



# Nowe funkcje w LOGO!

*Wprowadzone kilka miesięcy temu do sprzedaży nowe LOGO! wyposażono w cztery nowe funkcje specjalne, których sposób użycia przedstawimy w artykule.*

Najnowszą wersję LOGO!, oznaczoną symbolem 0BA6, prezentowaliśmy na łamach EP kilkakrotnie. Przypomnimy tylko, że zbudowano ją na bazie nowoczesnego mikrokontrolera z rodziny STR9, wyposażonego w 32-bitowy rdzeń ARM966-E. Znacznie większa niż we wcześniejszych wersjach moc obliczeniowa mikrokontrolera umożliwiła dodanie do listy wcześniej obsługiwanych funkcji specjalnych (**tab. 1**) cztery nowe:

- analog ramp control,
- pulse width modulator (PWM),
- analog math,
- analog math error detection.

Pozostałe nowe cechy i możliwości modułów LOGO! z serii 0BA6 w stosunku do modelu 0BA5 są następujące:

- bezpośrednia współpraca z panelem LOGO! TD, który umożliwia wyświetlanie komunikatów tekstowych oraz powieliła funkcje klawiatury LOGO!, ponadto wyposażono go w cztery przyciski uniwersalne o funkcjach programowanych przez użytkownika,

- współpraca z nowymi typami kart: baterijną (LOGO! Battery Card) oraz baterijną zintegrowaną z pamięcią nieulotną o pojemności 32 kB (LOGO! Combined Memory/Battery Card). Baterie zamontowane na kartach służą do podtrzymania pracy zegara RTC przez co najmniej 2 lata,
- kilka wersji LOGO! 0BA6 Basic wyposażono w dodatkowe wejścia analogowe oraz wejścia szybkiego zliczania,
- komunikaty tekstowe wyświetlane na LCD mają dodatkowe atrybuty: mogą migać, można wyświetlać bargrafy oraz wybrać jeden z dwóch zestawów wyświetlanych znaków. Komunikaty mogą być wyświetlane na wyświetlaczu modułu LOGO! Basic, na wyświetlaczu modułu LOGO! TD lub na obydwu wyświetlaczach jednocześnie,
- do programowania pamięci programu modułów LOGO! z serii 0 BA6 opracowano nowoczesny interfejs dołączany do USB komputera PC,

- możliwość obsługi prądowych wyjść analogowych 0/4...20 mA w modułach AM2 AQ,
- powiększona pamięć programu umożliwia tworzenie aplikacji składających się z maksymalnej liczby 200 bloków,
- powiększono liczbę parametrów i niektóre możliwości wybranych bloków funkcyjnych.

W tab. 1 zestawiono funkcje specjalne obsługiwane przez poszczególne wersje LOGO!

## **Analog ramp control**

Symbol graficzny tej funkcji pokazano na **rys. 1**, służy ona do generacji przebiegów analogowych zmieniających się w zadanym czasie od ustalonej przez użytkownika wartości początkowej do wartości końcowej (przykład na **rys. 2**). Wejście En służy do wyzwolenia generacji przebiegu, przy czym start następuje od wartości wynikającej ze wzoru:

$$\text{Offset B} + \text{StSp}$$

Pokazane na **rys. 2** odcinki o stałych wartościach, trwające po 100 ms wynikają ze sposobu obsługi funkcji przez LOGO!, wartości te nie są modyfikowalne.

Wartość offsetu, prędkości narastania napięcia wyjściowego, wzmocnienia, jak

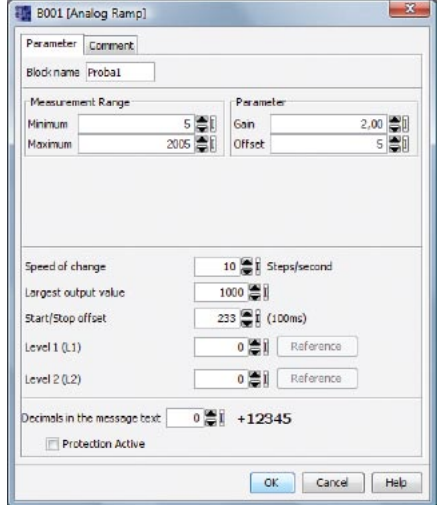
**Tab. 1. Zestawienie funkcji dostępnych w LOGO! wszystkich generacji**

Nazwa funkcji	OBA0 Standard	OBA 0...L	OBA 0...LB11	OBA1	OBA2	OBA3	OBA4	OBA5	OBA6
On delay	X	X	X	X	X	X	Xr	Xr	Xr
Off delay	X	X	X	X	X	X	Xr	Xr	Xr
Retentive on delay	X	X	X	X	X	X	Xr	Xr	Xr
On/off delay	–	–	–	–	X	X	Xr	Xr	Xr
Latching relay	X	Xr	Xr	Xr	Xr	Xr	Xr	Xr	Xr
Pulse relay	X	Xr	Xr	Xr	Xr	Xr	Xr	Xr	Xr
Wiping relay	–	X	X	X	X	X	Xr	Xr	Xr
Edge-triggered wiping relay	–	–	–	–	X	X	Xr	Xr	Xr
Weekly timer	X	X	X	X	X	X	X	X	X
Yearly timer	–	–	–	X	X	X	X	X	X
Up/down counter	X	Xr	Xr	Xr	Xr	Xr	Xr	Xr	Xr
Hours counter	–	Xr	Xr	Xr	Xr	Xr	Xr	Xr	Xr
Symmetrical pulse generator	X	X	X	X	X	X	X	X	X
Asynchronous pulse generator	–	–	–	X	X	X	Xr	Xr	Xr
Random generator	–	–	–	–	X	X	X	X	X
Analog threshold trigger	–	X	X	X	X	X	X	X	X
Analog trigger	–	–	–	–	X	X	X	X	X
Analog comparator	–	–	–	–	X	X	X	X	X
Stairway lighting switch	–	–	–	–	X	X	Xr	Xr	Xr
Multiple function switch	–	–	–	–	X	X	Xr	Xr	Xr
Message text	–	–	–	–	X	X	X	X	X
Softkey	–	–	–	–	–	Xr	Xr	Xr	Xr
Shift register	–	–	–	–	–	–	Xr	Xr	Xr
Analog value monitoring	–	–	–	–	–	–	Xr	Xr	Xr
Analog amplifier	–	–	–	–	–	–	X	X	X
Analog differential trigger SFB	–	–	–	–	–	–	X	X	X
Analog multiplexer	–	–	–	–	–	–	–	Xr	Xr
Controller PI	–	–	–	–	–	–	–	Xr	Xr
Analog ramp control	–	–	–	–	–	–	–	–	X
Pulse width modulator (PWM)	–	–	–	–	–	–	–	–	X
Analog math	–	–	–	–	–	–	–	–	X
Analog math error detection	–	–	–	–	–	–	–	–	X

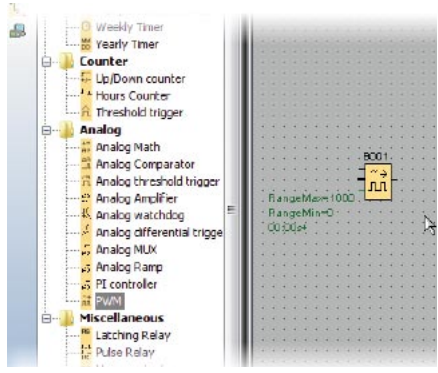
r – oznacza funkcję z możliwością podtrzymania stanu (jej parametry są przechowywane w nieulotnej pamięci sterownika)

**Tab. 2. Tabela prawdy opisująca działanie funkcji Analog math error detection**

Err	ZD	OF	Q
ZD	1	x	1
ZD	0	x	0
OF	x	1	1
OF	x	0	0
ZD/OF	1	0	1
ZD/OF	0	1	1
ZD/OF	1	1	1
ZD/OF	0	0	0



Rys. 3.



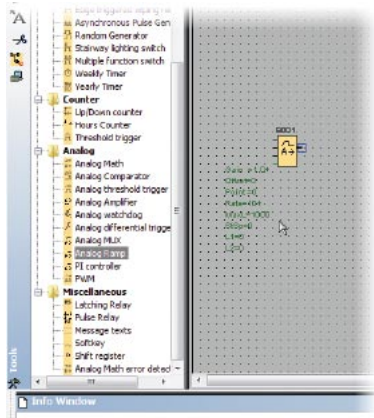
Rys. 4.

**Pulse width modulator (PWM)**

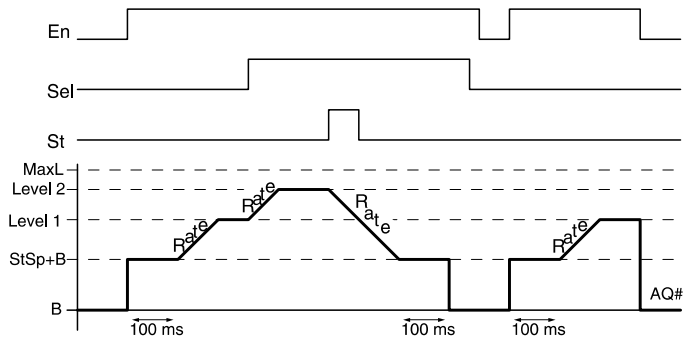
Symbol graficzny tej funkcji pokazano na rys. 4. Służy ona do konwersji sygnału analogowego na sygnał cyfrowy o zmiennym wypełnieniu. Czas trwania impulsu wyjściowego (jego szerokość) jest zależny proporcjo-

i pozostałych parametrów są konfigurowalne (w edytorze konfiguracji LOGO! Soft Com-

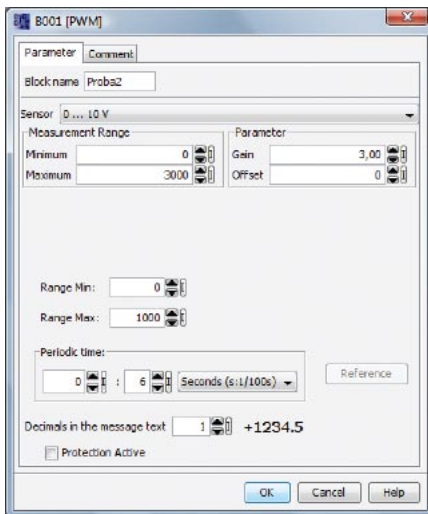
fort, za pomocą okna pokazanego na rys. 3). Dzięki zastosowaniu wielu parametrów wpływających na wynik końcowy, może on być wygodnie skalowany bez konieczności stosowania zewnętrznych kondycjonerów.



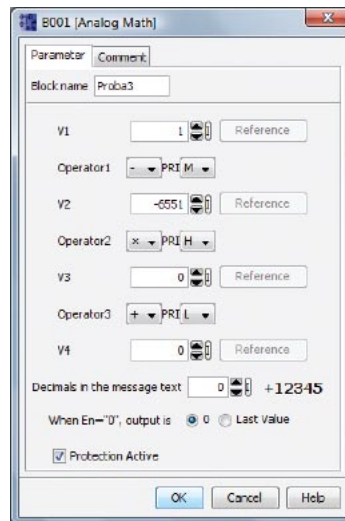
Rys. 1.



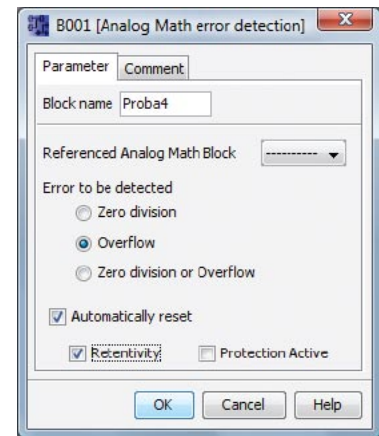
Rys. 2.



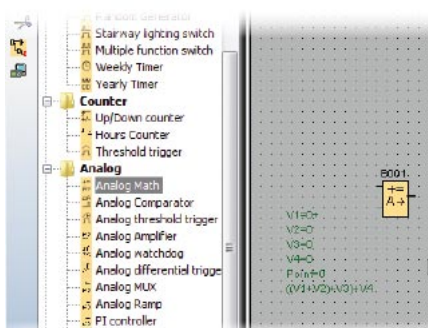
Rys. 5.



Rys. 7.



Rys. 9.



Rys. 6.

nalnie od napięcia podawanego na wejście Ax. Przetworzona wartość napięcia jest mnożona przez zadaną przez użytkownika wartość parametru A (wzmocnienie), natomiast wartość parametru B (offset) jest dodawana do wyniku mnożenia zgodnie ze wzorem:

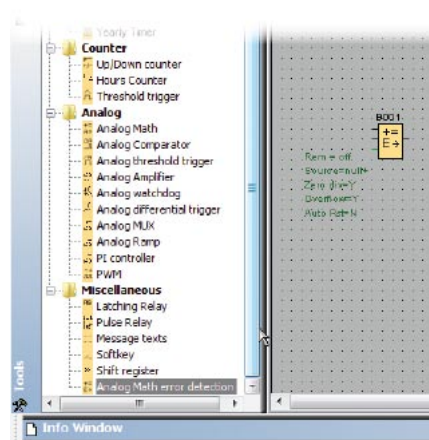
$$(Ax \cdot A) + B = \text{wynikowa wartość Ax}$$

Funkcja oblicza stosunek wynikowej wartości Ax do całego zakresu przetwarzania. Stan na wyjściu Q jest zmieniany na „1” na czas proporcjonalny do wyniku obliczeń i zmieniany na „0” na czas do końca okresu przebiegu wyjściowego.

Na rys. 5 pokazano okno konfiguracji generatora PWM.

### Analog math

Symbol graficzny tej funkcji pokazano na rys. 6. Funkcja umożliwia wykonanie operacji arytmetycznych na czterech argumentach z wykorzystaniem trzech wybranych przez użytkownika operatorów działań, według zbudowanego przez niego wzoru. W obliczeniach można korzystać z czterech standardowych operatorów: dodawania, odejmowania, mnożenia oraz dzielenia. Każdemu operatorowi można nadać jeden z trzech priorytetów:



Rys. 8.

High (H), Medium (M) lub Low (L). Działanie operatora o najwyższym priorytecie będzie wykonywane w pierwszej kolejności, następnie będzie wykonywane działanie o średnim priorytecie, na końcu o niskim priorytecie. Użytkownik musi pamiętać o tym, że każdy operator może mieć przypisany tylko jeden priorytet. Liczba argumentów i operatorów są stałe: 4 oraz 3. Jeżeli do wykonania działania jest potrzebna mniejsza liczba argumentów w miejscach niewykorzystanych należy używać konstrukcji typu +0 lub \*1.

W przypadku wykrycia próby dzielenia przez zero lub wystąpienia przepełnienia, funkcja automatycznie ustawia bity sygnalizujące rodzaj błędu. Dzięki blokowi funkcjonalnemu do detekcji błędów operacji arytmetycznych na sygnałach analogowych użytkownik może stwierdzić rodzaj błędu i ustalić odpowiednią reakcję programu. Każdy blok detekcji błędów współpracuje z jednym blokiem operacji arytmetycznych na sygnałach analogowych.

Na rys. 7 pokazano okno konfiguracji funkcji *Analog math*.

### Analog math error detection

Symbol graficzny tej funkcji pokazano na rys. 8. Na wyjściu bloku detekcji błędów operacji arytmetycznych na sygnałach analogowych pojawia się stan „1” w chwili wykrycia błędu obliczeniowego podczas pracy bloku *Analog math*. Dzięki niemu można wykryć błąd dzielenia przez zero, błąd przepełnienia lub obydwadwa rodzaje błędów. Jeżeli obliczenia przez blok operacji arytmetycznych na sygnałach analogowych nowej wartości zostało wykonane przed zainicjalizowaniem bloku detekcji błędów, sygnalizacja powstania błędu nastąpi w tym samym cyklu wykonywania programu. Jeżeli zainicjalizowanie bloku detekcji błędów nastąpi później, sygnalizacja wystąpienia błędu nastąpi w kolejnym cyklu. W tab. 2 pokazano sposób działania bloku *Analog math error detection*, przy czym:

- Q to wyjście funkcji,
- ZD odpowiada dzieleniu przez zero,
- OF odpowiada błędowi przepełnienia,
- ZD/OF odpowiada wystąpieniu błędu dzielenia przez zero lub błędowi przepełnienia.

Na rys. 9 pokazano okno konfiguracji funkcji *Analog math error detection*.

Przedstawione w artykule nowe funkcje dostępne w LOGO! trudno jest uznać za rewolucyjne, ale są one bez wątpienia (dobrze to widać w tab. 1) elementem konsekwentnego rozwoju możliwości wcześniejszych modeli. Zwiększone możliwości komunikacyjne (w tym możliwość bezpośredniej współpracy z modemami), dostępność interfejsu do programowania z USB, znacznie powiększona pojemność pamięci programu i inne pomniejsze udoskonalenia powodują, że LOGO! w wersji 0BA6 stało się jeszcze bardziej przyjazne użytkownikom.

Andrzej Gawryluk