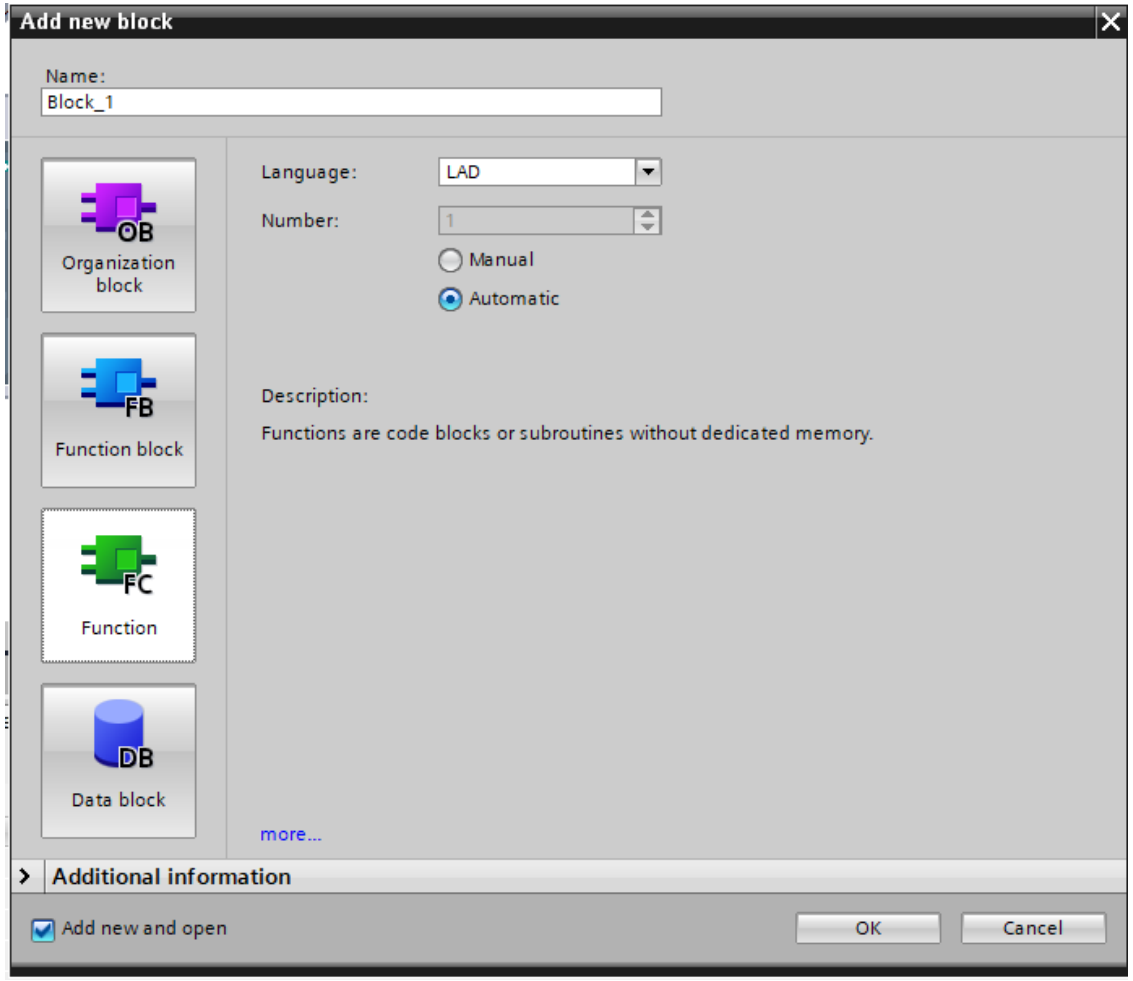
Politechnika Świętokrzyska

Funkcja

Funkcja (*Function*) to wydzielona część programu, która raz napisana może być wielokrotnie wywoływana w programie.

- funkcję można wywołać z parametrami wejściowymi,
- funkcja może zwracać parametry wyjściowe,
- poza parametrami wejściowymi i wyjściowymi pozostałe parametry używane wewnątrz funkcji są lokalne.

Dodanie funkcji



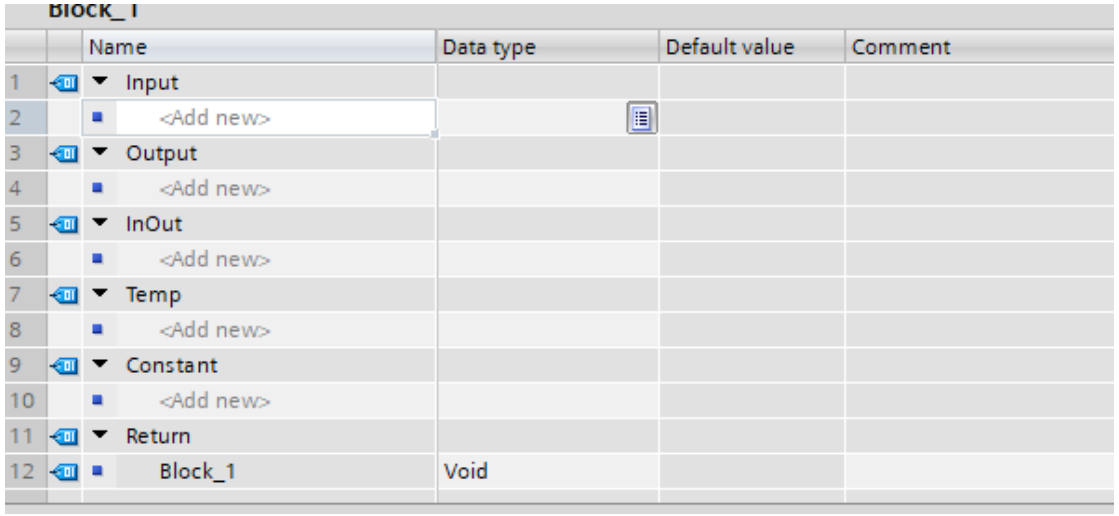
Parametry funkcji

Nazwę oraz typ danych należy określić przed pisaniem kodu programu.

Aby użyć danej w kodzie bloku funkcyjnego należy jej nazwę poprzedzić znakiem #.

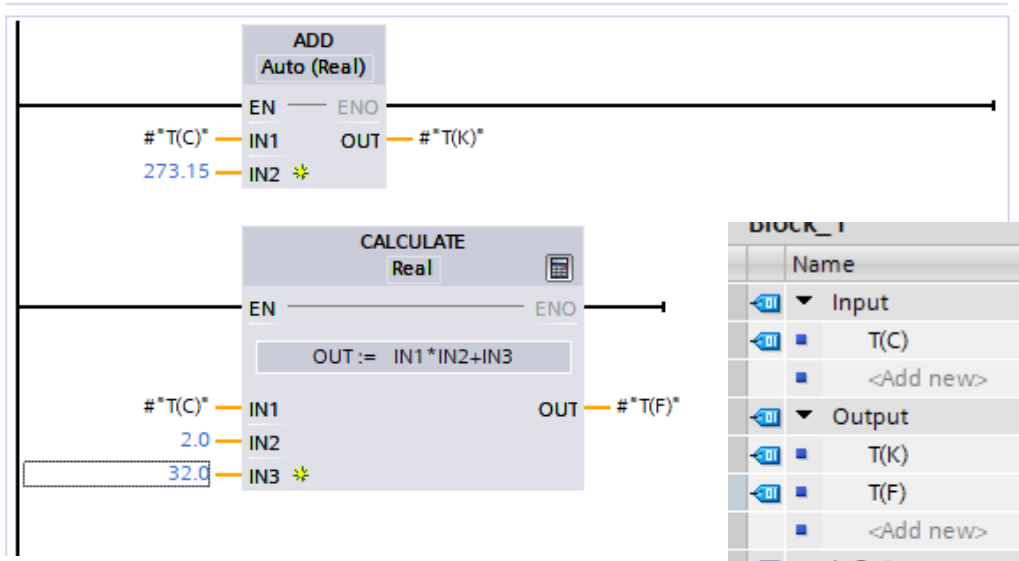
Parametry funkcji:

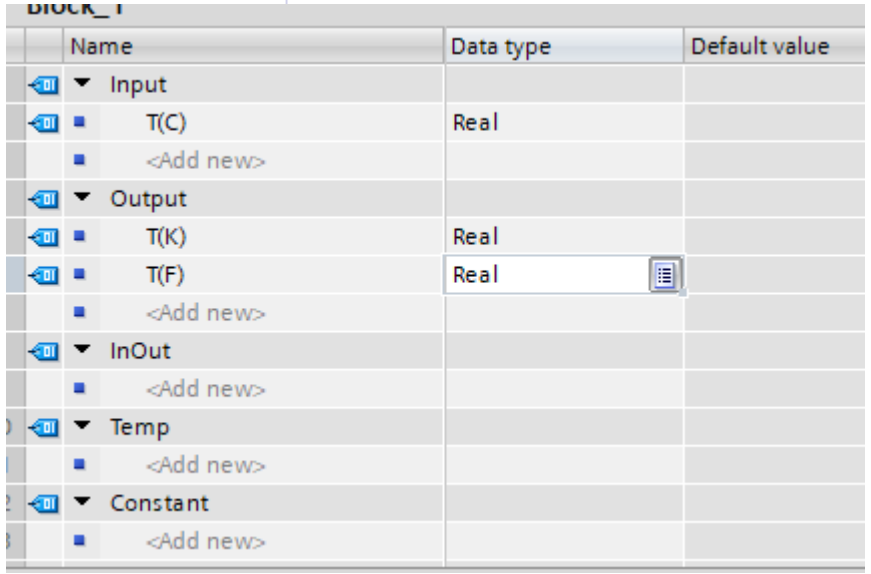
- Input - parametry wejściowe (argumenty).
- Output - parametry wyjściowe (zwracane).
- InOut - parametry wejściowe które mogą być zwracane na wyjściu
- Temp – zmienne lokalne,
- Constant – stałe.



Przykład funkcji

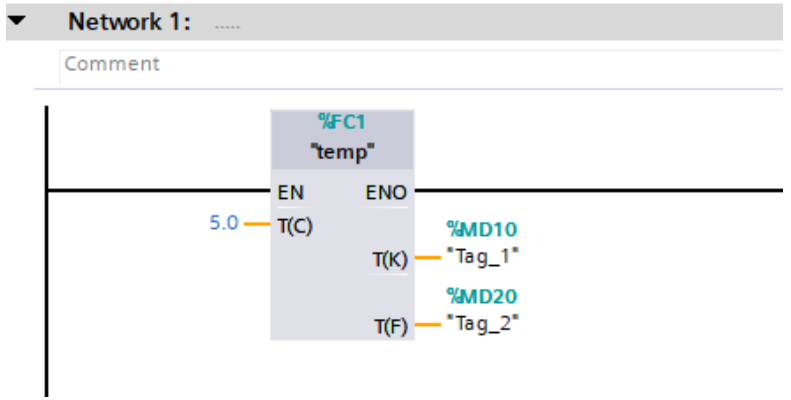
Funkcja do konwersji temperatury w stopniach Celsjusza na stopnie Farenheita i Kelwiny





$$T(K) = T(^\circ C) + 273{,}15$$

$$T(F) = 2 \times T(^\circ C) + 32$$

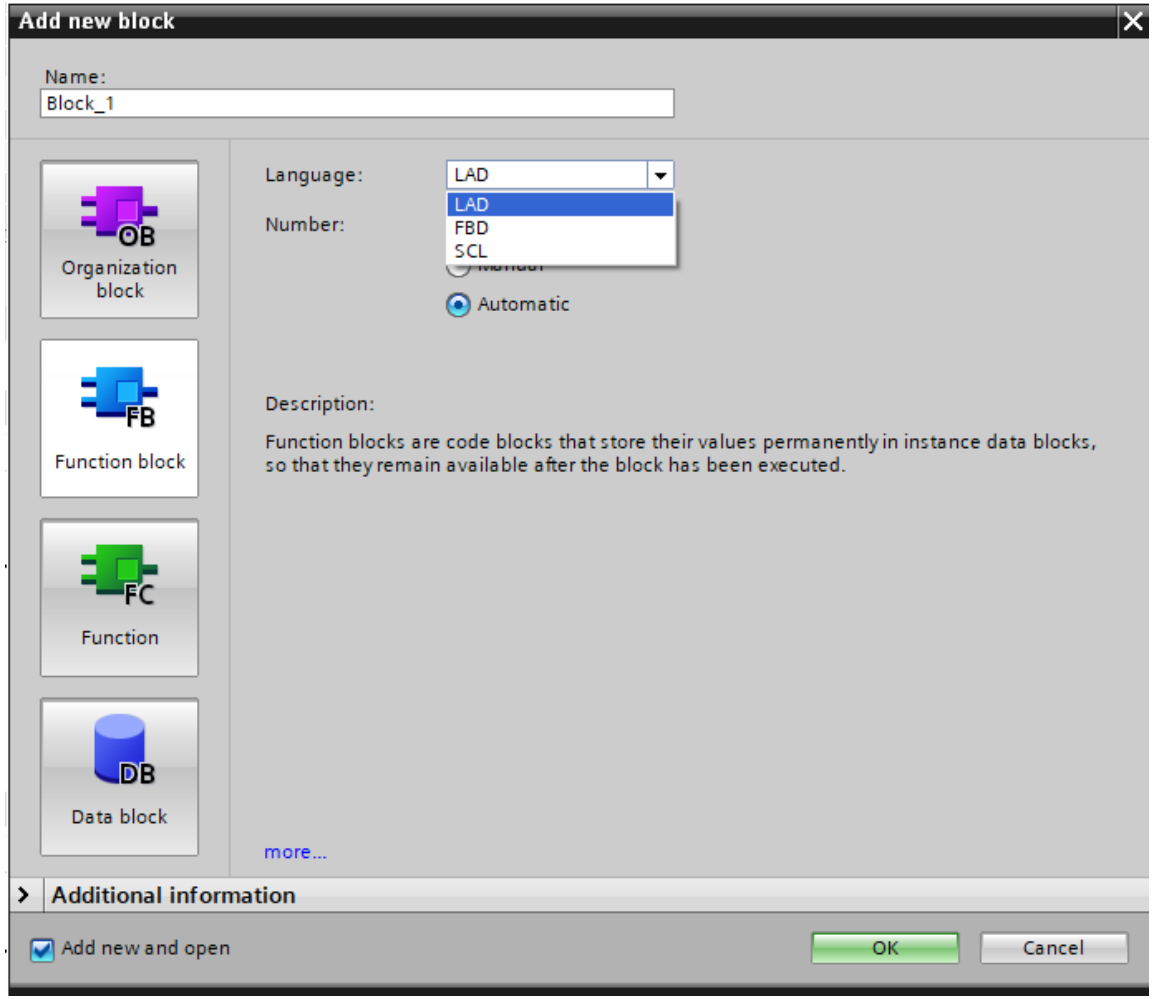


Blok funkcyjny

Blok funkcyjny (*Function block*) to wydzielona część programu, która raz napisana może być wielokrotnie wywoływana w programie.

- blok funkcyjny można wywołać z parametrami wejściowymi oraz może on zwracać parametry wyjściowe.
- blok funkcyjny tym się różni od funkcji, że zapamiętuje wartości wybranych parametrów przy każdym wywołaniu, zatem wymaga dodatkowego bloku danych, w którym te parametry są zapisane.
- każde użycie bloku funkcyjnego, to nowy blok danych, zatem możliwe jest zadeklarowanie innych danych początkowych każdego wywołania.

Dodawanie bloku funkcyjnego



Parametry bloku funkcyjnego

Nazwę oraz typ danych należy określić przed pisaniem kodu programu.

Aby użyć danej w kodzie bloku funkcyjnego należy jej nazwę poprzedzić znakiem #.

Parametry bloku funkcyjnego:

- Input - parametry wejściowe (argumenty).
- Output - parametry wyjściowe (zwracane).
- InOut - parametry wejściowe które mogą być zwracane na wyjściu
- Static - parametry dostępne tylko lokalnie (pamiętane w kolejnych wywołaniach OB)
- Temp – zmienne lokalne,
- Constant – stałe.

Przykład bloku funkcyjnego

Blok funkcyjny przelicza wartość bitową na napięcie dla różnych typów przetworników (liczba bitów i zakres pomiarowy)

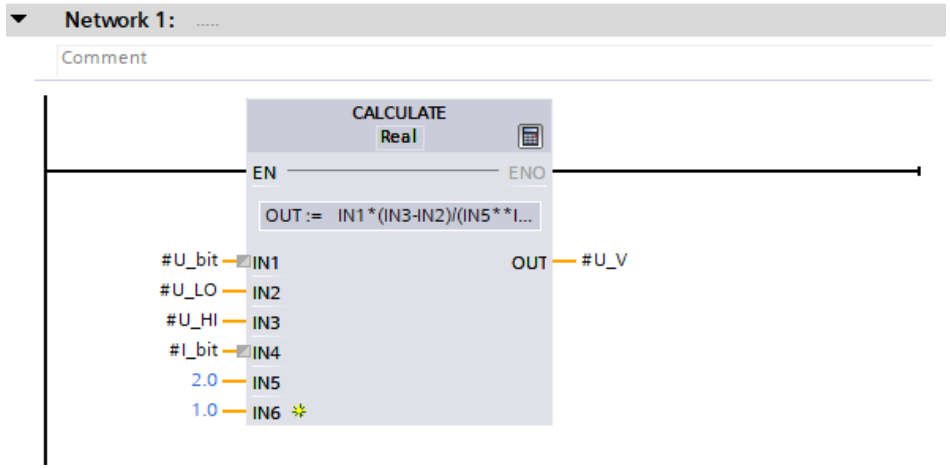
$$U_V = U_{bit}\frac{U_{HI} - U_{LO}}{2^{I\_bit} - 1} + U_{LO}$$

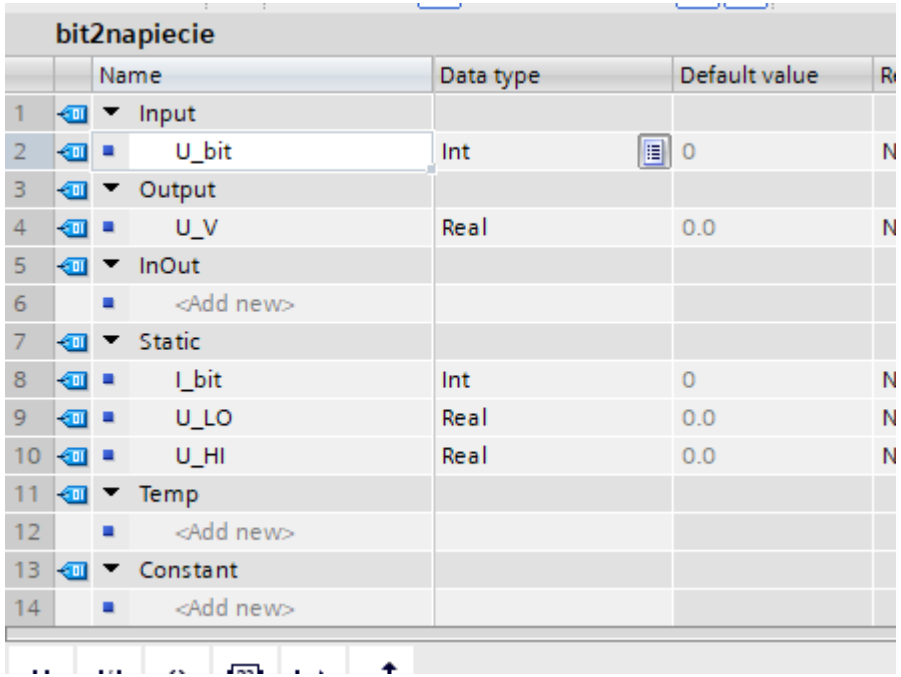
	Nazwa	Typ	Wart. pocz.	Komentarz
Input	U_bit	Int	0	wartość napięcia [bit]
Output	U_V	Real	0.0	wartość napięcia [V]
InOut	-	-	-	-
Static	I_bit	Int	0	Liczba bitów przetwornika A/C
	U_LO	Real	0.0	Dolny zakres pomiarowy [V]
	U_HI	Real	0.0	Górny zakres pomiarowy [V]
Temp	-	-	-	-
Constant	-	-	-	-

Źródło przykład: Programowanie Sterowniki Logiczne PLC, dr inż. Janusz Staszewski, prof. Uczelni. Wykład 6. Funkcje

Przykład bloku funkcyjnego

Blok funkcyjny przelicza wartość bitową na napięcie dla różnych typów przetworników (liczba bitów i zakres pomiarowy)





Źródło przykład: Programowanie Sterowniki Logiczne PLC, dr inż. Janusz Staszewski, prof. Uczelni. Wykład 6. Funkcje

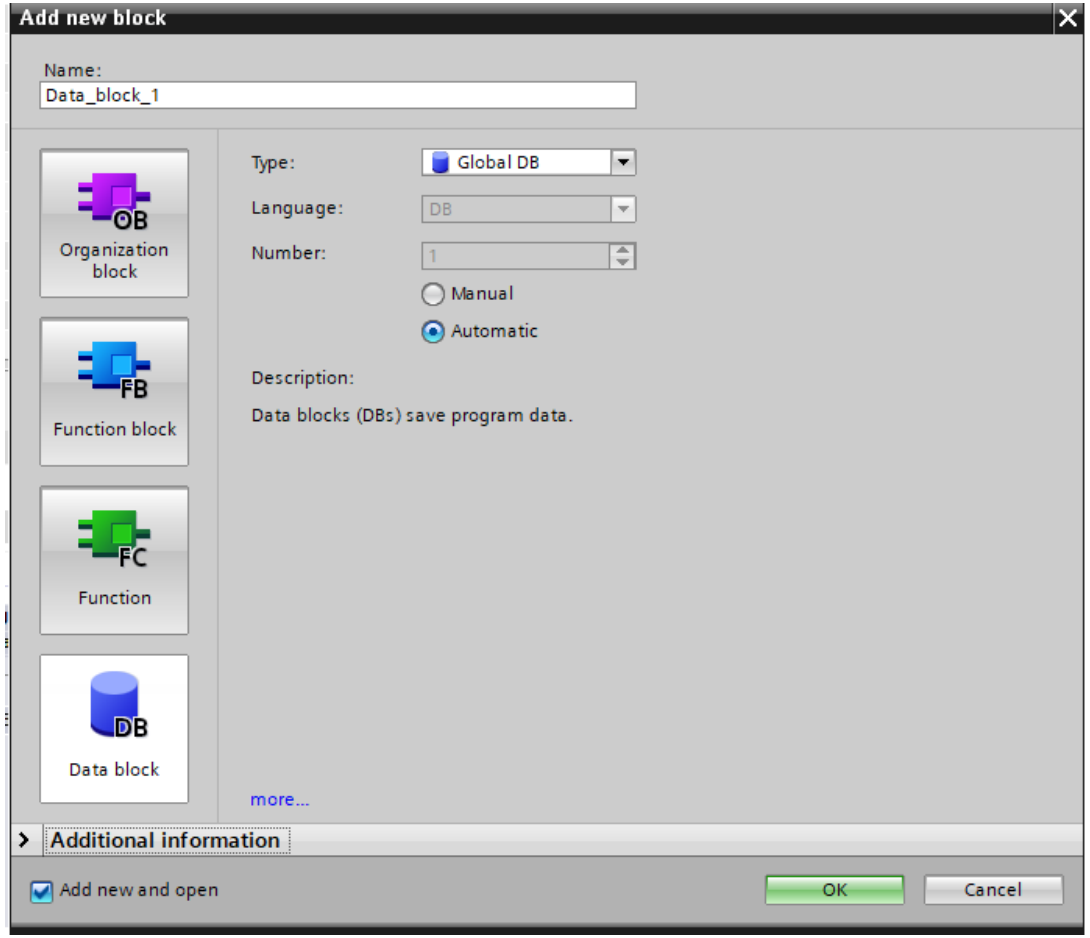
Bloki danych

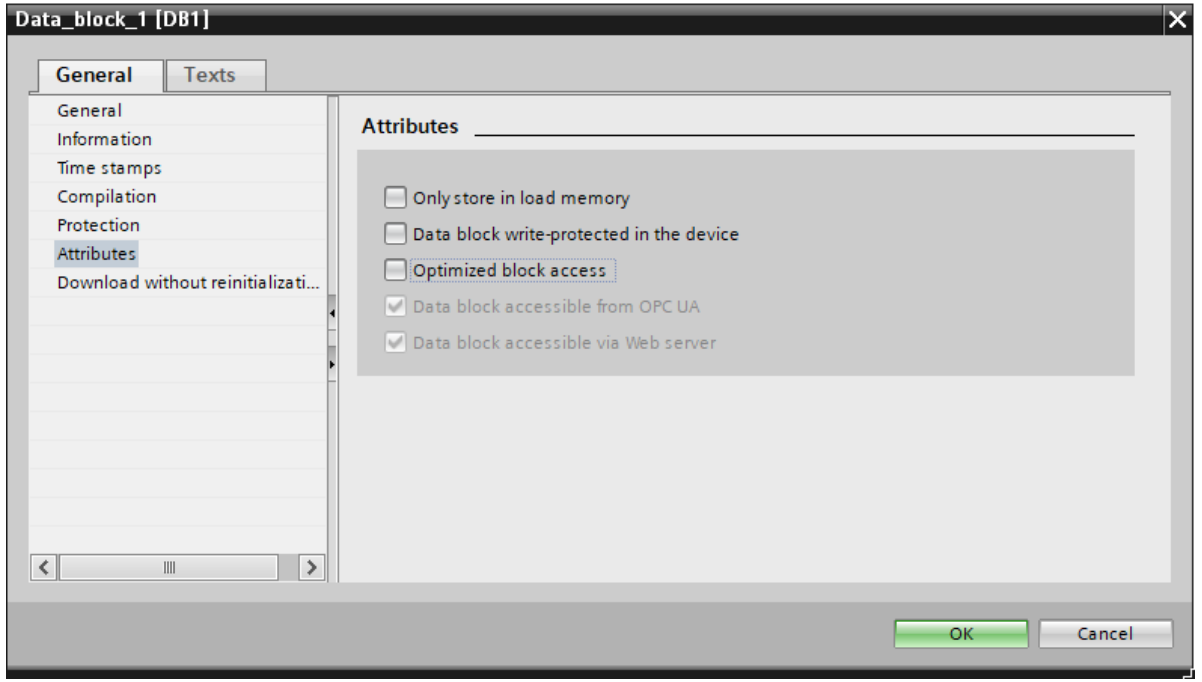
Zalety bloku danych względem pamięci M (definiowanie w tablicy tagów):

- optymalne wykorzystanie pamięci,
- brak ryzyka „nakładania” na siebie komórek pamięci,
- możliwość ustalania wartości początkowych,
- możliwość deklarowania tablic,
- możliwość deklarowania struktur.

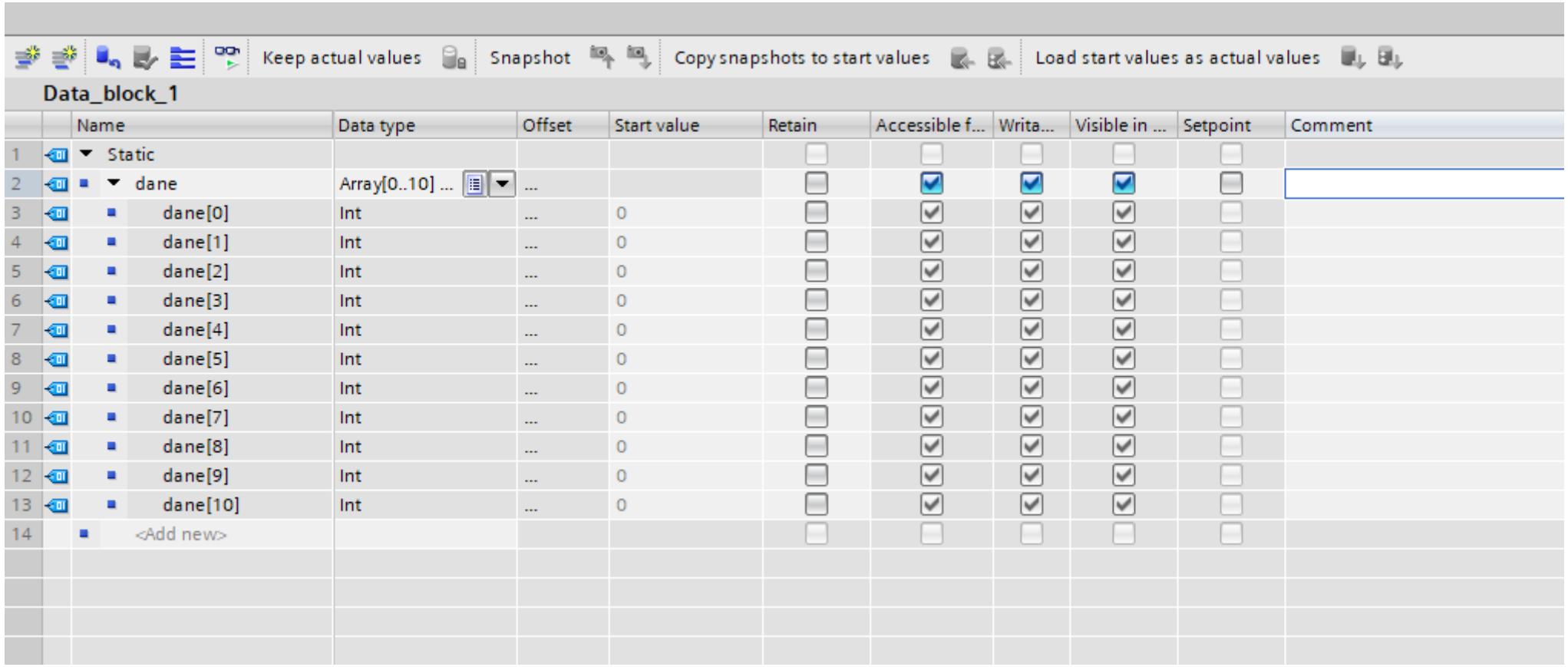
Źródło przykład: Programowanie Sterowniki Logiczne PLC, dr inż. Janusz Staszewski, prof. Uczelni. Wykład 6. Funkcje

Dodawanie bloku danych





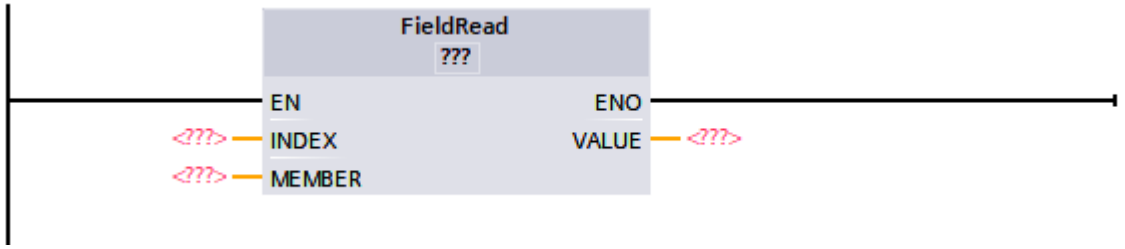
Tablice



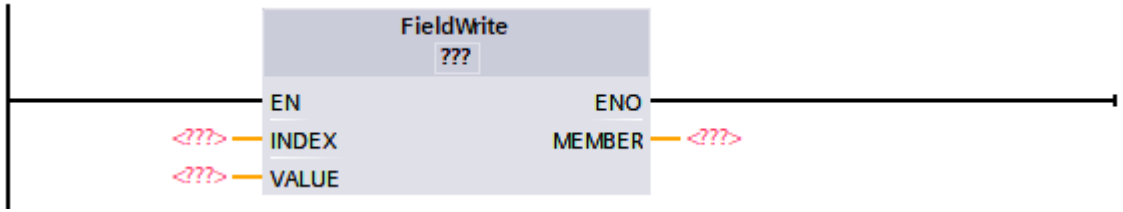
Data_block_1

	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static									
2	dane	Array[0..10] ...	...							
3	dane[0]	Int	...	0						
4	dane[1]	Int	...	0						
5	dane[2]	Int	...	0						
6	dane[3]	Int	...	0						
7	dane[4]	Int	...	0						
8	dane[5]	Int	...	0						
9	dane[6]	Int	...	0						
10	dane[7]	Int	...	0						
11	dane[8]	Int	...	0						
12	dane[9]	Int	...	0						
13	dane[10]	Int	...	0						
14	<Add new>									

Tablice – odczyt i zapis

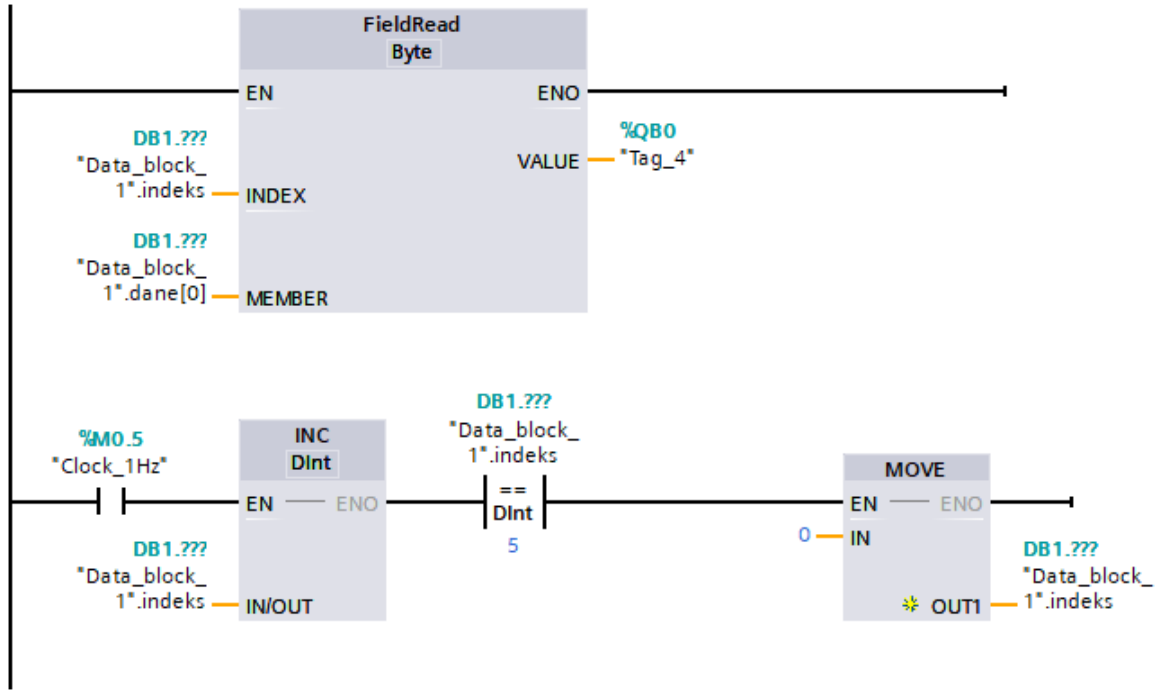


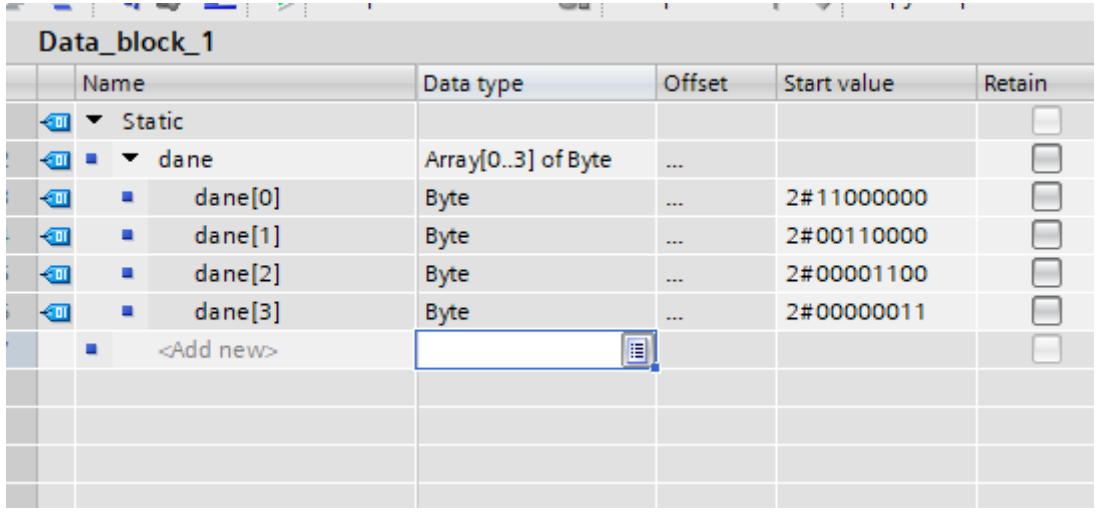
Funkcja FieldRead wpisuje VALUE element tablicy o indeksie INDEX, z tablicy której pierwszy element jest określony przez parametr MEMBER.

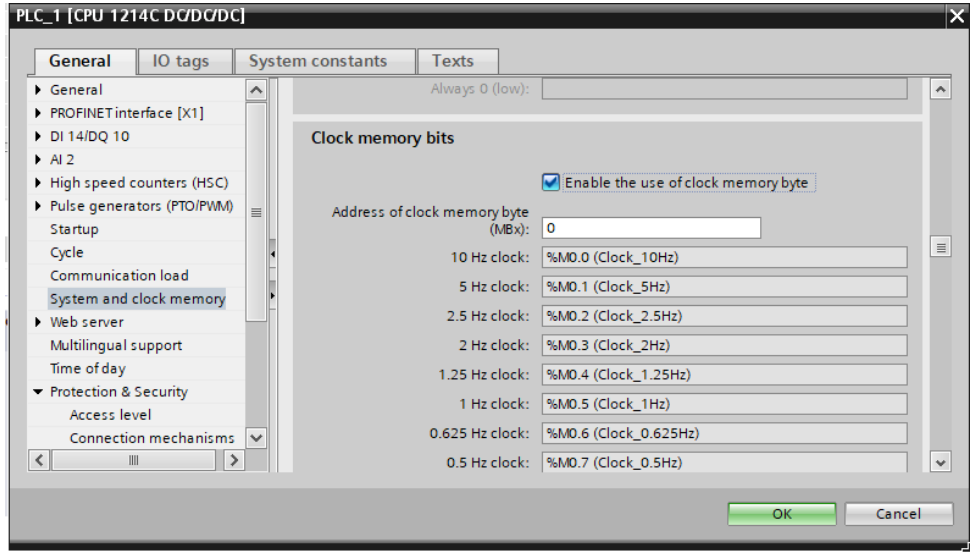


Funkcja FieldWrite zapisuje wartość VALUE do elementu tablicy, której pierwszy element jest określony przez MEMBER a element przez INDEX.

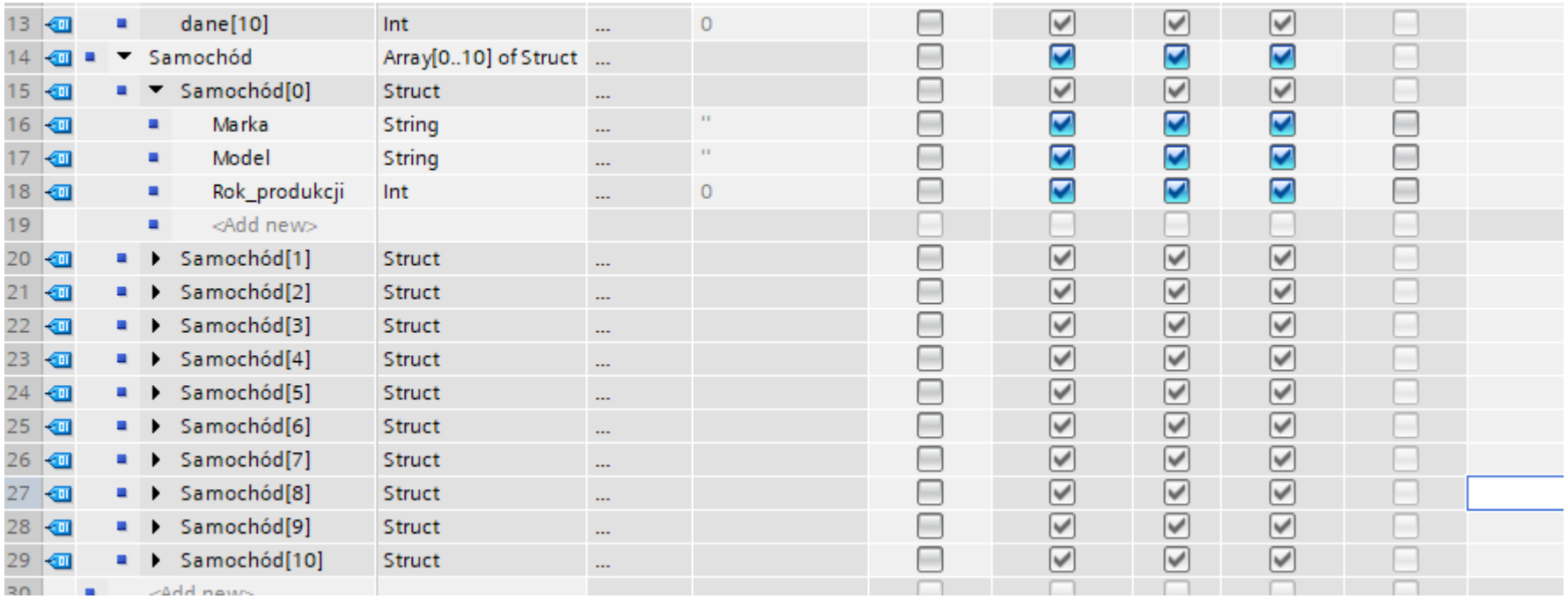
Tablice – przykład







Struktury



13		dane[10]	Int	...	0	☐	☑	☑	☑	☐
14	Samochód		Array[0..10] of Struct	...		☐	☑	☑	☑	☐
15		Samochód[0]	Struct	...		☐	☑	☑	☑	☐
16		Marka	String	...	''	☐	☑	☑	☑	☐
17		Model	String	...	''	☐	☑	☑	☑	☐
18		Rok_produkcji	Int	...	0	☐	☑	☑	☑	☐
19		<Add new>				☐	☐	☐	☐	☐
20		Samochód[1]	Struct	...		☐	☑	☑	☑	☐
21		Samochód[2]	Struct	...		☐	☑	☑	☑	☐
22		Samochód[3]	Struct	...		☐	☑	☑	☑	☐
23		Samochód[4]	Struct	...		☐	☑	☑	☑	☐
24		Samochód[5]	Struct	...		☐	☑	☑	☑	☐
25		Samochód[6]	Struct	...		☐	☑	☑	☑	☐
26		Samochód[7]	Struct	...		☐	☑	☑	☑	☐
27		Samochód[8]	Struct	...		☐	☑	☑	☑	☐
28		Samochód[9]	Struct	...		☐	☑	☑	☑	☐
29		Samochód[10]	Struct	...		☐	☑	☑	☑	☐
30		<Add new>				☐	☐	☐	☐	☐