



# ***GE Fanuc Automation***

---

***Sterowniki programowalne***

***Zestaw instrukcji dla jednostek centralnych  
sterowników 90™-30/20/Micro***

***Podręcznik użytkownika***

GFK-0467M-PL

Maj 2002

## Znaki ostrzegawcze i uwagi

### Niebezpieczeństwo

Symbolem tym oznaczono w niniejszym podręczniku informacje o niebezpiecznie wysokich napięciach, dużych prądach, temperaturach oraz innych czynnikach, związanych z urządzeniem lub współpracującym z nim sprzętem, które mogą spowodować obrażenia cielesne lub uszkodzenie urządzenia.

Miejsca, w których nieuwaga mogłaby spowodować obrażenia lub uszkodzenie urządzenia oznaczono słowem „Ostrzeżenie”

### Ostrzeżenie

Symbolem tym oznaczono informacje, których nieprzestrzeżenie może prowadzić do uszkodzenia urządzenia.

### Uwaga

Symbolem tym oznaczono informacje o szczególnie dużym znaczeniu dla zrozumienia zasad eksploatacji i użytkowania urządzenia.

Niniejszy podręcznik przygotowano w oparciu o informacje dostępne w czasie publikacji. Podjęto wszelkie starania, aby zamieszczone informacje były dokładne, nie mniej jednak nie można zagwarantować, że uwzględnione zostały wszystkie szczegółowe dane i zmiany wprowadzone w sprzęcie i oprogramowaniu, jak również nie jest możliwe uwzględnienie wszystkich sytuacji, które mogą wystąpić w czasie instalowania, obsługi lub konserwacji urządzenia. Mogą występować różnice pomiędzy opisami zamieszczonymi w niniejszym dokumencie a dostarczonymi urządzeniami i oprogramowaniem. GE Fanuc Automation nie zobowiązuje się do informowania właścicieli niniejszego podręcznika o wprowadzanych zmianach.

GE Fanuc Automation nie udziela żadnej gwarancji, jawnie sprecyzowanej lub domniemanej oraz nie ponosi odpowiedzialności za dokładność, kompletność, pełność oraz użyteczność informacji zawartych w niniejszym podręczniku. Nie jest udzielana gwarancja na przydatność handlową i techniczną.

Podane poniżej znaki towarowe są zastrzeżone przez GE Fanuc Automation North America, Inc.

Alarm Master	Genius	PROMACRO	Series Six
CIMPLICITY	Helpmate	PowerMotion	Series Three
CIMPLICITY 90-ADS	Logicmaster	PowerTRAC	VersaMax
CIMSTAR	Modelmaster	Series 90	VersaPro
Field Control	Motion Mate	Series Five	VuMaster
GENet	ProLoop	Series One	Workmaster

W niniejszej instrukcji opisano działanie instrukcje oprogramowania Logicmaster90™ dla sterowników serii 90™-30 i 90-20 i 90-Micro Sterowniki programowalne serii 90-30, 90-20 i 90 Micro wchodzą w skład większej rodziny sterowników serii 90 produkcji GE Fanuc Automation.

## Zmiany uwzględnione w Podręczniku

- Jednostka centralna 374 obsługuje połączenia z siecią Ethernet za pośrednictwem dwóch portów 10BaseT/100BaseTx z automatycznym trybem full duplex. Jednostki centralne CPU364 (wersja 9.10 lub nowsza) oraz CPU374 obsługują Ethernet Global Data (EGD). Należy zwrócić uwagę, że CPU374 jest obsługiwane tylko przez oprogramowanie działające w systemie Windows.
- Wprowadzono inne zmiany i poprawki, stosownie do potrzeb.

## Literatura uzupełniająca

*Logicmaster™ 90 Series 90™-30/20/Micro Programming Software User's Manual* (GFK-0466).

*VersaPro™ Programming Software User's Guide* (GFK-1670)

*CIMPLICITY® Machine Edition Getting Started* (GFK-1868)

*Series 90™-30 Programmable Controller Installation Manual* (GFK-0356)

*Series 90™-20 Programmable Controller Installation Manual* (GFK-0551)

*Series 90™-30 I/O Module Specifications Manual* (GFK-0898)

*Series 90™ Programmable Coprocessor Module and Support Software User's Manual* (GFK-0255)

*Series 90™ PCM Development Software (PCOP) User's Manual* (GFK-0487)

*CIMPLICITY™ 90-ADS Alphanumeric Display System User's Manual* (GFK-0499)

*CIMPLICITY™ 90-ADS Alphanumeric Display System Reference Manual* (GFK-0641)

*Series 90™-30 and 90-20 PLC Hand-Held Programmer User's Manual* (GFK-0402)

*Power Mate APM for Series 90™-30 PLC—Standard Mode User's Manual* (GFK-0840)

*Power Mate APM for Series 90™-30 PLC—Follower Mode User's Manual* (GFK-0781)

*Motion Mate™ DSM302 for Series 90™-30 PLCs User's Manual* (GFK-1464)

*Series 90™-30 High Speed Counter User's Manual* (GFK-0293)

*Series 90™-30 Genius Communications Module User's Manual* (GFK-0412)

*Series 90™-30 Genius™ Bus Controller User's Manual* (GFK-1034)

*Series 90™-70 FIP Bus Controller User's Manual* (GFK-1038)

*Series 90™-30 FIP Remote I/O Scanner User's Manual* (GFK-1037)

*Field Control™ Distributed I/O and Control System Genius™ Bus Interface Unit User's Manual (GFK-0825)*

*Series 90™ Micro Programmable Logic Controller User's Manual (GFK-1065)*

*Series 90™ PLC Serial Communications User's Manual (GFK-0582)*

<b>Rozdział 1</b>	<b>Wprowadzenie .....</b>	<b>1-1</b>
<b>Rozdział 2</b>	<b>Działanie systemu .....</b>	<b>2-1</b>
	<b>Część 1: Cykl pracy sterownika .....</b>	<b>2-2</b>
	Standardowy cykl pracy sterownika .....	2-2
	Obliczanie czasu trwania cyklu pracy sterownika .....	2-7
	Szczegółowy opis cyklu pracy sterownika .....	2-8
	Komunikacja modułu programowalnego procesora (PCM) ze sterownikiem (Modele 331 i wyższe).....	2-12
	Komunikacja modułu DSM ze sterownikiem.....	2-13
	Warianty standardowego cyklu pracy sterownika.....	2-13
	Tryb ze stałym czasem trwania cyklu pracy sterownika.....	2-13
	Cykl pracy sterownika w trybie STOP .....	2-14
	Tryby realizacji faz komunikacji .....	2-14
	Przełącznik w jednostkach centralnych serii 35x, 36x i 37x: Zmiany trybu i zabezpieczania przed zapisem .....	2-15
	<b>Część 2: Organizacja programu, typy danych i typy zmiennych.....</b>	<b>2-17</b>
	Bloki podprogramów (wyłącznie dla sterowników serii 90-30).....	2-18
	Przykłady zastosowania podprogramów.....	2-18
	Wywoływanie podprogramów w szczeblu programu sterującego .....	2-19
	Wykonywanie programu zawierającego podprogramy .....	2-19
	Podprogramy wykonywane okresowo .....	2-20
	Typy zmiennych .....	2-20
	Nazwy pomocnicze.....	2-22
	Wymuszenie zmiany wartości zmiennych.....	2-22
	Pamięć stanu .....	2-22
	Typy danych .....	2-23
	Zmienne systemowe statusu .....	2-24
	Podstawowe informacje o instrukcjach języka drabinkowego .....	2-26
	Format przekaźników i styków .....	2-26
	Format bloków funkcyjnych.....	2-26
	Parametry bloków funkcyjnych (instrukcji) .....	2-28
	Przepływ sygnału przez blok funkcyjny .....	2-29
	<b>Część 3: Rozruch i wyłączenie sterownika .....</b>	<b>2-31</b>
	Rozruch sterownika .....	2-31
	Wyłączanie sterownika.....	2-34
	<b>Część 4: Zegary i przekaźniki czasowe.....</b>	<b>2-35</b>
	Zegar odmierzający czas pracy sterownika .....	2-35
	Zegar czasu rzeczywistego .....	2-35
	Zegar alarmowy .....	2-36

Zegar czasu trwania przerwy w zasilaniu sterownika .....	2-36
Zegar cyklu pracy o stałym czasie trwania.....	2-36
Generator sygnału prostokątnego .....	2-37
<b>Część 5: Mechanizmy zabezpieczające .....</b>	<b>2-38</b>
Hasła.....	2-38
Zmiana poziomu dostępu.....	2-39
Blokowanie/odblokowywanie podprogramów .....	2-39
Trwała blokada podprogramów .....	2-39
<b>Część 6: Moduły wejść/wyjść sterowników serii 90-3090-30, 90-20 i Micro.....</b>	<b>2-40</b>
Moduł wejść/wyjść sterowników serii 90-30 .....	2-41
Formaty danych wejściowych/wyjściowych .....	2-43
Domyślne wartości dla modułów wyjściowych serii 90-30 .....	2-43
Dane diagnostyczne .....	2-44
Dane globalne .....	2-44
Dane globalne Genius.....	2-44
Sieć Ethernet.....	2-44
Moduł wejść/wyjść sterowników serii 90-20 .....	2-45
Konfigurowanie i programowanie.....	2-45
<b>Rozdział 3 Błędy działania sterownika - opis i ich usuwania.....</b>	<b>3-1</b>
<b>Część 1: Obsługa błędów.....</b>	<b>3-2</b>
Procesor Alarmowy .....	3-2
Klasy błędów .....	3-2
Reakcja systemu na błędy działania .....	3-3
Tabele błędów działania .....	3-3
Wagi błędów .....	3-4
Zmienne do obsługi błędów.....	3-4
Zmienne systemowe .....	3-4
Dodatkowe skutki wystąpienia błędów działania.....	3-5
Tabela błędów działania sterownika.....	3-5
Tabela błędów działania układów wejść/wyjść .....	3-5
Wyświetlanie dodatkowych informacji o błędzie.....	3-6
<b>Część: Tabela błędów działania sterownika.....</b>	<b>3-7</b>
Wagi błędów .....	3-8
Loss of, or Missing, Option Module (Brak lub uszkodzenie modułu dodatkowego) 3-8	
Reset of, Addition of, or Extra, Option Module (Ponowne uruchomienie, dodanie lub brak konfiguracji dla modułu).....	3-8
System Configuration Mismatch (Błędna konfiguracja systemu) .....	3-9

Option Module Software Failure (Błąd w oprogramowaniu wyspecjalizowanego modułu dodatkowego) .....	3-10
Program Block Checksum Failure (Błędna suma kontrolna programu).....	3-10
Low Battery Signal (Rozładowane baterie).....	3-10
Constant Sweep Time Exceeded (Przekroczony czas trwania cyklu pracy).....	3-11
Application Fault (Błąd w działaniu programu sterującego).....	3-11
No User Program Present (Brak programu sterującego) .....	3-12
Corrupted User Program on Power-Up (Wykrycie uszkodzenia programu sterującego podczas rozruchu).....	3-12
Password Access Failure (Niepoprawne hasło dostępu do sterownika).....	3-12
PLC CPU System Software Failure (Błąd w oprogramowaniu jednostki centralnej sterownika) .....	3-13
Communications Failure During Store (Awaria komunikacji podczas przesyłania danych do sterownika) .....	3-15
<b>Część 3: Tabela błędów działania układów wejść/wyjść .....</b>	<b>3-16</b>
Loss of I/O Module (Brak zadeklarowanego modułu wejść/wyjść).....	3-16
Addition of I/O Module (Dołączony nie zadeklarowany moduł wejść/wyjść)...	3-17
<b>Rozdział 4 Funkcje przekaźników.....</b>	<b>4-1</b>
Styki.....	4-1
Przekaźniki .....	4-2
Styk normalnie otwarty —   —.....	4-3
Styk normalnie zwarty — / — .....	4-3
Przykład .....	4-3
Przekaźnik —( )—.....	4-3
Przykład .....	4-3
Przekaźnik o stykach zwartych —(/)—.....	4-4
Przykład.....	4-4
Przekaźnik o stykach otwartych z pamięcią —(M)—.....	4-4
Przekaźnik o stykach zamkniętych, z pamięcią —(/M)—.....	4-4
Przekaźnik uaktywniany zboczem narastającym sygnału —(↑)—.....	4-4
Przekaźnik uaktywniany zboczem opadającym sygnału —(↓)—.....	4-5
Przykład .....	4-5
Przekaźnik ustawialny SET —(S) —.....	4-5
Przekaźnik ustawialny RESET —(R)—.....	4-5
Przykład .....	4-6
Przekaźnik SET z pamięcią —(SM)—.....	4-6
Przekaźnik RESET z pamięcią —(RM)—.....	4-6
Szyny .....	4-7
Przykład .....	4-7
Przekaźniki kontynuacji (—<+>) i styki (<+>—) kontynuacji .....	4-8

<b>Rozdział 5</b>	<b>Liczniki i przekaźniki czasowe.....</b>	<b>5-1</b>
	Blok danych sterujących przekaźnikami czasowymi i liczników .....	5-1
	ONDTR .....	5-3
	Parametry .....	5-4
	Dopuszczalne typy zmiennych .....	5-4
	Przykład .....	5-5
	TMR .....	5-5
	Parametry .....	5-6
	Dopuszczalne typy zmiennych .....	5-6
	Przykład .....	5-7
	OFDT .....	5-8
	Parametry .....	5-9
	Dopuszczalne typy zmiennych .....	5-10
	Przykłady .....	5-10
	UPCTR .....	5-11
	Parametry .....	5-11
	Dopuszczalne typy zmiennych .....	5-12
	Przykłady .....	5-12
	DNCTR.....	5-13
	Parametry .....	5-13
	Dopuszczalne typy zmiennych .....	5-14
	Przykłady .....	5-14
	Przykład .....	5-15
<b>Rozdział 6</b>	<b>Funkcje matematyczne .....</b>	<b>6-1</b>
	Standardowe funkcje matematyczne (ADD, SUB, MUL, DIV) .....	6-2
	Parametry .....	6-3
	Dopuszczalne typy zmiennych .....	6-3
	Przykładowy funkcji matematycznych .....	6-4
	Funkcje matematyczne i typy danych.....	6-5
	Przykład .....	6-6
	MOD (INT, DINT).....	6-7
	Parametry .....	6-7
	Dopuszczalne typy zmiennych .....	6-8
	Przykład .....	6-8
	SQRT (INT, DINT, REAL).....	6-9
	Parametry .....	6-9
	Dopuszczalne typy zmiennych .....	6-10
	Przykłady .....	6-10
	Funkcje trygonometryczne (SIN, COS, TAN, ASIN, ACOS, ATAN) .....	6-11
	Parametry .....	6-12
	Dopuszczalne typy zmiennych .....	6-12
	Przykład .....	6-12
	Funkcje logarytmiczne/wykładnicze (LOG, LN, EXP, EXPT).....	6-13



	Parametry .....	6-13
	Dopuszczalne typy zmiennych .....	6-14
	Przykład .....	6-14
	Konwersja miar kąta (RAD, DEG) .....	6-15
	Parametry .....	6-15
	Dopuszczalne typy zmiennych .....	6-15
	Przykład .....	6-16
<b>Rozdział 7</b>	<b>Funkcje relacji.....</b>	<b>7-1</b>
	Standardowe funkcje relacji (EQ, NE, GT, GE, LT, LE).....	7-2
	Parametry .....	7-2
	Rozszerzony opis .....	7-3
	Dopuszczalne typy zmiennych .....	7-3
	Przykład .....	7-3
	RANGE (INT, DINT, WORD) .....	7-4
	Parametry .....	7-5
	Dopuszczalne typy zmiennych .....	7-5
	Przykład 1 .....	7-5
	Przykład 2 .....	7-6
<b>Rozdział 8</b>	<b>Funkcje do operacji na bitach.....</b>	<b>8-1</b>
	AND i OR (WORD) .....	8-3
	Parametry .....	8-3
	Dopuszczalne typy zmiennych .....	8-4
	Przykład .....	8-4
	XOR (WORD).....	8-5
	Parametry .....	8-5
	Dopuszczalne typy zmiennych .....	8-6
	Przykład obwodu alarmowego.....	8-6
	NOT (WORD) .....	8-7
	Parametry .....	8-7
	Dopuszczalne typy zmiennych .....	8-7
	Przykład .....	8-7
	SHL i SHR (WORD).....	8-8
	Parametry .....	8-9
	Dopuszczalne typy zmiennych .....	8-9
	Przykład .....	8-9
	ROL i ROR (WORD).....	8-10
	Parametry .....	8-10
	Dopuszczalne typy zmiennych .....	8-11
	Przykład .....	8-11
	BTST (WORD).....	8-12
	Parametry .....	8-12
	Dopuszczalne typy zmiennych .....	8-13

Przykład .....	8-13
BSET i BCLR (WORD) .....	8-14
Parametry .....	8-14
Dopuszczalne typy zmiennych .....	8-15
Przykłady .....	8-15
BPOS (WORD) .....	8-16
Parametry .....	8-16
Dopuszczalne typy zmiennych .....	8-17
Przykład .....	8-17
MSKCMP (WORD, DWORD) .....	8-18
Parametry .....	8-19
Dopuszczalne typy zmiennych .....	8-19
Przykład 1 – instrukcja MSKCMP .....	8-20
Przykład 2 – Detekcja błędu z użyciem maski .....	8-21
<b>Rozdział 9</b>	
<b>Funkcje do operacji na danych .....</b>	<b>9-1</b>
MOVE (BIT, INT, WORD, REAL) .....	9-2
Parametry .....	9-3
Przykład 1 - Nakładanie się adresów (tylko CPU 311-341) .....	9-4
Przykład 2 – wszystkie jednostki centralne .....	9-4
BLKMOV (INT, WORD, REAL) .....	9-5
Parametry .....	9-5
Dopuszczalne typy zmiennych .....	9-6
Przykład .....	9-6
BLKCLR (WORD) .....	9-7
Parametry .....	9-7
Dopuszczalne typy zmiennych .....	9-7
Przykład .....	9-7
SHFR (BIT, WORD) .....	9-8
Parametry .....	9-9
Dopuszczalne typy zmiennych .....	9-9
Przykład 1 .....	9-10
Przykład 2 .....	9-10
BITSEQ (BIT) .....	9-11
Zapotrzebowanie na pamięć bloku funkcyjnego BIT_SEQ .....	9-12
Parametry .....	9-13
Dopuszczalne typy zmiennych .....	9-13
Przykład .....	9-14
COMMREQ .....	9-15
Blok danych sterujących .....	9-15
Parametry .....	9-16
Dopuszczalne typy zmiennych .....	9-16
Przykład .....	9-17

<b>Rozdział 10</b>	<b>Operacje tablicowe.....</b>	<b>10-1</b>
	ARRAY_MOVE (INT, DINT, BIT, BYTE, WORD).....	10-2
	Definiowanie tablic i elementów danych.....	10-2
	Numery indeksu.....	10-2
	Instrukcja Move Array.....	10-2
	Parametry.....	10-4
	Dopuszczalne typy zmiennych.....	10-4
	Przykład 1.....	10-5
	Przykład 2.....	10-5
	Przykład 3.....	10-6
	Funkcje do szukania danych.....	10-7
	Parametry.....	10-8
	Dopuszczalne typy zmiennych.....	10-8
	Przykład 1.....	10-9
	Przykład 2.....	10-10
<b>Rozdział 11</b>	<b>Funkcje konwersji.....</b>	<b>11-1</b>
	—>BCD-4 (INT).....	11-2
	Parametry.....	11-2
	Dopuszczalne typy zmiennych.....	11-2
	Przykład.....	11-2
	—>INT (BCD-4, REAL).....	11-3
	Parametry.....	11-3
	Dopuszczalne typy zmiennych.....	11-3
	Przykład 1 - konwersja danych typu BCD4 na INT.....	11-4
	Przykład 2 – konwersja danych REAL na INT.....	11-4
	—>DINT (REAL).....	11-5
	Parametry.....	11-5
	Dopuszczalne typy zmiennych.....	11-5
	Przykład.....	11-6
	—>REAL (INT, DINT, BCD-4, WORD).....	11-7
	Parametry.....	11-7
	Dopuszczalne typy zmiennych.....	11-7
	Przykład 1 – konwersja danych typu INT na REAL.....	11-8
	Przykład 2 – konwersja danych typu DINT na REAL.....	11-8
	—>WORD (REAL).....	11-9
	Parametry.....	11-9
	Dopuszczalne typy zmiennych.....	11-9
	Przykład – konwersji danych typu REAL na WORD.....	11-10
	TRUN (INT, DINT).....	11-11
	Parametry.....	11-11
	Dopuszczalne typy zmiennych.....	11-11
	Przykład 1 – Konwersja liczby typu REAL na INT (CPU352).....	11-12
	Przykład 2 – Konwersja liczby typu REAL na DINT (CPU352).....	11-12

<b>Rozdział 12</b>	<b>Funkcje sterujące .....</b>	<b>12-1</b>
	CALL.....	12-2
	Przykład .....	12-2
	DOIO .....	12-3
	Parametry .....	12-4
	Dopuszczalne typy zmiennych .....	12-4
	Przykład 1 - Uaktualniania stanu wejść .....	12-5
	Przykład 2 - Uaktualniania stanu wejść .....	12-5
	Przykład 1 - Uaktualniania stanu wyjść .....	12-6
	Przykład 2 - Uaktualniania stanu wyjść .....	12-6
	Rozszerzona funkcja DO I/O dla jednostek centralnych 331 i nowszych .....	12-7
	SER.....	12-8
	Parametry .....	12-9
	Dopuszczalne typy zmiennych .....	12-9
	Blok danych sterujących .....	12-10
	Słowo Status Extra Data .....	12-12
	Blok danych funkcji SER .....	12-13
	Tryby próbkowania.....	12-14
	Próbkowanie sterowane sygnałem wyzwalania .....	12-14
	Formaty stępli czasowych funkcji SER .....	12-17
	Przykład bloku danych sterujących.....	12-19
	END .....	12-23
	Przykład .....	12-23
	MCRN/MCR .....	12-24
	Przegląd funkcji MCR i MCRN .....	12-24
	Kompatybilność z jednostkami centralnymi.....	12-25
	Możliwe problemy z kompatybilnością .....	12-25
	Zagnieżdżanie funkcji MCRN .....	12-25
	Opis działania funkcji MCR .....	12-26
	Parametry .....	12-26
	Różnice pomiędzy funkcjami MCR/MCRN a JUMP .....	12-27
	Przykład 1 .....	12-28
	Przykład 2 .....	12-29
	ENDMCRN/ENDMCR .....	12-30
	Przykład .....	12-30
	JUMP .....	12-31
	Przykłady .....	12-32
	LABEL .....	12-33
	Przykład .....	12-33
	COMMENT .....	12-34
	SVCREQ .....	12-35
	Podstawowe informacji o bloku funkcyjnym SVCREQ .....	12-36
	SVCREQ #1: Zmiana/odczyt czasu trwania cyklu pracy sterownika w trybie ze stałym czasem trwania cyklu .....	12-38

	SVCREQ #2: Odczyt wartości z programatora .....	12-41
	SVCREQ #3: Zmiana trybu i czasu trwania komunikacji z programatorem .....	12-43
	SVCREQ #4: Zmiana trybu komunikacji systemowej oraz czasu trwania komunikacji systemowej.....	12-45
	SVCREQ #6: Odczyt/zmiana liczby słów sumy kontrolnej programu sterującego.....	12-47
	SVCREQ #7: Odczyt/zmiana wskazań zegara czasu rzeczywistego.....	12-49
	SVCREQ #8: Zerowanie zegara alarmowego .....	12-53
	SVCREQ #9: Odczyt czasu trwania cyklu .....	12-54
	SVCREQ #10: Odczyt nazwy folderu .....	12-55
	SVCREQ #11: Odczyt identyfikatora sterownika .....	12-56
	SVCREQ #12: Odczyt trybu pracy sterownika .....	12-57
	SVCREQ #13: Zatrzymanie sterownika .....	12-58
	SVCREQ #14: Wymazanie komunikatów z tablicy błędów sterownika i układów wejść/wyjść.....	12-59
	SVCREQ #15: Odczyt ostatnio zarejestrowanego komunikatu o błędzie działania.....	12-60
	SVCREQ #16: Zegar odmierzający czas pracy sterownika.....	12-64
	SVCREQ #18: Kontrola występowania wymuszeń zmiany wartości zmiennych wejściowych i wyjściowych .....	12-65
	SVCREQ #23: Odczyt sumy kontrolnej programu sterującego i konfiguracji.....	12-66
	SVCREQ #24: Zerowanie modułu dodatkowego .....	12-67
	SVCREQ #26/30: Porównanie rzeczywistej konfiguracji modułów wejść/ wyjść sterownika ze zdefiniowaną.....	12-68
	SVCREQ #29: Zegar czasu trwania przerwy w zasilaniu sterownika.....	12-69
	SVCREQ #45: Pominięcie obsługi wejść/wyjść w następnym cyklu .....	12-70
	SVCREQ #46: Szybki dostęp do statusu komunikacji przez szynę komunikacyjną.....	12-71
	SVCREQ #48: Ponowne uruchomienie po błędzie krytycznym automatycznego zerowania.....	12-77
	SVCREQ 49 Automatyczne zerowanie statystyk.....	12-79
	PID.....	12-80
	Parametry .....	12-81
	Dopuszczalne typy zmiennych .....	12-81
	Blok parametrów funkcji PID.....	12-82
	Opis działania bloku funkcyjnego PID .....	12-84
<b>Załącznik A</b>	<b>Czasy wykonywania instrukcji .....</b>	<b>A-1</b>
	Czasy wykonywania dla przekaźników i styków .....	A-15
	Rozmiary instrukcji – CPU350 - 374 .....	A-15
<b>Załącznik B</b>	<b>Interpretowanie tabeli błędów .....</b>	<b>B-1</b>
	Tabela błędów działania sterownika.....	B-1
	Przykład .....	B-2
	Tabela błędów działania wejść/wyjść.....	B-8
<b>Załącznik C</b>	<b>Nazwy mnemoniczne.....</b>	<b>C-1</b>
<b>Załącznik D</b>	<b>Skróty klawiszowe.....</b>	<b>D-1</b>

<b>Załącznik E</b>	<b>Operacje na liczbach zmiennoprzecinkowych .....</b>	<b>E-1</b>
	Liczby zmiennoprzecinkowe .....	E-1
	Format liczb rzeczywistych .....	E-2
	Wartości liczb zmiennoprzecinkowych .....	E-4
	Wprowadzanie i wyświetlanie liczb zmiennoprzecinkowych .....	E-5
	Błędy w operacjach i liczbach zmiennoprzecinkowych .....	E-6
	NaN .....	E-6
	Przepływ sygnału sterującego dla liczb nieskończonych i NaN .....	E-8
<b>Załącznik F</b>	<b>Porównanie oprogramowania .....</b>	<b>F-1</b>

Rysunek 2-1. Cykl pracy sterownika.....	2-3
Rysunek 2-2. Schemat blokowy komunikacji z programatorem.....	2-10
Rysunek 2-3. System Communications Window Flow ChartSchemat blokowy komunikacji systemowej..	2-11
Rysunek 2-4 Komunikacja modułu programowalnego procesora (PCM) ze sterownikiem .....	2-12
Rysunek 2-5. Sekwencja rozruchu .....	2-32
Rysunek 2-6. Schemat pracy generatora sygnału prostokątnego .....	2-37
Rysunek 2-7. Struktura wejść/wyjść sterowników serii 90-30.....	2-40
Rysunek 2-9. Moduły wejść/wyjść serii 90-30 .....	2-41
Rysunek 12-1. Przykład próbkowania w trybie Rejestrowanie próbek do sygnału wyzwolenia (dla 512 próbek) .....	12-15
Rysunek 12-2. Przykład próbkowania do zebrania określonej ilości próbek po sygnale wyzwolenia (dla 512 próbek) .....	12-15
Rysunek 12-3. Przykład próbkowania do zebrania określonej ilości próbek po sygnale wyzwolenia (dla 512 próbek) .....	12-16
Rysunek 12-4. Regulator PID o niezależnych wyrazach (PIDIND) .....	12-89

Tabela 2-1. Składniki czasu trwania cyklu pracy sterownika.....	2-4
Tabela 2-2. Czasy odczytywania wejść i ustawiania wyjść dla sterowników 90-30, jednostki centralne 35x, 36x i 37x (w ms) .....	2-5
Tabela 2-3. Czasy odczytywania wejść i ustawiania wyjść dla sterowników serii 90-30, jednostki centralne CPU311 do CPU341.....	2-6
Tabela 2-4. Zmienne rejestrowe .....	2-20
Tabela 2-5. Zmienne dyskretne .....	2-21
Tabela 2-6. Typy danych.....	2-23
Tabela 2-7. Zmienne systemowe .....	2-24
Tabela 2-8. Moduły wejść/wyjść serii 90-30 - kontynuacja.....	2-42
Tabela 2-8. Moduły wejść/wyjść serii 90-30 - kontynuacja.....	2-43
Tabela 3-1. Zestawienie błędów .....	3-3
Tabela 3-2. Wagi błędów i działania podejmowane w momencie ich wystąpienia .....	3-4
Tabela 4-1. Typy styków .....	4-1
Tabela 4-2. Typy przekaźników .....	4-2
Tabela 12-1. Blok danych sterujących dla przykładowej funkcji SER .....	12-19
Tabela 12-2. Zawartość próbki dla przykładowego bloku funkcyjnego SER .....	12-21
Tabela 12-3. Blok danych dla przykładowej funkcji SER .....	12-21
Tabela 12-4. Specjalne funkcje sterownika.....	12-35
Tabela 12-6. Parametry bloku dla funkcji odczytu dodatkowych danych .....	12-72
Tabela 12-8. Parametry bloku dla funkcji zapisu danych.....	12-73
Tabela 12-9. Parametry bloku dla funkcji odczyt/zapis danych.....	12-74
Tabela 12-10. Kody błędów .....	12-75
Tabela 12-11. Parametry bloku uruchomienie po błędzie krytycznym.....	12-78
Tabela 12-12. Definicje statusu raportowania .....	12-78
Tabela 12-13. Blok parametrów funkcji.....	12-79
Tabela 12-14. Definicje statusu raportowania.....	12-79
Tabela 12-15. Zestawienie parametrów bloku funkcyjnego PID .....	12-82
Tabela 12-15. Zestawienie parametrów bloku funkcyjnego PID (kontynuacja).....	12-83
Tabela 12-16. Szczegółowy opis parametrów bloku funkcyjnego PID.....	12-85
Tabela 12-16. Zestawienie parametrów bloku funkcyjnego PID (kontynuacja).....	12-86
Tabela 12-16. Zestawienie parametrów bloku funkcyjnego PID (kontynuacja).....	12-87
Tabela A-1. Czasy wykonywania instrukcji.....	A-2
Tabela A-1. Czasy wykonywania instrukcji - kontynuacja .....	A-3
Tabela A-1. Czasy wykonywania instrukcji - kontynuacja .....	A-4
Tabela A-1. Czasy wykonywania instrukcji - kontynuacja .....	A-5
Tabela A-2. Czasy wykonywania instrukcji CPU35x/36x.....	A-6
Tabela A-2. Czasy wykonywania instrukcji CPU35x/36x - kontynuacja .....	A-7
Tabela A-2. Czasy wykonywania instrukcji CPU35x/36x - kontynuacja .....	A-8



Tabela A-2 Czasy wykonywania instrukcji CPU35x/36x - kontynuacja .....	A-9
Tabela A-3 Czas wykonywania bloku funkcyjnego SER.....	A-10
Tabela A-4 Czasy wykonywania instrukcji – CPU 37x .....	A-11
Tabela A-4 Czasy wykonywania instrukcji – CPU37x - kontynuacja .....	A-12
Tabela A-4 Czasy wykonywania instrukcji – CPU37x - kontynuacja .....	A-13
Tabela A-4 Czasy wykonywania instrukcji – CPU37x - kontynuacja .....	A-14
Tabela A-5 Czasy wykonywania dla przekaźników i styków .....	A-15
Tabela B-1 Grupy błędów działania sterownika: .....	B-4
Tabela B-2 Tabela wag błędów działania sterownika: .....	B-5
Tabela B-3 Kody błędów alarmowych dla grupy PLC CPU Software .....	B-5
Tabela B-4 Kody błędów alarmowych dla sterownika .....	B-6
Tabela B-5 Informacje o błędzie działania sterownika - Wykryty niewłaściwy kod logiczny OPCODE .....	B-7
Tabela B-6 Czas i data wystąpienia błędu działania sterownika .....	B-7
Tabela B-7 Długość bloku szczegółowych informacji o błędzie .....	B-9
Tabela B-8 Adres elementu wejść/wyjść.....	B-9
Tabela B-9 Typ pamięci adresu elementu wejść/wyjść.....	B-9
Tabela B-10 Grupy błędów działania układów wejść/wyjść.....	B-11
Tabela B-11 Wagi błędów działania układów wejść/wyjść .....	B-12
Tabela B-12 Blok szczegółowych informacji o błędach działania układów wejść/wyjść .....	B-12
Tabela B-13 Czas i data wystąpienia błędu działania układów wejść/wyjść .....	B-13
Tabela E-1 Przepływ sygnału przy operacjach zmiennoprzecinkowych.....	E-8



Sterowniki serii 90-30 90-20 i Micro są częścią większej rodziny sterowników programowalnych serii 90. Odznaczają się one łatwością instalowania i konfigurowania, oferują zaawansowane mechanizmy programowania oraz są kompatybilne ze sterownikami serii 90-70.

W modelach 341 i słabszych sterownikach 90-30 i sterownikach 90-20 wykorzystywany jest mikroprocesor 80188. Modele 35x i 36x sterowników serii 90-30 korzystają z mikroprocesora 80386EX. W modelach 37x sterowników 90-30 zastosowano mikroprocesor 586. Sterowniki serii 90 korzystają z mikroprocesora H8. Mikroprocesory te pozwalają zarówno na wykonywanie programu jak i realizację innych zadań, takich jak: przeprowadzanie czynności diagnostycznych, skanowanie wejść/wyjść i przetwarzanie alarmów. Oprogramowanie systemowe zawiera również procedury do komunikacji z programatorem. Procedury te pozwalają na usuwanie i wczytywanie programów sterujących, odczytywanie informacji o statusie oraz zapewniają kontrolę nad sterownikami.

W sterownikach serii 90-30, programy sterujące napisane przez użytkownika (służące do sterowania przebiegiem procesu, do obsługi, którego zastosowano sterownik), obsługiwane są poprzez specjalizowany koprocesor (ISCP). Koprocesor ten zrealizowany jest sprzętowo w systemach z jednostką centralną CPU Model 313 i wyższych, natomiast w systemach z jednostką centralną 311 i w sterownikach serii Micro. Mikroprocesor i koprocesor ISCP pracują równolegle, przy czym mikroprocesor odpowiedzialny jest za komunikację sterownika, a koprocesor wykonuje większość instrukcji. Nie mniej jednak mikroprocesor musi również wykonywać zawarte w programie sterującym bloki funkcyjne, odpowiadające za operacje inne niż logiczne.

Wystąpienie pewnych uszkodzeń lub okoliczności mających wpływ na pracę i charakterystyki robocze systemu sygnalizowane jest w sterownikach serii 90-30, 90-20 i Micro w postaci błędów. Błędy te mogą uniemożliwić sterowanie maszyną czy procesem. Można też wyróżnić błędy mające na celu wyłącznie zwrócenie uwagi użytkownika, jak na przykład informacja o niskim stanie naładowania akumulatorów podtrzymujących zawartość pamięci i związanej z tym konieczności ich wymiany.

Błędy te obsługiwane są przez programową funkcję alarmową, która zapisuje je w dwóch tablicach (osobno dla sterownika, osobno dla układów wejścia/wyjścia). (W jednostce centralnej 331 i wyższych, zapisywany jest również czas wystąpienia błędu). Tabele te można wyświetlić na ekranie komputera z oprogramowaniem Logicmaster 90-30/20/Micro.

### Uwaga

Operacje zmiennoprzecinkowe dostępne są **wyłącznie** w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 i CPU374.

Jednostki centralne CPU364 (wersja 9.10 lub nowsza) oraz CPU374 obsługują Ethernet Global Data (EGD).

Sterowniki serii 90-20 są efektywną ekonomicznie platformą dla aplikacji o małej liczbie wejść/wyjść. Seria sterowników 90-20 to:

- Małe sterowniki odznaczające się łatwością użytkowania, instalowania, aktualizowania i utrzymywania.
- Efektywna ekonomicznie rodziny kompatybilnych sterowników PLC.
- Łatwa integracja systemu poprzez standardowe rozwiązania sprzętowe i protokoły do komunikacji.

Sterowniki serii 90 Micro są efektywną ekonomicznie platformą dla aplikacji o małej liczbie wejść/wyjść. Przy opracowywaniu sterowników serii Micro postawione były takie same cele, jak dla serii 90-30. Dodatkowo, serię Micro charakteryzuje:

- Zabudowane w jednej kompaktowej obudowie: mikroprocesor, zasilacz, układ wejść/wyjść.
- Większość z modeli posiada również na wyposażeniu licznik wysokiej częstotliwości.
- Łatwość konfiguracji urządzenia zawierającego mikroprocesor, moduł zasilania i układ wejść/wyjść.

#### **Uwaga**

Dodatkowe informacje podane są w załącznikach do niniejszego podręcznika.

- W Załączniku A podano rozmiar pamięci w bajtach oraz czas wykonywania w mikrosekundach każdej z instrukcji programowania.
- W Załączniku B opisano reprezentację komunikatów o błędach, zapisywanych w tabeli błędów sterownika i tabeli błędów układów wejścia/wyjścia.
- W załączniku C podano nazwy mnemoniczne instrukcji, wykorzystywane przy edycji programu sterującego.
- W załączniku D podano skróty klawiaturowe wykorzystywane w oprogramowaniu Logicmaster 90-30/20/Micro.
- Appendix E describes the use of floating-point math operations.

## **Uwagi dla użytkowników oprogramowania pracującego w systemach MS Windows**

Podręcznik ten został napisany dla użytkowników pracujących z oprogramowaniem Logicmaster (pracującym w systemie operacyjnym DOS). Informacje o oprogramowaniu pracującym w środowisku Windows tj. CIMPLICITY® Machine Edition Logic Developer i VersaPro®, można znaleźć w ich systemach pomocy on-line. Użytkownicy pracujący w środowisku Windows powinni zwrócić uwagę, że poszczególne instrukcje są wyświetlane w inny sposób niż ma to miejsce w programie Logicmaster, (pomimo że dotyczą one tego samego sterownika). W systemie pomocy oprogramowania pracującego w środowisku Windows znajdują się dokładne informacje o poszczególnych instrukcjach. W załączniku F podsumowano główne różnice pomiędzy poszczególnymi rodzajami oprogramowania.

# Rozdział 2

## *Działanie systemu*

---

---

W rozdziale tym opisano pewne zagadnienia, związane z działaniem sterowników serii 90-30, 90-20 i Micro. Omówiono między innymi:

- Cykl pracy sterowników (Część 1). .....2-1
- Organizację programu sterującego, typy danych i typy zmiennych (Część 2). .....2-16
- Działania wykonywane przez sterownik podczas włączania i wyłączania zasilania (Część 3). .....2-30
- Liczniki i przekaźniki czasowe (Część 4). .....2-35
- Mechanizm zabezpieczenia za pomocą haseł (Część 5) .....2-38
- Moduły wejść/wyjść sterowników serii 90-30 (Część 6). .....2-40

## Część 1: Cykl pracy sterownika

Programy sterujące w sterownikach serii 90-30, 90-20 i Micro wykonywane są cyklicznie, aż do momentu zatrzymania za pośrednictwem instrukcji z komputera- programatora lub z innego urządzenia zewnętrznego. Ciąg operacji koniecznych do jednorazowego wykonania programu sterującego jest nazywany cyklem pracy sterownika. Oprócz wykonania części logicznej programu sterującego, cykl pracy sterownika zawiera również fazy gromadzenia danych z urządzeń wejściowych, wysłania informacji do urządzeń wyjściowych, przeprowadzenia wewnętrznej inicjalizacji sterownika, obsługi programatora oraz komunikacji z innymi urządzeniami.

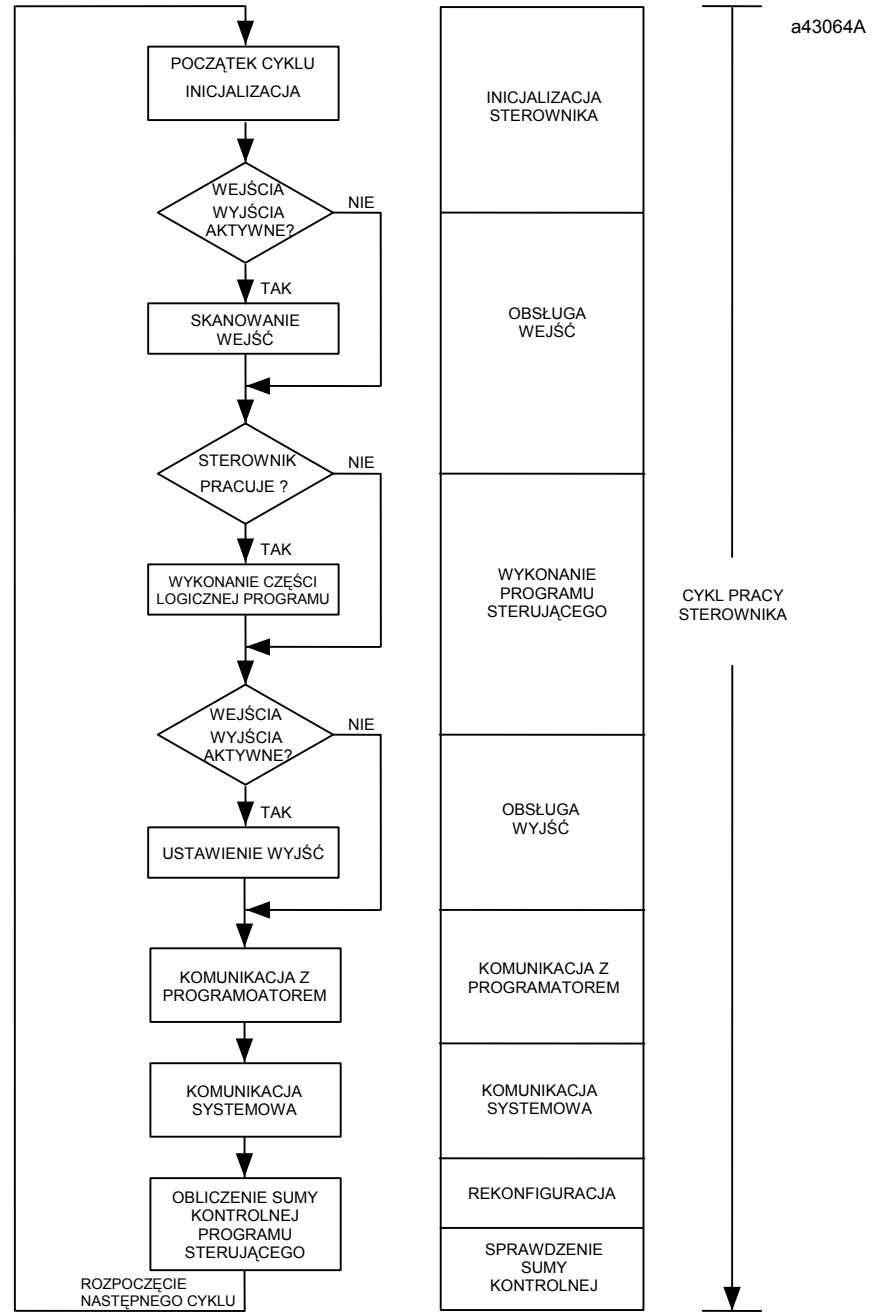
Sterowniki serii 90-30, 90-20 i Micro zwykle pracują w trybie **STANDARD PROGRAM SWEEP** (standardowy cykl pracy). Inne możliwe tryby pracy to tryb zatrzymania sterownika z nieaktywnymi wejściami i wyjściami (**STOP WITH I/O DISABLED**), tryb zatrzymania sterownika z odczytywaniem wejść i ustawianiem wyjść (**STOP WITH I/O ENABLED**), oraz tryb ze stałym czasem trwania cyklu pracy sterownika (**CONSTANT SWEEP**). Każdy z tych trybów jest uaktywniany przez parametry konfiguracyjne i wystąpienie pewnych zdarzeń w trakcie pracy systemu. Tryb pracy jest ustalany na początku każdego cyklu.

### Standardowy cykl pracy sterownika

**STANDARD PROGRAM SWEEP** jest najczęściej wykorzystywanym trybem pracy sterownika. W czasie pracy, jednostka centralna wykonuje program sterujący, uaktualnia stan wejść i wyjść, realizuje komunikację, itp. Czynności te powtarzane są cyklicznie. Standardowy tryb pracy składa się z siedmiu faz:

1. Inicjalizacji cyklu
2. Obsługi wejść
3. Wykonania programu sterującego
4. Obsługi wyjść
5. Komunikacja z programatorem
6. Komunikacja systemowa
7. Diagnostyka

Wszystkie te fazy wykonywane są podczas każdego cyklu. Komunikacja z programatorem realizowana jest jedynie w przypadku wykrycia błędu działania układu lub zgłoszenia żądania komunikacji przez programator. Na zamieszczonym poniżej rysunku pokazano sekwencje faz standardowego cyklu pracy sterownika.



Rysunek 2-1. Cykl pracy sterownika

As shown in the PLC sweep sequence, several items are included in the sweep. Zgodnie z powyższym schematem, każdy z cykli składa się z kilku faz. These items contribute to the total sweep time as shown in the following table. Czasy trwania tych faz podano w zamieszczonej poniżej tabeli.

Tabela 2-1. Składniki czasu trwania cyklu pracy sterownika

Sweep Element	Opis	Czas trwania (ms) <sup>4</sup>							
		Micro	211	311/313	331	34x	35x/36x	37x	
Inicjalizacja cyklu sterownika	<ul style="list-style-type: none"> <li>Obliczenie czasu trwania cyklu</li> <li>Zaplanowanie czasu rozpoczęcia następnego cyklu.</li> <li>Określenie trybu pracy sterownika w następnym cyklu.</li> <li>Uaktualnienie tabel błędów.</li> <li>Wyzerowanie zegara alarmowego</li> </ul>	0.368	0.898	0.714	0.705	0.424	0.279	0.027	
Obsługa wejść	Wprowadzenie danych wejściowych, uzyskanych z modułów wejść oraz z modułów opcjonalnych.	Uwaga 5	Proszę porównać z tabelą 2-2 i 2-3.						
Wykonanie programu sterującego	Wykonanie części logicznej programu sterującego.	Czas wykonania programu zależy od długości programu i rodzaju instrukcji zastosowanych w programie. Czasy wykonania poszczególnych instrukcji podano w Załączniku A.							
Obsługa wyjść	Dane wyjściowe są wysyłane do modułów wyjść oraz do modułów opcjonalnych.	1.656	Proszę porównać z tabelą 2-2 i 2-3.						
Komunikacja z programatorem i komunikacja systemowa	Przetwarzanie żądań nadesłanych przez programator oraz wyspecjalizowane moduły.	Prog. ręczny	1.93	6.526	4.426	4.524	2.476	0.334	Brak
		Programator	0.380	3.536	2.383	2.454	1.248	0.517	0.026
		PCM <sup>2</sup>	Brak	Brak	Brak	3.337	1.943	0.482	0.029
Ponowna konfiguracja	Monitorowanie gniazd z uszkodzonymi modułami oraz gniazd bez modułów.	N/A <sup>6</sup>	Brak	0.458	0.639	0.463	0.319	0.243	
Diagnostyka	Sprawdzenie integralności programu (czas potrzebny na sprawdzenie sumy kontrolnej każdego cyklu) <sup>3</sup>	N/A <sup>7</sup>	0.083	0.050	0.048	0.031	0.010	0.022	

- Czas obsługi urządzeń zewnętrznych zależy od trybu pracy okna do komunikacji, w którym realizowana jest obsługa. W trybie LIMITED, czas trwania obsługi może wynosić maksymalnie 8 milisekund dla jednostek centralnych 311, 313, 323 i 331 oraz 6 milisekund dla jednostek centralnych serii 340 i wyższych. W trybie **RUN-TO-COMPLETION** czas trwania obsługi może wynosić maksymalnie 50 ms, w zależności od liczby jednocześnie wysłanych żądań.
- Pomiary przeprowadzono przy zainstalowanym fizycznie module programowalnego koprocesora (PCM), ale bez jego konfigurowania oraz bez uruchomionej aplikacji na tym module PCM.
- Liczba słów sumy kontrolnej dla każdego z cykli może być zmieniona za pomocą funkcji SVCREQ.
- Pomiary przeprowadzono bez programu sterującego oraz przy domyślnych parametrach konfiguracyjnych. Sterowniki serii 90-30 były zamontowane w kasecie z 10-oma gniazdami, bez podłączonych dodatkowych kaset rozszerzających. Ponadto dla czasów podanych w tabeli założono, że nie jest uruchomiony żaden podprogram wykonywany okresowo.
- Czas wprowadzania danych wejściowych w sterowniku Micro można obliczyć wg następującego wzoru:  $0.365\text{ms} (\text{fixed scan}) + 0.036\text{ms} (\text{filter time}) \times (\text{total sweep time}) / 0.5\text{ms}$ .
- Ponieważ sterownik Micro dysponuje statycznym zestawem wejść i wyjść, nie jest wymagane jego ponowne konfigurowanie.
- Program dla sterownika Micro przechowywany jest w pamięci ulotnej, w związku z czym nie jest sprawdzana jego integralność.



Tabela 2-2. Czasy odczytywania wejść i ustawiania wyjść dla sterowników 90-30, jednostki centralne 35x, 36x i 37x (w ms)

Typ modułu	Jednostki centralne 35x i 36x			37x Series CPUs			
	Kaseta główna	Kaseta rozszerzająca	Kaseta oddalona	Kaseta główna	Kaseta rozszerzająca	Kaseta oddalona	
Moduł wejść dyskretnych, 8 punktów	.030	.055	.206	.030	.055	.206	
Moduł wejść dyskretnych, 16 punktów	.030	.055	.206	.030	.055	.206	
Moduł wejść dyskretnych, 32 punktów	.043	.073	.269	.048	.075	.272	
Moduł wyjść dyskretnych, 8 punktów	.030	.053	.197	.024	.052	.198	
Moduł wyjść dyskretnych, 16 punktów	.030	.053	.197	.030	.052	.199	
Moduł wyjść dyskretnych, 32 punktów	.042	.070	.259	.047	.069	.258	
Zintegrowany moduł wejść/wyjść dyskretnych	.060	.112	.405	.052	.110	.408	
Moduł wejść analogowych, 4 kanały	.075	.105	.396	.085	.109	.403	
Moduł wyjść analogowych, 2 kanałów	.058	.114	.402	.046	.101	.393	
Moduł wejść analogowych (prądowych lub napięciowych), 16 kanałów	.978	1.446	3.999	.423	.700	1.741	
Moduł wyjść analogowych, 8 kanałów	1.274	1.988	4.472	.873	1.492	3.635	
Zintegrowany moduł wejść/wyjść analogowych	1.220	1.999	4.338	.862	1.487	4.103	
Licznik impulsów o wysokiej częstotliwości	1.381	2.106	5.221	1.142	1.808	5.234	
I/O Processor (procesor wejść/wyjść)	1.574	2.402	6.388	1.270	2.125	6.269	
Interfejs Ethernet (bez połączenia)	.7129	2.067	3.681	.426	.795	2.302	
Moduł Power Mate APM (1 oś)	1.527	2.581	6.388	1.236	2.073	6.032	
Moduł Power Mate APM (2 osie)	1.807	2.864	7.805	1.539	2.439	7.369	
DSM 302 *	40 AI, 6 AQ	2.143	3.315	9.527	1.801	2.963	9.275
	50AI, 9 AQ	2.427	3.732	11.092	2.075	3.373	10.840
	64 AI, 12 AQ	2.864	4.317	13.138	2.441	3.931	12.881
DSM314 *	1 oś	1.6	2.6	6.9	1.330	2.337	6.905
	2 osie	2.2	3.8	9.9	1.888	3.148	9.917
	3 osie	2.8	4.3	13.0	2.421	3.953	12.929
	4 osie	3.3	5.2	15.9	2.969	4.761	15.982
GCM	8 urządzeń po 32 bity	8.826	16.932	21.179	7.386	9.520	20.591
GCM+	bez urządzeń	.567	.866	1.830	.457	.759	1.743
	32 urządzenia po 64 słowa	19.497	25.588	80.871	17.036	24.390	80.044
GBC	bez urządzeń	.798	1.202	2.540	.544	.908	2.209
	16 urządzenia po 64 słowa	29.976	40.570	131.702	26.976	38.564	130.639
PCM 3xx	bez konfiguracji lub bez zadania dla aplikacji	.476	Brak	Brak	.195	Brak	Brak
	przetwarzanie 20Kb programu	1.746	Brak	Brak	.538	Brak	Brak
ADC (bez zadania)		.476	Brak	Brak	.193	Brak	Brak
Połączenie We/Wy Master	bez urządzeń	.569	.865	1.932	.996	1.618	3.749
	16 urządzeń 64 punktowych	4.948	7.003	19.908	5.924	8.240	26.637
Połączenie We/Wy Slave	32 punkty	.087	.146	.553	.095	.149	.540
	64 punkty	.154	.213	.789	.165	.219	.803

\* W przypadku systemów, w których czas poświęcony na komunikację z modulem DSM może powodować zakłócenia w pracy sterowanej maszyny, można przesyłać dane do/ odczytywać dane z modułu DSM za pomocą bloków funkcyjnych DO I/O, Suspend I/O i SVC\_REQ #46, dzięki czemu, nie wszystkie dane będą odbierane w każdym cyklu. Szczegółowe informacje podano w podręczniku *Motion Mate DSM302 for Series 90-30 PLCs User's Manual*, GFK1464. Szczegółowe informacje podano w podręczniku *Motion Mate DSM302 for Series 90-30 PLCs User's Manual*, GFK1464. UWAGA: Moduł DSM314 pracuje tylko w systemach z CPU350, 352, 360, 363, 364 i 374 z oprogramowaniem systemowym 10.00 lub nowszym.

Tabela 2-3. Czasy odczytywania wejść i ustawiania wyjść dla sterowników serii 90-30, jednostki centralne CPU311 do CPU341

Typ modułu		Model jednostki centralnej						
		311/313/ 323	331			340/341		
			Kaseta główna	Kaseta rozszerzająca	Kaseta oddalona	Kaseta główna	Kaseta rozszerzająca	Kaseta oddalona
Moduł wejść dyskretnych, 8 punkty		.076	.054	.095	.255	.048	.089	.249
Moduł wejść dyskretnych, 16 punkty		.075	.055	.097	.257	.048	.091	.250
Moduł wejść dyskretnych, 32 punkty		.094	.094	.126	.335	.073	.115	.321
Moduł wyjść dyskretnych, 8 punktów		.084	.059	.097	.252	.053	.090	.246
Moduł wyjść dyskretnych, 16 punktów		.083	.061	.097	.253	.054	.090	.248
Moduł wyjść dyskretnych, 32 punktów		.109	.075	.129	.333	.079	.114	.320
Kombinowany moduł wejść/wyjść, 8 punktów		.165	.141	.218	.529	.098	.176	.489
Moduł wejść analogowych, 4 kanały		.151	.132	.183	.490	.117	.160	.462
Moduł wyjść analogowych, 2 kanały		.161	.138	.182	.428	.099	.148	.392
Licznik impulsów o wysokiej częstotliwości		2.070	2.190	2.868	5.587	1.580	2.175	4.897
Moduł specjalistyczny Power Mate APM (1 oś)		2.330	2.460	3.175	6.647	1.750	2.506	5.899
Moduł specjalistyczny Power Mate APM (2 oś)		3.181	3.647	4.497	9.303	2.154	3.097	7.729
DSM 302 *	40 AI, 6 AQ	3.613	4.081	5.239	11.430	2.552	3.648	9.697
	50AI, 9 AQ	4.127	4.611	5.899	13.310	2.911	4.170	11.406
	64 AI, 12 AQ	4.715	5.276	6.759	15.747	3.354	4.840	13.615
GCM	bez urządzeń	.041	.054	.063	.128	.038	.048	.085
	8 urządzeń 64 punktowych	11.420	11.570	13.247	21.288	9.536	10.648	19.485
GCM+	bez urządzeń	.887	.967	1.164	1.920	.666	.901	1.626
	32 urządzeń 64 punktowych	4.120	6.250	8.529	21.352	5.043	7.146	20.052
PCM 3xx	bez konfiguracji lub bez zadania dla aplikacji	Brak	3.350	Brak	Brak	1.684	Brak	Brak
	odczyt w jak najkrótszym czasie 128%R	Brak	4.900	Brak	Brak	2.052	Brak	Brak
ADC 311		Brak	3.340	Brak	Brak	1.678	Brak	Brak
Moduł wejść analogowych (prądowych lub napięciowych), 16 kanałów		1.370	1.450	1.937	4.186	1.092	1.570	3.796
Połączenie We/Wy Master	bez urządzeń	1.910	2.030	1.169	1.925	.678	.904	1.628
	szesnaście urządzeń 64 punktowych	6.020	6.170	8.399	21.291	4.992	6.985	20.010
Połączenie We/Wy Slave	32 punkty	.206	.222	.289	.689	.146	.226	.636
	64 punkty	.331	.350	.409	1.009	.244	.321	.926

\* W przypadku systemów, w których czas poświęcony na komunikację z modułem DSM może powodować zakłócenia w pracy sterowanej maszyny, można przesyłać dane do/ odczytywać dane z modułu DSM za pomocą bloków funkcyjnych DO I/O, Suspend I/O i SVC\_REQ #46, dzięki czemu, nie wszystkie dane będą odbierane w każdym cyklu. Szczegółowe informacje podano w podręczniku *Motion Mate DSM302 for Series 90-30 PLCs User's Manual*, GFK1464.  
UWAGA: The DSM314 is not supported by the 311 through 341 CPUs.

## Obliczanie czasu trwania cyklu pracy sterownika

W tabeli 2-1 wyszczególniono siedem faz, które składają się na czas trwania cyklu pracy sterownika. Czas ten zawiera składniki stałe (jak inicjalizacja cyklu i diagnostyka) oraz zmienne. Składniki zmienne zależą od konfiguracji układów wejść/wyjść, rozmiarów programu sterującego napisanego przez użytkownika oraz typu urządzenia programującego podłączonego do sterownika (może to być komputer-programator wyposażony w oprogramowanie Logicmaster 90 lub programator ręczny; sterownik może też pracować bez żadnego podłączonego urządzenia programującego).

### Przykład obliczania czasu trwania cyklu pracy sterownika

W zamieszczonej poniżej tabeli podano przykład obliczania czasu trwania cyklu pracy dla sterownika serii 90-30 z jednostką centralną CPU 331.

Obliczenia przeprowadzono dla następujących modułów i instrukcji:

- Moduły wejść: pięć szesnasto- punktowych modułów wejść sterownika serii 90-30.
- Moduły wyjść: cztery 16-o punktowych modułów wyjść serii 90-30.
- Instrukcje programu sterującego Program składający się z 1200 kroków, zawierający 700 instrukcji logicznych (typu LD, AND, OR, itp.), 300 przełączników (OUT, OUTM, itp.) oraz 200 funkcji matematycznych (ADD, SUB, itp.).

Faza cyklu	Obliczenia	Czas trwania		
		Bez programatora	Z programatorem ręcznym	Z oprogramowaniem Logicmaster
Inicjalizacja cyklu sterownika	0.705ms	0.705ms	0.705ms	0.705ms
Obsługa wejść	$0.055 \times 5 = 0.275\text{ms}$	0.275ms	0.275ms	0.275ms
Wykonanie programu sterującego	$1000 \times 0.4\mu\text{s}^* + 200 \times 89\mu\text{s}^{**} + 18.2\text{ms}$	18.2ms	18.2ms	18.2ms
Obsługa wyjść	$0.061 \times 4 = 0.244\text{ms}$	0.244ms	0.244ms	0.244ms
Obsługa programatora	$0.4\text{ms} + \text{czas pracy programatora} + 0.6\text{ms}$	0ms	4.524ms	2.454ms
Bez obsługi programatora	Brak w rozważanym przykładzie	0ms	0ms	0ms
Ponowna konfiguracja	0.639ms	0.639ms	0.639ms	0.639ms
Diagnostyka	0.048ms	0.048ms	0.048ms	0.048ms
Czas trwania cyklu pracy sterownika	Inicjalizacja sterownika + Obsługa wejść + Wykonanie programu + Obsługa wyjść + Obsługa programatora + Bez obsługi programatora + Diagnostyka	12.611ms	17.135ms	15.065ms

## Szczegółowy opis cyklu pracy sterownika

W części tej szczegółowo opisano główne fazy cyklu pracy sterownika.

1. Inicjalizacja cyklu sterownika
2. Obsługa wejść
3. Wykonywanie programu sterującego
4. Obsługa wyjść
5. Obsługa programatora
6. Komunikacja systemowa
7. Ponowna konfiguracja
8. Obliczanie sumy kontrolnej

### 1. Inicjalizacja pracy sterownika

The housekeeping portion of the sweep performs all of the tasks necessary to prepare for the start of the sweep. Jeśli sterownik pracuje w trybie ze stałym czasem cyklu (**CONSTANT SWEEP**), rozpoczęcie nowego cyklu jest opóźniane do momentu, aż upłynie czas przewidziany na wykonanie poprzedniego cyklu. Jeśli wymagany czas upłynął, a poprzedni cykl nie zakończył się, wartość zmiennej systemowej %SA0002, przypisanej stykowi OV\_SWP, jest ustawiana na 1, po czym nowy cykl jest rozpoczynany bez opóźnienia. Następnie uaktualniane są wartości zmiennych przypisanych generatorom sygnału prostokątnego (o okresie 0.01, 0.1 i 1 sekunda), poprzez obliczenie różnicy pomiędzy czasem rozpoczęcia poprzedniego i bieżącego cyklu. W celu zachowania dokładności, faktyczny czas rozpoczęcia cyklu zapisywany jest z dokładnością 100 mikrosekund. Każdy generator sygnału prostokątnego posiada parametr podający liczbę okresów o długości 100 mikrosekund, które upłynęły od momentu ostatniego inkrementowania wartości generatora sygnału prostokątnego.

### 2. Skanowanie wejść

W czasie przeznaczonym na obsługę wejść dokonywane jest odczytywanie stanu wejść sterownika, ma to miejsce bezpośrednio przed wykonaniem części logicznej programu sterującego. W fazie tej odczytywany jest stan wejść wszystkich modułów sterowników 90-30, a odczytane wartości zapisywane są odpowiednio w pamięci adresowanej poprzez zmienne %I (wejścia dyskretne) oraz %AI (wejścia analogowe). Wszystkie dane globalne otrzymane za pośrednictwem modułu komunikacyjnego GENIUS (GCM), modułu Enhanced Genius (GCM+) lub za pośrednictwem sterownika szyny Genius (GBC) zostają zapisane w pamięci adresowanej przez zmienne globalne %G.

Wejścia są odczytywane zgodnie z rosnącymi adresami zmiennych, rozpoczynając od modułu komunikacyjnego GENIUS, następnie przechodząc do modułów wejść dyskretnych i ostatecznie do modułów wejść analogowych.

Jeśli sterownik znajduje się w trybie zatrzymania (**STOP**) i jest skonfigurowany tak, aby nie przeprowadzać odczytywania wejść i wyjść w tym trybie (**STOP/NO IO**), opisana tu faza zostaje pominięta.

### 3. Wykonywanie programu sterującego

Rozpoczęcie wykonywania programu logicznego następuje bezpośrednio po zakończeniu skanowania wejść. Program sterujący ma za wypełnia dwa podstawowe zadania: (1) rozwiązanie/wykonanie programu sterującego oraz (2) uaktualnienie danych w pamięci %Q, %AI, i %AQ. (Modułów wyjść,

jednakże dopiero w czasie obsługi wyjść). Program sterujący wykonywany jest od lewej do prawej strony drabinki logicznej począwszy od pierwszego szczebla, jednakże kierunek wykonywania programu może zostać zmieniony przez podprogramy lub funkcje Jump. Wykonywanie programu jest zatrzymywane po napotkaniu instrukcji END lub po osiągnięciu domyślnego końca programu END OF PROGRAM LOGIC.

Jednostki centralne 311 i wyższe koprocesor ISCP wykonujący funkcje logiczne oraz mikroprocesor 80C188,80386 lub AMD SC 520 odpowiadający za wykonywanie bloków funkcyjnych i czasowych (liczniki, timery). W jednostkach centralnych CPU 311 i sterownikach serii 90-20, wszystkie wymienione wyżej instrukcje wykonywane są przez procesor 80C188. W sterownikach Micro, procesor H8 wykonuje wszystkie operacje logiczne oraz bloki funkcyjne.

Czasy wykonywania poszczególnych instrukcji logicznych programu sterującego podano w Załączniku A.

## 4. Wysyłanie danych wejściowych

W czasie przeznaczonym na obsługę wyjść następuje przypisanie stanu zmiennych wyjściowych fizycznym wyjściom sterownika, etap ten realizowany jest natychmiast po zakończeniu wykonywania programu sterującego. Stan wyjść jest uaktualniany na podstawie wartości zmiennych %Q (wyjścia dyskretne) oraz %AQ (wyjścia analogowe). Jeśli moduł komunikacyjny GENIUS lub Genius Bus Controller jest skonfigurowany do przesyłania danych globalnych, dane z komórek pamięci adresowanych przez zmienne globalne %G zostają przesłane do modułu GCM, GCM+ lub GBC. W sterownikach serii Micro obsługiwane są wyłącznie wyjścia dyskretne.

We wszystkich sterownikach serii 90-30 moduły wyjściowe obsługiwane są zgodnie z rosnącymi adresami zmiennych. Zakończenie fazy obsługi wyjść następuje w momencie, gdy wartości zmiennych przypisanych wyjściom zostaną przesłane do wszystkich modułów wyjściowych sterowników 90-30.

Jeśli sterownik znajduje się w trybie **STOP** i parametr *IPScan-Stop* w CPU nie jest ustawiony faza ustawiania wyjść zostaje pominięta.

### Ostrzeżenie

**Jeśli został ustawiony w CPU parametr *IPScan-Stop* wyjścia mogą zostać załączone nawet, jeśli sterownik znajduje się w trybie STOP, ponieważ sterownik będzie zapisywał aktualne wartości z tablicy wyjść do poszczególnych wyjść modułów w czasie obsługi wyjść.**

## 5. Komunikacja z programatorem

Ta faza cyklu przeznaczona jest na komunikację z programatorem. Faza ta jest wykonywana, gdy do sterownika podłączony jest programator, a w systemie znajduje się moduł wymagający konfiguracji. Jeśli żaden z tych warunków nie jest spełniony, faza komunikacji z programatorem nie jest wykonywana. Podczas jednego cyklu może zostać skonfigurowany tylko jeden moduł.

Sterownik umożliwia podłączenie programatora ręcznego lub innych programatorów, mogących komunikować się za pomocą złącza szeregowego i korzystających z protokołu SNP. Zapewniana jest również obsługa komunikacji programatora z wyspecjalizowanymi modułami dodatkowymi.

### Tryby komunikacji z programatorem

- **Limited Mode.** W domyślnym trybie komunikacji ograniczonej czasowo, jednostka centralna sterownika wykonuje jedną operację związaną z programatorem podczas każdego cyklu, co oznacza, że odpowiada na jedno żądanie komunikacji lub na naciśnięcie klawisza. Jeśli programator wysyła żądanie komunikacji, którego realizacja zabiera więcej niż 6 milisekund (lub 8 w zależności od wersji jednostki centralnej - proszę porównać z uwagą poniżej), to jego

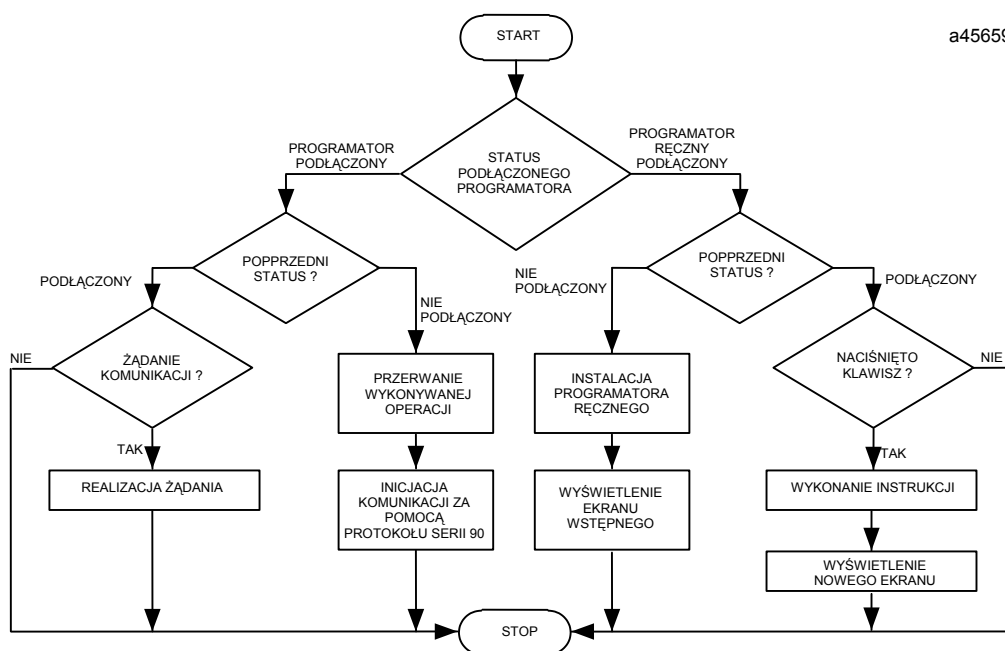
realizacja zostaje podzielona na kilka cykli tak, aby żaden z nich nie trwał więcej niż maksymalna 6 milisekund (lub 8 w zależności od wersji jednostki centralnej - proszę porównać z uwagą).

### Uwaga

Maksymalny, dopuszczalny czas trwania komunikacji wynosi 6 milisekund dla jednostki centralnej 340 i wyższych, oraz 8 milisekund dla jednostek centralnych 311, 313, 323 i 331.

- **Complete Mode.** W trybie tym jednostka centralna będzie komunikować się z programatorem do momentu przesłania wszystkich danych lub aż do upłynięcia 50 ms.

Poniżej przedstawiono schemat blokowy części cyklu przeznaczonej na komunikację z programatorem.

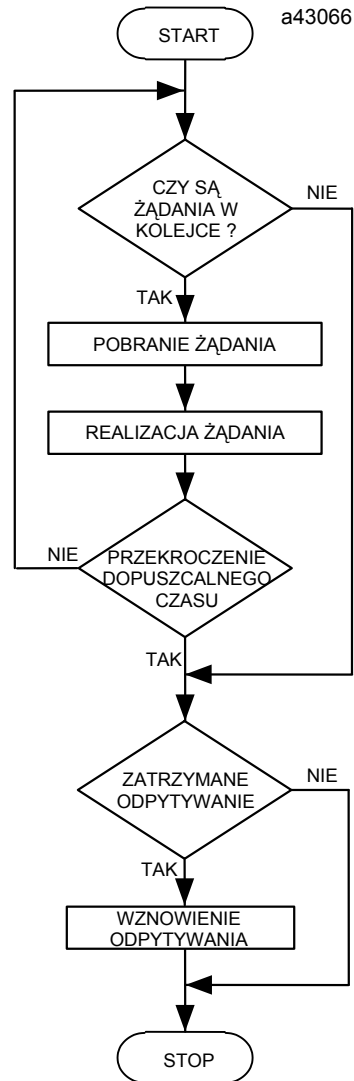


Rysunek 2-2. Schemat blokowy komunikacji z programatorem

## 6. Komunikacja systemowa (modele 331 i wyższe)

Podczas fazy komunikacji systemowej realizowane są żądania komunikacji nadsyłane przez wyspecjalizowane moduły, np. moduł programowalnego koprocessora lub DSM (proszę porównać ze schematem). Żądania komunikacji są obsługiwane w kolejności napływania. Ponieważ moduły dodatkowe są zapytywane metoda określona, żaden z tych modułów nie jest uprzywilejowany w stosunku do innych.

Jeżeli sterownik pracuje w trybie **Run - to - Completion**, czas trwania fazy komunikacji systemowej ograniczony jest do 50 ms. Jeśli czas realizacji żądania komunikacji wysłanego przez moduł dodatkowy przekracza 50 milisekund, jest ono realizowane w kilku cyklach tak, że żaden cykl nie zostaje przedłużony o więcej niż 50 ms.



Rysunek 2-3. System Communications Window Flow Chart Schemat blokowy komunikacji systemowej

## 7. Ponowna konfiguracja

Podczas tej fazy cyklu jednostka centralna porównuje aktualną konfigurację modułów z konfiguracją zapamiętaną. W sterowniku znajdują się gniazda ze skonfigurowanymi modułami ale również puste gniazda oraz moduły z błędami, które nie będą skanowane przez jednostkę centralną (np. jednostka centralna nie odczyta/wyśle danych do modułu lub gniazda). Jeśli w czasie trwania tej fazy moduły, w których występowały błędy a obecnie nie występują lub jeśli zostały dodane moduły do sterownika jednostka centralna rozpocznie ich skanowanie.

Rekonfiguracja umożliwia:

- Rozpoznanie i zapisanie zmian konfiguracji.
- Ignorowanie potencjalnie nieprawidłowych danych wejściowych z modułów.
- Unikanie wysłania danych do zepsutych modułów.

## 8. Obliczanie sumy kontrolnej

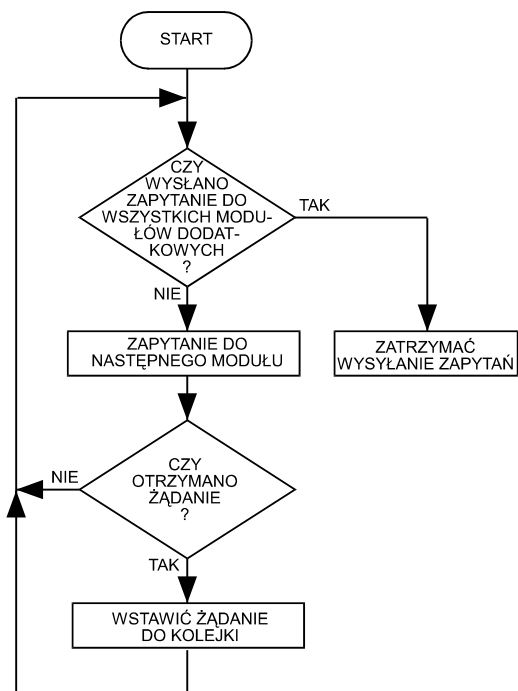
Na końcu każdego cyklu obliczana jest suma kontrolna programu sterującego użytkownika. Ponieważ obliczanie sumy kontrolnej dla całego programu jest bardzo czasochłonne, można określić liczbę słów z zakresu 0 do 32, dla których ma być obliczana suma kontrolna.

Jeżeli obliczona suma kontrolna nie jest zgodna z zapamiętaną, następuje ustawienie znacznika błędu. Powoduje to wprowadzenie nowej pozycji do tabeli błędów sterownika oraz przejście do trybu **STOP**. Jeżeli suma kontrolna nie zostanie obliczona, do okna komunikacji z programatorem nie są wprowadzane żadne informacje. Domyślna liczba słów do obliczania sumy kontrolnej jest równa 8.

## Komunikacja modułu programowalnego procesora (PCM) ze sterownikiem (Modele 331 i wyższe)

Wyspecjalizowane moduły dodatkowe (IOM), jak programowany moduł koprocera PCM, nie są w stanie przerwać pracy jednostki centralnej w celu nawiązania komunikacji. Jednostka centralna musi sama sprawdzić (okresowo), czy któryś z modułów nie żąda komunikacji. Sprawdzanie to może być realizowane asynchronicznie w czasie trwania wykonywania cyklu (proszę porównać z zamieszczonym poniżej schematem).

Jeżeli po wysłaniu zapytania do modułu dodatkowego, w odpowiedzi uzyskane zostanie żądanie komunikacji, jest ono wstawiane do kolejki w czasie fazy komunikacji systemowej.



Rysunek 2-4 Komunikacja modułu programowalnego procesora (PCM) ze sterownikiem



## Komunikacja modułu DSM ze sterownikiem

Moduły DSM302 i DSM314 są inteligentnymi modułami współpracującymi asynchronicznie z jednostkami centralnymi serii 90-30. Moduł DSM automatycznie wymienia dane z jednostką centralną za pośrednictwem zmiennych %Q, %I, %AQ i %AI. Wymiana danych poprzez magistralę, pomiędzy jednostką centralną sterownika a modułem DSM wymaga pewnego czasu. W tabeli 2-2 pokazano wpływ różnych konfiguracji modułu DSM na czas trwania cyklu pracy. Więcej informacji o modułach DSM można znaleźć w następujących podręcznikach:

- *Motion Mate DSM302 for Series 90-30 PLCs User's Manual*, GFK-1464.
- *Motion Mate DSM314 for Series 90-30 PLCs User's Manual*, GFK-1742.

## Warianty standardowego cyklu pracy sterownika

W standardowym cyklu pracy sterownika można wprowadzić lub wymusić pewne zmiany. Zmiany te, opisane w następnych akapitach, mogą być dokonane za pomocą oprogramowania.

### Tryb ze stałym czasem trwania cyklu pracy sterownika

W standardowym trybie pracy każdy cykl jest wykonywany tak szybko, jak to jest możliwe, co powoduje, że czasy trwania poszczególnych cykli mogą się różnić. Trybem alternatywnym jest tryb ze stałym czasem trwania (**CONSTANT SWEEP TIME**), w którym każdy cykl trwa przez taki sam okres czasu. Tryb ten przyjmowany jest domyślnie po ustawieniu parametru Configured Constant Sweep, będzie on zawsze ustawiany po przejściu z trybu zatrzymania (**STOP**) do trybu pracy (**RUN**). Tryb **CONSTANT SWEEP TIME** może dla jednostek centralnych CPU311-341 trwać od 5 do 200 ms, a dla jednostek centralnych CPU350-364 i 374 – od 5 do 500 ms.

Z powodu zmian czasu trwania poszczególnych faz cyklu w różnych cyklach, czas ten powinien być co najmniej 10 ms wyższy od czasu trwania cyklu wyświetlanego w linii statusu sterownika podczas jego pracy w trybie standardowym (**NORMAL SWEEP**). Ma to na celu uniknięcie wystąpienia błędów działania sterownika, powodowanych przedłużeniem cyklu poza określony limit.

Omawiany tryb pracy ma zastosowanie, jeżeli punkty wejścia/wyjścia lub wartości rejestrów muszą być sprawdzane ze stałą częstotliwością, jak to ma na przykład miejsce w algorytmach sterowania. Innym, może być wprowadzenie pewnego odstępu czasowego pomiędzy fazą obsługi wyjść sterownika a fazą obsługi wejść w następnym cyklu, co umożliwia ustalenie się stanu wejść po otrzymaniu danych wyjściowych z programu.

Jeśli czas przeznaczony na jeden cykl upłynie przed zakończeniem wykonywania cyklu, cały cykl, włącznie z fazami komunikacji z programatorem i innymi urządzeniami, zostaje dokończony. Jednakże na początku następnego cyklu zostanie zarejestrowany błąd przekroczenia czasu trwania cyklu.

## Konfigurowanie trybu o stałym czasie trwania

Istnieją dwa sposoby konfigurowania trybu Constant Sweep:

- W oprogramowaniu Logicmaster na ekranie konfiguracji CPU można ustawić parametry trwania cyklu. Po ustaleniu parametrów cyklu należy je zapisać w sterowniku, można tego dokonać jeśli sterownik znajduje się w trybie **STOP**. Po wprowadzeniu nowej konfiguracji, tryb ten będzie przyjmowany jako domyślny.
- W oprogramowaniu Logicmaster w sekcji Table Sweep Table w menu PLC Control and Status można ustawić tryb pracy sterownika oraz parametry czasowe. Parametry wyświetlone na tym ekranie są dostępne tylko gdy sterownik jest w trybie **RUN**. Parametry te są zapisywane tylko w sterowniku i są użyteczne tylko jeśli sterownik pozostaje w trybie **RUN**. Po zatrzymaniu, sterownik przechodzi w domyślny tryb pracy i pozostaje w nim po włączeniu. Taka metoda chwilowego konfigurowania cyklu pracy jest przydatna dla projektantów i w czasie usuwania błędów z programu.

## Cykl pracy sterownika w trybie STOP

W trybie **STOP** pracy sterownika, program sterujący nie jest wykonywany. Kontynuowana jest komunikacja z programatorem oraz ze specjalizowanymi modułami urządzeń dodatkowych. Dodatkowo w trybie **STOP** kontynuowane jest pobieranie danych z modułów wejść/wyjść oraz ich kontrola pod kątem wykrycia ewentualnych błędów w działaniu. W celu zwiększenia efektywności pracy, system operacyjny przeznaczony na komunikację przedziały czasu większe niż przypadku trybu **RUN** (zwykle około 50 ms na każdą fazę komunikacji). Wprowadzając odpowiednią konfigurację, można określić czy mają być obsługiwane wejścia i wyjścia modułów. Wejścia i wyjścia modułów są obsługiwane w trybie stop jeżeli parametr *IOScan-Stop* na ekranie ze szczegółowymi parametrami jednostki centralnej CPU zostanie ustawiony na wartość **YES**.

### Ostrzeżenie

**Jeśli został ustawiony w CPU parametr *IOScan-Stop* wyjścia mogą zostać załączone nawet, jeśli sterownik znajduje się w trybie STOP, ponieważ sterownik będzie zapisywał aktualne wartości z tablicy wyjść do poszczególnych wyjść modułów w czasie obsługi wyjść.**

## Tryby realizacji faz komunikacji

Domyślnie, czas przeznaczony na komunikację z programatorem jest ograniczony. Oznacza to, że jeśli czas realizacji żądania przekracza 6 milisekund, jest ono realizowane w kilku cyklach, tak że żaden cykl nie zostaje przedłużony o więcej niż 6 ms. Dla jednostek centralnych CPU 313, 323 i 331, w trybie programowania on-line podczas pracy (run), cykl może być przedłużony o maksymalnie 12 milisekund. Tryb realizacji faz komunikacji można zmienić na ekranie "Sweep Control" oprogramowania Logicmaster, zgodnie z instrukcjami podanymi w Rozdziale 5, "Sterowanie i status sterownika" podręcznika *Logicmaster 90™ Series 90™-30/20/Micro Programming Software User's Manual* (GFK-0466).

### Uwaga

Jeżeli czas przeznaczony na komunikację z programatorem jest ograniczony, dodatkowe moduły takie jak moduł PCM i GBC, obsługiwane w fazie komunikacji systemowej, będą w mniejszym stopniu wpływać na czas realizacji cyklu, ale spowoduje to wolniejszą obsługę wysyłanych przez nie żądań.

## Przełącznik w jednostkach centralnych serii 35x, 36x i 37x: Zmiany trybu i zabezpieczania przed zapisem

Wszystkie jednostki centralne 350-374 posiadają przełącznik (CPU 311-341 nie posiadają przełącznika), jednakże niektóre wersje oprogramowania systemowego nie obsługują wszystkich jego funkcjonalności. Dokładniejszy opis znajduje się w następnej części. Należy zwrócić uwagę, że na niektórych jednostkach centralnych jest on opisany ON/RUN i OFF/STOP, a na innych ON, OFF. Niezależnie od opisu funkcje przełącznika opisano poniżej.

### Zabezpieczenie pamięci Flash

Ta sprzętowa nie konfigurowalna funkcjonalność może być wykorzystywana do zabezpieczenia pamięci Flash przed zmianą przez nieupoważniony personel. Jeśli przełącznik jest ustawiony w pozycji ON, nie można dokonywać zapisu do pamięci Flash. Zapisywanie do pamięci Flash można dokonywać tylko jeśli przełącznik jest w pozycji OFF. Funkcjonalność ta pozostaje aktywna niezależnie od dwu konfigurowalnych ustawień.

### Run/Stop (konfigurowalne)

Ta funkcjonalność została wprowadzona w jednostkach centralnych z oprogramowaniem systemowym 7.00. Jest ona konfigurowana parametrem **R/S Switch** na ekranie konfiguracyjnym CPU. Parametr ten domyślnie jest ustawiony na *Disabled (nieaktywny)*. Jeśli parametr ten jest ustawiony na *Enabled (aktywny)*, można zatrzymać sterownik przekręcając przełącznik na OFF i uruchomić przekręcając na ON (jeśli nie ma błędów). W przypadku występowania błędów, może nastąpić:

- **Jeśli nie ma błędów krytycznych**, przełączenie z pozycji OFF na ON spowoduje przejście do trybu run i zapalenie diody RUN, nie nastąpi wyzerowanie tablicy błędów.
- **Jeśli występują błędy krytyczne**, przełączenie z pozycji OFF na ON spowoduje miganie diody RUN przez okres 5 sekund, a sterownik nie przejdzie w tryb run. Miganie diody oznacza, że w tablicy błędów zapisano jeden lub więcej błędów krytycznych. Można spróbować wyczyścić tabelę błędów przełączając przełącznik jeszcze raz z pozycji OFF na ON w czasie migania diody przez 5 sekund. (Jeśli miganie diody po 5 sekundach ustanie, ponowne przełączenie z pozycji OFF na ON rozpocznie odliczanie nowego czasu 5 sekund). Jeśli w ten sposób nie uda się uruchomić sterownika, należy znaleźć przyczynę powodującą występowanie błędu przed ponowieniem tej procedury. Dodatkowe informacje na ten temat zawarto w Rozdziale 3.

### Inne funkcje przełącznika Run/Stop

- Jeśli parametr **R/S Switch** ustawiono na *Enabled* a przełącznik jest w pozycji OFF, sterownik będzie w trybie STOP, a za pomocą oprogramowania nie będzie możliwe przełączenie sterownika w tryb RUN.
- Jeśli parametr **R/S Switch** ustawiono na *Enabled* a przełącznik jest w pozycji On oraz nie występują błędy krytyczne, za pomocą oprogramowania będzie możliwe przełączanie sterownika z trybu RUN w tryb STOP i odwrotnie.
- Jeśli parametr **R/S Switch** ustawiono na *Enabled*, przełącznik jest w pozycji ON, a sterownik jest zatrzymany można przejść do trybu RUN zarówno za pomocą przełącznika przekręcając go na OFF i z powrotem na ON lub wykorzystując oprogramowanie.

## Pamięć RAM i zabezpieczenie przed wymuszeniem stanu (konfigurowalne)

Ta funkcjonalność została wprowadzona w jednostkach centralnych z oprogramowaniem systemowym 8.00. Jest ona ustawiana parametrem **Mem Protect** na ekranie konfiguracyjnym CPU. Parametr ten domyślnie jest ustawiony na *Disabled (nieaktywny)*.

Jeśli parametr **Mem Protect** jest ustawiony na *Enabled*, a przełącznik znajduje się pozycji ON, wtedy:

- Pamięć RAM użytkownika (program i konfiguracja) nie może zostać zmieniona.
- Nie można wymusić zmiany stanu punktów dyskretnych.
- Nie można zmienić zegara TOD (time of day) za pomocą programatora ręcznego (można zmienić jego ustawienia za pomocą oprogramowania konfiguracyjnego).

### Zabezpieczenie kluczyków

Każde nowa CPU 350 – 374 jest dostarczana z dwoma kluczykami do przełącznika. Należy przechowywać kluczyki w bezpiecznym miejscu, w wykorzystywaniu funkcji zabezpieczających przełącznika. Zgubienie lub kradzież kluczyków może oznaczać zablokowanie sterownika lub umożliwienie dostępu do niego nieupoważnionym osobom. Istnieje możliwość zakupienia dodatkowych kluczyków. W zestawie znajdują się trzy zestawy kluczyków i są dostępne u dystrybutorów GE Fanuc. Zestawy mają numer katalogowy 44A736756-G01. Wszystkie CPU 350 – 374 posiadają takie same kluczyki.

### Blokowanie przełącznika

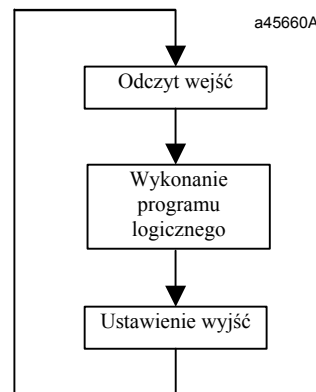
Gdy przełącznik kluczowy nie jest wykorzystywany, można wyłączyć wszystkie jego funkcjonalności. Aby zablokować przełącznik należy ustawić go w pozycji OFF oraz parametry **R/S Switch** i **Mem.Protect** ustawić na *Disabled* (ustawienie domyślne). W takim przypadku funkcje przełącznika będą zablokowane, a dostęp do sterownika nie będzie wymagał zastosowania kluczyka.

## Część 2: Organizacja programu, typy danych i typy zmiennych

Zamieszczona poniżej tabela zawiera maksymalną, dopuszczalną wielkość pamięci użytkownika w sterownikach serii 90-30.

Rozmiar pamięci użytkownika	
Model jednostki centralnej	Pamięć użytkownika (Kb)
CPU311	6
CPU313, CPU323	12
CPU331	16
CPU340	32
CPU341	80
CPU350	80 (wersja 9.00 i nowsze) 32 (przed wersją 9.00)
CPU351, CPU352, CPU360, CPU363, CPU364, CPU374	240 (wersja 9.00 i nowsze) 80 (przed wersją 9.00)

Rozpoczynając od oprogramowania systemowego 9.00 w CPU 351, 352, 360, 363, 364 i 374 rozmiary pamięci dla %R, %AI, i %AQ są konfigurowalne. (Więcej informacji zawarto w podręczniku *Logicmaster 90™ Series 90™-30/20/Micro Programming Software User's Manual*, GFK-0466K lub późniejszym oraz w podręczniku użytkownika danego oprogramowania konfiguracyjnego). W sterownikach 90-20 z CPU 211 program może mieć rozmiar do 2 KB, a maksymalna ilość szczebli w bloku logicznym może wynosić 3000 (program główny oraz podprogram). W sterownikach 90-30 maksymalny rozmiar wynosi 80 kilobajtów dla bloków C i 16 kilobajtów dla drabinki logicznej i bloków SFC; jednakże bloki SFC cześć z 16 kilobajtów pamięci wykorzystują dla wewnętrznych bloków danych. Program sterujący jest wykonywany wielokrotnie przez sterownik dopóki znajduje się on w trybie RUN, jak pokazano na poniższym rysunku.



Rozmiary programów oraz limity adresów zmiennych poszczególnych typów dla każdej z jednostek centralnych podano w podręczniku *Series 90-30 Programmable Controller Installation and Hardware Manual*, GFK-0356 lub w podręczniku *Series 90-20 Programmable Controller User's Manual*, GFK-0551.

Wszystkie programy rozpoczynają się tabelą deklaracji zmiennych, w której wyszczególnione są nazwy pomocnicze i objaśnienia zmiennych, przypisane odpowiednim elementom logicznym programu sterującego.

Edytor deklaracji bloków programu podaje wykaz podprogramów zadeklarowanych w programie.

## Bloki podprogramów (wyłącznie dla sterowników serii 90-30)

Program może w trakcie wykonywania wywołać podprogram. Podprogram musi zostać zadeklarowany przy pomocy edytora deklaracji bloków programu; dopiero wtedy podprogram ten można wywołać za pomocą instrukcji CALL. Maksymalnie w programie mogą zostać zadeklarowane 64 podprogramy, a dla każdego z bloków programu sterującego dozwolone są 64 instrukcje CALL. Maksymalny rozmiar podprogramów to 16 kB lub 3000 szczebli, lecz program główny wraz ze wszystkimi podprogramami musi zmieścić się w granicach obowiązujących dla poszczególnych jednostek centralnych.

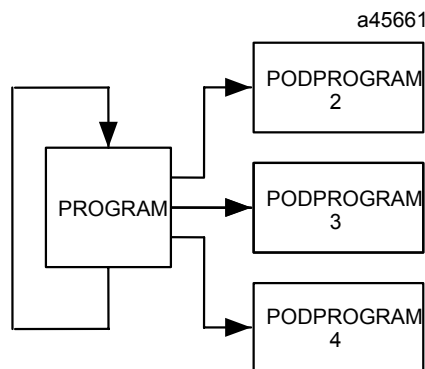
### Uwaga

Nie można korzystać z podprogramów w przypadku sterowników serii 90-20 oraz Micro.

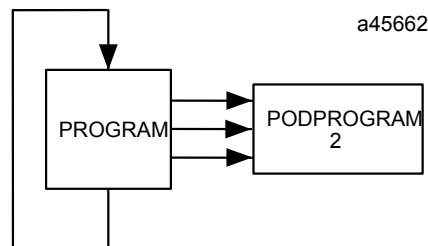
Używanie podprogramów jest opcjonalne. Podzielenie jednak programu na mniejsze bloki może uprościć proces programowania, ułatwić zrozumienie algorytmu sterowania oraz zmniejszyć globalną liczbę elementów logicznych w programie.

### Przykłady zastosowania podprogramów

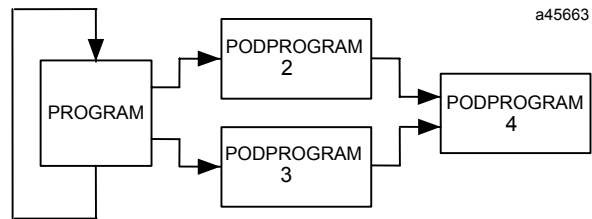
Przykładem wykorzystania podprogramów może być podział części logicznej programu na trzy podprogramy, z których każdy może być wywoływany w razie potrzeby z programu głównego. Główny blok programu sterującego może natomiast zawierać małą liczbę instrukcji logicznych, służących głównie do obsługi podprogramów.



Podprogram może być wywoływany dowolną liczbą razy podczas wykonywania programu sterującego. Bloki logiczne, które powtarzają się w programie sterującym, mogą być zatem wprowadzone w postaci podprogramów. Można je następnie wywoływać w odpowiednich miejscach programu. Pozwala to na zmniejszenie całkowitych rozmiarów programu.



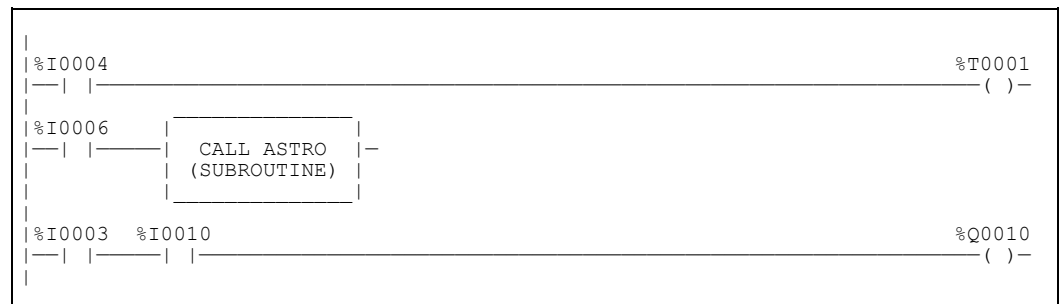
Dodatkowo, podprogramy mogą być też wywoływane z poziomu innych podprogramów. Podprogramy mogą być nawet wywoływane rekurencyjnie.



Sterownik dopuszcza jedynie osiem zagnieżdżonych (jedno wewnątrz drugiego) wywołań, a po przekroczeniu tego limitu zostanie zarejestrowany błąd działania sterownika i zostanie on zatrzymany (przejdzie w tryb **Stop/Fault**). Przy obliczaniu poziomu wywołania podprogramów, program główny traktowany jest jako poziom 1.

## Wywoływanie podprogramów w szczelbu programu sterującego

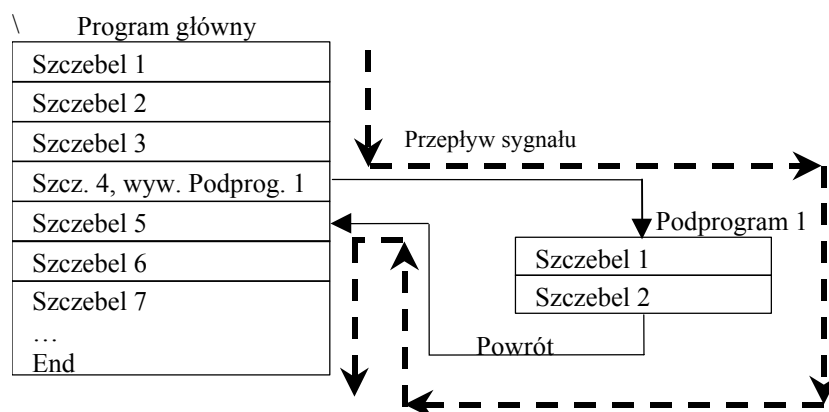
Podprogram jest wykonywany po wywołaniu z bloku głównego programu lub z poziomu innego podprogramu.



Przykład ten ilustruje wywoływanie podprogramu z bloku programu głównego.

## Wykonywanie programu zawierającego podprogramy

Podprogram wywołany z programu lub innego podprogramu będzie wykonywany aż do jego końca, po czym dalej będzie wykonywany program lub podprogram, z którego został on wywołany. Wykonywanie programu zostanie rozpoczęte od następnego szczelbu po szczelbu, w którym został wywołany podprogram. W przykładzie zamieszczonym poniżej linią kreskowaną oznaczono kierunek przepływu sygnału (kolejność wykonywania programu). W przykładzie tym prosty dwu szczelbowy podprogram jest wywoływany w szczelbu 4 programu głównego. Po wykonaniu podprogramu, sygnał powraca do programu głównego w szczelbu 5.



## Podprogramy wykonywane okresowo

Podprogramy wykonywane okresowo można wykorzystywać w systemach z jednostką centralną CPU 340 lub nowszą, w wersji 4.20 lub późniejszej. Należy pamiętać o następujących ograniczeniach:

1. Bloki funkcyjne przełączników czasowych (TMR, ONDTR i OFDTR) nie są realizowane poprawnie w podprogramach wykonywanych okresowo. Zastosowanie bloków funkcyjnych DOIO w podprogramach wykonywanych okresowo z odwołaniami do wyspecjalizowanych modułów wejścia/wyjścia (HSC, PowerMate, APM, Genius, itp.) powoduje przerwanie komunikacji pomiędzy tymi modułami a jednostką centralną. Zmienne systemowe %S1 (FST\_SCN - pierwszy cykl pracy sterownika) i %S2 (LST\_SCN - ostatni cykl pracy sterownika) podczas wykonywania takiego podprogramu nie mają określonego stanu. Podprogramy wykonywane okresowo nie mogą być wywoływane z poziomu innych podprogramów, jak również nie można z nich wywoływać innych podprogramów.
2. Opóźnienie wykonania podprogramu (czyli maksymalna zwłoka czasowa pomiędzy planowanym momentem wykonania podprogramu a rzeczywistym wykonaniem podprogramu) wynosi około 0.35 ms, jeśli w kasecie podstawowej nie ma zainstalowanego modułu programowalnego koprocatora (PCM), modułu komunikacyjnego (CMM) lub modułu koprocatora wizualizacji (ADC). Jeśli którykolwiek z tych modułów znajduje się w kasecie podstawowej (nawet jeśli nie jest używany i nie jest skonfigurowany), opóźnienie może wynieść nawet 2.25 ms. Z tego powodu stosowanie podprogramów wykonywanych okresowo w systemach z wymienionymi modułami nie jest zalecane.

## Typy zmiennych

Dane wykorzystywane w programie sterującym przechowywane są jako zmienne rejestrowe lub zmienne dyskretne.

Tabela 2-4. Zmienne rejestrowe

Typ	Opis
%R	Zmienna 16-bitowa oznaczająca rejestr, w którym można przechowywać dane programu sterującego (np. wyniki obliczeń). Symbolowi powinien towarzyszyć adres rejestru (np.%R00201).
%AI	Przedrostek %AI oznacza rejestr wejścia analogowego. Po przedrostku podawany jest adres rejestru (np.%AI0015). Zmienna ta przechowuje wartość jednego rejestru wejścia analogowego lub inną wartość.
%AQ	Przedrostek %AQ oznacza rejestr wyjścia analogowego. Po przedrostku podawany jest adres rejestru (np.%AQ0056). Zmienna ta przechowuje wartość jednego rejestru wyjścia analogowego lub inną wartość.



## Uwaga

Wartości wszystkich zmiennych rejestrowych są przechowywane w przypadku wyłączenia zasilania sterownika i po ponownym włączeniu pozostają takie same.

Tabela 2-5. Zmienne dyskretne

Typ	Opis
%I	Zmienne poprzedzone tym przedrostkiem reprezentują fizyczne wejścia dyskretne. Po symbolu podawany jest adres zmiennej w tabeli stanu wejść (np.%I00121). Wartości wszystkich zmiennych typu %I są umieszczane w tabeli stanu wejść, przechowywany jest w niej aktualny stan wszystkich wejść sterownika, uzyskany z modułów wejściowych podczas ostatniego odczytu. Adres zmiennej jest przyporządkowywany modułom wejść dyskretnych za pomocą oprogramowania konfiguracyjnego lub programatora ręcznego. Do momentu przyporządkowania adresu zmiennej moduł wejść nie przesyła żadnych danych do sterownika. Zmienne typu %I mogą posiadać pamięć stanu (tzn. zachowywać poprzednią wartość po wyłączeniu zasilania sterownika i ponownym jego włączeniu)*.
%Q	Zmienne poprzedzone tym przedrostkiem reprezentują fizyczne wyjścia dyskretne. Funkcja sprawdzania przełączników oprogramowania Logicmaster 90-30/Micro sprawdza, czy zmienne %Q nie są wielokrotnie wykorzystywane w funkcjach obsługi przełączników i wyjść. Począwszy od wersji 3 oprogramowania, można wybrać żądany poziom sprawdzania przełączników (SINGLE, WARN MULTIPLE, lub MULTIPLE). Więcej informacji o tym mechanizmie podano w podręczniku <i>Logicmaster 90-30/20Micro Programming Software User's Manual, GFK-0466K</i> . Po symbolu podawany jest adres zmiennej w tabeli stanu wyjść (np.%Q00016). Wartości wszystkich zmiennych typu %Q umieszczane są w tabeli stanu wyjść, przechowywany jest w niej aktualny stan wszystkich wyjść sterownika, ostatnio ustawionych przez program sterujący. Wartości te są przesyłane do modułów wyjściowych podczas fazy obsługi wyjść. Adres zmiennej jest przyporządkowywany modułom wyjść dyskretnych za pomocą oprogramowania konfiguracyjnego lub programatora ręcznego. Do momentu przyporządkowania adresu zmiennej, nie są przesyłane żadne dane do modułu. Zmienne typu %Q mogą posiadać pamięć stanu (tzn. zachowywać poprzednią wartość po wyłączeniu zasilania sterownika i ponownym jego włączeniu). *
%M	Zmienne poprzedzone tym przedrostkiem reprezentują wewnętrzne zmienne dyskretne programu sterującego. Funkcja sprawdzania przełączników oprogramowania Logicmaster 90-30/Micro sprawdza, czy zmienne %M nie są wielokrotnie wykorzystywane w funkcjach obsługi przełączników i wyjść. Począwszy od wersji 3 oprogramowania, można wybrać żądany poziom sprawdzania przełączników (SINGLE, WARN MULTIPLE, lub MULTIPLE). Szczegółowe informacje na ten temat podano w podręczniku GFK-0466. Zmienne typu %M mogą posiadać pamięć stanu (tzn. zachowywać poprzednią wartość po wyłączeniu zasilania sterownika i ponownym jego włączeniu). *
%T	Zmienne poprzedzone tym przedrostkiem reprezentują zmienne chwilowe. Zmienne te nie są sprawdzane pod kątem liczby wykorzystania w przełącznikach, przez co można je wielokrotnie wykorzystywać w programie, nawet jeżeli włączony jest mechanizm sprawdzania przełączników. Zmienne typu %T mogą być stosowane w celu uniknięcia konfliktu przełączników przy posługiwaniu się funkcjami wytnij/ wklej i funkcjami zapisu/ dołączania do pliku. Ponieważ pamięć przeznaczona na te zmienne nie jest podtrzymywana po wyłączeniu zasilania, lub po przejściu z trybu <b>RUN</b> do <b>STOP</b> , lub <b>STOP</b> do <b>RUN</b> , zmienne te nie mogą być wykorzystywane w przełącznikach z pamięcią.
%S	Przedrostek %S umieszczany jest przed zmiennymi systemowymi. Umożliwiają one dostęp do danych systemowych, takich jak informacje o błędach działania sterownika, pracy modułów wejść/wyjść. Wśród zmiennych systemowych można wyróżnić cztery grupy, oznaczone odpowiednio symbolami %S, %SA, %SB oraz %SC.  Zmienne %S, %SA, %SB i %SC mogą być przypisane dowolnym stykom.  Zmienne %SA, %SB i %SC mogą być przypisane przełącznikom z pamięcią stanu -(M)-. Zmienne %S mogą być wykorzystane jako parametry wejściowe (typu słowo lub słowo bitowe) bloku funkcyjnego.  Zmienne %SA, %SB i %SC mogą być wykorzystane jako parametry wejściowe lub wyjściowe (typu słowo lub słowo bitowe) bloku funkcyjnego.
%G	Przedrostek %G umieszczany jest przed zmiennymi globalnymi. Umożliwiają one dostęp do danych wspólnie wykorzystywanych przez kilka sterowników. Zmienne typu %G zawsze posiadają pamięć stanu i mogą być przypisywane stykom i przełącznikom z pamięcią stanu. Nie mogą być natomiast przypisywane do przełączników bez pamięci.

\* Posiadanie przez zmienną pamięci stanu uzależnione jest od rodzaju przełącznika. Więcej informacji podano w punkcie "Pamięć stanu", na następnej stronie.

## Nazwy pomocnicze

Użytkownik może dodatkowo ze zmienną powiązać nazwę pomocniczą. Nazwa pomocnicza jest przydatna ze względu na możliwość przekazywania informacji o przeznaczeniu funkcji lub adresu. Na przykład w systemie sterowania przekaźnik wyjściowy %Q0001 jest wykorzystywany do załączania rozrusznika silnika sterującego pompą, może zostać nazwy „pompa numer 1”. Powiązanie nazwy pomocniczej POMPA1 z %Q0001 może pomóc pracownikom konserwującym system w rozpoznaniu przeznaczenia przekaźnika %Q0001.

Nazwy pomocnicze mogą mieć długość do siedmiu znaków i muszą rozpoczynać się literą. Aby rozróżniać adresy pamięci (zmiennie), a nazwy pomocnicze, pierwszym znakiem w adresie jest znak „%”. Dla przykładu M1 oznacza nazwę pomocniczą, ale %M1 oznacza adres w pamięci. Więcej informacji na temat nazw pomocniczych można znaleźć w podręczniku GFK-0466 (the Logicmaster user's manual for the Series 90-30 PLC).

## Wymuszenie zmiany wartości zmiennych

Zmienne typu %I, %Q, %M i %G posiadają możliwość zmiany wartości wywołanej z zewnątrz (chwilowe przełączenie na wartość przeciwną - transition lub przełączenie trwale z blokadą wartości - override). Zmienne typu %T, %S, %SA, %SB i %SC posiadają możliwość chwilowego przełączenia wartości, lecz nie posiadają możliwości zmiany trwałej z blokadą wartości. Chwilowe przełączenie na wartość przeciwną wykorzystywane jest przez jednostkę centralną dla liczników i przekaźników uaktywnianych zboczem sygnału. Należy zwrócić uwagę, że liczniki nie korzystają z tego samego typu bitów chwilowo przełączanych na wartość przeciwną co przekaźniki. Bity chwilowo przełączane na wartość przeciwną pamiętane są w specjalnym, wewnętrznym miejscu, niedostępnym bezpośrednio dla programisty, przyporządkowanym do każdego adresu zmiennej.

Jednostki centralne CPU 331 i wyższe dopuszczają możliwość trwałego ustawiania wartości bitów sterujących z zewnątrz. Po ustawieniu wartości takiego bitu, wartość odpowiadającej mu zmiennej nie może zostać zmieniona przez program sterujący ani też przez urządzenie wejściowe; a jedynie poprzez rozkaz programatora. W jednostkach 323, 321, 313, 311 i Micro nie jest dostępna możliwość trwałego ustawienia wartości bitów sterujących z zewnątrz.

## Pamięć stanu

Zmienna posiada pamięć stanu, jeśli jej wartość jest przechowywana po wyłączeniu sterownika. Sterowniki serii 90 przechowują część logiczną programu sterującego, tabele błędów działania i diagnostyczne, zablokowane wartości zmiennych, wartości zmiennych rejestrowych (typu %R, %AI, %AQ), wartości zmiennych dyskretnych (typu %I, %S, %G, bity - znaczniki błędów działania) oraz wartości zmiennych dyskretnych typu %M i %Q w przypadku, gdy są one przypisane przekaźnikom z pamięcią. Wartości zmiennych %T nie są zapamiętywane. Chociaż, zgodnie z podanymi powyżej informacjami, pamiętana jest wartość zmiennych %SC, domyślnie, nie jest pamiętana wartość dla zmiennych %S, %SA i %SB.

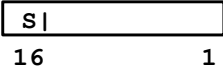

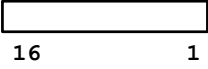

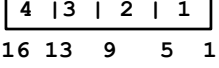

Zmienne typu %Q i %M, przypisane przekaźnikom bez pamięci, nie posiadają pamięci stanu (tzn. ich wartość nie jest pamiętana po przejściu sterownika z trybu **STOP** do trybu **RUN**). Przekaźniki bez pamięci to przekaźniki o następujących symbolach: —( )—, —(/)—, —(S)— oraz —(R)—.

Gdy zmienne typu %Q i %M zostaną przypisane przekaźnikom z pamięcią lub wykorzystane jako zmienne wyjściowe bloków funkcyjnych, ich wartość jest przechowywana po wyłączeniu zasilania sterownika oraz po przejściu z trybu pracy **RUN** w tryb **STOP** i ponownie w tryb **RUN**. Przekaźniki z pamięcią to przekaźniki o następujących symbolach: —(M)—, —(/M)—, —(SM)— oraz —(RM)—.

Stan zapamiętanej i przechowywanej wartości zmiennej typu %Q lub %M zostaje określony w miejscu jej ostatniego wykorzystania. Przykładowo, jeżeli zmienna %Q0001 zostanie przypisana przekaźnikowi z pamięcią, jej wartość będzie pamiętana. Nie mniej jednak, jeżeli zmienna %Q0001 zostanie przypisana przekaźnikowi bez pamięci, nie będzie ona posiadała pamięci stanu.

# Typy danych

Tabela 2-6. Typy danych

Typ	Nazwa	Opis	Format zapisu
INT	Liczby całkowite ze znakiem	Liczby całkowite ze znakiem zajmują 16 bitów pamięci i są zapisywane w formacie dopełnienia do dwóch dla liczb ujemnych. (Bit 16 to bit znaku.) Zakres: -32,768 do +32,767.	<p>Rejestr 1</p>  <p>(16 pozycji bitów)</p>
DINT	Liczby całkowite podwójnej precyzji ze znakiem	Liczby całkowite podwójnej precyzji ze znakiem są przechowywane w 32 bitach pamięci i są zapisywane w formacie dopełnienia do dwóch dla liczb ujemnych. (Bit 32 to bit znaku.) Zakres: od -2 147 483 647 do +2,147,483,867.	<p>Rejestr 2                      Rejestr 1</p>  <p>(Dopełnienie do dwóch)</p>
BIT	Bit	Dana zajmująca najmniejszą komórkę pamięci. Może przyjmować wartość 1 lub 0. Słowo bitowe może mieć długość N.	
BYTE	Bajt	Dana zawierająca 8 bitów. Zakres: od 0 do 255 (0 do FF w systemie heksadecymalnym).	
WORD	Rejestr	Słowo zajmuje 16 kolejnych bitów pamięci sterownika, ale w przeciwieństwie do ciągu bitów reprezentującego w pamięci liczbę, bity mogą być niezależne od siebie odczytywane i zmieniane. Każdy bit posiada swój własny stan logiczny (1 lub 0). Zakres wartości: 0 do FFFF (w systemie heksadecymalnym)	<p>Rejestr 1</p>  <p>(16 pozycji bitów)</p>
DWORD	Słowo podwójnej długości	Słowo podwójnej długości posiada taką samą charakterystykę jak typ WORD, jedna różnica to wykorzystywanie do przechowywania 32 kolejnych bitów, a nie 16 kolejnych bitów. Zakres wartości: 0 do FFFF (w systemie heksadecymalnym)	<p>Rejestr 2                      Rejestr 1</p>  <p>(32 bity stanów)</p>
BCD-4	Czterocyfrowa liczba dziesiętna zakodowana w formacie BCD	Czterocyfrowe liczby dziesiętne zakodowane w formacie BCD zajmują 16 bitów pamięci. Każda z czterech cyfr tej liczby jest zakodowana w czterech bitach i może reprezentować cyfrę z zakresu od 0 do 9. Zakres wartości: od 0 do 9999.	<p>Rejestr 1</p>  <p>(4 cyfry BCD)</p>
REAL	Liczba rzeczywista	Liczby rzeczywiste zajmują 32 kolejne bity pamięci (w rzeczywistości są to dwie kolejne komórki pamięci po 16 bitów każda). Zakres wartości: ± 1.401298E-45 do ± 3.402823E+38.	<p>Rejestr 2                      Rejestr 1</p>  <p>(Dopełnienie do dwóch)</p>

Z = Bit znaku (0 = dodatni, 1 = ujemny).

## Zmienne systemowe statusu

Zmienne typu %S, %SA, %SB oraz %SC są zmiennymi systemowymi. Każda z tych zmiennych posiada nazwę pomocniczą. Przykładowe nazwy zmiennych sterujących podstawą czasu generatora sygnału prostokątnego to: T\_10MS, T\_100MS, T\_SEC oraz T\_MIN. Przykładowe nazwy zmiennych ułatwiających programowanie sterownika to: FST\_SCN, ALW\_ON i ALW\_OFF.

### Uwaga

Wartości zmiennych systemowych %S można jedynie odczytywać, nie można ich zmieniać. Można jednakże zmieniać wartości zmiennych %SA, %SB i %SC.

Poniżej podano zmienne systemowe, które można wykorzystywać w programie sterującym. Podczas wprowadzania elementów logicznych można używać zarówno adresów zmiennych, jak i nazw pomocniczych. Więcej informacji na temat błędów i sposobów ich usuwania podano w Rozdziale 3, "Błędy działania sterownika - opis i ich usuwanie". Nie można wykorzystywać tych specjalnych nazw pomocniczych do innych zmiennych.

Tabela 2-7. Zmienne systemowe

Zmienna	Nazwa	Definicja
%S0001	FST_SCN	Jeżeli bieżący cykl jest pierwszym cyklem pracy sterownika, ustawiana jest wartość 1.
%S0002	LST_SCN	Jeżeli bieżący cykl jest ostatnim cyklem pracy sterownika, ustawiana jest wartość 0.
%S0003	T_10MS	Podstawa czasu generatora sygnału prostokątnego: 0,01 s.
%S0004	T_100MS	Podstawa czasu generatora sygnału prostokątnego: 0,1 s.
%S0005	T_SEC	Podstawa czasu generatora sygnału prostokątnego: 1.0 s.
%S0006	T_MIN	Podstawa czasu generatora sygnału prostokątnego: 1.0 minuta.
%S0007	ALW_ON	Styk zawsze zwarty.
%S0008	ALW_OFF	Styk zawsze otwarty.
%S0009	SY_FULL	Jeżeli tabela błędów działania sterownika jest wypełniona całkowicie, ustawiana jest wartość 1. Wartość zmiennej jest ustawiana ponownie na 0, gdy z tabeli zostanie usunięta część informacji lub cała zawartość tabeli zostanie wymazana.
%S0010	IO_FULL	Jeżeli tabela błędów działania układów wejść/wyjść jest wypełniona całkowicie, ustawiana jest wartość 1. Wartość zmiennej jest ustawiana ponownie na 0, gdy z tabeli zostanie usunięta część informacji lub cała zawartość tabeli zostanie wymazana.
%S0011	OVR_PRE	Wymuszona z zewnątrz zmiana wartości zmiennej typu %I, %Q, %M lub %G.
%S0013	PRG_CHK	Ustawiana na 1 po włączeniu kontroli programu.
%S0014	PLC_BAT	Jeżeli bateria jednostki centralnej w wersji 4 lub późniejszej jest rozładowana, zmienna przyjmuje wartość 1. Zmienna ta uaktualniana jest jeden raz w ciągu cyklu.
%S0017	SNP_XACT	Do jednostki centralnej przyłączony jest aktywny master protokołu SNP-X.
%S0018	SNP_X_RD	Master protokołu SNP-X odczytał dane z jednostki centralnej.
%S0019	SNP_X_WT	Master protokołu SNP-X zapisał dane do jednostki centralnej.
%S0020		Zmienna ustawiana na 1 po pomyślnym wykonaniu relacji matematycznej operującej na danych typu rzeczywistego. Jeżeli parametr wejściowy ma wartość NaN, bit ten jest zerowany.
%S0032		Zarezerwowana dla oprogramowania.
%SA0001	PB_SUM	Suma kontrolna wyliczona dla programu sterującego nie jest równa wartości porównawczej. Jeśli błąd ten został spowodowany przez chwilową usterkę, można go usunąć poprzez ponowne zapisanie programu w pamięci sterownika. Jeśli natomiast uszkodzona została pamięć RAM, należy wymienić jednostkę centralną sterownika.
%SA0002	OV_SWP	Poprzedni cykl pracy sterownika trwał dłużej niż zostało to wyszczególnione przez użytkownika. Wartość tej zmiennej jest ustawiana ponownie na 0, gdy czas trwania cyklu pracy sterownika nie przekroczy zadanej wartości. Wartość 0 jest również ustawiana w czasie przejścia z trybu zatrzymania <b>STOP</b> do trybu pracy <b>RUN</b> . Zmienna ta zachowuje ważność wyłącznie w trybie ze stałym czasem trwania cyklu.
%SA0003	APL_FLT	Błąd w działaniu programu sterującego. Wartość tej zmiennej zostaje ponownie ustawiona na 0 po przejściu sterownika z trybu <b>STOP</b> do trybu <b>RUN</b> .
%SA0009	CFG_MM	Wartość tej zmiennej ustawiana jest na 1, jeżeli konfiguracja zapisana w

Zmienna	Nazwa	Definicja
		oprogramowaniu, sprawdzana podczas zapisu konfiguracji lub włączania zasilania układu, nie jest zgodna z konfiguracją sprzętową. Wartość tej zmiennej jest ustawiana ponownie na 0, gdy konfiguracje zgadzają się.
%SA0010	HRD_CPU	Wartość tej zmiennej ustawiana jest na 1, po wykryciu przez procedury diagnostyczne uszkodzenia jednostki centralnej. Wartość tej zmiennej jest ustawiana ponownie na 0 po wymianie jednostki centralnej.
%SA0011	LOW_BAT	Niskie napięcie baterii zasilających. Wartość tej zmiennej jest ustawiana ponownie na 0 po wymianie baterii.
%SA0014	LOS_IOM	Utrata komunikacji pomiędzy jednostką centralną a modulem wejść/wyjść. Wartość tej zmiennej ustawiana jest ponownie na 0 po wymianie modułu, wyłączeniu i ponownym włączeniu zasilania.
%SA0015	LOS_SIO	Utrata komunikacji z jednym ze specjalizowanych modułów. Wartość tej zmiennej ustawiana jest ponownie na 0 po wymianie modułu, wyłączeniu i ponownym włączeniu zasilania.
%SA0019	ADD_IOM	Wartość tej zmiennej jest ustawiana na 1, po dodaniu do kasy modułu wejść/wyjść. Wartość 0 jest ponownie ustawiana po wyłączeniu i ponownym włączeniu zasilania sterownika, gdy zapamiętana konfiguracja odpowiada rzeczywistej.
%SA0020	ADD_SIO	Wartość tej zmiennej jest ustawiana na 1, jeżeli do kasy dodano jeden z opcjonalnych modułów. Wartość 0 jest ponownie ustawiana po wyłączeniu i ponownym włączeniu zasilania sterownika, gdy zapamiętana konfiguracja odpowiada rzeczywistej.
%SA0027	HRD_SIO	Wartość tej zmiennej ustawiana jest na 1, jeżeli wykryte zostało uszkodzenie sprzętowe w jednym z modułów opcjonalnych. Wartość tej zmiennej ustawiana jest ponownie na 0 po wymianie modułu, wyłączeniu i ponownym włączeniu zasilania.
%SA0031	SFT_SIO	Wartość tej zmiennej ustawiana jest na 1, jeżeli wykryty został niepoprawialny błąd programowy w jednym z modułów opcjonalnych. Wartość 0 jest ponownie ustawiana po wyłączeniu i ponownym włączeniu zasilania sterownika, gdy zapamiętana konfiguracja odpowiada rzeczywistej.
%SB0010	BAD_RAM	Ustawiana na 1, jeżeli jednostka centralna wykryła uszkodzenie pamięci RAM podczas rozruchu. Ustawiana na 0, jeżeli jednostka centralna nie wykryje uszkodzenia pamięci RAM podczas rozruchu.
%SB0011	BAD_PWD	Ustawiana na 1, w przypadku wprowadzenia błędnego hasła dostępu do sterownika. Ustawiana na 0 po usunięciu zawartości tabeli błędów działania sterownika.
%SB0013	SFT_CPU	Ustawiana na 1, jeżeli jednostka centralna sterownika wykryje nienaprawialny błąd w oprogramowaniu. Ustawiana na 0 po usunięciu zawartości tabeli błędów działania sterownika.
%SB0014	STOR_ER	Ustawiana na 1, jeżeli w czasie operacji ładowania programu sterującego z programatora wykryty zostanie błąd. Ustawiana na 0 po udanym załadowaniu programu.
%SC0009	ANY_FLT	Ustawiana na 1 w przypadku wykrycia jakichkolwiek błędów działania sterownika lub układów wejść/wyjść. Ustawiana na 0, gdy obie tabele błędów działania nie zawierają komunikatów o błędach.
%SC0010	SY_FLT	Ustawiana na 1 jeżeli wystąpiło uszkodzenie powodujące wprowadzenie nowej pozycji do tabeli błędów działania sterownika. Ustawiana na 0 jeżeli tabela ta nie zawiera żadnych komunikatów o błędach.
%SC0011	IO_FLT	Ustawiana na 1 jeżeli wystąpił jakikolwiek błąd działania układów wejść/wyjść sterownika, zapisywany w tablicy błędów działania wejść/wyjść. Ustawiana na 0 jeżeli tabela ta nie zawiera żadnych komunikatów o błędach.
%SC0012	SY_PRES	Wartość tej zmiennej jest równa 1 przez cały czas, jeśli tylko w tabeli błędów działania sterownika znajduje się choć jedna informacja o błędzie. Ustawiana na 0 jeżeli tabela ta nie zawiera żadnych komunikatów o błędach.
%SC0013	IO_PRES	Wartość tej zmiennej jest równa 1 przez cały czas, jeśli tylko w tabeli błędów działania układów wejść/wyjść znajduje się choć jedna informacja o błędzie. Ustawiana na 0 jeżeli tabela ta nie zawiera żadnych komunikatów o błędach.
%SC0014	HRD_FLT	Wartość ustawiana na 1 w momencie wystąpienia uszkodzenia sprzętowego. Ustawiana na 0, gdy obie tabele błędów działania nie zawierają komunikatów o błędach.
%SC0015	SFT_FLT	Wartość ustawiana na 1 w momencie pojawienia się błędu oprogramowania. Ustawiana na 0, gdy obie tabele błędów działania nie zawierają komunikatów o błędach.

**Uwaga:** Wszystkie pozostałe zmienne %S, nie wymienione w powyższej tabeli są zmiennymi zarezerwowanymi i nie powinny być wykorzystywane w części logicznej programu sterującego.

## Podstawowe informacje o instrukcjach języka drabinkowego

Każdy szczebel drabiny logicznej programu sterującego składa się z jednego lub większej liczby elementów logicznych. Mogą to być zarówno proste przełączniki, jak i bardziej złożone bloki funkcyjne.

### Format przełączników i styków

Oprogramowanie narzędziowe udostępnia kilka typów przełączników oraz styków. Elementy te zapewniają podstawową kontrolę przepływu sygnału w programie sterującym. Przykładem mogą być styki otwarte i przełączniki o stykach zamkniętych. Każdy z tych styków i przełączników posiada jedno wejście i jedno wyjście. Połączone, umożliwiają sterowanie przepływem sygnału.

Każdemu przełącznikowi i stykowi musi być przypisana zmienna, która jest wprowadzana równocześnie z wprowadzeniem tego elementu do szczebla. W przypadku styków, zmienna ta reprezentuje bit w pamięci sterownika, którego wartość określa przepływ sygnału przez styk. W poniższym przykładzie, jeśli wartość zmiennej %I0122 jest ustawiona na 1, przez styk przepływa sygnał:

```
%I0122
-| |-
```

W przypadku przełączników, zmienna ta reprezentuje bit w pamięci sterownika, którego wartość jest ustawiana w zależności od tego, czy do przełącznika dopływa sygnał. W poniższym przykładzie, jeśli do przełącznika dopływa sygnał, wartość zmiennej %Q0004 jest ustawiana na 1:

```
%Q0004
-( )-
```

Oprogramowanie oraz programator ręczny posiadają możliwość kontroli wielokrotnego wykorzystania tych samych zmiennych typu %Q i %M, przypisanych różnym przełącznikom i wyjściom bloków funkcyjnych.

### Format bloków funkcyjnych

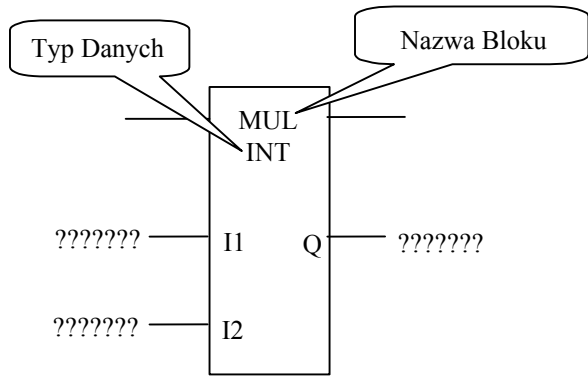
Niektóre funkcje stosowane w programie sterującym mają bardzo prostą postać, np. funkcja MCR, pokazana poniżej:

```
-[ MCR ]-
```

Większość funkcji posiada bardziej złożoną postać. Mogą wymagać podania kilku danych (parametrów).

Przykładem takiej funkcji jest instrukcja mnożenia (MUL). Jej parametry są typowe dla wielu funkcji. Jednakże, ilość i rodzaj parametrów jest różny dla różnych bloków funkcyjnych. Górna część bloku zawiera zwykle nazwę funkcji. W niektórych przypadkach może tu być również podany typ danych, w niniejszym przypadku liczba całkowita ze znakiem.

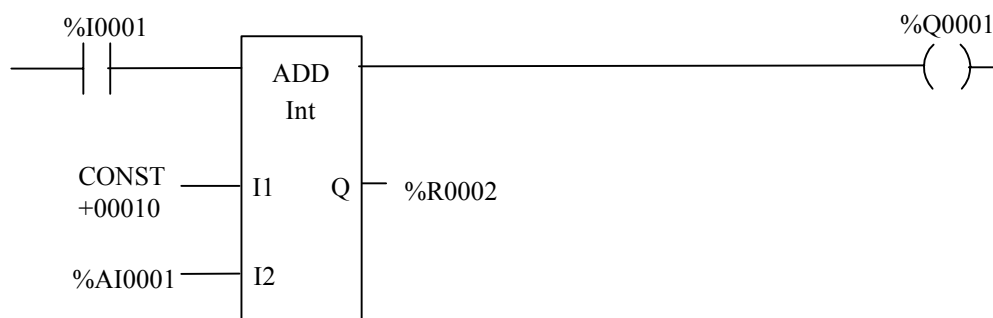
Wiele funkcji (instrukcji) umożliwia wybór lub zmianę typu danych, na których funkcja wykonuje operacje, już po wyborze funkcji. Przykładowo, typ danych INT dla funkcji MUL można zmienić na DINT. Typy danych opisano we wcześniejszej części niniejszego rozdziału.



## Parametry bloków funkcyjnych (instrukcji)

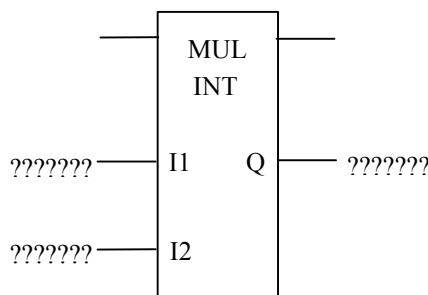
Każda linia dochodząca do lewej strony bloku funkcyjnego reprezentuje wejście tej funkcji. Rozróżnia się dwa rodzaje wejść bloków funkcyjnych – dyskretne i analogowe. Wejścia dyskretne mogą być załączone lub wyłączone. Na poniższym rysunku styk %I0001 jest przykładem wejścia dyskretnego. Wejścia analogowe mogą być zarówno stałe jak i zmienne. Stała jest wartością występującą w postaci jawnej. Zmienna jest identyfikowana poprzez jej adres. Ogólnie można przyjąć, że zmienne są wykorzystywane jeśli dane wejściowe zmieniają się. Na przykład zmienna może być adresem wejścia z danymi z urządzenia pomiarowego.

W poniższym przykładzie (blok funkcyjny dodawania ADD) parametr wejściowy I1 jest stałą, natomiast parametr I2 zmienną.



Każda linia wychodząca z prawej strony bloku funkcyjnego reprezentuje wyjście. Wyjścia mogą być zarówno dyskretne jak i analogowe. W przypadku wejść/wyjść analogowych ich wartości umieszczane są w rejestrach. W powyższym przykładzie sygnał dyskretny OK z bloku funkcyjnego kontroluje przekaźnik %Q0001. W tym przykładzie, dane z wyjścia Q pomimo, że reprezentują wynik działania matematycznego, to są zapisywane w rejestrze %R0002.

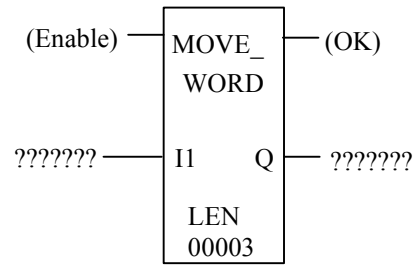
Jeżeli z lewej strony bloku funkcyjnego podawany jest znak zapytania, można wprowadzić bezpośrednio samą wartość, adres pod którym znajduje się wartość lub zmienną, określającą adres, gdzie znajduje się wartość parametru. Jeżeli znak zapytania umieszczony jest z prawej strony bloku funkcyjnego, należy podać adres, pod który ma być zapisana wartość wyjściowa z bloku funkcyjnego albo zmienną, określającą adres, pod który ma być zapisana wartość wyjściowa z bloku funkcyjnego.



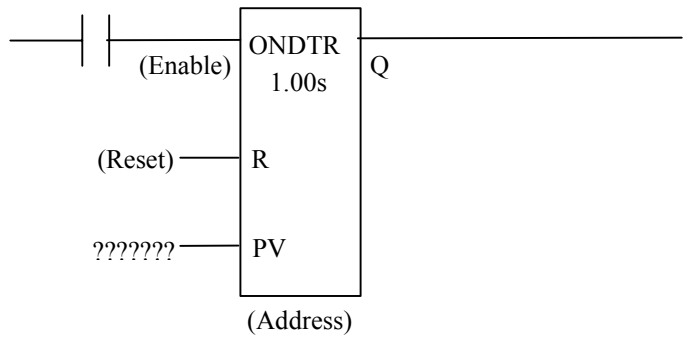
Większość bloków funkcyjnych nie powoduje zmian wartości wejść, wykorzystuje dane wejściowe podczas wykonywania operacji i zapisuje wynik wykonanej operacji w zmiennej wyjściowej.

W przypadku funkcji operujących na grupach adresów, można ustalić długość. W zamieszczonym poniżej bloku funkcyjnym, parametr LEN podaje liczbę słów wejściowych do przesunięcia.



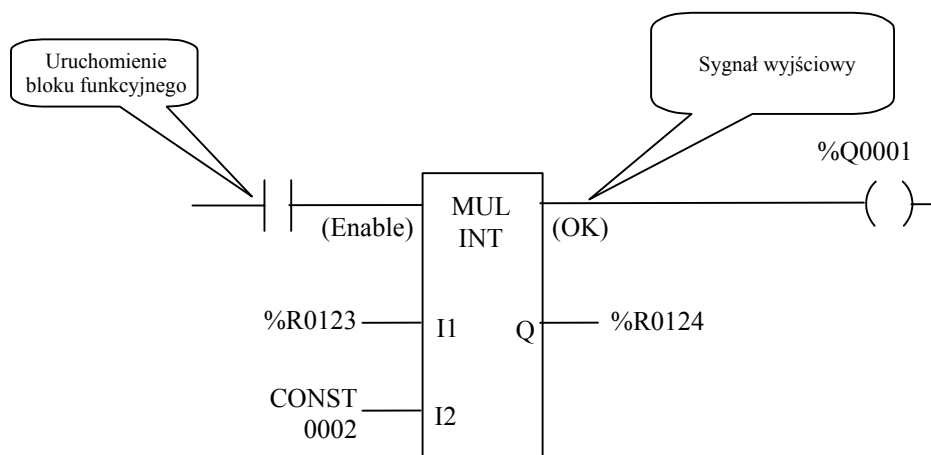


Liczniki, przekaźniki czasowe, BITSEQ i PID wymagają wprowadzenia adresu początkowego obszaru pamięci o długości 3 słów (rejestrów), w którym przechowywane są: wartość zadana, bieżąca oraz słowo kontrolne funkcji. Adres pierwszego z trzech słów jest wyświetlany na ekranie pod blokiem funkcyjnym, w miejscu oznaczonym „(Address)”.



### Przepływ sygnału przez blok funkcyjny

Sygnal dopływa do bloku z jego lewej strony u góry do wejścia Enable. Większość bloków funkcyjnych posiada wyjście sygnału nazywane “OK”. Jeśli blok funkcyjny pracuje prawidłowo, wyjście OK przepuszcza sygnał. Jeśli z wyjściem OK jest połączone inne urządzenie takie jak przekaźnik, urządzenie to zostanie załączone, tak jak na poniższym rysunku. Jednakże w większości przypadków wykorzystywanie wyjścia OK jest opcjonalne, gdyż jego głównym zadaniem jest potwierdzenie wykonania operacji.



### Uwaga

Korzystając z oprogramowania Logicmaster nie jest możliwe podłączenie bloków funkcyjnych bezpośrednio do szyny zasilającej znajdującej się po lewej stronie ekranu. Jeżeli funkcja ma być wywoływana w każdym cyklu, można skorzystać ze zmiennej %S7, ALW\_ON (zawsze włączona) o stykach otwartych połączonej z symboliczną linią zasilającą po lewej stronie ekranu.

Sygnał wyjściowy bloku funkcyjnego na schemacie umieszczony jest z prawej strony tego bloku, u samej góry. Może on być wykorzystany przez inne elementy logiczne programu lub przekaźniki (opcjonalnie). Bloki funkcyjne przesyłają sygnał wyjściowy po pomyślnym wykonaniu operacji.

## Część 3: Rozruch i wyłączenie sterownika

Rozruch sterowników serii 90-30 może przebiegać na dwa sposoby: jest to tzw. rozruch "na zimno" oraz "na gorąco". Jednostka centralna zazwyczaj przeprowadza rozruch "na zimno", jednakże w przypadku jednostek CPU 331 i wyższych, jeśli czas pomiędzy wyłączeniem zasilania a ponownym jego włączeniem jest krótszy niż 5 sekund, przeprowadzany jest rozruch "na gorąco".

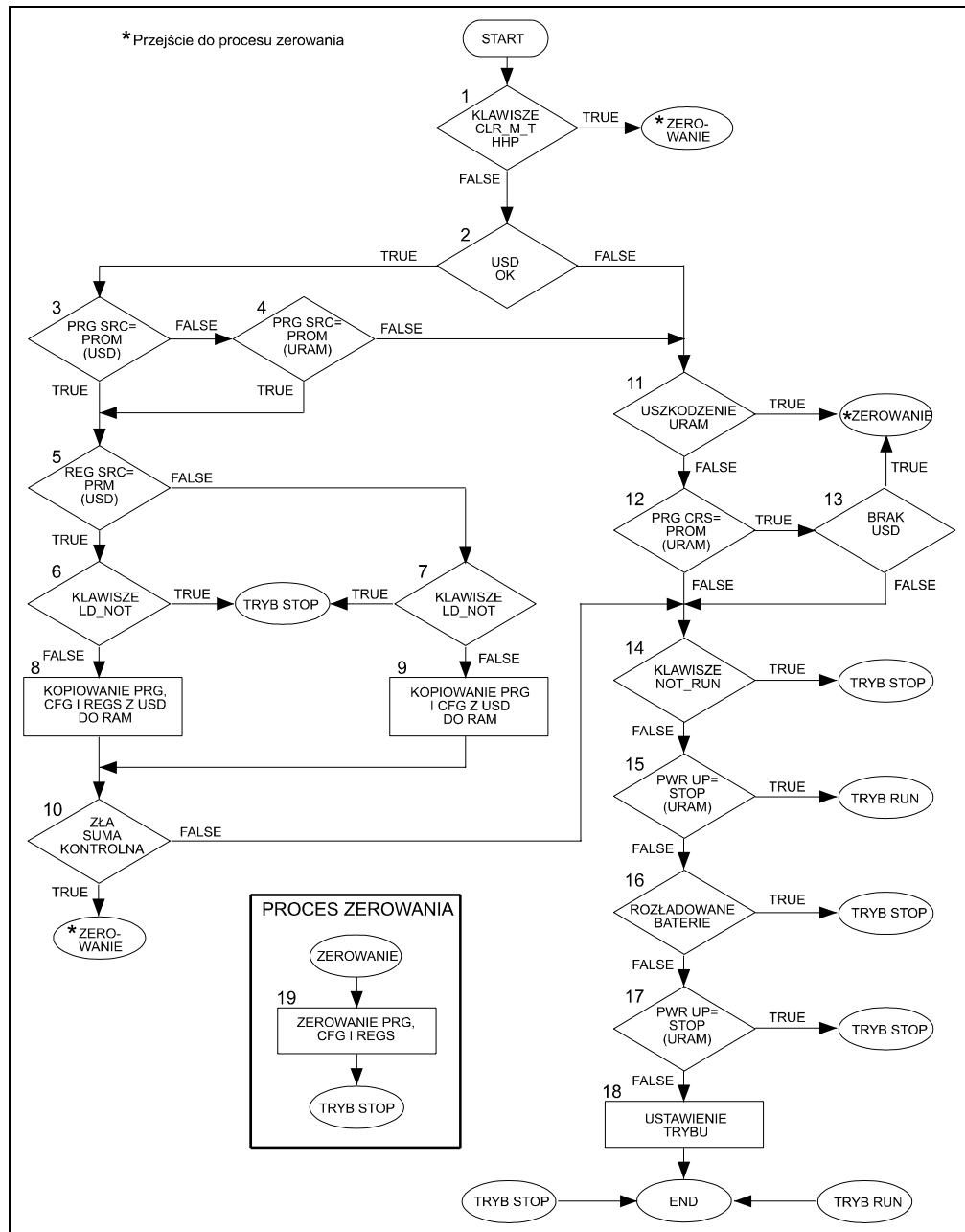
### Rozruch sterownika

Rozruch "na zimno" składa się z wyszczególnionego poniżej ciągu operacji. Rozruch "na gorąco" różni się od rozruchu "na zimno" jedynie pominięciem operacji opisanych w kroku 1.

1. Jednostka centralna przeprowadza autodiagnostykę. W kroku tym zawarta jest między innymi kontrola części pamięci RAM podtrzymywanej bateryjnie w celu stwierdzenia, czy zawiera ona poprawne dane.
2. Jeśli obecna jest pamięć EPROM lub EEPROM i konfiguracja sterownika nakazuje wykorzystanie zawartości tej pamięci, to jest ona kopiowana do pamięci RAM. Jeśli pamięć EPROM lub EEPROM nie jest obecna, zawartość pamięci RAM pozostaje bez zmian.
3. Jednostka centralna sprawdza każde gniazdo kasety w celu określenia, jakie moduły są zainstalowane.
4. Konfiguracja sprzętowa jest porównywana z konfiguracją zaprogramowaną (powinny być one takie same). Wszystkie wykryte niezgodności traktowane są jako błędy. Na przykład brak modułów lub obecność modułów innych niż zadeklarowane traktowane są jako błędy działania i zostają zakomunikowane użytkownikowi.
5. Jeśli nie zapisano konfiguracji, jednostka centralna wybiera automatycznie konfigurację standardową.
6. Jednostka centralna wybiera kanał komunikacyjny pomiędzy nią a wyspecjalizowanymi modułami dodatkowymi.
7. Ostatecznie, na podstawie konfiguracji jednostki centralnej, określany jest tryb pracy sterownika (RUN lub STOP) dla pierwszego cyklu. Na Rys. 2-5 zamieszczonym na następnej stronie przedstawiono sekwencje decyzji podejmowanych przez jednostkę centralną w celu określenia, czy ma być kopiowana zawartość pamięci PROM, oraz jaki tryb pracy sterownika ma być wybrany (**STOP** lub **RUN**).

#### Uwaga

Kroki 2 do 7 opisane powyżej nie mają zastosowania w odniesieniu do sterownika serii 90 Micro. Więcej informacji o sekwencjach rozruchu i wyłączenia dla sterownika Micro podano w Rozdziale 5 "Działanie systemu", punkt "Rozruch i wyłączenie" podręcznika *Series 90 Micro PLC User's Manual* (GFK-1065).



Rysunek 2-5. Sekwencja rozruchu

Przed wykonaniem instrukcji START zamieszczonej na schemacie blokowym rozruchu, realizowane są czynności diagnostyczne, w czasie, których testowane są różne urządzenia zewnętrzne, wykorzystywane przez jednostkę centralną oraz testowana jest pamięć RAM. Po zakończeniu czynności diagnostycznych następuje inicjalizacja wewnętrznych struktur danych i urządzeń zewnętrznych wykorzystywanych przez jednostkę centralną. Następnie jednostka centralna sprawdza, czy nie została uszkodzona pamięć RAM wykorzystywana przez użytkownika. Jeżeli stwierdzono zostało uszkodzenie pamięci RAM, następuje usunięcie programu sterującego, wyzerowanie wszystkich rejestrów użytkownika oraz wprowadzenie konfiguracji domyślnej.

## OBJAŚNIENIE SKRÓTÓW STOSOWANYCH NA SCHEMACIE

PRG = Program sterujący (PRG SRC = Źródło programu)

CFG = Konfiguracja użytkownika

REGS = Rejestry użytkownika (zmiennie %I, %Q, %M, %G, %R, %AI, i %AQ).

USD = Urządzenie zapisujące dane (pamięć) użytkownika, może to być zarówno EPROM jak i urządzenie z możliwością ponownego zapisu (EEPROM).

URAM = Pamięć podtrzymywana, zawierająca program sterujący, konfigurację i rejestry.

HHP = Programator ręczny

PU = Rozruch sterownika

CLR = Zerowanie

BATT = Bateria

### Pełna wersja skróconych napisów:

- (1) Czy w czasie rozruchu wciśnięto klawisze <CLR> i <M\_T> programatora ręcznego w celu wyczyszczenia pamięci URAM
- (2) Czy zainstalowana jest pamięć USD (user storage device) i czy zapisane w niej informacje są poprawne?
- (3) Czy parametr PRG SRC z zestawu USD ma wartość Prom, co oznacza wczytywanie programu sterującego i konfiguracji z urządzenia USD?
- (4) Czy parametr PRG SRC z zestawu USD ma wartość Prom, co oznacza wczytywanie programu sterującego i konfiguracji z urządzenia USD?
- (5) Czy parametr REG SRC z zestawu USD ma wartość Prom, co oznacza wczytywanie programu sterującego i konfiguracji z urządzenia USD?
- (6 & 7) Czy w czasie rozruchu wciśnięto klawisze <LD> i <NOT> programatora ręcznego w celu uniemożliwienia wczytywania programu sterującego i rejestrów z urządzenia USD?
- (8) Kopiowanie programu sterującego, konfiguracji i rejestrów z urządzenia USD do URAM.
- (9) Kopiowanie programu sterującego i konfiguracji z urządzenia USD do URAM.
- (10) Czy sumy kontrolne dla programu sterującego i konfiguracji, wczytane z urządzenia USD są poprawne?
- (11) Czy zawartość pamięci URAM jest poprawna? Przyczyna uszkodzenia zawartości może być włączenie bez włożonej baterii lub z baterią wyczerpaną. Zakłócenia tego typu mogą być również spowodowane przez uaktualnienie oprogramowania systemowego.
- (12) Czy parametr PRG SRC z zestawu USD ma wartość Prom, co oznacza wczytywanie programu sterującego i konfiguracji z urządzenia USD?
- (13) Czy zainstalowana jest pamięć USD? W przypadku jednostek centralnych 311-314. Dla CPU 350-364 i 374 zakłada się obecność USD.
- (14) Czy w czasie rozruchu wciśnięto na programatorze ręcznym klawisze i w celu bezwarunkowego rozpoczęcia pracy w trybie zatrzymania (STOP).
- (15) Czy parametr PWR UP z zestawu URAM ma wartość **run**?
- (16) Czy baterie są rozładowane?
- (17) Czy parametr PWR UP z zestawu URAM ma wartość **STOP**?
- (18) Ustawienie trybu RUN/STOP na tryb obowiązujący w momencie wyłączenia sterownika.
- (19) Zerowanie programu sterującego, konfiguracji i rejestrów.

### Uwaga

Pierwsza część schematu 2-6, zamieszczonego na poprzedniej stronie, nie dotyczy sterowników serii 90 Micro. Więcej informacji o sekwencjach rozruchu i wyłączenia dla sterownika Micro podano w Rozdziale 5 "Działanie systemu", punkt "Rozruch i wyłączenie" podręcznika *Series 90 Micro PLC User's Manual* (GFK-1065).

---

## Wyłączanie sterownika

Wyłączenie systemowe (samoczynne) sterownika następuje w przypadku zaniku napięcia w sieci zasilającej prądu przemiennego na okres dłuższy niż jeden jego cykl lub gdy napięcie zasilania sterownika prądem stałym (nominalnie wynoszące 5 V) spadnie poniżej 4.9 V.

## Część 4: Zegary i przekaźniki czasowe

Sterowniki serii 90-30 posiadają kilka rodzajów zegarów: zegar odmierzający czas pracy sterownika, zegar czasu rzeczywistego (modele 331, 340/341, 350-374 i 28-o punktowy sterownik Micro), zegar alarmowy oraz zegar cyklu pracy o stałym czasie trwania. Trzy bloki funkcyjne przekaźników czasowych to przekaźnik czasowy, przekaźnik czasowy z zanegowanym wejściem oraz przekaźnik czasowy z pamięcią (nazywany również przekaźnikiem kontrolnym). Dostępne są również cztery generatory sygnału prostokątnego o podstawie 0.01 s, 0.1 s, 1.0 s i 1 minuty.

### Zegar odmierzający czas pracy sterownika

Zegar ten wykorzystuje 100 ms impulsy do odmierzania czasu, który upłynął od momentu rozruchu jednostki centralnej. Zegar ten nie przechowuje odmierzonej wartości po wyłączeniu zasilania, zostaje on wyzerowany przy każdym ponownym rozruchu. Co sekundę zegar komunikuje się z jednostką centralną w celu zapisania odmierzonej wartości (w sekundach). Odmierzanie sekund może trwać do około 100 lat po włączeniu zegara.

Ponieważ zegar odmierzający czas pracy sterownika jest podstawą operacji programowych oraz bloków funkcyjnych przekaźników czasowych, nie może on zostać wyzerowany z poziomu programu sterującego ani z programatora. Jednakże program sterujący może odczytać aktualną wartość czasu odmierzonego przez zegar za pomocą funkcji SVCREQ #16.

### Zegar czasu rzeczywistego

Ten zegar sprzętowy posiadają jednostki centralne CPU 331 i wyższe oraz 28 punktowe Micro. Posiada on następujących siedem funkcji:

- Rok (dwie cyfry)
- Miesiąc
- Dzień miesiąca
- Godzina
- Minuta
- Sekunda
- Dzień tygodnia

Zegar odmierzający aktualną datę i czas (TOD) zasilany jest bateryjnie i pracuje w przypadku awarii lub wyłączenia zasilania sterownika. Aby jednak skorzystać z odmierzanych przezeń wartości, należy dokonać jego inicjalizacji. Przy użyciu funkcji SVCREQ7 można w sposób programowy odczytywać i ustawiać wskazania tego zegara.

Wskazania zegara TOD mogą być odczytywane i ustawiane za pomocą programatora ręcznego (HHP). Jednakże w jednostkach centralnych 350-364 z oprogramowaniem systemowym 8.00 z ustawionym parametrem *Mem. Protect* oraz ustawionym przełącznikiem w pozycji ON nie jest możliwa zmiana ustawień zegara TOD. Należy zwrócić uwagę, że tylko w jednostki centralne 350-374 posiadają zabezpieczenie przełącznikiem kluczowym.

Zegar podtrzymujący aktualną datę i czas jest zaprojektowany zgodnie z kalendarzem. Uwzględniane są automatycznie lata przestępne, do roku 2079.

## Zegar alarmowy

Zegar alarmowy sterowników serii 90-30 przeznaczony jest do wykrywania błędów krytycznych, które powodują znaczną wydłużenie czasu trwania cyklu pracy sterownika. Zadana wartość porównawcza dla zegara alarmowego wynosi 200 ms (500 milisekund dla jednostek centralnych serii 350 i 374), i nie może zostać zmieniona. Zegar ten zostaje wyzerowany na początku każdego cyklu pracy.

W sterownikach 90-30, w jednostkach centralnych serii 331 i niższych, po przekroczeniu wartości porównawczej, dioda LED oznaczona "OK" gaśnie, jednostka centralna zostaje zatrzymana; wyjścia przechodzą w stan standardowy, przyjmowany samoczynnie. Wszystkie mikroprocesory modułów zostają zatrzymane i ustaje wszelka komunikacja. Aby uruchomić sterownik ponownie, należy wyłączyć i włączyć zasilanie kasety sterownika zawierającej jednostkę centralną. W jednostkach centralnych sterowników serii 90 Micro, 90-20 i 90-30 modele 340 i wyższe, zegar alarmowy powoduje wyzerowanie jednostki centralnej, wykonanie programu logicznego dla rozruchu, wygenerowanie błędu o uszkodzeniu zegara alarmowego i przejście do trybu zatrzymania **STOP**.

## Zegar czasu trwania przerwy w zasilaniu sterownika

Zegar czasu trwania przerwy w zasilaniu sterownika podaje czas trwania braku zasilania sterownika. Po wyłączeniu zasilania, zegar ten jest zerowany, po czym rozpoczyna zliczanie. Po włączeniu zasilania, zliczanie czasu jest wstrzymywane, a naliczona wartość zostaje zapamiętana. Wskazania tego zegara można odczytać za pomocą bloku funkcyjnego SVC\_REQ #29, opisanego w rozdziale 12.

### Uwaga

Funkcja ta dostępna jest wyłącznie w jednostkach centralnych 331 i wyższych sterowników serii 90-30.

## Zegar cyklu pracy o stałym czasie trwania

Zegar ten kontroluje czas trwania cyklu, jeżeli sterownik serii 90-30 pracuje w trybie ze stałą długością cyklu (**CONSTANT SWEEP TIME**). W trybie tym, czas realizacji każdego cyklu jest taki sam. Stały czas trwania cyklu jest ustawiany za pomocą programatora i może wynosić od 5 do wartości porównawczej zegara alarmowego. Czas trwania tego cyklu wynosi 100 ms. Zwykle, w przypadku większości programów sterujących, fazy odczytywania wejść, wykonywania części logicznej programu sterującego oraz ustawiania wyjść nie wymagają takiej samej ilości czasu w każdym z cykli.

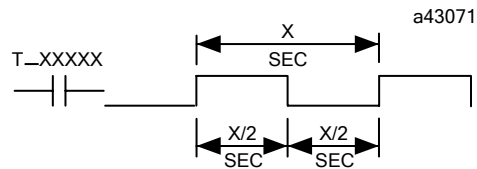
Jeśli cykl nie zakończy się przed upływem zadanego czasu trwania, a czas trwania poprzedniego cyklu nie został przekroczony, w tabeli błędów działania sterownika zostaje umieszczony komunikat alarmowy. Na początku kolejnego cyklu wartość zmiennej systemowej OV\_SWP zostaje ustawiona na 1. Wartość tej zmiennej zostaje ponownie ustawiona na 0, gdy sterownik zostanie przełączony z trybu pracy ze stałym czasem trwania cyklu (**CONSTANT SWEEP TIME**) na inny lub gdy poprzedni cykl nie przekroczy zadanego limitu czasu.



## Generator sygnału prostokątnego

Sterowniki serii 90 posiadają cztery styki generatora sygnału prostokątnego, o podstawach czasu 0.01 s, 0.1 s, 1 s oraz 1 min. Stan tych styków zmienia się podczas inicjalizacji sterownika. Poprzez styki te wysyłany jest sygnał o takim samym czasie włączenia jak i wyłączenia. Stykom tym przypisane są zmienne systemowe (odpowiednio T\_10MS, T\_100MS, T\_SEC oraz T\_MIN).

Działanie generatora objaśnia zamieszczony poniżej rysunek.



Rysunek 2-6. Schemat pracy generatora sygnału prostokątnego

## Część 5: Mechanizmy zabezpieczające

Mechanizmy zabezpieczające w sterownikach serii 90-30, 90-20 i Micro, mają na celu uniemożliwienie dokonania zmian w sterowniku osobom nieupoważnionym. W sterowniku dostępne są cztery poziomy uprawnień. Na pierwszym poziomie, który jest zawsze dostępny, można odczytywać dane ze sterownika, nie można wprowadzać żadnych zmian w aplikacji. Pozostałe trzy poziomy zabezpieczane są hasłem.

Wraz ze wzrostem poziomu wzrasta zakres zmian, które mogą być wprowadzane. Uprawnienia dostępne na danym poziomie są sumą uprawnień dla tego poziomu, plus uprawnień dostępnych na poziomach niższych. Dostępne są następujące poziomy uprawnień:

Poziom uprawnień	Opis
Poziom 1	Możliwość odczytywania wszystkich danych, za wyjątkiem haseł. Można więc odczytywać pamięć (%I, %Q, %AQ, %R, itp.), tabele błędów oraz wszystkie typy bloków programu (dane, wartości i stałe). Nie jest dostępna możliwość wprowadzania zmian w sterowniku.
Poziom 2	Poziom ten pozwala na zapisywanie danych do pamięci (%I, %R, itp.).
Poziom 3	Poziom ten pozwala na zapisywanie zmian w programie sterującym wyłącznie w trybie zatrzymania <b>STOP</b> .
Poziom 4	Jest to poziom przyjmowany domyślnie dla systemów, w których nie ustawiono hasła. Domyślnym poziomem dla systemu z ustawionymi hasłami jest najwyższy poziom bez ustawionego hasła. Poziom ten, najwyższy, pozwala na odczytywanie i zapisywanie danych do wszystkich pamięci, włączając w to hasła, zarówno w trybie pracy <b>RUN</b> jak i w trybie zatrzymania <b>STOP</b> . (Nie można zmieniać konfiguracji w trybie <b>RUN</b> ).

## Hasła

Każdy poziom uprawnień sterownika posiada swoje hasło. (Nie można ustawić hasła dla 1 poziomu uprawnień). Dla każdego poziomu może być wprowadzone inne hasło, można też wprowadzić takie samo hasło dla więcej niż jednego poziomu. Hasła mogą mieć długość od jednego do czterech znaków ASCII, można je wprowadzać i zmieniać za pomocą oprogramowania, lub za pomocą programatora ręcznego.

Poziom uprawnień obowiązuje tak długo, jak długo trwa komunikacja pomiędzy sterownikiem a programatorem. Nie jest konieczne wykonywanie jakichkolwiek działań, nie mniej jednak komunikacja nie może zostać przerwana. W przypadku braku komunikacji przez 15 minut, poziom uprawnień zostaje zmieniony na najwyższy, nie zabezpieczony poziom.

Po podłączeniu do sterownika, oprogramowania żąda podania statusu zabezpieczenia każdego z poziomów uprawnień. Następnie, oprogramowanie wysyła do sterownika żądanie przejścia do najwyższego, niezabezpieczonego poziomu, nie określając tym samym, który poziom to jest. Jeżeli do sterownika podłączony jest programator ręczny, sterownik przełącza się do najwyższego, niezabezpieczonego poziomu dostępu.

## Zmiana poziomu dostępu

W celu zmiany poziomu dostępu należy wprowadzić za pomocą programatora nowy poziom dostępu oraz hasło dla tego poziomu. Żądanie przejścia na inny poziom dostępu jest odrzucane, jeżeli wprowadzone hasło nie jest zgodne z hasłem zapisanym dla tego poziomu w tabeli haseł dostępu. W sytuacji takiej poziom dostępu nie jest zmieniany, nie są wprowadzane żadne zmiany. Jeżeli podjęta zostanie próba wprowadzenia zmian w sterowniku za pomocą programatora ręcznego, bez posiadania uprawnień, programator wyświetli komunikat o błędzie, informujący o braku dostępu.

## Blokowanie/odblokowywanie podprogramów

Za pomocą funkcji udostępnianych przez oprogramowanie, można blokować i odblokowywać podprogramy. Dostępne są dwa rodzaje blokad:

Typ blokady	Opis
Blokada przeglądania	Po zablokowaniu, nie można wyświetlić zawartości podprogramu.
Blokada edycji	Po zablokowaniu podprogram może być wyświetlony, ale nie można go edytować.

Zablokowanie podglądu i edycji podprogramów można usunąć z poziomu edytora deklaracji bloków programu, o ile nie jest to blokada podglądu i edycji wprowadzona na stałe.

W podprogramach z zablokowanym podglądem można korzystać z funkcji edycyjnych do szukania i zastępowania. Jeżeli szukany tekst zostanie odnaleziony w podprogramie z zablokowanym podglądem, zamiast programu sterującego wyświetlony zostanie następujący komunikat:

```
Found in locked block <block_name> (Continue/Quit)
(Znalezione miejsca wystąpienia w zablokowanym bloku <nazwa_bloku>
(kontynuacja/koniec))
```

lub

```
Cannot write to locked block <block_name> (Continue/Quit)
(Nie można zapisywać do zablokowanego bloku <nazwa_bloku> (kontynuacja/koniec))
```

Operacja szukania można kontynuować (Continue) lub zakończyć (Quit).

Można usuwać zawartość i same foldery z zablokowanymi podprogramami. Jeżeli folder zawiera zablokowane podprogramy, bloki te pozostają zablokowane w czasie wywoływania funkcji oprogramowania Copy, Backup i Restore.

## Trwała blokada podprogramów

Poza blokadą podglądu (VIEW LOCK) i edycji (EDIT LOCK) dostępne są dwa rodzaje blokad stałych. Po założeniu stałej blokady podglądu (PERMANENT VIEW LOCK), wszelkie próby podglądu podprogramu są odrzucane. Po założeniu stałej blokady edycji (PERMANENT EDIT LOCK), wszelkie próby edycji podprogramu są odrzucane.

### Ostrzeżenie

**Po założeniu blokad stałych, w odróżnieniu od normalnych blokad podglądu i edycji, nie można ich już usunąć.**

Po założeniu stałej blokady edycji (PERMANENT EDIT LOCK), można ją jedynie zmienić na stałą blokadę podglądu (PERMANENT VIEW LOCK). Jeżeli została założona blokada PERMANENT VIEW LOCK, nie można jej zmienić na żadną, innego typu blokadę.

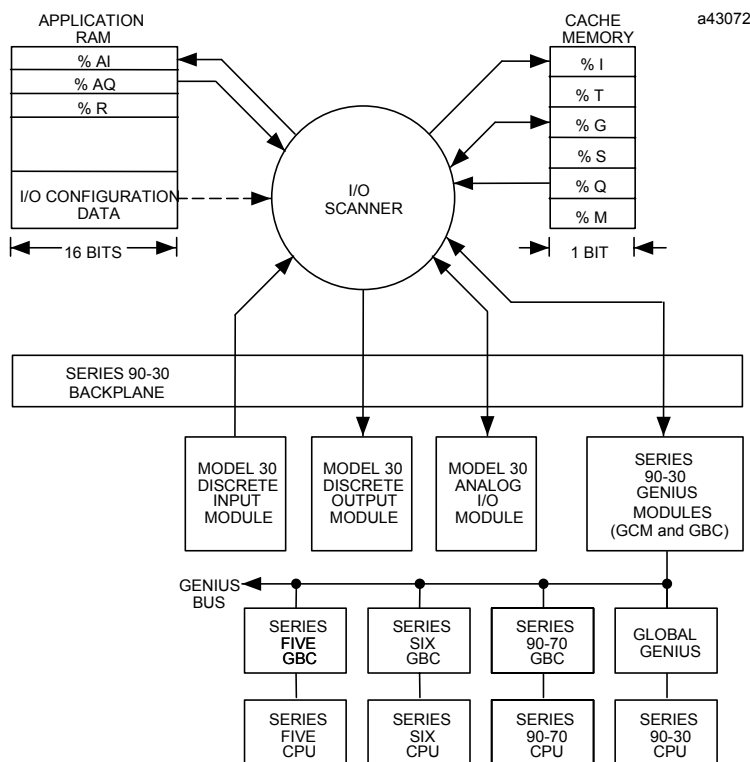
## Część 6: Moduły wejść/wyjść sterowników serii 90-3090-30, 90-20 i Micro

Moduły wejść i wyjść sterownika zapewniają łączność z urządzeniami i osprzętem użytkownika. W sterownikach serii 90-30 moduły wejść/wyjść są bezpośrednio wpinane w gniazda kasety bazowej. Ilość obsługiwanych modułów jest zależna od modelu jednostki centralnej:

- Jednostki centralne 350-374 obsługują do 79 modułów wejść/wyjść. Te jednostki centralne obsługują do 8 kaset bazowych (7 kaset rozszerzających plus 1 kasetę z jednostką centralną).
- Jednostki centralne 331, 340 i 341 obsługują do 49 modułów wejść/wyjść. Te jednostki centralne obsługują do 5 kaset bazowych (4 kasety rozszerzające plus 1 kasetę z jednostką centralną).
- Jednostki centralne 311 i 313 (kasetę 5 gniazdową) obsługują do 5 modułów wejść/wyjść. Jednostka centralna 323 (kasetę 10 gniazdową) obsługuje do 10 modułów wejść/wyjść. Jednostki te nie obsługują kaset rozszerzających.

Na zamieszczonym poniżej rysunku pokazano strukturę wejść i wyjść dla sterowników serii 90-30.

### Moduły wejść/wyjść sterownika



Rysunek 2-7. Struktura wejść/wyjść sterowników serii 90-30

### Uwaga

Na zamieszczonym powyżej rysunku pokazano typową dla sterowników 90-30 strukturę systemu wejść i wyjść. Moduły wyspecjalizowane i dodatkowe nie są obsługiwane w czasie skanowania wejść/wyjść; korzystają one z komunikacji systemowej. Dodatkowe informacje na temat modułów wejść/wyjść sterowników serii 90-20 podano w podręczniku *Series 90™-20 Programmable Controller User's Manual* (GFK-0551). Więcej informacji na temat struktury wejść/wyjść sterowników Micro podano w podręczniku *Series 90™ Micro PLC User's Manual* (GFK-1065).

## Moduł wejść/wyjść sterowników serii 90-30

Dostępnych jest następujących pięć rodzajów modułów wejść i wyjść serii 90-30: moduły wejść dyskretnych, wyjść dyskretnych, wejść analogowych, wyjść analogowych oraz specjalizowane moduły dodatkowe. W zamieszczonej poniżej tabeli zestawiono moduły wejść/wyjść serii 90-30, numery katalogowe, liczbę punktów wejść/wyjść oraz krótki opis każdego z modułów.

### Uwaga

Informacje na temat dostępnych modułów można uzyskać u najbliższego dystrybutora firmy GE Fanuc. W kolumnie "publikacja" zamieszczono numer publikacji, w której można znaleźć opis techniczny, sposób podłączania modułów wejść/wyjść.

Rysunek 2-8. Moduły wejść/wyjść serii 90-30

Numer katalogowy	Liczba punktów	Opis	Publikacja
<i>Moduły wejść dyskretnych</i>			
IC693MDL230	8	120 V VAC, izolowane	GFK-0898
IC693MDL231	8	240 V VAC, izolowane	GFK-0898
IC693MDL240	16	120 V VAC	GFK-0898
IC693MDL241	16	4 V VAC/ DC/ logika dodatnia/ujemna	GFK-0898
IC693MDL630	8	24 V DC, logika dodatnia	GFK-0898
IC693MDL632	8	125 V DC, logika dodatnia/ujemna	GFK-0898
IC693MDL633	8	24 V DC, logika ujemna	GFK-0898
IC693MDL634	8	24 V DC, logika dodatnia/ujemna	GFK-0898
IC693MDL640	16	24 V DC, logika dodatnia	GFK-0898
IC693MDL641	16	24 V DC, logika ujemna	GFK-0898
IC693MDL643	16	24 V DC, logika dodatnia, FAST	GFK-0898
IC693MDL644	16	24 V DC, logika ujemna, FAST	GFK-0898
IC693MDL645	16	24 V DC, logika dodatnia/ujemna	GFK-0898
IC693MDL646	16	24 V DC, logika dodatnia/ujemna, FAST	GFK-0898
IC693MDL652	32	24 V DC, logika dodatnia/ujemna	GFK-0898
IC693MDL653	32	24 V DC, logika dodatnia/ujemna, FAST	GFK-0898
IC693MDL654	32	5/12 V DC (TTL), logika dodatnia/ujemna	GFK-0898
IC693MDL655	32	24 V DC, logika dodatnia/ujemna	GFK-0898
IC693ACC300	8/16	Symulator wejść	GFK-0898

Tabela 2-8. Moduły wejść/wyjść serii 90-30 - kontynuacja

Numer katalogowy	Liczba punktów	Opis	Publikacja
<i>Moduły wyjść dyskretnych</i>			
IC693MDL310	12	120 V AC, 0.5 A	GFK-0898
IC693MDL330	8	120/240 V AC, 2A	GFK-0898
IC693MDL340	16	120 V AC, 0.5 A	GFK-0898
IC693MDL390	5	120/240 V AC, Izolowane, 2A	GFK-0898
IC693MDL730	8	12/24 V DC, logika dodatnia, 2A	GFK-0898
IC693MDL731	8	12/24 V DC, logika ujemna, 2A	GFK-0898
IC693MDL732	8	12/24 V DC, logika dodatnia, 0.5 A	GFK-0898
IC693MDL733	8	12/24 V DC, logika ujemna, 0,5A	GFK-0898
IC693MDL734	6	125 V DC, logika ujemna, 2 A	GFK-0898
IC693MDL740	16	12/24 V DC, logika dodatnia, 0.5 A	GFK-0898
IC693MDL741	16	12/24 V DC, logika ujemna, 0,5A	GFK-0898
IC693MDL742	16	12/24 V DC, logika dodatnia, 1 A	GFK-0898
IC693MDL750	32	12/24 V DC, logika ujemna	GFK-0898
IC693MDL751	32	12/24 V DC, logika dodatnia, 0,3A	GFK-0898
IC693MDL752	32	5/24 V DC (TTL), logika ujemna, 0,5 A	GFK-0898
IC693MDL753	32	12/24 V DC, logika dodatnia/ujemna, 0,5 A	GFK-0898
IC693MDL760	16	11 wyjść pneumatycznych, 5 wyjść 0,5A logika dodatnia	GFK-1881
IC693MDL930	8	Przełącznik, normalnie otwarty, 4 A, Izolowany	GFK-0898
IC693MDL931	8	Przełącznik, BC, Izolowany	GFK-0898
IC693MDL940	16	Przełącznik, normalnie otwarty, 2 A	GFK-0898
<i>Moduły Wejść/Wyjść</i>			
IC693MDR390	8/8	Wejście 24 V DC, Przełącznik wyjściowy	GFK-0898
IC693MAR590	8/8	Wejście 120 V AC, Przełącznik wyjściowy	GFK-0898
<i>Moduły analogowe</i>			
IC693ALG220	4 kanał.	Wejście analogowe, Napięcie	GFK-0898
IC693ALG220	4 kanał.	Wejście analogowe, Prądowe	GFK-0898
IC693ALG222	16	Wejście analogowe, Napięcie	GFK-0898
IC693ALG223	16	Wejście analogowe, Prądowe	GFK-0898
IC693ALG390	2 kanał.	Wyjście analogowe, Napięciowe	GFK-0898
IC693ALG391	2 kanał.	Wyjście analogowe, Prądowe	GFK-0898
IC693ALG392	8 kanał.	Wyjście analogowe, Prądowo/Napięciowe	GFK-0898
IC693ALG442	4/2	Wejście/wyjście analogowe, prądowo-napięciowe,	GFK-0898

Tabela 2-8. Moduły wejść/wyjść serii 90-30 - kontynuacja

Numer katalogowy	Opis	Pub Number
	<i>Moduły dodatkowe</i>	
IC693APU300	Licznik impulsów o wysokiej częstotliwości	GFK-0293
IC693APU301	Moduł pozycjonowania 1 osiowy, nadążny	GFK-0781
IC693APU301	Moduł pozycjonowania 1 osiowy, standardowy	GFK-0840
IC693APU302	Moduł pozycjonowania 2 osiowy, nadążny	GFK-0781
IC693APU302	Moduł pozycjonowania 2 osiowy, standardowy	GFK-0840
IC693MCS001/002*	System sterowania ruchem (1 i 2 osie)	GFK-1256
IC693DSM302	Moduł kontrolera ruchu	GFK-1464
IC693DSM314	Moduł kontrolera ruchu	GFK-1742
IC693APU305	Moduł procesora wejść/wyjść	GFK-1028
IC693CMM321	Moduł sieci Ethernet	GFK-1541
IC693ADC311	Koprocessor wyświetlacza alfanumerycznego	GFK-0521
IC693BEM331	Kontroler sieci Genius	GFK-1034
IC693BEM320	Moduł interfejsu z systemem CNC (typ slave)	GFK-0631
IC693BEM321	Moduł interfejsu z systemem CNC (typ master)	GFK-0823
IC693CMM311	Moduł koprocessora komunikacyjnego	GFK-0582
IC693CMM301	Moduł komunikacyjny Genius	GFK-0412
IC693CMM302	Moduł komunikacyjny Enhanced Genius	GFK-0695
IC693PBM200	Moduł sieci Profibus - Master	GFK-2121
IC693PBS201	Moduł sieci Profibus - Slave	GFK-2193
IC693PCM300	Moduł programowalnego koprocessora PCM, 160 KB (35 kB dla programu w języku MegaBasic)	GFK-0255
IC693PCM301	Moduł programowalnego koprocessora PCM, 192 KB (47 kB dla programu w języku MegaBasic)	GFK-0255
IC693PCM311	Moduł programowalnego koprocessora PCM, 640 KB (190 kB dla programu w języku MegaBasic)	GFK-0255
IC693PTM100/101	Moduł do kontrolowania sieci energetycznych	GFK-1734
IC693TCM302/303	Specjalizowany moduł do stosowania we wtryskarkach, ośmio kanałowy	GFK-1466

\* Przystarzały. Umieszczony tylko jako odniesienie.

## Formaty danych wejściowych/wyjściowych

Stan wejść i wyjść dyskretnych jest przechowywany w postaci bitów w pamięci podręcznej (cache) sterownika. Dane dotyczące wejść i wyjść analogowych są zapamiętywane jako słowa bitowe w części pamięci RAM przeznaczonej do tego celu.

## Domyślne wartości dla modułów wyjściowych serii 90-30

Przy rozruchu sterownika moduły wyjść dyskretnych sterowników 90-30 ustawiają samoczynnie wartości wyjść na 0. Pozostają one w tym stanie do momentu obsługi wyjść w pierwszym cyklu pracy sterownika. Wyjścia analogowe można skonfigurować za pomocą przełącznika umieszczonego w module tak, aby ustawione zostały one przy rozruchu na 0 lub zatrzymywały ostatnią wartość. Moduły wyjść analogowych mogą być zasilane z zewnętrznego źródła zasilania, co pozwala na pamiętanie wartości przy wyłączonym zasilaniu sterownika.

## Dane diagnostyczne

W komórkach pamięci adresowanych przez określone zmienne systemowe (patrz tabela zmiennych systemowych) znajdują się informacje dotyczące ewentualnych błędów działania modułów wejść i wyjść sterownika lub niezgodności konfiguracji ze stanem faktycznym. Nie są dostępne informacje diagnostyczne odnośnie poszczególnych punktów wejść lub wyjść. Więcej informacji na temat obsługi błędów podano w Rozdziale 3, "Błędy działania sterownika - opis i ich usuwanie."

## Dane globalne

### Dane globalne Genius

Sterowniki serii 90-30 pozwalają na komunikowanie się wielu jednostek centralnych, poprzez moduł komunikacyjny GENIUS i dyskretne zmienne globalne (%G). Sterownik szyny Genius, IC693BEM331 w jednostkach centralnych CPU, wersja 5 lub późniejsza oraz moduł komunikacyjny Genius, IC693CMM302, umożliwiają przesyłanie do 128 bajtów danych do innych sterowników lub komputerów. Mogą one również odbierać dane o wielkości do 128 bajtów z/do 30 innych sterowników Genius, połączonych w sieć. Dane mogą być przesyłane i odbierane nie tylko jako dyskretne zmienne globalne %G, ale również jako zmienne innego typu.

Oryginalny moduł komunikacyjny Genius, IC693CMM301, posiada ograniczenia do stałych adresów globalnych %G i pozwala tylko na wymianę do 32 bitów poprzez magistralę szeregową od SBA 16 do 23. Nie jest zalecane stosowanie tego modułu ponieważ moduł komunikacyjny Enhanced Genius posiada znacznie zwiększone możliwości.

Dane globalne mogą być wymieniane pomiędzy sterownikami Five, Six i serii 90, podłączonych do tej samej szyny wejść/wyjść Genius.

### Sieć Ethernet

W jednostkach centralnych CPU364 (wersja 9.0 i nowsze) można przysyłać dane siecią Ethernet za pomocą jednego z dwóch, wbudowanych portów Ethernet (ale nie przez obydwa jednocześnie). Jednostki te wyposażone są w port AAUI i port 10BaseT. Jednostka centralna 374 obsługuje połączenia z siecią Ethernet za pośrednictwem dwóch portów 10BaseT/100BaseTx z automatycznym trybem full duplex.

Jednostki centralne CPU364 i CPU374 obsługują dane globalne Ethernet (EGD – Ethernet Global Data), które są podobne do danych globalnych Genius, pozwalają jednemu urządzeniu (producent) przysyłać dane do jednego lub więcej urządzeń sieciowych (konsumentów). Dane globalne Ethernet nie są obsługiwane przez oprogramowanie Logicmaster 90 (wymagany jest pakiet do programowania sterowników serii 90, pracujący w środowisku Windows).



## Moduł wejść/wyjść sterowników serii 90-20

Poniżej przedstawiono moduły wejść/wyjść oferowane dla sterowników 90-20. Dla każdego modułu podano numer katalogowy, liczbę punktów wejść/wyjść, oraz krótki opis. Moduł wejść/wyjść jest umieszczony na płycie montażowej wspólnie z zasilaczem. Parametry techniczne i sposób podłączania każdego z modułów omówiono w rozdziale 5 podręcznika *Series 90-20 Programmable Controller User's Manual*, GFK-0551.

Numer katalogowy	Opis	Liczba punktów wejść/wyjść
IC692MAA541	Moduł wejść/wyjść i zasilacza Wejście 120 VAC/ Wyjście 120 VAC/ Zasilacz 120 VAC	16 wejść/12 wyjść
IC692MDR541	Moduł wejść/wyjść i zasilacza Wejście 24 VAC/ Wyjście przekaźnikowe/ Zasilacz 120 VAC	16 wejść/12 wyjść
IC692MDR741	Moduł wejść/wyjść i zasilacza Wejście 24 VDC/ Wyjście przekaźnikowe/ Zasilacz 240VAC	16 wejść/12 wyjść
IC692CPU211	Moduł jednostki centralnej CPU, model CPU 211	Nie dotyczy

## Konfigurowanie i programowanie

Konfigurowanie jest procesem polegającym na przypisywaniu adresów logicznych oraz innych parametrów do modułów sprzętowych systemu. Proces ten można zrealizować zarówno przed jak i po programowaniu, za pomocą oprogramowania konfiguracyjnego lub programatora ręcznego, nie mniej jednak zalecane jest przeprowadzenie konfigurowania przed rozpoczęciem programowania. W podręczniku użytkownika opisano proces tworzenia, drukowania oraz przesyłania programu do sterownika. W rozdziałach 4 do 12 opisano instrukcje programowania, które można wykorzystać do budowy programu sterującego dla sterowników serii 90-30 i 90-20.



# Rozdział 3

## Błędy działania sterownika - opis i ich usuwania

W rozdziale tym zawarto informacje odnośnie sygnalizowanych błędów działania sterowników 90-30, 90-20 i Micro. Podano opis dla błędów, zapisywanych w tabeli błędów działania sterownika oraz kategorie dla błędów, zapisywanych w tabeli błędów układów wejść i wyjść.

Przy omawianiu błędów w niniejszym rozdziale, dla każdego z nich podano opis zamieszczony w tabeli błędów sterownika oraz kategorię błędu w tabeli błędów działania sterownika. W celu uzyskania dodatkowych informacji o błędach należy odszukać w niniejszym podręczniku odpowiedni opis lub kategorię, wyświetlane do ekranie programatora. Dla każdego błędu podawana jest również przyczyna i instrukcje odnośnie usuwania.

Rozdział 3 składa się z następujących rozdziałów:

<b>Część</b>	<b>Tytuł</b>	<b>Opis</b>	<b>Strona</b>
1	Obsługa błędów	Opis typu błędów działania, które mogą wystąpić w sterownikach 90-30 oraz sposób ich wyświetlania w tabeli błędów. Zamieszczono również opis tabeli błędów wejść/wyjść i sterownika.	3-2
2	Tabela błędów działania sterownika	Dostarcza informacje o błędach sterownika i sposób ich usuwania.	3-7
3	Tabela błędów działania układów wejść/wyjść	Opisuje następujące kategorie błędów: brak zadeklarowanego modułu i wykrycie nie zadeklarowanego modułu.	3-16

## Część 1: Obsługa błędów

### Uwaga

Informacje o obsłudze błędów dotyczą systemów programowanych za pomocą oprogramowania Logicmaster 90-30/Micro.

Wystąpienie pewnych błędów lub okoliczności, mających wpływ na pracę systemu, sygnalizowane jest w sterownikach serii 90-30, 90-20 i Micro w postaci błędów. Błędy te, jak na przykład brak modułu wejść/wyjść lub kasety, mogą uniemożliwiać sterowanie maszyną lub procesem. Można też wyróżnić błędy mające na celu wyłącznie zwrócenie uwagi użytkownika, na przykład informacja o niskim stanie naładowania baterii podtrzymujących zawartość pamięci i związanej z tym konieczności jej wymiany. Jednak, niektóre stany zapisywane w tablicy błędów nie mają związku z awariami. Na przykład w przypadku zainstalowania nowego modułu taki stan zostanie przedstawiony w tablicy błędów jako "Addition of I/O Module" (nie zadeklarowany moduł).

## Procesor Alarmowy

**Błąd** jest pewnym warunkiem lub awarią samą w sobie. Jeżeli informacja o uszkodzeniu zostanie odebrana i przetworzona przez jednostkę centralną, uszkodzenie takie nazywane jest **alarmem**. Oprogramowanie systemowe jednostki centralnej zajmujące się obsługą błędów jest nazywane procesorem alarmowym. Użytkownik komunikuje się z procesorem alarmowym za pomocą oprogramowania. Błędy działania sterownika i układów wejść/wyjść są zapisywane w odpowiednich tabelach oprogramowania.

## Klasy błędów

Sterowniki 90-30, 90-20 i Micro są wykrywają różnego typu błędy. Są to usterki wewnętrzne, usterki zewnętrznych modułów wejść/wyjść oraz błędy operacyjne.

Rodzaj błędu	Przykłady
Usterki wewnętrzne	Nie zgłaszające się moduły Słabe baterie podtrzymujące pamięć Błędna suma kontrolna pamięci sterownika.
Usterki zewnętrznych modułów wejść/wyjść	Brak zgłoszonej kasety lub modułu. Dołączenie zgłoszonej kasety lub modułu.
Błędy operacyjne	Błędy komunikacji. Niegodność konfiguracji. Próby użycia błędnego hasła dostępu do sterownika.

### Note

Więcej informacji na temat obsługi błędów w sterownikach Micro podano w podręczniku *Series 90 Micro PLC User's Manual* (GFK-1065).

## Reakcja systemu na błędy działania

Zazwyczaj błędy sprzętowe wymagają zatrzymania sterownika lub też błąd jest tolerowany. Usterki zewnętrznych modułów wejść/wyjść mogą być tolerowane przez sterownik, lecz mogą stanowić zagrożenie dla sterowanego procesu. Błędy operacyjne są normalnie tolerowane. Błędy działania sterowników serii 90-30, 90-20 i Micro posiadają dwa atrybuty:

Atrybut	Opis
Wykorzystywana tabela błędów	Tabela działania wejść/wyjść tabela działania sterownika
Waga błędu	Krytyczny Diagnostyczny Informacyjny

### Tabele błędów działania

Błędy związane z układami wejść/wyjść zapisywane są w tabeli błędów działania układów wejść/wyjść, pozostałe błędy zapisywane są w tabeli błędów działania sterownika. W zamieszczonej poniżej tabeli podano grupy błędów, wagi, tabele błędów oraz nazwy dyskretnych zmiennych systemowych %S, których wartość jest zmieniana w przypadku wystąpienia błędów.

**Tabela 3-1 Zestawienie błędów**

Grupa błędów	Waga błędu	Tabele błędów	Zmienne systemowe			
			io_flt	any_flt	io_pres	los_iom
Loss of or Missing I/O Module	Diagnostyczny	Wejścia/Wyjścia	io_flt	any_flt	io_pres	los_iom
Loss of or Missing Option Module	Diagnostyczny	Sterownika	sy_flt	any_flt	sy_pres	los_sio
System Configuration Mismatch	Krytyczny	Sterownika	sy_flt	any_flt	sy_pres	cfg_mm
PLC CPU Hardware Failure	Krytyczny	Sterownika	sy_flt	any_flt	sy_pres	hrd_cpu
Program Checksum Failure	Krytyczny	Sterownika	sy_flt	any_flt	sy_pres	pb_sum
Low Battery	Diagnostyczny	Sterownika	sy_flt	any_flt	sy_pres	low_bat
PLC Fault Table Full	Diagnostyczny	—	sy_full			
I/O Fault Table Full	Diagnostyczny	—	io_full			
Application Fault	Diagnostyczny	Sterownika	sy_flt	any_flt	sy_pres	apl_flt
No User Program	Informacyjny	Sterownika	sy_flt	any_flt	sy_pres	no_prog
Corrupted User RAM	Krytyczny	Sterownika	sy_flt	any_flt	sy_pres	bad_ram
Password Access Failure	Diagnostyczny	Sterownika	sy_flt	any_flt	sy_pres	bad_pwd
PLC Software Failure	Krytyczny	Sterownika	sy_flt	any_flt	sy_pres	sft_cpu
PLC Store Failure	Krytyczny	Sterownika	sy_flt	any_flt	sy_pres	stor_er
Constant Sweep Time Exceeded	Diagnostyczny	Sterownika	sy_flt	any_flt	sy_pres	ov_swp
Unknown PLC Fault	Krytyczny	Sterownika	sy_flt	any_flt	sy_pres	
Unknown I/O Fault	Krytyczny	Wejścia/Wyjścia	io_flt	any_flt	io_pres	

## Wagi błędów

Błędy można podzielić na krytyczne, diagnostyczne oraz informacyjne.

Wystąpienie błędu krytycznego powoduje zapisanie tego błędu w odpowiedniej tabeli, ustawienie wartości odpowiednich zmiennych systemowych oraz zatrzymanie systemu. W przypadku błędów diagnostycznych, są one zapisywane w odpowiedniej tabeli oraz ustawiania jest wartość zmiennych diagnostycznych. Błędy informacyjne są zapisywane wyłącznie w odpowiedniej tabeli.

Działania podejmowane, w zależności od wagi błędów, przedstawiono w zamieszczonej poniżej tabeli:

**Tabela 3-2 Wagi błędów i działania podejmowane w momencie ich wystąpienia**

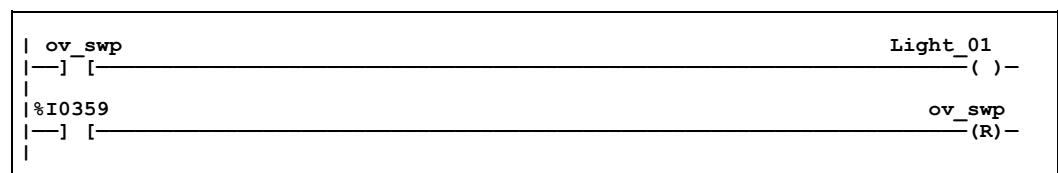
Waga błędu	Podejmowane działania
Krytyczny	Zarejestrowanie błędu w tabeli błędów Ustawienie wartości zmiennych systemowych na 1. Zatrzymanie sterownika (tryb <b>STOP</b> ).
Diagnostyczny	Zarejestrowanie błędu w tabeli błędów Ustawienie wartości zmiennych systemowych na 1.
Informacyjny	Zarejestrowanie błędu w tabeli błędów

W przypadku wystąpienia błędu, jednostka centralna podejmuje odpowiednie działania, w zależności od wagi błędu. Nie można zmieniać wagi błędów w sterownikach 90-30, 90-20 i Micro.

## Zmienne do obsługi błędów

W sterownikach 90-30 istnieją zmienne (%S, %Sx), sygnalizujące wystąpienie błędów w pracy systemu. Zmienne te zawierają informacje o typie błędu, który wystąpił. Zmienna opisująca błąd pozostaje ustawiona (1), aż do momentu wyzerowania sterownika lub zmianie wartości przez program sterujący.

Poniżej zamieszczono przykład, w którym wartość bitu informującego o błędzie ustawiana po czym na zerowana. W przykładzie tym przekaźnik Light\_01 jest załączany po zwarceniu styku systemowego OV\_SWP (%SA0002), co oznacza że przekroczony został czas trwania cyklu. Styk OV\_SWP oraz przekaźnik Light\_01 są wyłączane po zwarceniu styku %I0359, ponieważ zwarcie styku %I0359 wyłącza styk OV\_SWP.



## Zmienne systemowe

Procesor alarmowy pamięta wartość 128 bitów w obszarze pamięci adresowanym przez zmienne systemowe. Wiele z tych zmiennych określa miejsce wystąpienia błędu i jego rodzaj. Zmienne %S, %SA,

%SB i %SC posiadają nazwy pomocnicze. Na przykład bit %SA0009 posiada nazwę pomocniczą CFG\_MM i pojawienie się go informuje o błędnej konfiguracji. Mogą one być wykorzystywane przez program sterujący. Pełne zestawienie zmiennych systemowych podano w Rozdziale 2 "Działanie systemu".

## Dodatkowe skutki wystąpienia błędów działania

Wystąpienie jednego z dwóch podanych poniżej błędów działania, powoduje wystąpienie dodatkowych skutków. Skutki te są opisane w zamieszczonej poniżej tabeli:

Błąd	Opis skutków dodatkowych
PLC CPU Software Failure	Po każdorazowym zarejestrowaniu takiego błędu, jednostka centralna natychmiast przechodzi w specjalny tryb pracy <b>ERROR SWEEP</b> . W trybie tym nie są dozwolone jakiegokolwiek operacje. Jediną metodą anulowania tego trybu jest wyłączenie zasilania sterownika i jego ponowne włączenie.
PLC Sequence Store Failure	Jeśli podczas ładowania programu do pamięci nastąpi przerwa w komunikacji sterownika z urządzeniem, z którego ładowany jest program lub też nastąpi wystąpienie innego zakłócenia przerywającego ładowanie, rejestrowany jest błąd ładowania programu. Tak długo, jak błąd ten obecny jest w systemie, sterownik nie może zostać uruchomiony (nie może przejść w tryb <b>RUN</b> ). Aby móc ponownie ładowanie programu należy wyzerować błąd. Można to uczynić za pośrednictwem oprogramowania konfiguracyjnego.

## Tabela błędów działania sterownika

Na ekranie wyświetlania tabeli błędów sterownika pokazywane są informacje o takich błędach jak nieprawidłowe hasła, niezgodność konfiguracji, błędy sumy kontrolnej oraz błędy komunikacji.

Błędy są zapisywane w sterowniku, dlatego też jeżeli oprogramowanie konfiguracyjne znajduje się w trybie **OFFLINE** nie ma możliwości wyświetlenia tablicy błędów. Jeśli oprogramowanie konfiguracyjne znajduje się w trybie **ONLINE** lub **MONITOR** tablice błędów działania sterownika są dostępne. W trybie **ONLINE**, można usuwać błędy (funkcja ta może być zabezpieczona hasłem).

Po usunięciu informacji o błędzie pomimo, że błąd nadal występuje, nie jest on rejestrowany w tabeli błędów (za wyjątkiem informacji o rozładowaniu baterii) aż do momentu zapisania nowej konfiguracji lub wyłączenia i ponownego włączenia sterownika.

## Tabela błędów działania układów wejść/wyjść

Na ekranie wyświetlania tabeli błędów układów wejść/wyjść pokazywane są informacje o takich błędach jak uszkodzenie obwodów, konflikty adresów, wymuszeniach i uszkodzeniach magistrali wejść/wyjść.

Błędy są zapisywane w sterowniku, dlatego też jeżeli oprogramowanie konfiguracyjne znajduje się w trybie **OFFLINE** nie ma możliwości wyświetlenia tablicy błędów. Jeśli oprogramowanie konfiguracyjne znajduje się w trybie **ONLINE** lub **MONITOR** tablice błędów wejść/wyjść są dostępne. W trybie **ONLINE**, można usuwać błędy (funkcja ta może być zabezpieczona hasłem).

Po usunięciu informacji o błędzie pomimo, że błąd nadal występuje, nie jest on rejestrowany w tabeli błędów aż do momentu zapisania nowej konfiguracji lub wyłączenia i ponownego włączenia sterownika.







## Wagi błędów

- **Krytyczne** powodują przejście sterownika w tryb zatrzymania (**STOP**), po zakończeniu cyklu, w którym błąd wystąpił.
- **Diagnostyczne** są rejestrowane, a odpowiednia zmienna systemowa jest ustawiana; sterownik pozostaje w trybie **RUN**.
- **Informacyjne** są tylko rejestrowane w tabeli błędów działania sterownika; sterownik pozostaje w trybie **RUN**.

### Loss of, or Missing, Option Module (Brak lub uszkodzenie modułu dodatkowego)

Błąd **Loss of, or Missing Option Module** występuje w przypadku braku komunikacji z modułami dodatkowymi. Błąd może zostać zarejestrowany przy rozruchu sterownika, jeśli modułu nie ma lub podczas pracy sterownika, gdy moduł nie odpowiada na żądanie komunikacji. Jest to błąd diagnostyczny.

<b>Kod błędu:</b>	1, 42
<b>Nazwa:</b>	Option Module Soft Reset Failed
<b>Opis:</b>	Jednostka centralna nie może nawiązać komunikacji z modułem dodatkowym.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Powtórzyć procedurę uruchomienia dla tego modułu.</li> <li>(2) Wymienić moduł, który nie odpowiada na żądanie komunikacji.</li> <li>(3) Wyłączyć zasilanie systemu. Sprawdzić, czy odpowiadający moduł jest poprawnie wpięty w kasetę oraz czy wszystkie kable są poprawnie i trwale podłączone.</li> <li>(4) Sprawdzić i ewentualnie poprawić okablowanie.</li> </ol>
<b>Kod błędu:</b>	Wszystkie pozostałe.
<b>Nazwa:</b>	Module Failure During Configuration
<b>Opis:</b>	Oprogramowanie konfiguracyjne rejestruje błąd, gdy moduł jest niesprawny w czasie konfiguracji systemu lub podczas rozruchu sterownika.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Wyłączyć zasilanie systemu. Wymienić moduł umieszczony w odpowiednim gnieździe kasety sterownika.</li> </ol>

### Reset of, Addition of, or Extra, Option Module (Ponowne uruchomienie, dodanie lub brak konfiguracji dla modułu)

Błędy z grupy **Reset of, Addition of, or Extra Option Module** (Ponowne uruchomienie, dodanie lub brak konfiguracji dla modułu) występują w przypadku, gdy moduł dodatkowy (PCM, ADC, itp.) zostanie włączony, wyzerowany lub wykryty w gnieździe, ale nie jest określony w konfiguracji. Jest to błąd diagnostyczny.

<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Uaktualnienie danych konfiguracyjnych poprzez uwzględnienie dodatkowego modułu.</li> <li>(2) Usunięcie modułu z systemu.</li> </ol>
------------------	--

## System Configuration Mismatch (Błędna konfiguracja systemu)

Błędy z grupy **Configuration Mismatch** występują, jeżeli stwierdzona zostanie niezgodność pomiędzy modulem wpiętym w gniazdo, a informacjami wprowadzonymi w czasie konfiguracji. Jest to błąd **krytyczny**.

<b>Kod błędu:</b>	1
<b>Nazwa:</b>	System Configuration Mismatch
<b>Opis:</b>	Oprogramowanie systemowe sterownika rejestruje ten błąd, gdy konfiguracja rzeczywista nie jest zgodna z konfiguracją zaprogramowaną.
<b>Usuwanie:</b>	Zidentyfikować niezgodność a następnie ponownie skonfigurować moduł lub kasetę.
<b>Kod błędu:</b>	6
<b>Nazwa:</b>	System Configuration Mismatch
<b>Opis:</b>	Błąd ten wykazuje podobieństwo do błędu o kodzie nr 1, ponieważ występuje, jeżeli moduł zainstalowany w gnieździe nie jest zgodny z konfiguracją lub jeżeli skonfigurowany typ kasety nie jest zgodny z rzeczywiście zainstalowaną kasetą.
<b>Usuwanie:</b>	Zidentyfikować niezgodność a następnie ponownie skonfigurować moduł lub kasetę.
<b>Kod błędu:</b>	18
<b>Nazwa:</b>	Unsupported Hardware
<b>Opis:</b>	W kasecie z jednostką centralną 311, 313, 323 lub w kasecie rozszerzającej zainstalowany jest moduł programowalnego koprocatora PCM lub moduł tego typu.
<b>Usuwanie:</b>	Usunąć moduł PCM lub moduł typu PCM, albo zainstalować jednostkę centralną obsługującą taki moduł. UWAGA: Moduły tego typu można instalować w kasetach z jednostką centralną obsługującą ten typ modułu.
<b>Kod błędu:</b>	26
<b>Nazwa:</b>	Module busy—config not yet accept by module
<b>Opis:</b>	Nie można w danym momencie wprowadzić nowej konfiguracji dla modułu, ponieważ moduł ten zajęty jest przez inny proces.
<b>Usuwanie:</b>	Odczekać do momentu kiedy moduł zakończy realizowaną operację, po czym ponownie wprowadzić konfigurację.
<b>Kod błędu:</b>	51
<b>Nazwa:</b>	END Function Executed from Sequential Function Chart (SFC) Action
<b>Opis:</b>	Wystąpienie tego błędu powodowane jest przez umieszczenie w programie sterującym SFC lub w programie wywoływanym z poziomu programu SFC funkcji END.
<b>Usuwanie:</b>	Usunąć funkcję END z programu sterującego SFC lub z programu wywoływanego z poziomu programu SFC.

## Option Module Software Failure (Błąd w oprogramowaniu wyspecjalizowanego modułu dodatkowego)

Powodem występowania błędów z grupy **Option Module Software Failure (Błąd w oprogramowaniu wyspecjalizowanego modułu dodatkowego)** jest wykrycie nieusuwalnego błędu w oprogramowaniu modułu PCM lub ADC. Jest to błąd **krytyczny**.

<b>Kod błędu:</b>	Wszystkie
<b>Nazwa:</b>	COMMREQ Frequency Too High
<b>Opis:</b>	Sygnały żądania komunikacji są wysyłane do modułu z częstotliwością większą, niż zdolność modułu do ich przetwarzania.
<b>Usuwanie:</b>	Dokonać zmian w programie sterującym, aby sterownik wysyłał do modułu sygnały żądania komunikacji z mniejszą częstotliwością.

## Program Block Checksum Failure (Błędna suma kontrolna programu)

Błąd z grupy **Program Block Checksum Failure** występuje, gdy jednostka centralna sterownika wykryje warunki odpowiadające błędom w blokach programu sterującego przesłanego do sterownika. Jest on również rejestrowany, gdy jednostka centralna sterownika wykryje niezgodność sumy kontrolnej na etapie sprawdzania pamięci podczas rozruchu lub w trakcie pracy sterownika (w trybie **RUN**). Jest to błąd **krytyczny**.

<b>Kod błędu:</b>	Wszystkie
<b>Nazwa:</b>	Program Block Checksum Failure
<b>Opis:</b>	Oprogramowanie systemowe sterownika wykrywa ten błąd, gdy uszkodzony jest jeden z bloków programu sterującego.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Wyczyścić pamięć sterownika i powtórzyć ładowanie programu.</li> <li>(2) Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc podając wszystkie informacje dotyczące błędu.</li> </ol>

## Low Battery Signal (Rozładowane baterie)

Błędy z grupy **Low Battery Signal** występują w przypadku wykrycia zbyt niskiego napięcia baterii w jednostce centralnej lub w przypadku zasygnalizowania przez jeden z modułów (np. PCM) niskiego napięcia baterii. Jest to błąd **diagnostyczny**.

<b>Kod błędu:</b>	0
<b>Nazwa:</b>	Failed Battery Signal
<b>Opis:</b>	Bateria jednostki centralnej (lub innego modułu posiadającego baterię) jest wyczerpana.
<b>Usuwanie:</b>	Wymienić baterię. Nie wyłączać zasilania kasety sterownika.
<b>Kod błędu:</b>	1
<b>Nazwa:</b>	Low Battery Signal
<b>Opis:</b>	Bateria jednostki centralnej (lub innego modułu posiadającego baterię) daje zbyt niskie napięcie.
<b>Usuwanie:</b>	Wymienić baterię. Nie wyłączać zasilania kasety sterownika.

## Constant Sweep Time Exceeded (Przekroczony czas trwania cyklu pracy)

Błędy z grupy **Constant Sweep Time Exceeded** występują, gdy jednostka centralna sterownika pracuje w trybie ze stałym czasem trwania cyklu pracy, a czas trwania cyklu przekroczył zadaną wartość. Dodatkowe dane w opisie błędu zawierają rzeczywisty czas trwania cyklu (w pierwszych dwóch bajtach) i nazwę programu sterującego (w następnych ośmiu bajtach). Jest to błąd **diagnostyczny**.

<b>Usuwanie:</b>	(1) Zwiększyć zadawany stały czas trwania cyklu pracy sterownika.
	(2) Usunąć fragment części logicznej programu sterującego.

## Application Fault (Błąd w działaniu programu sterującego)

Błędy z grupy **Application Fault** występują, gdy jednostka centralna sterownika wykryje błąd w programie sterującym (błąd struktury programu). Jest to błąd **diagnostyczny** za wyjątkiem błędu Subroutine Call Stack Exceeded, w którym to przypadku jest to błąd **krytyczny**.

<b>Kod błędu:</b>	7
<b>Nazwa:</b>	Subroutine Call Stack Exceeded
<b>Opis:</b>	Podprogramy mogą być wywoływane z poziomu z innych podprogramów, dopuszczalne jest osiem poziomów zagnieżdżeń. Z poziomu jednego podprogramu można wywołać następny podprogram, która z kolei znowu może wywoływać następny podprogram, aż do dojścia do 8 poziomu.
<b>Usuwanie:</b>	Zmodyfikować program sterujący tak, aby ilość poziomów przywołań podprogramów nie przekraczała ośmiu.
<b>Kod błędu:</b>	1B
<b>Nazwa:</b>	Comm Req Not Processed Due To PLC Memory Limitations
<b>Opis:</b>	Żądania bezzwłocznej komunikacji mogą być umieszczane w kolejce z większą prędkością niż są obsługiwane (np. jedno na cykl). W przypadku takiej sytuacji, w momencie nagromadzenia się liczby żądań powodującej spadek dostępnej pamięci poniżej dopuszczalnej, minimalnej wartości, żądania takie traktowane są jako błędy i nie są przetwarzane.
<b>Usuwanie:</b>	Zmniejszyć liczbę żądań komunikacji lub zredukować ilość danych wymienianych w systemie.
<b>Kod błędu:</b>	5A
<b>Nazwa:</b>	User Shut Down Requested
<b>Opis:</b>	Oprogramowanie systemowe sterownika (bloki funkcyjne) generują ten błąd informacyjny, gdy w programie sterującym zostanie wykonana funkcja SVCREQ #13 (powodująca zatrzymanie sterownika - przejście w tryb STOP).
<b>Usuwanie:</b>	Żadne działanie nie jest konieczne. Jest to jedynie błąd informacyjny.

## No User Program Present (Brak programu sterującego)

Błędy z grupy **No User Program Present** występują, gdy sterownik otrzyma instrukcję przejścia z trybu zatrzymania (**STOP**) do trybu (**RUN**), a w pamięci sterownika nie ma programu sterującego. Jednostka centralna sprawdza poprawność programu sterującego w momencie rozruchu. Jest to błąd **Informacyjny**.

<b>Usuwanie:</b>	Załadować program sterujący przed przejściem do trybu <b>RUN</b> .
------------------	--

## Corrupted User Program on Power-Up (Wykrycie uszkodzenia programu sterującego podczas rozruchu)

Błędy z grupy **Corrupted User Program on Power-Up** występują, jeżeli podczas rozruchu stwierdzone zostanie uszkodzenie pamięci RAM. Jednostka sterująca sterownika będzie pozostawać w trybie **STOP**, aż do momentu załadowania poprawnego programu sterującego i pliku konfiguracyjnego. Jest to błąd **krytyczny**.

<b>Kod błędu:</b>	1
<b>Nazwa:</b>	Corrupted User RAM on Power-Up
<b>Opis:</b>	Oprogramowanie systemowe sterownika rejestruje ten błąd przy wykryciu błędów pamięci RAM podczas rozruchu.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Załadować ponownie program sterujący, plik konfiguracyjny oraz ewentualne wartości zmiennych.</li> <li>(2) Wymienić baterię jednostki centralnej sterownika.</li> <li>(3) Wymienić płytę z rozszerzeniem pamięci jednostki centralnej sterownika.</li> <li>(4) Wymienić jednostkę centralną sterownika.</li> </ol>
<b>Kod błędu:</b>	2
<b>Nazwa:</b>	Illegal Boolean OpCode Detected
<b>Opis:</b>	Oprogramowanie systemowe sterownika rejestruje ten błąd przy wykryciu niewłaściwych instrukcji w programie sterującym.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Załadować ponownie program sterujący oraz ewentualne wartości zmiennych.</li> <li>(2) Wymienić płytę z rozszerzeniem pamięci jednostki centralnej sterownika.</li> <li>(3) Wymienić jednostkę centralną sterownika.</li> </ol>

## Password Access Failure (Niepoprawne hasło dostępu do sterownika)

Błędy z grupy **Password Access Failure** są rejestrowane w przypadku wprowadzenia niewłaściwego hasła dostępu dla jednego z poziomów dostępu do sterownika. Jest to błąd **Informacyjny**.

<b>Usuwanie:</b>	Ponowić próbę używając poprawnego hasła dostępu.
------------------	--

## PLC CPU System Software Failure (Błąd w oprogramowaniu jednostki centralnej sterownika)

Błędy z grupy **PLC CPU System Software Failure** są rejestrowane wskutek wielu różnych przyczyn związanych z działaniem oprogramowania systemowego sterowników 90-30, 90-20 i Micro. Błędy te mogą występować w różnych momentach pracy systemu. Są to błędy **krytyczne** powodujące natychmiastowe przejście jednostki centralnej sterownika w tryb pracy uniemożliwiający wykonanie jakiegokolwiek operacji (**ERROR SWEEP**) oprócz komunikacji z komputerem-programatorem. Jedynym sposobem anulowania tego stanu jest wyłączenie i włączenie zasilania sterownika. Jest to błąd **krytyczny**.

<b>Kod błędu:</b>	1 do B
<b>Nazwa:</b>	User Memory Could Not Be Allocated
<b>Opis:</b>	Oprogramowanie systemowe zarządzające pamięcią sterownika rejestruje ten błąd, gdy oprogramowanie żąda od programu zarządzającego pamięcią przydzielenia niedopuszczalnych bloków pamięci RAM. Błędy te nie powinny występować; są one spotykane tylko w czasie prac nad oprogramowaniem systemowym.
<b>Usuwanie:</b>	Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc, podając wszystkie informacje dotyczące błędu.
<b>Kod błędu:</b>	D
<b>Nazwa:</b>	System Memory Unavailable
<b>Opis:</b>	Oprogramowanie sterownika (skaner wejść i wyjść) rejestruje ten błąd, gdy program zarządzający pamięcią odmawia przydzielenia pamięci systemowej z powodu braku miejsca na stosie systemowym. Jest to jedynie błąd <i>informacyjny</i> jeśli błąd pojawia się w czasie wykonywania funkcji DO I/O. Jest to błąd <b>krytyczny</b> , jeśli pojawia się podczas rozruchu sterownika lub w czasie autokonfiguracji.
<b>Usuwanie:</b>	Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc, podając wszystkie informacje dotyczące błędu.
<b>Kod błędu:</b>	E
<b>Nazwa:</b>	System Memory Could Not Be Freed
<b>Opis:</b>	Oprogramowanie sterownika (skaner wejść i wyjść) rejestruje ten błąd, gdy program zarządzający pamięcią odmawia zwolnienia bloku pamięci systemowej. Błąd taki może pojawić się jedynie w czasie wykonywania funkcji DO I/O.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc podając wszystkie informacje dotyczące błędu.</li> <li>(2) Skontrolować pamięć sterownika.</li> </ol>
<b>Kod błędu:</b>	10
<b>Nazwa:</b>	Invalid Scan Request of the I/O Scanner
<b>Opis:</b>	Oprogramowanie sterownika (skaner wejść i wyjść) rejestruje ten błąd, gdy system operacyjny lub funkcja DO I/O nie żądają ani pełnego, ani częściowego skanowania wejść i wyjść.
<b>Usuwanie:</b>	Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc, podając wszystkie informacje dotyczące błędu.
<b>Kod błędu:</b>	13
<b>Nazwa:</b>	PLC Operating Software Error
<b>Opis:</b>	Błąd ten dotyczy pewnych problemów w funkcjonowaniu oprogramowania sterownika. Błąd ten nie powinien występować; jest one spotykany tylko w czasie prac nad oprogramowaniem systemowym.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc podając wszystkie informacje dotyczące błędu.</li> <li>(2) Skontrolować pamięć sterownika.</li> </ol>

<b>Kod błędu:</b>	14, 27
<b>Nazwa:</b>	Corrupted PLC Program Memory
<b>Opis:</b>	Oprogramowanie sterownika rejestruje ten błąd, gdy pojawiają się pewne problemy związane z funkcjonowaniem pamięci. Błędy te nie powinny występować; są one spotykane tylko w czasie prac nad oprogramowaniem systemowym.
<b>Usuwanie:</b>	(1) Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc podając wszystkie informacje dotyczące błędu. (2) Skontrolować pamięć sterownika.
<b>Kod błędu:</b>	27 do 4E
<b>Nazwa:</b>	PLC Operating Software Error
<b>Opis:</b>	Oprogramowanie sterownika rejestruje ten błąd, gdy pojawiają się pewne problemy związane z funkcjonowaniem pamięci. Błędy te nie powinny występować; są one spotykane tylko w czasie prac nad oprogramowaniem systemowym.
<b>Usuwanie:</b>	Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc, podając wszystkie informacje dotyczące błędu.
<b>Kod błędu:</b>	4F
<b>Nazwa:</b>	Communications Failed
<b>Opis:</b>	Oprogramowanie sterownika (procesor obsługi żądań komunikacji) rejestruje ten błąd, gdy na wysłane żądanie otrzymuje odmowę nawiązania komunikacji.
<b>Usuwanie:</b>	(1) Sprawdzić, czy szyna komunikacyjna funkcjonuje poprawnie. (2) Wymienić wyspecjalizowany moduł dodatkowy, który nie odpowiada na żądanie komunikacji.
<b>Kod błędu:</b>	50, 51, 53
<b>Nazwa:</b>	System Memory Errors
<b>Opis:</b>	Oprogramowanie sterownika rejestruje ten błąd, gdy program zarządzający pamięcią nie jest w stanie przydzielić bloku pamięci systemowej z powodu braku wolnej pamięci lub jej uszkodzeń.
<b>Usuwanie:</b>	(1) Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc podając wszystkie informacje dotyczące błędu. (2) Skontrolować pamięć sterownika.
<b>Kod błędu:</b>	52
<b>Nazwa:</b>	Backplane Communications Failed
<b>Opis:</b>	Oprogramowanie sterownika (procesor obsługi żądań komunikacji) rejestruje ten błąd, gdy nie jest w stanie nawiązać komunikacji przez szynę komunikacyjną.
<b>Usuwanie:</b>	(1) Sprawdzić, czy szyna komunikacyjna funkcjonuje poprawnie. (2) Wymienić wyspecjalizowany moduł dodatkowy, który nie odpowiada na żądanie komunikacji. (3) Sprawdzić, czy kabel równoległy programatora jest prawidłowo podłączony.
<b>Kod błędu:</b>	Wszystkie pozostałe.
<b>Nazwa:</b>	PLC CPU Internal System Error
<b>Opis:</b>	Nastąpił wewnętrzny błąd systemowy, który <b>nie</b> powinien pojawić się w systemie.
<b>Usuwanie:</b>	Wyświetlić tabelę błędów działania sterownika na ekranie komputera-programatora. Skontaktować się z serwisem firmy GE Fanuc, podając wszystkie informacje dotyczące błędu.



## Communications Failure During Store (Awaria komunikacji podczas przesyłania danych do sterownika)

Błędy z grupy **Communications Failure During Store** występują podczas ładowania bloków programu sterującego i innych danych do pamięci sterownika. Gdy podczas procesu transmisji danych nastąpi przerwa w komunikacji sterownika z urządzeniem, z którego ładowany jest program lub też nastąpią inne zakłócenia transmisji, zarejestrowany zostaje błąd. Tak długo, jak błąd ten obecny jest w systemie, sterownik nie może zostać uruchomiony (nie może przejść w tryb **RUN**).

Błąd ten *nie* jest automatycznie likwidowany przy rozruchu sterownika; użytkownik musi zrobić to sam. Jest to błąd **krytyczny**. Więcej informacji na temat tego błędu można znaleźć w części “Dodatkowe skutki wystąpienia błędu” w tym rozdziale.

<b>Usuwanie:</b>	Wyczyścić komunikat o błędzie i spróbować ponownie załadować program lub plik konfiguracyjny.
------------------	---



## Addition of I/O Module (Dołączony nie zadeklarowany moduł wejść/wyjść)

Kategoria błędów **Addition of I/O Module** dotyczy modułów wejść i wyjść analogowych oraz dyskretnych sterowników serii 90-30. Dla kategorii tej nie są podawane typy błędów i opisy błędów. Jest to błąd **diagnostyczny**.

<b>Opis:</b>	Oprogramowanie sterownika rejestruje ten błąd, gdy uszkodzony wcześniej moduł wejść/wyjść powraca do pracy.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Niepotrzebne jest żadne działanie, gdy moduł został usunięty lub wymieniony, lub wyłączono i włączono zasilanie kasety dodatkowej.</li> <li>(2) Uaktualnić dane konfiguracyjne w celu uwzględnienia dodatkowego modułu lub usunąć moduł z systemu.</li> </ol>
<b>Opis:</b>	Oprogramowanie sterownika rejestruje ten błąd, gdy wykryta zostaje fizyczna obecność modułu wejść/wyjść sterowników serii 90-30, który nie został zadeklarowany w pliku konfiguracyjnym sterownika.
<b>Usuwanie:</b>	<ol style="list-style-type: none"> <li>(1) Usunąć moduł jeśli został wpięty omyłkowo.</li> <li>(2) Uaktualnić dane konfiguracyjne w celu uwzględnienia dodatkowego modułu.</li> </ol>



# Rozdział 4

## Funkcje przekaźników

W niniejszym rozdziale podano informacje związane ze stosowaniem styków, przekaźników i połączeń w szczeblach programu sterującego w języku drabinkowym.

Funkcja	Strona
Przekaźniki o stykach otwartych i przekaźniki o stykach zamkniętych	4-2
Styki otwarte i styki zamknięte	4-1
Przekaźniki o stykach otwartych i przekaźniki o stykach zamkniętych, z pamięcią	4-4
Przekaźniki uaktywniane zboczem narastającym sygnału i przekaźniki uaktywniane zboczem opadającym sygnału	4-5
Przekaźniki ustawialne SET i RESET	4-6
Przekaźniki ustawialne SET i RESET z pamięcią	4-7
Połączenia poziome i pionowe	4-7
Przekaźniki i styki kontynuacji	4-8

## Styki

Styki są stosowane do monitorowania stanu zmiennych. Od stanu zmiennej przypisanej stykowi oraz od typu styku zależy, czy będzie on przewodził sygnał. Zmienna jest ustawiona (ON) jeżeli jej wartość jest równa 1, zmienna jest nie ustawiona (OFF) jeżeli jej wartość jest równa 0.

Tabela 4-1 Typy styków

Typ	Wyświetlacz	Dopływ sygnału do prawej strony
Przekaźnik o stykach otwartych	—  —	Jeżeli zmienna ma wartość 1
Styk zamknięty	—/ —	Jeżeli zmienna ma wartość 0
Styk kontynuacji	<+>——	Jeżeli zmienna związana z poprzednim szczeblem ma wartość 1.

## Przełączniki

Przełączniki są wykorzystywane do sterowania zmiennymi takimi jak %Q czy %M. Dopyły sygnału do przełącznika musi być sterowany przez inne elementy logiczne. Przełączniki natychmiast zmieniają wartość zmiennych, nie przesyłają one sygnału do swojej prawej strony. Jeżeli określony stan zmiennej przypisanej przełącznikowi ma decydować o wykonaniu pewnej części programu sterującego, należy tam zastosować zmienną wewnętrzną lub przełącznik i styk kontynuacji.

Przełączniki są zawsze umieszczane skrajnie, po prawej stronie szczebla programu sterującego. Szczebel może zawierać do ośmiu przełączników.

Typ przełącznika należy dobrać stosownie dożądanego działania programu sterującego. Stan przełączników z pamięcią jest zapamiętywany po wyłączeniu zasilania lub po przejściu sterownika z trybu zatrzymania (**STOP**) do trybu pracy (**RUN**). Stan przełączników bez pamięci jest ustawiany na zero po wyłączeniu zasilania lub po przejściu sterownika z trybu zatrzymania (**STOP**) do trybu pracy (**RUN**).

Tabela 4-2 Typy przełączników

Typy przełącznika	Wyświetlacz	Sygnal do przełącznika	Działanie
Przełącznik o stykach otwartych	—( )—	1	Ustawia wartość zmiennej na 1
		0	Ustawia wartość zmiennej na 0.
Przełącznik o stykach zamkniętych	—(/)—	1	Ustawia wartość zmiennej na 0.
		0	Ustawia wartość zmiennej na 1
Przełącznik o stykach otwartych, z pamięcią	—(M)—	1	Ustawia wartość zmiennej na 1, z pamięcią.
		0	Ustawia wartość zmiennej na 0, z pamięcią.
Przełącznik o stykach zamkniętych Przełącznik o stykach otwartych, z pamięcią	—(/M)—	1	Ustawia wartość zmiennej na 0, z pamięcią.
		0	Ustawia wartość zmiennej na 1, z pamięcią.
Przełącznik uaktywniany zboczem narastającym sygnału	—(↑)—	OFF→ON	Jeżeli zmienna ma wartość 0, jej wartość jest zmieniana na 1 na okres jednego cyklu.
Przełącznik uaktywniany zboczem opadającym sygnału	—(↓)—	ON←OFF	Jeżeli zmienna ma wartość 0, jej wartość jest zmieniana na 1 na okres jednego cyklu.
Przełącznik ustawialny SET	—(S)—	1	Ustawia zmienną na 1, aż do momentu jej wyzerowania poprzez —(R)—.
		0	Nie zmienia stanu przełącznika.
Przełącznik ustawialny RESET	—(R)—	1	Ustawia zmienną na 0, aż do momentu zmiany jej wartości na 1 poprzez przełącznik —(S)—.
		0	Nie zmienia stanu przełącznika.
Przełącznik ustawialny SET, z pamięcią	—(SM)—	1	Ustawia zmienną na 1, aż do momentu jej wyzerowania poprzez —(RM)—.
		0	Nie zmienia stanu przełącznika.
Przełącznik o stykach otwartych, z pamięcią Przełącznik ustawialny RESET	—(RM)—	1	Ustawia zmienną na 0, aż do momentu zmiany jej wartości na 1 poprzez przełącznik —(SM)—.
		0	Nie zmienia stanu przełącznika.
Przełącznik kontynuacji	—<+>	1	Zwarcie następującego po nim styku kontynuacji.
		0	Rozwarcie następującego po nim styku kontynuacji.

## Styk normalnie otwarty —| |—

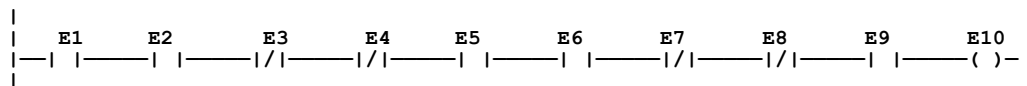
Styk taki działa jak wyłącznik, który przewodzi sygnał (zwiera styki), gdy wartość logiczna przypisanej mu zmiennej wynosi 1.

## Styk normalnie zwarty —||—

Styk taki działa jak wyłącznik, który przewodzi sygnał (zwiera styki), gdy wartość logiczna przypisanej mu zmiennej wynosi 0.

### Przykład

Poniżej pokazano przykładowy szczebel z 10 elementami, o nazwach pomocniczych od E1 do E10 (w rozdziale 2 zamieszczono więcej informacji o nazwach pomocniczych). Przekaznik E10 ma wartość logiczną 1, jeżeli zmienne E1, E2, E3, E5, E6 i E9 są równe 1, a zmienne E3, E4, E7 i E8 są równe 0.

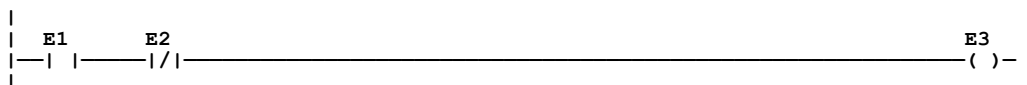


## Przekaznik —( )—

Przekaznik ustawia wartość przypisanej zmiennej na jeden, gdy dopłynie do niego sygnał. Jest to przekaznik bez pamięci, nie może on być zatem zastosowany wraz ze zmiennymi, które cechuje pamięć stanu (%SA, %SB, %SC lub %G).

### Przykład

W zamieszczonym poniżej przykładzie, przekaznik ustawia wartość przypisanej mu zmiennej na 1, jeżeli wartość E1 jest równa 1, a wartość E2 jest równa 0.

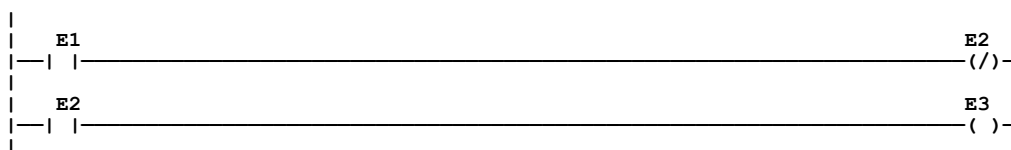


## Przełącznik o stykach zwartych —(I)—

Przełącznik ten ustawia wartość przypisanej zmiennej dyskretnej na jeden, gdy nie dopływa do niego sygnał. Jest to przełącznik bez pamięci, nie może on być zatem zastosowany wraz ze zmiennymi, które cechuje pamięć stanu (%SA, %SB, %SC lub %G).

### Przykład

W zamieszczonym poniżej przykładzie, przełącznik E3 ustawia wartość przypisanej mu zmiennej na 1, jeżeli wartość E1 jest równa 0.



## Przełącznik o stykach otwartych z pamięcią —(M)—

Podobnie jak zwykły przełącznik o stykach otwartych, przełącznik ten ustawia wartość przypisanej zmiennej na 1, gdy dopływie do niego sygnał. Stan przełącznika zostaje podtrzymany w przypadku zaniku zasilania sterownika. Z tego powodu, nie może być on zatem zastosowany wraz ze zmiennymi, które nie posiadają pamięci stanu (%T)

## Przełącznik o stykach zamkniętych, z pamięcią —(/M)—

Przełącznik taki ustawia wartość przypisanej zmiennej dyskretnej na 1, gdy nie dopływa do niego sygnał. Stan przełącznika zostaje podtrzymany w przypadku zaniku zasilania sterownika. Z tego powodu, nie może być on zatem zastosowany wraz ze zmiennymi, które nie posiadają pamięci stanu (%T)

## Przełącznik uaktywniany zboczem narastającym sygnału —(↑)—

Jeżeli zmienna przypisana do przełącznika uaktywnianego zboczem narastającym sygnału ma wartość 0, doprowadzenie sygnału do przełącznika powoduje ustawienie wartości tej zmiennej na 1. Styki powiązane z takim przełącznikiem zmieniają swój stan na okres jednego cyklu pracy sterownika. (Jeżeli szczebel zawierający przełącznik zostanie pominięty w czasie kolejnych cykli, wartość tej zmiennej jest nadal równa 1, przez cały czas trwania tych cykli). Przełącznik ten może zostać wykorzystany do ustalania wartości zmiennej na okres jednego cyklu.

Każda zmienna powinna być wykorzystywana w programie sterującym jako przełącznik uaktywniany zboczem sygnału tylko jeden raz, co pozwoli na uniknięcie charakterystycznego dla tego przełącznika ustawiania wartości na okres jednego cyklu.

Przełączniki ustawiane zboczem sygnału mogą być stosowane ze zmiennymi z pamięcią stanu lub nie posiadającymi tej cechy (tzn. %Q, %M, %T, %G, %SA, %SB lub %SC).



## Przełącznik uaktywniany zboczem opadającym sygnału —(↓)—

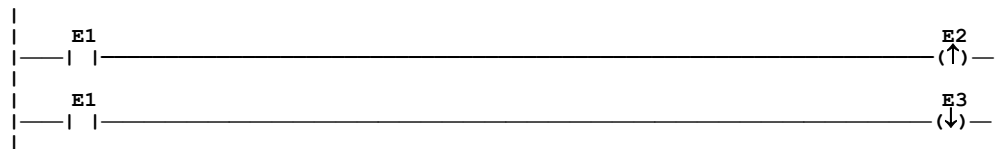
Jeżeli zmienna przypisana do tego przełącznika ma wartość 0, po zaprzestaniu doprowadzania sygnału do przełącznika, zmienna jest ustawiana na 1, a styki połączone z tym przełącznikiem zmieniają stan na okres jednego cyklu pracy sterownika.

Zmienna powinna być wykorzystywana w programie sterującym jako przełącznik uaktywniany zboczem sygnału tylko jeden raz, co pozwoli na uniknięcie charakterystycznego dla tego przełącznika ustawiania wartości na okres jednego cyklu.

Przełączniki ustawiane zboczem sygnału mogą być stosowane ze zmiennymi z pamięcią stanu lub nie posiadającymi tej cechy (tzn. %Q, %M, %T, %G, %SA, %SB lub %SC).

### Przykład

W zamieszczonym poniżej przykładzie zmienna E1 zmienia wartość z 0 na 1, sygnał dopływa do przełączników E2 i E3, a zmienna związana z przełącznikiem E2 przyjmuje wartość 1 na okres jednego cyklu. Jeżeli zmienna związana z E1 zmieni wartość z 1 na 0, sygnał przestaje dopływać do E2 i E3, przez co zmienna związana z przełącznikiem E3 przyjmuje wartość 1 na okres jednego cyklu.



## Przełącznik ustawialny SET —(S)—

Przełączniki SET i RESET są przełącznikami bez pamięci, które można zastosować do ustawiania na pewien czas wartości im przypisanej. Gdy do przełącznika SET dopływnie sygnał, wartość przypisanej zmiennej zostaje ustawiona na 1 i jest utrzymywana do momentu, aż sygnał dopływnie do przełącznika RESET do którego przypisano tę samą zmienną, niezależnie od tego, czy przez ten czas nadal dopływnie sygnał do przełącznika SET, czy nie.

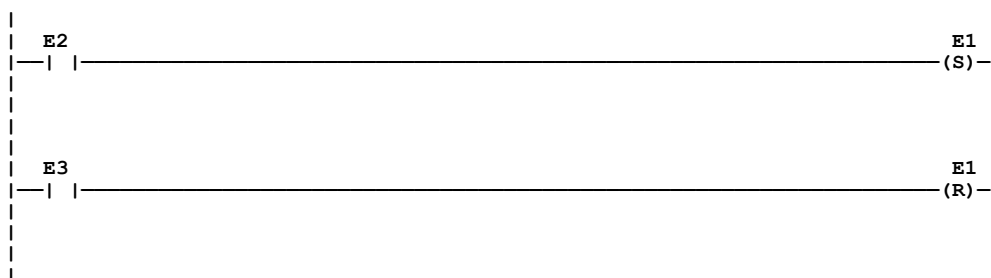
## Przełącznik ustawialny RESET —(R)—

Gdy sygnał dopływnie do przełącznika RESET, wartość zmiennej jest ustawiana na 0. Zmienna ma wartość 0 do momentu, aż do momentu zmiany jej wartości przez inny przełącznik. Jej wartość jest zmieniana stosownie do przełącznika, do którego dopływnie sygnał.

## Przykład

W poniższym przykładzie zmienna przypisana do przekaźnika E1(S) zostaje ustawiona na 1 w momencie załączenia cewki E2. Nawet po wyzerowaniu zmiennej przypisanej do styku E2, przekaźnik E1(R) pozostaje załączony do momentu ustawienia styku E3.

UWAGA: W przypadku jednoczesnego załączenia styków E2 i E3, przekaźnik E1 pozostanie wyłączony. Dzieje się tak ponieważ program jest wykonywany od góry do dołu, a stan przekaźnika reset w drugim szczelbli będzie zapisany później. Jeśli kolejność szczelbli byłaby odwrotna, przekaźnik set byłby skanowany ostatni, a E1 byłby załączony przy jednoczesnym załączeniu E2 i E3.



### Uwaga

Jeżeli kontrola wielokrotnego wykorzystania zmiennych w przekaźnikach ustawiona jest na SINGLE, można korzystać ze zmiennych typu %M i %Q, wyłącznie z jednym przekaźnikiem, ale można z nich korzystać jednocześnie z jednym przekaźnikiem SET i jednym przekaźnikiem RESET. Jeżeli kontrola wielokrotnego wykorzystania zmiennych w przekaźnikach jest ustawiona na WARN MULTIPLE lub MULTIPLE, każda ze zmiennych może być używana z wieloma normalnymi przekaźnikami lub przekaźnikami Set i przekaźnikami RESET. W przypadku wielokrotnego wykorzystywania, wartość zmiennej można ustawić na 1 przy pomocy zarówno przekaźnika SET jak i normalnego przekaźnika, a następnie zmienić jej wartość na 0 przy pomocy przekaźnika RESET lub normalnego przekaźnika.

## Przekaźnik SET z pamięcią —(SM)—

Przekaźniki SET i RESET z pamięcią są podobne w działaniu do przekaźników SET i RESET bez pamięci, lecz ich stan jest podtrzymywany w przypadku wyłączenia zasilania sterownika, lub po przejściu sterownika z trybu **STOP** do trybu **RUN**. Gdy sygnał dopłynie do przekaźnika SET z pamięcią, wartość związanej z nim zmiennej jest ustawiana na 1. Wartość ta pozostaje równa 1 do momentu wyzerowania za pomocą przekaźnika RESET z pamięcią.

Przekaźnik SET z pamięcią ustawia przypadkową wartość bitu chwilowego przełączenia na wartość przeciwną danej zmiennej. (Proszę porównać z informacjami podanymi w "Wymuszone zmiany wartości zmiennych" w punkcie 2 "Działanie systemu").

## Przekaźnik RESET z pamięcią —(RM)—

Przekaźnik taki ustawia wartość przypisanej zmiennej na zero, gdy dopłynie do niego sygnał. Wartość ta pozostaje równa 0, do momentu ustawienia wartości 1 za pomocą przekaźnika SET z pamięcią. Stan tego

przełącznika jest zapamiętywany po wyłączeniu zasilania, lub po przejściu sterownika z trybu zatrzymania (**STOP**) do trybu pracy (**RUN**).

Przełącznik RESET z pamięcią ustawia przypadkową wartość bitu chwilowego przełączenia na wartość przeciwną danej zmiennej. (Proszę porównać z informacjami podanymi w "Wymuszone zmiany wartości zmiennych" w punkcie 2 "Działanie systemu").

## Szyny

Połączenia poziome i pionowe (widoczne jako proste linie) służą do łączenia elementów logicznych programu sterującego. Umożliwiają przepływ sygnału sterującego z lewej do prawej strony drabiny logicznej programu sterującego.

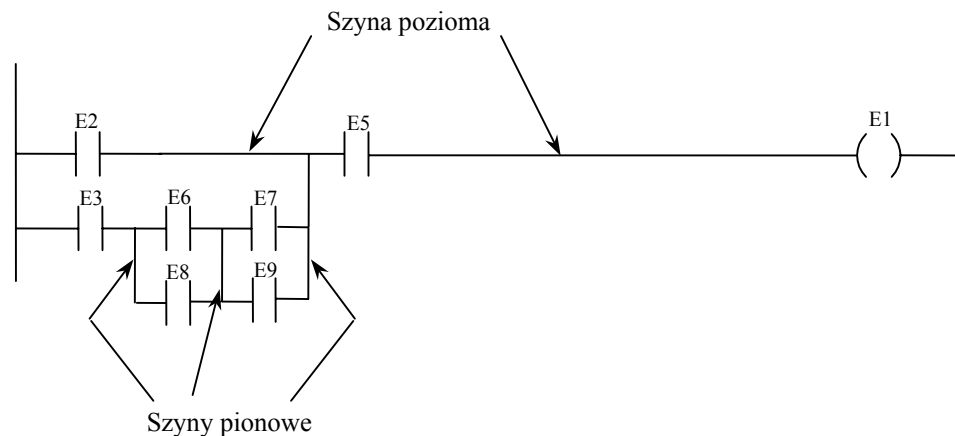
### Uwaga

Nie można za pomocą połączenia poziomego połączyć funkcji lub przełącznika z lewą stroną sygnału. Jeżeli funkcja ma być wywoływana w każdym cyklu, można skorzystać ze zmiennej %S7, ALW\_ON (zawsze włączona) powiązanej ze stykiem otwartym.

## Przykład

Poniżej pokazano zastosowanie szyn:

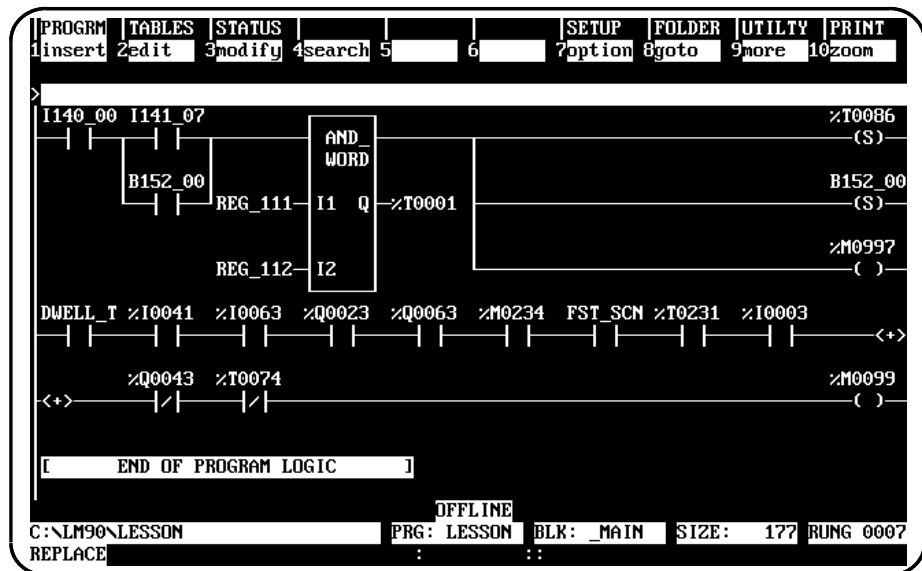
- Szyna pozioma łączy styk E2 z E5 oraz styk E5 z przełącznikiem E1.
- Szyny pionowe łączą styk E8 z E6, E9 z E7 oraz prawą stronę styków E7/E9 ze stykami E2 i E5.



## Przełączniki kontynuacji (—<+>) i styki (<+>—) kontynuacji

Przełączniki kontynuacji (—<+>) i styki kontynuacji (<+>—) służą do przedłużenia szeregu drabiny logicznej programu sterującego poza limit dziesięciu kolumn. Stan ostatniego przełącznika kontynuacji jest automatycznie przenoszony na następujący po nim styk. Warunkiem wykonania części logicznej styku kontynuacji jest umieszczenie przed nim przełącznika kontynuacji. Stan styku kontynuacji jest zerowany w przypadku przejścia sterownika z trybu **STOP** do trybu **RUN**, a sygnał nie będzie przepływał aż do momentu kiedy wartość zmiennej związanej z przełącznikiem uaktywnianym zbroczem sygnału zostanie ustawiona na 1, od momentu przejścia w tryb run .

W szeregu może wystąpić tylko jeden przełącznik kontynuacji; styk kontynuacji może zajmować wyłącznie pozycję w pierwszej kolumnie szeregu, a przełącznik kontynuacji musi być umieszczony w 10 kolumnie szeregu. Poniżej pokazano przykład zastosowania przełącznika i styku kontynuacji.



# Rozdział 5

## Liczniki i przełączniki czasowe

This chapter explains how to use on-delay and stopwatch-type timers, up counters, and down counters. W punkcie tym objaśniono sposób stosowania przełączników czasowych oraz liczników zliczających w górę i w dół. The data associated with these functions is retentive through power cycles. Dane związane z tymi funkcjami pamiętane są przez cały okres zasilania.

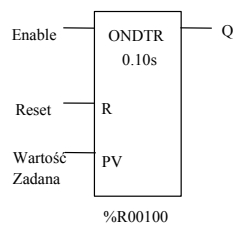
Oznaczenie skrótowe	FunctionFunkcja	Strona
ONDTR	Przełącznik czasowy z pamięcią	5-2
TMR	Przełącznik czasowy z pamięcią	5-5
OFDT	Przełącznik czasowy bez pamięci, z zanegowanym wejściem	5-8
UPCTR	Licznik zliczający w górę	5-11
DNCTR	Licznik zliczający w dół	5-13

### Blok danych sterujących przełącznikami czasowymi i licznikami

Każdy przełącznik czasowy i licznik wykorzystuje trzy słowa (rejstry) pamięci typu %R, do zapamiętywania następujących parametrów:

wartość bieżąca (CV)	słowo 1
wartość zadana (PV)	słowo 2
słowo sterujące	słowo 3

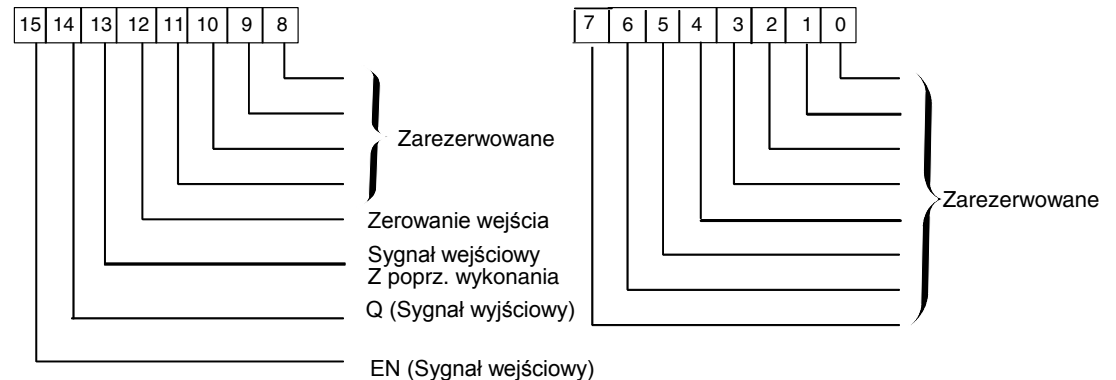
W bloku funkcyjnym przełącznika czasowego lub licznika należy podać adres początkowy dla tych trzech słów (rejestrów), bezpośredni pod symbolem graficznym wykorzystywanego bloku. Na poniższym przykładzie początkowym adresem jest %R00100.



#### Uwaga

Należy upewnić się, że blok trzech słów nie jest wykorzystywany w innym miejscu programu (duplikowanie adresów nazywa się „overlapping”). Oprogramowanie Logimaster *nie* sprawdza, ani też nie ostrzega w przypadku pokrywania się bloków rejestrów. Liczniki i przełączniki czasowe nie pracują poprawnie jeśli wykorzystują zduplikowane rejestry.

Słowo sterujące (trzecie słowo w bloku) zapamiętuje stan wejść i wyjść dyskretnych, powiązanych z danym blokiem funkcyjnym, zgodnie z przedstawionym poniżej formatem:



Bit 0 do 11 są zarezerwowane dla zapewnienia dokładności przełączników czasowych, bity te nie są wykorzystywane w przypadku liczników.

### Uwaga

Należy zachować ostrożność w przypadku wykorzystywania tego samego adresu dla wartości zadanej PV (Preset Value) i drugiego słowa w bloku trzech słów. Jeżeli wartość zadana PV nie jest stałą, wartość PV jest zwykle ustawiana w innym miejscu niż to drugie słowo. Niektóre programy sterujące korzystają z adresu drugiego słowa w celu uzyskania wartości PV, jak na przykład wywołanie %R0102, jeżeli blok danych rozpoczyna się od %R0101. Pozwala to na zmianę wartości PV w czasie pracy przełącznika czasowego czy licznika. Programy sterujące mogą odczytywać pierwsze słowo zawierające wartość bieżącą CV i trzecie słowo sterujące, ale nie mogą zmieniać ich wartości, ponieważ funkcja nie będzie poprawnie wykonana.

### Specjalna uwaga odnośnie niektórych operacji bitowych

**W przypadku korzystania z funkcji BTST, BSET, BCLR i BPOS**, bity numerowane są od 1 do 16, a nie od 0 do 15, jak w przykładzie powyżej.

## ONDTR

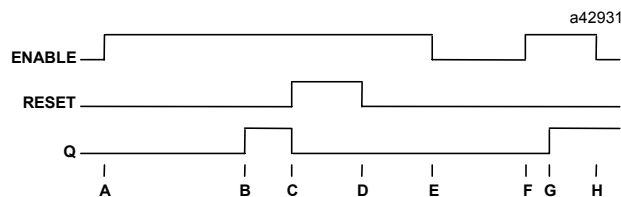
Przełącznik czasowy z pamięcią zlicza czas, gdy dopływa do niego sygnał i zatrzymuje naliczoną wartość, gdy sygnał przestaje dopływać. Czas może być zliczany w dziesiątych, setnych lub tysięcznych częściach sekundy. Zakres dopuszczalnych wartości wynosi 0 do +32 767, a więc zliczany czas może wynosić od 0.001 do 3 276.7 sekund. Wartość bieżąca przełącznika jest przechowywana w przypadku awarii zasilania sterownika.

Gdy do przełącznika dopłynie sygnał po raz pierwszy, rozpoczyna on naliczanie czasu (wartość bieżąca). Po napotkaniu przełącznika czasowego w drabinie programu sterującego, wartość bieżąca jest aktualizowana.

### Uwaga

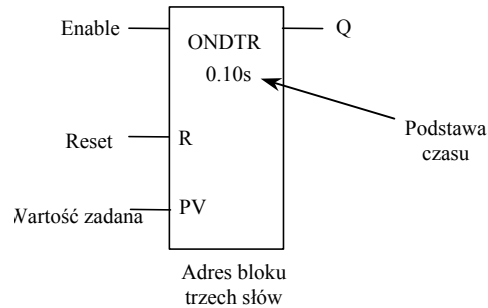
Jeżeli w programie sterującym znajdują się dwa lub więcej przełączników czasowych o tych samych przypisanych zmiennych, to ich bieżące wartości będą takie same.

Gdy bieżąca wartość zrówna się lub przekroczy wartość zadaną PV, na wyjściu Q zostaje przesłany sygnał. Przez cały czas dopływu sygnału do przełącznika, wartość ta jest inkrementowana, aż do momentu dojścia do wartości maksymalnej. Po dojściu do wartości maksymalnej 32,767 zliczanie zostaje zatrzymane, a sygnał pozostaje na wyjściu Q bez zmian.



- A = Na wejściu ENABLE pojawia się sygnał. Przełącznik rozpoczyna zliczanie
- B = Wartość bieżąca osiąga wartość zadaną PV. Na wyjściu Q pojawia się sygnał.
- C = Na wejściu RESET pojawia się sygnał. Na wyjściu Q nie jest przesyłany sygnał. Czas zliczany zostaje wyzerowany.
- D = Na wejście RESET nie jest przesyłany sygnał. Przełącznik rozpoczyna zliczanie czasu od początku.
- E = Sygnał przestaje być doprowadzany do wejścia Enable. Przełącznik przerywa zliczanie czasu. Wartość bieżąca pozostaje zapamiętana.
- F = Na wejściu ENABLE pojawia się znowu sygnał. Przełącznik kontynuuje zliczanie czasu.
- G = Wartość bieżąca osiąga wartość zadaną PV. Na wyjściu pojawia się sygnał. Przełączniki pracują aż do momentu gdy wyłączenia ENABLE, lub pojawienia się sygnału RESET lub aż wartość CV zrówna się z wartością zadaną.
- H = Na wejście ENABLE przestaje być podawany sygnał. Przełącznik przerywa zliczanie czasu.

W momencie gdy do przełącznika przestaje dopływać sygnał, zliczanie czasu zostaje zatrzymane i zapamiętany zostaje stan licznika. Jeżeli do wyjścia Q jest doprowadzany sygnał pozostaje ono w tym stanie. Jeżeli na wejściu znowu pojawi się sygnał, wartość bieżąca jest ponownie inkrementowana, począwszy od zapamiętanej wcześniej wartości. Doprowadzenie sygnału do wejścia R powoduje ustawienie bieżącej wartości na 0 i zaprzestanie przesyłania sygnału na wyjściu Q. W sterownikach serii 35x, 36x i 37x, jeżeli do wejścia Enable nie jest podawany sygnał, PV=0 i doprowadzany jest sygnał do wejścia R, na wyjściu nie jest przesyłany sygnał. Należy jednak pamiętać, że w przypadku sterowników 311-341, w takich samych warunkach, na wyjściu będzie przesyłany sygnał.



## Parametry

Parametr	Opis
Adres bloku trzech słów	Przełącznik czasowy ONDTR wykorzystuje trzy słowa (rejstry) pamięci typu %R, do zapamiętywania następujących parametrów: <ul style="list-style-type: none"> <li>• Wartości bieżącej (CV) = słowo 1.</li> <li>• Wartości zadanej (PV) = słowo 2.</li> <li>• Słowo sterujące = słowo 3.</li> </ul> <p>Bezpośrednio pod blokiem ONDTR należy podać adres pierwszego z trzech słów (rejestrów).</p> <p><b>Ostrzeżenie:</b> Nie można wykorzystywać tych adresów przez inne funkcje. Wykorzystywanie tych adresów przez inne elementy logiczne powoduje zakłócenia w pracy przełącznika.</p>
Enable	Sygnal uruchamiający funkcję zliczania czasu.
R	Sygnal zerujący. Sygnal zerujący wartość bieżącą. Wejście R musi być połączone przez jeden lub więcej styków do szyny zasilającej. Blok ONDTR musi być pierwszym blokiem w szczeblu drabiny logicznej (pierwszy z lewej strony).
PV	Wartość zadana. Wartość PV jest kopiowana do drugiego rejestru (adres+1) w momencie zerowania lub uruchomienia przełącznika. Przełącznik załączy wyjście Q w momencie zliczenia czasu równemu wartości PV.
Q	Sygnal wyjściowy, pojawiający się, gdy wartość bieżąca (CV) jest większa lub równa od wartości zadanej (PV).
Podstawa czasu	Wartość określająca przyrost czasu w dziesiątych (0.1), setnych (0.01) lub tysięcznych (0.001) częściach sekundy. Wartość podstawy czasu jest mnożona przez liczbę z wejścia PV, w celu określenia aktualnej wartości.

## Dopuszczalne typy zmiennych

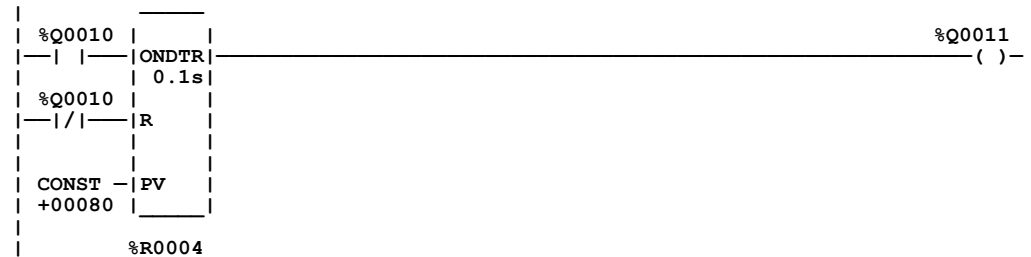
Parametr	flow	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
adres								•				
enable	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.



## Przykład

W zamieszczonym poniżej przykładzie, przełącznik czasowy z pamięcią wykorzystany jest to wysłania sygnału (%Q0011) po upływie 8.0 sekund od momentu ustawienia zmiennej %Q0010 na wartość 1 oraz zanikającego po ustawieniu wartości zmiennej %Q0010 na 0. Dzieje się tak gdyż wyłączając %Q0010 - styk normalnie zwarty przepuszcza sygnał do wejścia zerującego R. Czas 8 sekund jest uzyskiwany przez mnożenie wartości PV 80 przez podstawę czasu 0.1 s.



## TMR

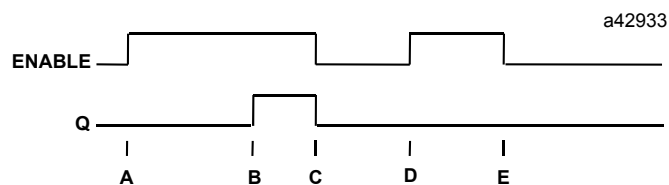
Przełącznik czasowy bez pamięci (TMR) zlicza czas, gdy dopływa do niego sygnał, zostaje wyzerowany, gdy sygnał przestaje dopływać. Czas może być zliczany w dziesiątych, setnych lub tysięcznych częściach sekundy. Zakres dopuszczalnych wartości wynosi 0 do +32 767, a więc zliczany czas może wynosić od 0.001 do 3 276.7 s. Wartość bieżąca przełącznika jest przechowywana w przypadku awarii zasilania sterownika.

Gdy do przełącznika dopłynie sygnał po raz pierwszy, rozpoczyna on naliczanie czasu. Bieżąca wartość jest aktualizowana po jej wywołaniu w programie sterującym, co pozwala na odczytanie czasu, który upłynął od momentu, kiedy na wejściu Reset pojawił się sygnał.

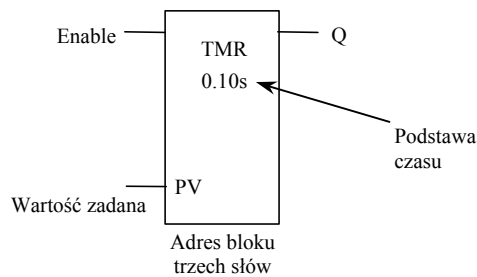
### Uwaga

Jeżeli w programie sterującym znajdują się dwa lub więcej przełączników czasowych o tych samych przypisanych zmiennych, to ich bieżące wartości będą takie same.

Wartość CV jest odmierzana tak długo jak długo dopływa sygnał Enable. Jeżeli wartość bieżąca CV jest równa lub większa od wartości zadanej PV, na wyjściu Q pojawia się sygnał. Przełącznik kontynuuje odliczanie czasu, do momentu dojścia do wartości maksymalnej (32 767 jednostek czasu). Jeżeli do wejścia Enable przestanie dopływać sygnał, przełącznik wstrzymuje zliczanie czasu, a wartość bieżąca zostaje wyzerowana.



- A = Na wejściu ENABLE pojawia się znowu sygnał. Przełącznik rozpoczyna zliczanie czasu.
- B = Wartość bieżąca osiąga wartość zadaną PV. Na wyjściu Q pojawia się sygnał, a przełącznik nadal kontynuuje odliczanie czasu.
- C = Na wejście ENABLE przestaje być podawany sygnał, na wyjście nie jest przesyłany sygnał, zatrzymane zostaje zliczanie czasu oraz zerowana jest wartość bieżąca.
- D = Na wejściu ENABLE znowu pojawia się sygnał. Przełącznik rozpoczyna zliczanie czasu.
- E = Na wejście ENABLE przestaje być podawany sygnał zanim wartość bieżąca osiągnie wartość zadaną. Na wyjście nie jest nadal przesyłany sygnał. Przełącznik przerywa zliczanie czasu, zerując wartość bieżącą.



## Parametry

Parametr	Opis
Adres bloku trzech słów	Przełącznik czasowy TMR wykorzystuje trzy słowa (rejstry) pamięci typu %R, do zachowania następujących parametrów: <ul style="list-style-type: none"> <li>• Wartości bieżącej (CV) = słowo 1.</li> <li>• Wartości zadanej (PV) = słowo 2.</li> <li>• Słowo sterujące = słowo 3.</li> </ul> Bezpośrednio pod blokiem ONDTR należy podać adres pierwszego z trzech słów (rejestrów). <b>Ostrzeżenie:</b> Nie można wykorzystywać tych adresów przez inne funkcje. Wykorzystywanie tych adresów przez inne elementy logiczne powoduje zakłócenia w pracy przełącznika.
Enable	Sygnal uruchamiający funkcję zliczania czasu. Jeżeli do wejścia Enable nie jest doprowadzany sygnał, wartość bieżąca jest zerowana a na wyjście Q nie jest przesyłany sygnał.
PV	Wartość zadana. Wartość PV jest kopiowana do drugiego rejestru (adres+1) w momencie zerowania lub uruchomienia przełącznika. Przełącznik załączy wyjście Q w momencie zliczenia czasu równemu wartości PV.
Q	Sygnal wyjściowy, pojawiający się, gdy wartość bieżąca (CV) jest większa lub równa od wartości zadanej (PV).
Podstawa czasu	Wartość określająca dokładność zliczania czasu w dziesiątych (0.1), setnych (0.01) lub tysięcznych (0.001) częściach sekundy. Wartość podstawy czasu jest mnożona przez liczbę z wejścia PV, w celu określenia aktualnej wartości.

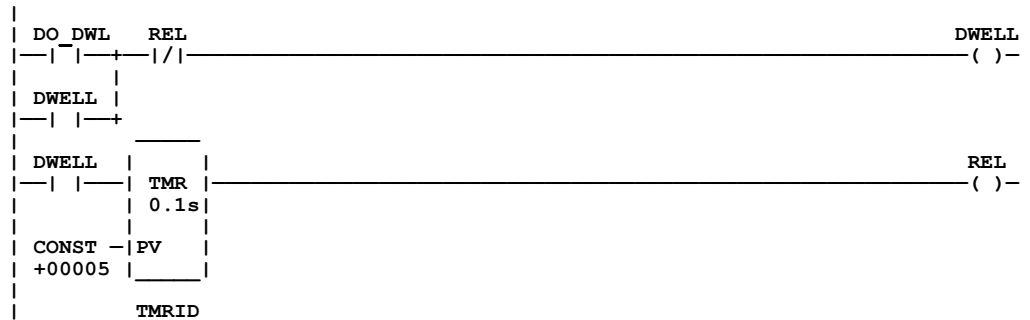
## Dopuszczalne typy zmiennych

Parametr	sygnal	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
adres								•				
enable	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład

W zamieszczonym poniżej przykładzie, przekaźnik czasowy TMR wykorzystywany jest do sterowania czasem załączenia przekaźnika DWELL. Zliczanie czasu rozpoczyna się załączeniu normalnie otwartego styku DO\_DWL, załączającego przekaźnik DWELL. Styk DWELL utrzymuje załączony przekaźnik DWELL ("zatrzaśnięty") w czasie gdy styk DO\_DWL pozostaje otwarty, inny styk DWELL załącza przekaźnik czasowy. Gdy przekaźnik czasowy odliczy wartość zadaną, złączony zostanie przekaźnik REL. Normalnie otwarty styk REL zostanie otwarty wyłączając jednocześnie przekaźnik DWELL. Styk DWELL na wejściu Enable powoduje przerwanie zliczenia czasu i wyzerowanie przekaźnika czasowego oraz wyłączenie przekaźnika REL. Obwód jest w tym momencie gotowy do następnego zwarcia styku DO\_DWL.



## OFDT

Przełącznik czasowy bez pamięci, z zanegowanym wejściem (OFDT) zlicza czas, gdy nie dopływa do niego sygnał i zostaje wyzerowany, gdy sygnał zacznie dopływać. Czas może być zliczany w dziesiątych, setnych lub tysięcznych częściach sekundy. Zakres dopuszczalnych wartości wynosi 0 do +32 767, a więc zliczany czas może wynosić od 0.001 do 3 276.7 s. Wartość bieżąca przełącznika jest przechowywana w przypadku awarii zasilania sterownika.

Gdy do przełącznika dopłynie sygnał po raz pierwszy, bieżąca wartość zostaje ustawiona na zero i sygnał zostaje przesłany na wyjście Q. (Przełącznik czasowy OFDT wykorzystuje słowo I (rejestr) do przechowywania wartości bieżącej CV - proszę porównać z punktem "Blok danych sterujących". Sygnał podawany jest na wyjście tak długo, jak długo doprowadzany jest sygnał do tego bloku funkcyjnego. Gdy do bloku przestanie dopływać sygnał z lewej strony, przełącznik rozpoczyna zliczanie czasu, po zliczeniu czasu równego wartości zadanej, zliczanie zostaje przerwane.

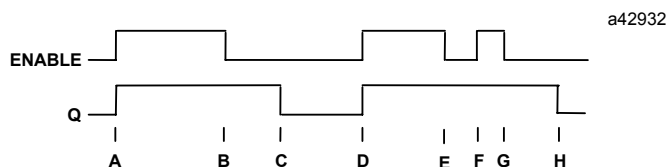
### Uwaga

Jeżeli w programie sterującym znajdują się dwa lub więcej przełączników czasowych o tych samych przypisanych zmiennych, to ich bieżące wartości będą takie same.

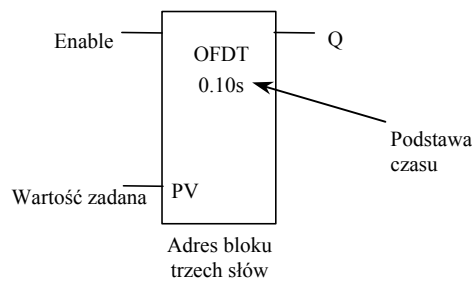
Przełącznik nie przesyła sygnału na wyjście, w przypadku gdy parametr PV jest równy zero lub ma wartość ujemną.

Jeżeli wykonany zostanie ten blok funkcyjny, a do wejścia Enable nie jest doprowadzany sygnał, bieżąca wartość zostaje zaktualizowana i będzie ona podawać czas, który upłynął od momentu zaprzestania doprowadzania sygnału do przełącznika. Jeżeli wartość bieżąca CV jest równa lub większa od wartości zadanej PV, zostaje wstrzymane przesyłanie sygnału na wyjście Q. Przełącznik zatrzymuje wtedy odliczanie czasu - proszę porównać z Częścią C wykresu zamieszczonego poniżej.

Przy ponownym doprowadzeniu sygnału, wartość bieżąca zostaje wyzerowana.



- A = Na wejściu Enable pojawia się sygnał. Przełącznik zostaje wyzerowany (CV=0), a na wyjściu również pojawia się sygnał.
- B = Na wejście ENABLE przestaje być podawany sygnał. Przełącznik rozpoczyna zliczanie czasu.
- C = Wartość bieżąca CV osiąga wartość zadaną PV. Na wyjście nie jest przesyłany sygnał. Przełącznik przerywa zliczanie czasu.
- D = Na wejściu Enable pojawia się sygnał. Przełącznik zostaje wyzerowany (CV=0), wyjście Q zostaje załączone.
- E = Na wejście ENABLE przestaje być podawany sygnał. Przełącznik rozpoczyna zliczanie czasu. Wyjście Q pozostaje załączone.
- F = Na wejściu Enable pojawia się sygnał. Przełącznik zostaje wyzerowany (CV=0), wyjście Q pozostaje załączone.
- G = Na wejście ENABLE przestaje być podawany sygnał. Przełącznik rozpoczyna zliczanie czasu. Wyjście Q pozostaje załączone.
- H = Wartość bieżąca CV osiąga wartość zadaną PV. Na wyjście nie jest przesyłany sygnał. Przełącznik przerywa zliczanie czasu.



Gdy przełącznik czasowy OFDT jest wykorzystywany w bloku programu, który **nie jest** wywoływany w każdym cyklu pracy sterownika, pomiędzy kolejnymi wywołaniami bloku programu zlicza on czas aż do momentu wyzerowania. Oznacza to, że pracuje on w programie jako przełącznik o znacznie dłuższym cyklu niż przełącznik w głównym bloku programu. Jeżeli więc blok programu będzie nieaktywny przez dłuższy czas, przełącznik powinien być tak zaprogramowany, aby skorzystać z tej jego cechy. Przykładowo, jeżeli przełącznik w bloku programu jest wyzerowany i ten blok programu nie jest wywoływany (jest nieaktywny) przez cztery minuty, po wywołaniu bloku, odliczone zostaną cztery minuty. Jest to czas, który upłynął od momentu doprowadzenia sygnału do przełącznika, o ile nie został on wcześniej wyzerowany.

## Parametry

Parametr	Opis
Adres bloku trzech słów	Przełącznik czasowy OFDT wykorzystuje trzy słowa (rejstry) pamięci typu %R, do zapamiętywania następujących parametrów: <ul style="list-style-type: none"> <li>• Wartości bieżącej (CV) = słowo 1.</li> <li>• Wartości zadanej (PV) = słowo 2.</li> <li>• Słowo sterujące = słowo 3.</li> </ul> <p>Bezpośrednio pod blokiem OFDT należy podać adres pierwszego z trzech słów (rejestrów).</p> <p><b>Ostrzeżenie:</b> Nie można wykorzystywać tych adresów przez inne funkcje. Wykorzystywanie tych adresów przez inne elementy logiczne powoduje zakłócenia w pracy przełącznika.</p>
Enable	W czasie gdy wejście Enable pozostaje załączone, również wyjście Q pozostaje załączone, a wartość chwilowa CV pozostaje zero. Po wyłączeniu wejścia Enable, rozpoczyna się zliczanie czasu. W momencie zrównania się wartości chwilowej CV z wartością zadaną PV, zliczanie czasu zostaje przerwane i wyjście Q zostaje wyłączone.
PV	Wartość zadana. Wartość PV jest kopiowana do drugiego rejestru (adres+1) w momencie zerowania lub uruchomienia przełącznika. Przełącznik wyłączy wyjście Q w momencie zliczenia czasu równemu wartości PV.
Q	Wyjście Q pozostaje załączone, w czasie gdy wejście Enable jest załączone, wartość chwilowa CV jest mniejsza niż wartość zadane PV.
Podstawa czasu	Wartość określająca dokładność zliczania czasu w dziesiątych (0.1), setnych (0.01) lub tysięcznych (0.001) częściach sekundy. Wartość podstawy czasu jest mnożona przez liczbę z wejścia PV, w celu określenia aktualnej wartości.

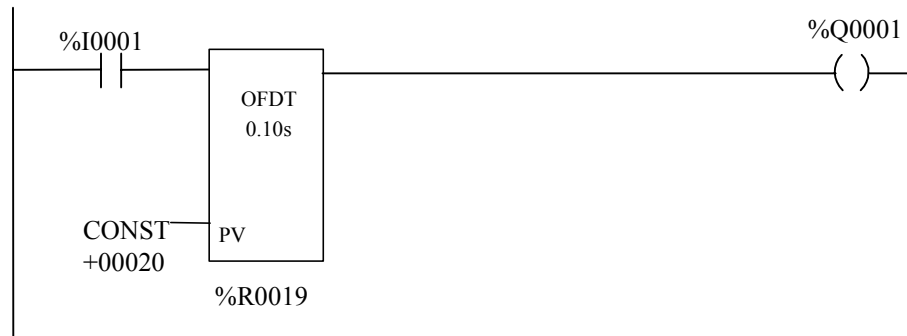
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
adres								•				
enable	•											
PV	•	•	•	•	•		•	•	•	•	•	•
Q	•											•

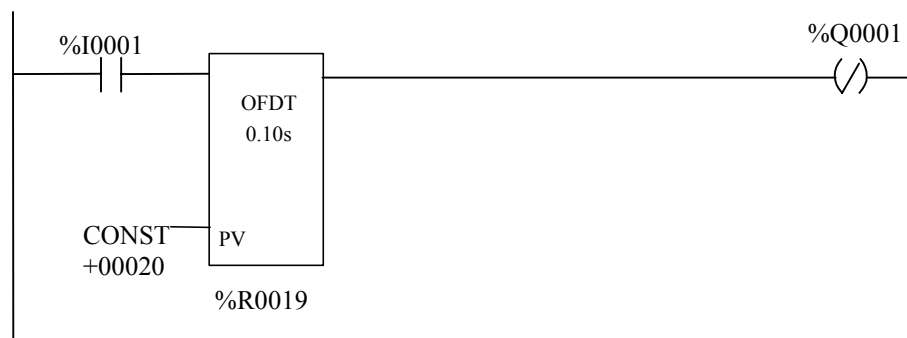
- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykłady

W poniższym przykładzie, po doprowadzeniu sygnału do styku %I0001 blok funkcyjny OFDT ustawia przełącznik wyjściowy %Q0001 na 0. Po zaprzestaniu doprowadzania sygnału do styku %I0001, przełącznik wyjściowy %Q0001 ma wartość 1 przez 2 sekundy, a następnie zostaje wyzerowany.



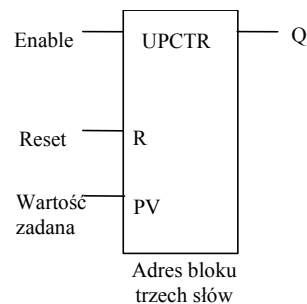
W tym przykładzie stan wyjścia jest zmieniany poprzez zastosowanie przełącznika o stykach zamkniętych. Zawsze, po doprowadzeniu sygnału do styku %I0001 blok funkcyjny OFDT ustawia przełącznik wyjściowy %Q0001 na 0. Po rozwarciu styku %I0001, przełącznik wyjściowy %Q0001 ma wartość 0 przez 2 sekundy, a następnie zostaje ustawiony na 1.



## UPCTR

Licznik zliczający w górę służy do zliczania impulsów sygnału od 0 do zadanej wartości. Zakres licznika wynosi od 0 do +32,767 impulsów. Pojawienie się na wejściu Reset powoduje wyzerowanie licznika. Zbocze narastające sygnału wejściowego (zmiana stanu sygnału wejściowego z 0 na 1) powoduje zwiększenie wartości bieżącej o 1. Sygnał wyjściowy jest wysyłany zawsze, gdy wartość bieżąca jest większa lub równa od wartości zadanej.

Wartość bieżąca licznika jest przechowywana w przypadku awarii zasilania sterownika, przy rozruchu nie następuje automatyczna inicjalizacja.



## Parametry

Parametr	Opis
Adres bloku trzech słów	<p>Licznik zliczający w górę UPCTR wykorzystuje trzy słowa (rejstry) pamięci typu %R, do zapamiętywania następujących parametrów:</p> <ul style="list-style-type: none"> <li>• Wartości bieżącej (CV) = słowo 1.</li> <li>• Wartości zadanej (PV) = słowo 2.</li> <li>• Słowo sterujące = słowo 3.</li> </ul> <p>Bezpośrednio pod blokiem UPCTR należy podać adres pierwszego z trzech słów (rejestrów).</p> <p><b>Ostrzeżenie:</b> Nie można wykorzystywać tych adresów przez inne funkcje. Wykorzystywanie tych adresów przez inne elementy logiczne powoduje zakłócenia w pracy przekaźnika.</p>
Enable	Każde zbocze narastające sygnału (przejścia z 0 na 1) powoduje wzrost wartości bieżącej (CV) o 1.
PV	Wartość zadana. Wartość zadana, kopiowana z wejścia PV do drugiego rejestru (adres+1) w momencie zerowania lub uruchomienia licznika. Licznik załączy wyjście Q w momencie zliczy wartość równą wartości PV. Jeśli wartość zadana jest stałą jej wartość musi zawierać się w przedziale od 0 32 767.
Q	Sygnał wyjściowy, pojawiający się, gdy wartość bieżąca (CV) jest większa lub równa od wartości zadanej (PV).
R	Sygnał zerujący. Po załączeniu wejścia R wartość bieżąca (CV) jest zerowana.

## Dopuszczalne typy zmiennych

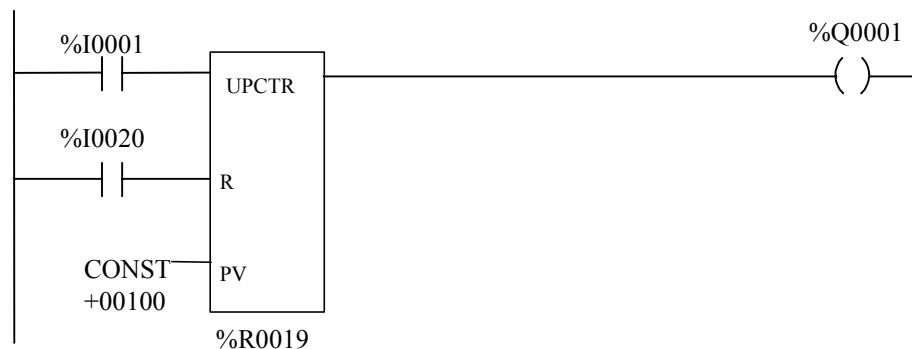
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
adres								•				
enable	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykłady

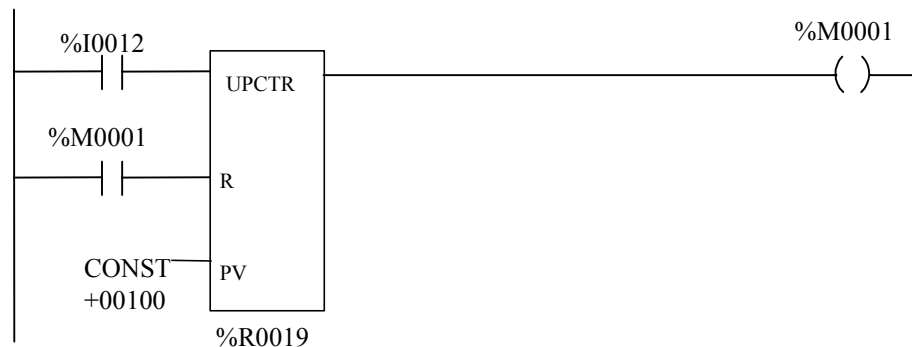
### Podstawowy obwód licznika

W poniższym przykładzie licznik UPCTR będzie zwiększał o jeden wartość bieżącą (CV) przy każdej zmianie wartości w rejestrze %I0001 z 0 na 1 na wejściu. Wejście PV ustawia wartość zadaną na 100. Po zliczeniu do 100 zostanie załączony przekaźnik %Q0001. Licznik będzie zliczał zmiany wartości na %I001 nawet po osiągnięciu wartości zadanej (100) dopóki nie osiągnie wartości maksymalnej (32 767) lub na wejściu R nie pojawi się sygnał zerujący licznik. Wyjście %Q0001 będzie załączone zawsze gdy wartość CV będzie równa lub większa od wartości PV.



### Obwód licznika samoczynnie zerujący się

W tym przykładzie, każde zbocze narastające sygnału wejściowego %I0012 powoduje zwiększenie wartości licznika o 1. Sygnał jest doprowadzany do wewnętrznego przekaźnika %M001 zawsze po odliczeniu 100 impulsów. Po załączeniu przekaźnika %M0001 wartość zliczona jest zerowana stykiem %M0001 na wejściu R.

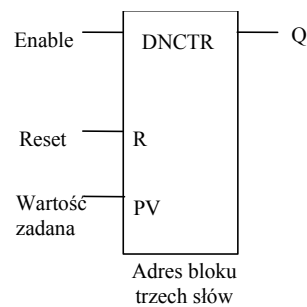




## DNCTR

Licznik zliczający w dół (DNCTR) służy do odliczania impulsów sygnału od zadanej wartości do 0. Minimalna wartość zadana może być równa zero, a maksymalna +32 767. Minimalna wartość bieżąca to –32 768 impulsów. Podczas zerowania bieżąca wartość licznika jest ustawiana na wartość zadaną PV. Zbocze narastające sygnału wejściowego (zmiana stanu sygnału wejściowego z 0 na 1) powoduje zmniejszenie wartości bieżącej o 1. Sygnał wyjściowy jest wysyłany, gdy wartość bieżąca jest większa lub równa zero.

Wartość bieżąca licznika jest przechowywana w przypadku awarii zasilania sterownika, przy rozruchu nie następuje automatyczna inicjalizacja.



## Parametry

Parametr	Opis
Adres bloku trzech słów	<p>Licznik zliczający w dół DNCTR wykorzystuje trzy słowa (rejstry) pamięci typu %R, do zapamiętywania następujących parametrów:</p> <ul style="list-style-type: none"> <li>• Wartości bieżącej (CV) = słowo 1.</li> <li>• Wartości zadanej (PV) = słowo 2.</li> <li>• Słowo sterujące = słowo 3.</li> </ul> <p>Bezpośrednio pod blokiem DNCTR należy podać adres pierwszego z trzech słów (rejestrów).</p> <p><b>Ostrzeżenie:</b> Nie można wykorzystywać tych adresów przez inne funkcje. Wykorzystywanie tego adresu przez inne elementy logiczne powoduje zakłócenia w pracy licznika.</p>
Enable	Każde zbocze narastające sygnału (przejścia z 0 na 1) powoduje zmniejszanie wartości bieżącej (CV) o 1.
PV	Wartość zadana. Wartość zadana, kopiowana do wejścia PV oraz wejścia CV w momencie zerowania lub uruchomienia licznika. Licznik załączy wyjście Q w momencie zliczenia od wartości zadanej do zera.
Q	Sygnał wyjściowy, pojawiający się, gdy wartość bieżąca jest mniejsza lub równa zero.
R	Sygnał zerujący. Po załączeniu wejścia R, wartość bieżąca (CV) jest ustawiana do wartości zadanej (PV) a wyjście Q zostaje wyłączone.

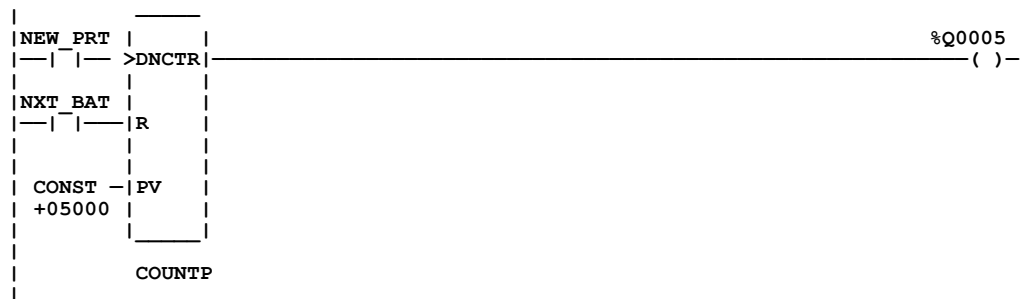
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
adres								.				
enable	.											
R	.											
PV		.	.	.	.		.	.	.	.	.	.
Q	.											.

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

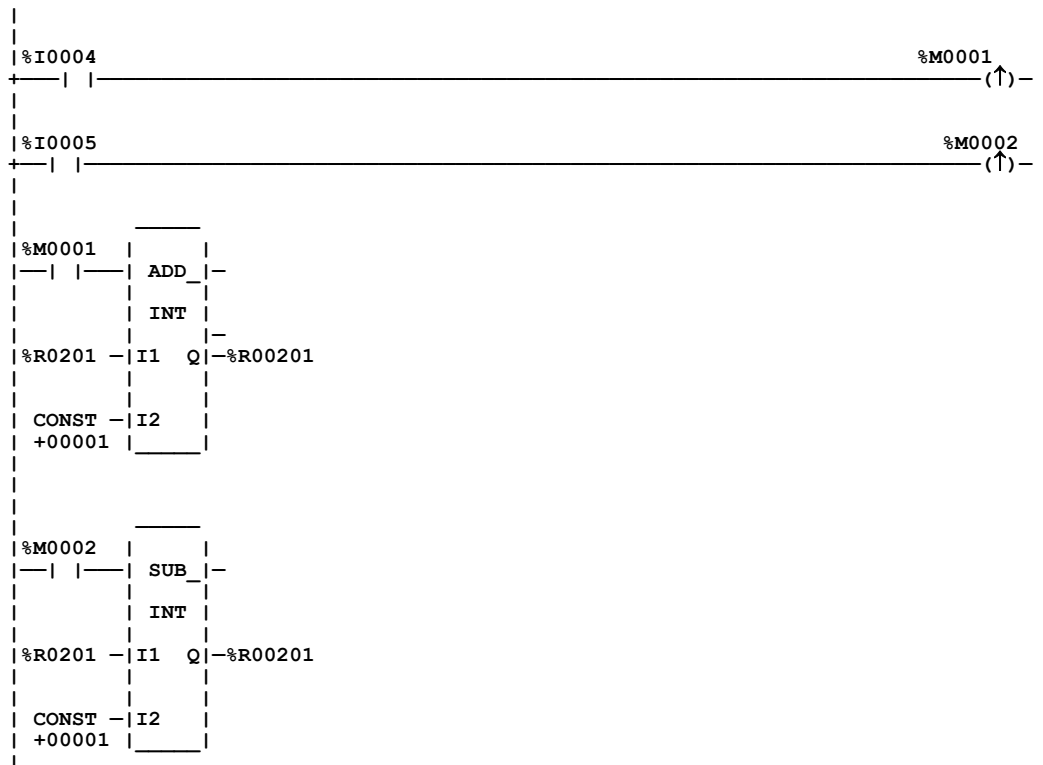
## Przykłady

W zamieszczonym poniżej przykładzie, licznik zliczający w dół COUNTP odlicza 5000 nowych impulsów przed ustawieniem wartości zmiennej %Q0005 na 1.





Poniżej zamieszczono drugą metodę rozwiązania takiego zadania wykorzystującą bloki ADD i SUB współdzielące rejestr %R00201. Zwieranie styku %I0004, powoduje wzrost wartości bloku ADD zapisywanej w rejestrze %R00201. Zwieranie styku %I0005, powoduje zmniejszanie wartości bloku SUB zapisywanej w rejestrze %R00201. W tym przypadku zmiana stanu styków jest wykorzystywana jako wejścia jednostopniowe funkcji ADD i SUB. Jeśli wejścia nie były by jednostopniowe to funkcje ADD i SUB były by wykonywane raz na cykl po ich załączeniu. (Zmiana stanu styków nie jest wymagana w przypadku funkcji UPCTR I DNCTR). W rozdziale 6 zamieszczono opis funkcji ADD I SUB.



W rozdziale tym opisano funkcje matematyczne, dostępne w zestawie instrukcji sterowników 90-30/20/Micro.

Oznaczenie skrótowe	Funkcja	Opis	Strona
ADD	Dodawanie	Dodawanie dwóch liczb.	6-1
SUB	Odejmowanie	Odejmowanie dwóch liczb	6-1
MUL	Mnożenie	Mnożenie dwóch liczb.	6-1
DIV	Dzielenie bez reszty	Część całkowita z dzielenia dwóch liczb.	6-1
MOD	Dzielenie modulo	Reszta z dzielenia dwóch liczb.	6-6
SQRT	Pierwiastek kwadratowy	Obliczanie pierwiastka kwadratowego z liczby całkowitej lub rzeczywistej.	6-8
SIN, COS, TAN, ASIN, ACOS, ATAN	Funkcje trygonometryczne†	Obliczenie odpowiedniej funkcji dla parametru będącego liczbą rzeczywistą, zadanego wejściu IN.	6-11
LOG, LN EXP, EXPT	Funkcje logarytmiczne/ wykładnicze †	Obliczenie odpowiedniej funkcji dla parametru będącego liczbą rzeczywistą, zadanego wejściu IN.	6-13
RAD, DEG	Konwersja wartości kąta	Obliczenie odpowiedniej funkcji dla parametru będącego liczbą rzeczywistą, zadanego wejściu IN.	6-15

† Funkcje trygonometryczne, logarytmiczne/ wykładnicze oraz funkcje do konwersji wartości kąta dostępne są wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 i CPU37x.

### Uwaga

Division and modulo division are similar functions that differ in their output; division finds a quotient, while modulo division finds a remainder. Funkcje DIV i MOD są podobne, różnica pomiędzy nimi polega na zwracanych wartościach. Funkcja DIV zwraca część całkowitą z dzielenia, a funkcja MOD resztę z dzielenia.

## Standardowe funkcje matematyczne (ADD, SUB, MUL, DIV)

Funkcje wykonują cztery podstawowe działania matematyczne: dodawanie, odejmowanie, mnożenie i dzielenie. Po doprowadzeniu sygnału do funkcji, wykonywane jest odpowiednie działanie matematyczne na dwóch liczbach I1 i I2, które są parametrami wejściowymi bloku funkcyjnego. Obydwa parametry wejściowe muszą być takiego samego typu. Parametr wyjściowy Q jest też tego samego typu.

### Reguły matematyczne

<b>Znak wyniku</b>	Podstawowe zasady matematyczne określają znak wyniku z działania.
<b>Dodawanie</b>	Instrukcja ADD wykorzystuje wzór $I1 + I2 = Q$ .
<b>Odejmowanie</b>	Instrukcja SUB wykorzystuje wzór $I1 - I2 = Q$ .
<b>Mnożenie</b>	Instrukcja MUL wykorzystuje wzór $I1 \times I2 = Q$ .
<b>Dzielenie bez reszty</b>	Instrukcja DIV wykorzystuje wzór $I1 \div I2 = Q$ . <b>Typ danych INT i DINT.</b> Funkcja DIV zaokrągla w dół iloraz z działania (reszta jest odrzucana) w przypadku liczb INT i DINT; wynik z działania nie jest zaokrąglany w dół. Na przykład, $53 / 5 = 10$ (reszta 3 jest odrzucana). <b>Typ danych REAL.</b> W przypadku liczb rzeczywistych funkcja zwraca ułamek dziesiętny.
<b>Dzielenie modulo</b>	Funkcja MOD może pracować na danych typu INT i DINT (dane typu REAL nie są obsługiwane). Funkcja MOD działa według wzoru $I1 \div I2 = Q$ . Jednakże funkcja ta zwraca tylko resztę z dzielenia a odrzuca iloraz. Na przykład, $53 / 5 = 3$ (iloraz 10 jest odrzucony).

### Typy danych standardowych funkcji matematycznych

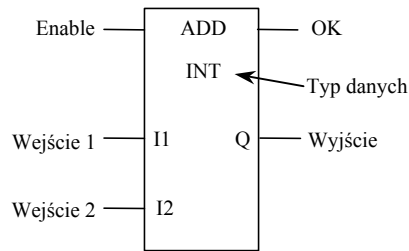
Po wprowadzeniu bloku funkcyjnego należy określić typ danych, na których będą przeprowadzane działania. Typ danych zostanie wyświetlony pod nazwą funkcji (patrz następny przykład). Poniżej zamieszczono typy danych, na których mogą pracować funkcje matematyczne:

Typ	Opis
INT	Liczba całkowita ze znakiem (16 bitowa)
DINT	Liczba całkowita podwójnej precyzji ze znakiem (32 bitowa).
REAL*	Liczba rzeczywista

\* Operacje na liczbach rzeczywistych można realizować wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x.

Domyślny typ danych to liczba całkowita ze znakiem. Więcej informacji o typach danych podano w Rozdziale 2, punkcie 2 "Organizacja programu. Typy danych i typy zmiennych."

Jeśli wynik operacji przekracza dopuszczalny zakres wartości dla stosowanego typu danych, parametr wyjściowy przyjmuje największą dopuszczalną wartość, a sygnał wyjściowy nie jest przesyłany. Dla liczb ze znakiem, znak jest ustawiany tak, aby wskazywał kierunek przekroczenia zakresu. Jeśli wynik operacji mieści się w dopuszczalnym zakresie wartości, blok funkcyjny przesyła sygnał wyjściowy. W przypadku gdy parametry wejściowe są liczbami całkowitymi ze znakiem lub liczbami całkowitymi podwójnej precyzji, znak wyniku operacji zależy od znaków parametrów wejściowych I1 i I2.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
I1	Wartość stała lub adres zmiennej, będącej pierwszym parametrem wykonywanego działania matematycznego. (Parametr I1 znajduje się z lewej strony działania matematycznego, jak na przykład w I1- I2).
I2	Wartość stała lub adres zmiennej, będącej drugim parametrem wykonywanego działania matematycznego. (Parametr I2 znajduje się z prawej strony działania matematycznego, jak na przykład w I1- I2).
ok	Sygnal wyjściowy, wysyłany gdy działanie zostało poprawnie wykonane, jego wynik mieści się w dopuszczalnym przedziale wartości i nie ma próby wykonania działania niewykonalnego.
Q	Wynik działania.

## Dopuszczalne typy zmiennych

Parametr	sygnal	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu INT. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu DINT i REAL.
- † Symbol oznaczający, że wykorzystując oprogramowanie Logicmaster możliwe będzie wprowadzenie danych DINT z zakresu -32 768 od +32 767 Wykorzystując oprogramowanie VersaPro możliwe jest wprowadzanie danych z całego zakresu.

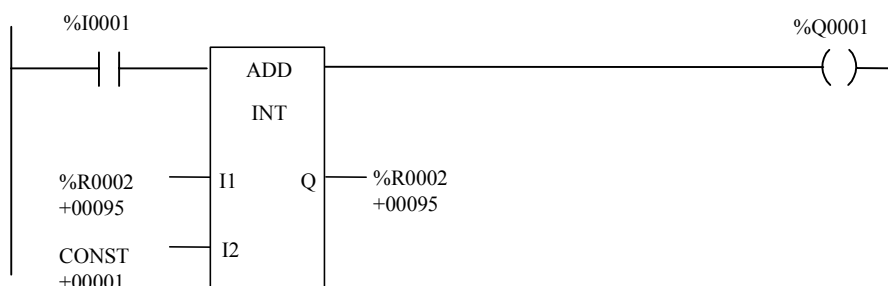
### Uwaga

Domyślnym typem danych dla funkcji operujących na liczbach 16 bitowych lub pojedynczych rejestrach jest typ INT. W celu zmiany tego typu na typ DINT, podwójne słowo o długości 32 bitów lub REAL (możliwość dostępna wyłącznie w jednostkach centralnych 35x, 36x i 37x), należy wcisnąć **F10**. Wartości typu INT zajmują pojedynczy, 16 bitowy rejestr, %R, %AI lub %AQ. Wartości typu DINT zajmują dwa sąsiadujące ze sobą rejestry, w pierwszym z nich przechowywane jest 16 młodszych bitów, a w drugim 16 starszych bitów. W jednostkach centralnych serii 35x i 36x (wersja 9 lub nowsza), oraz wszystkich wersjach jednostek 352 i 374 wartości typu REAL zajmują również dwa rejestry, w sumie 32 bity, znak przechowywany jest w najstarszym bicie, a następnie pamiętane są wykładnik i podstawa.

## Przykładowy funkcji matematycznych

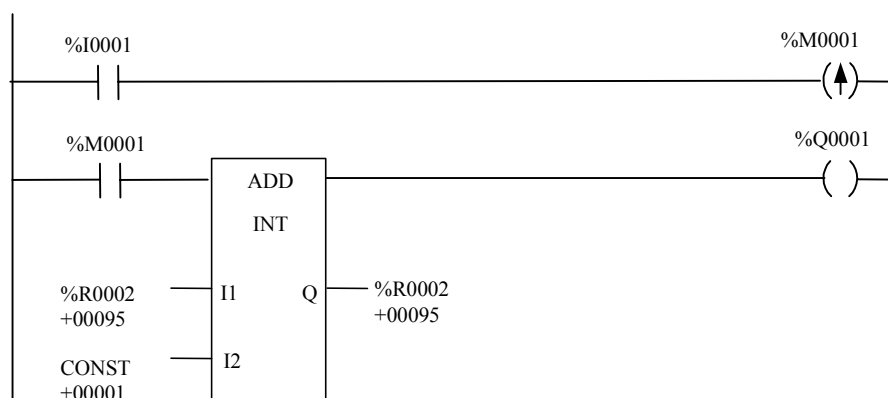
### Układ z blokiem ADD

Poniższy przykład przedstawia nieudaną próbę realizacji obwodu zliczającego ile razy został zamknięty przełącznik %I0001. Wynik jest przechowywany w rejestrze %R0002. W czasie projektowania tego obwodu założono, że po zamknięciu przełącznika %I0001, blok funkcyjny ADD będzie odczytywał wartość z rejestru %R0002 (wejście 2), inkrementował ją o 1 i zapisywał otrzymaną w ten sposób wartość z powrotem do tego rejestru. Przyczyną błędu jest fakt, że blok funkcyjny ADD będzie wykonywany w każdym cyklu pracy sterownika, w którym przełącznik %I0001 będzie zamknięty. Jeżeli więc, przykładowo, przełącznik %I0001 będzie zamknięty przez pięć cykli pracy jednostki centralnej, wynik zostanie zwiększony o pięć, nawet jeżeli przełącznik %I0001 zostanie w tym czasie zamknięty tylko jeden raz. Problem ten zostanie rozwiązany, jeżeli do bloku funkcyjnego ADD będzie doprowadzony sygnał wejściowy z przekaźnika uaktywnianego zboczem sygnału. Rozwiązanie takie przedstawiono poniżej.



W poniższym obwodzie, przełącznik %I0001 steruje przekaźnikiem uaktywnianym zboczem sygnału %M0001, który z kolei dostarcza sygnał wejściowy do bloku funkcyjnego ADD tylko przez jeden cykl pracy sterownika. Sygnał ten jest przesyłany po każdym zwarceniu przełącznika %I0001. Warunkiem ponownego zwarcia styków przekaźnika %M0001 jest rozwarcie i ponowne zwarcie styków przełącznika %I0001.

### Prawidłowy układ





## Funkcje matematyczne i typy danych

Funkcja	Działanie	Format wyświetlania
ADD INT	$Q(16 \text{ bit}) = I1(16 \text{ bit}) + I2(16 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 5 cyfrowej
ADD DINT	$Q(32 \text{ bit}) = I1(32 \text{ bit}) + I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 8 cyfrowej
ADD REAL*	$Q(32 \text{ bit}) = I1(32 \text{ bit}) + I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem i częścią dziesiętną o podstawie 7 cyfrowej
SUB INT	$Q(16 \text{ bit}) = I1(16 \text{ bit}) - I2(16 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 5 cyfrowej
SUB DINT	$Q(32 \text{ bit}) = I1(32 \text{ bit}) - I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 8 cyfrowej
SUB REAL*	$Q(32 \text{ bit}) = I1(32 \text{ bit}) - I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem i częścią dziesiętną o podstawie 7 cyfrowej
MUL INT	$Q(16 \text{ bit}) = I1(16 \text{ bit}) * I2(16 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 5 cyfrowej
MUL DINT	$Q(32 \text{ bit}) = I1(32 \text{ bit}) * I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 8 cyfrowej
MUL REAL*	$Q(32 \text{ bit}) = I1(32 \text{ bit}) * I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem i częścią dziesiętną o podstawie 7 cyfrowej
DIV INT	$Q(16 \text{ bit}) = I1(16 \text{ bit}) / I2(16 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 5 cyfrowej
DIV DINT	$Q(32 \text{ bit}) = I1(32 \text{ bit}) / I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem o podstawie 8 cyfrowej
DIV REAL*	$Q(32 \text{ bit}) = I1(32 \text{ bit}) / I2(32 \text{ bit})$	Liczba 10 cyfrowa ze znakiem i częścią dziesiętną o podstawie 7 cyfrowej

\* dotyczy wyłącznie jednostek centralnych 35x i 36x, wersja 9 lub nowsza oraz wszystkich wersji jednostek centralnych CPU352 i CPU37x

### Uwaga

Typy danych wejściowych i wyjściowych nie muszą być takie same. Sterowniki serii 90-30 nie obsługują funkcji MUL i DIV na różnych typach danych. Na przykład, wynikiem działania MUL INT na dwóch liczbach 16 bitowych jest liczba 16 bitowa (nie 32 bitowa). Aby w wyniku działania funkcji MUL DINT otrzymać liczbę 32 bitową, obydwa parametry wejściowe muszą być liczbami 32 bitowymi. W wyniku działania funkcji DIV INT, której parametrami są liczby 16 bitowe otrzymywana jest liczba 16 bitowa, natomiast w wyniku działania funkcji DIV DINT, której parametrami są liczby 32 bitowe otrzymywana jest liczba 32 bitowa.

Funkcje te przesyłają sygnał wyjściowy, pod warunkiem, że nie nastąpi przekroczenie dopuszczalnego zakresu wartości. W przypadku przekroczenia dopuszczalnego zakresu wartości, wynik jest równy najwyższej, dopuszczalnej wartości i posiada poprawny znak, ale nie jest wysyłany sygnał wyjściowy.

Należy zwrócić uwagę, aby przy stosowaniu funkcji MUL i DIV nie nastąpiło przekroczenie dopuszczalnego zakresu wartości. Przy konwertowaniu typu INT na DINT należy pamiętać, że jednostka centralna korzysta ze standardowego formatu liczby: dopełnienia do dwóch ze znakiem. Należy sprawdzić znak młodszego słowa 16 bitowego i zastosować go do drugiego słowa 16 bitowego. Jeżeli najbardziej znaczący bit w 16 bitowym słowie INT jest równy 0 (dodatni), należy przesunąć 0 do drugiego słowa. Jeżeli najbardziej znaczący bit w 16 bitowym słowie INT jest równy 1 (ujemny), należy przesunąć 1 do drugiego słowa. Konwersja typu DINT na INT jest łatwiejsza, ponieważ młodsze, 16 bitowe słowo jest częścią INT 32 bitowego słowa DINT. Starsze 16 bitów lub drugie słowo, powinny być równe 0 (wartość dodatnia) lub -1 (wartość ujemna), a jeżeli tak nie jest, liczba jest zbyt duża aby można ją było przekonwertować na liczbę 16 bitową.



## MOD (INT, DINT)

Funkcja Modulo (MOD) dzieli dwie liczby tego samego typu, a zwraca resztę z dzielenia. Znak liczby wynikowej jest zawsze taki sam jak znak parametru wejściowego I1.

Parametry funkcji MOD muszą mieć taki sam typ:

Typ	Opis
INT	Liczba całkowita ze znakiem (16 bitowa)
DINT	Liczba całkowita podwójnej precyzji ze znakiem (32 bitowa).

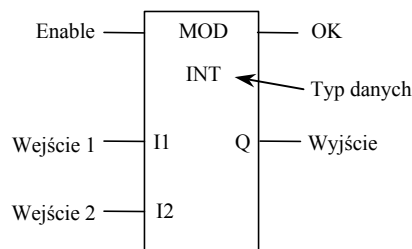
Domyślnym typem danych (wybieranym automatycznie przez oprogramowanie) jest INT, jednakże może on zostać zmieniony po dokonaniu wyboru bloku funkcyjnego i wprowadzeniu go do szczebla drabiny logicznej programu sterującego. Więcej informacji o typach danych podano w Rozdziale 2, części 2 "Organizacja programu. Typy danych i typy zmiennych."

Doprowadzenie sygnału do bloku funkcyjnego powoduje podzielenie parametru wejściowego I1 przez I2. Obydwa parametry wejściowe muszą być takiego samego typu. Wynik (parametr Q) jest obliczany według podanego poniżej wzoru:

$$Q = I1 - ([I1 \text{ DIV } I2] * I2)$$

gdzie wynikiem dzielenia DIV jest liczba całkowita. Parametr wyjściowy Q reprezentuje ten sam typ danych, co parametry wejściowe I1 oraz I2.

Sygnal wyjściowy o wartości 1 jest przesyłany zawsze po otrzymaniu sygnału wejściowego, chyba że nastąpi próba dzielenia przez zero. W takim wypadku sygnał wyjściowy nie jest przesyłany.



## Parametry

Parametr	Opis
Enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
I1	I1 - wartość stała lub adres zmiennej, będącej dzielną.
I2	I2 - wartość stała lub adres zmiennej, będącej dzielnikiem.
OK	Sygnal wyjściowy, wysyłany gdy działanie zostało poprawnie wykonane, a jego wynik mieści się w dopuszczalnym przedziale wartości.
Q	Na wyjściu Q dostępna jest reszta z dzielenia. If the value in I2 is an even multiple of I1, output Q will be zero, indicating no remainder.

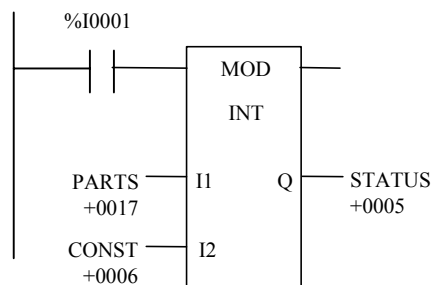
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

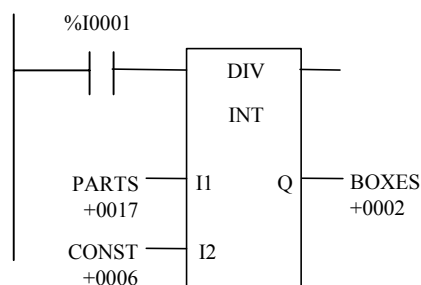
- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Dopuszczalne jest stosowanie wyłącznie zmiennych typu INT, nie można stosować zmiennych typu DINT.
- † Oznacza, że dla operacji wykonywanych na zmiennych typu DINT wartości stałe są ograniczone do przedziału <-32768, +32767>.

## Przykład

W poniższym przykładzie, pudełka są automatycznie wypełniane częściami. W jednym pudełku znajduje się sześć części. Obwód ten określa wypełnienie bieżącego pudełka wykorzystując funkcję dzielenia modulo. Uaktywnienie funkcji MOD powoduje dzielenie wartości z rejestru PARTS, zawierającego ilość wytworzonych części przez sześć. Wyjście STATUS bloku MOD określa ile części (od 1 do 5) zostało zapakowane do bieżącego pudełka. Po załadowaniu bieżącego pudełka wyjście Q równa się zero; jeśli bieżące pudełko jest tylko częściowo napełnione wyjście Q będzie informowało o ilości części znajdujących się w pudełku. W przykładzie tym ilość części wyprodukowanych wynosi 17, a w pudełku znajduje się 5 części. (Pozostałe 12 części zapełni dwa pudełka).



Aby określić ilość wypełnionych pudełek, wykorzystany zostanie instrukcja DIV.



Jednym możliwym problemem związanym z zastosowaniem tych bloków jest rejestr PARTS, w którym można przechować tylko 32 767 wpisów. Jeśli wymagane jest zliczanie wartości większych niż 32 767 należy (1) wyzerować rejestr PARTS zanim zostanie on zapełniony, (2) zapisać ilości zapełnionych opakowań przed wyzerowaniem rejestru PARTS, (3) wyzerować rejestr PARTS w momencie gdy rejestr STATUS będzie miał wartość zero co pozwala stwierdzić że odliczanie jest prawidłowe.

## SQRT (INT, DINT, REAL)

Blok funkcyjny SQRT wylicza pierwiastek kwadratowy z liczby. Gdy do bloku dociera sygnał, parametr wyjściowy Q przyjmuje wartość równą części całkowitej pierwiastka z liczby zadanej parametrem wejściowym IN. Wyjście Q musi zawierać ten sam typ danych co parametr wejściowy IN.

Funkcja SQRT operuje na następujących typach danych:

Typ	Opis
INT	Liczba całkowita ze znakiem (16 bitowa)
DINT	Liczba całkowita podwójnej precyzji ze znakiem (32 bitowa).
REAL	Liczba rzeczywista

W przypadku danych typu INT i DINT, tylko część całkowita wyniku działania funkcji będzie wyświetlana. Część ułamkowa zostanie odrzucona. Na przykład pierwiastek kwadratowy z liczby 2 lub 3 będzie równy 1, a z liczb 5, 6, 7 będzie równy 2.

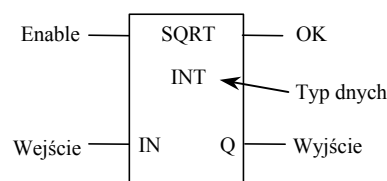
### Uwaga

Operacje na liczbach rzeczywistych można realizować wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x.

Domyślnym typem danych (wybieranym automatycznie przez oprogramowanie) jest INT, jednakże może on zostać zmieniony po dokonaniu wyboru bloku funkcyjnego i wprowadzeniu go do szczebla drabiny logicznej programu sterującego. Więcej informacji o typach danych podano w Rozdziale 2, części 2 "Organizacja programu. Typy danych i typy zmiennych."

Sygnał OK jest ustawiany w momencie prawidłowego wykonania operacji. Sygnał OK nie zostanie ustawiony jeśli wystąpi jedno z poniższych:

- $IN < 0$
- IN is NaN (Not a Number)



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
IN	Wartość stała lub adres zmiennej, dla której obliczany jest pierwiastek kwadratowy. Jeżeli parametr IN jest mniejszy od zera, funkcja nie powoduje przesłania sygnału Q.
ok	Sygnał wyjściowy, wysyłany gdy działanie zostało poprawnie wykonane, jego wynik mieści się w dopuszczalnym przedziale wartości i nie ma próby wykonania działania niewykonalnego.
Q	Wyjście Q zawiera pierwiastek kwadratowy obliczony z wejścia IN. Jednak w przypadku danych INT i DINT wyjście zawiera tylko część całkowitą wyniku działania, część ułamkowa jest odrzucana.

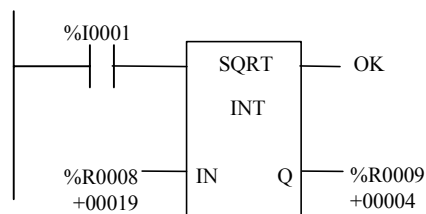
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

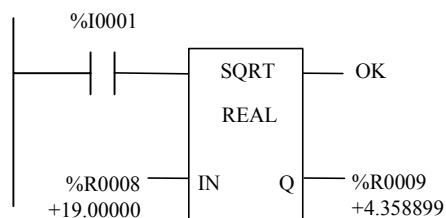
- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu INT. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu DINT i REAL.
- † Oznacza, że dla operacji wykonywanych na zmiennych typu DINT wartości stałe są ograniczone do przedziału <-32768, +32767>.

## Przykłady

W zamieszczonym poniżej przykładzie, zawsze gdy zmienna %I0001 jest równa 1, obliczany będzie pierwiastek kwadratowy z liczby %R0008, a wynik zapisywany będzie do zmiennej %R0009.



Alternatywnym rozwiązaniem dającym dokładniejszy wynik jest zastosowanie funkcji REAL –SQRT.



## Funkcje trygonometryczne (SIN, COS, TAN, ASIN, ACOS, ATAN)

Funkcje SIN, COS i TAN służą do obliczania odpowiednio sinusa, cosinusa i tangensa dla zadanej liczby. Po doprowadzeniu sygnału do jednej z tych funkcji, obliczany jest sinus (cosinus lub tangens) dla liczby podanej jako parametr wejściowy IN, a wynik zapisywany jest w zmiennej wyjściowej Q. Zarówno liczby IN i Q są zmiennoprzecinkowe.

Funkcje ASIN, ACOS i ATAN służą do obliczania odpowiednio arcus sinus, arcus cosinus i arcus tangens dla zadanej liczby. Po doprowadzeniu sygnału do jednej z tych funkcji, obliczana jest odpowiednia funkcja dla liczby podanej jako parametr wejściowy IN, a wynik w radianach zapisywany jest w zmiennej wyjściowej Q. Zarówno parametr IN jak i Q są liczbami rzeczywistymi.

Funkcje SIN, COS i TAN dopuszczają szeroki zakres wartości wejściowych:  $-2^{63} < IN < +2^{63}$ , ( $2^{63} \approx 9,22 \times 10^{18}$ ).

Funkcje ASIN oraz ACOS dopuszczają wąski zakres danych wejściowych ( $-1 \leq IN \leq 1$ ). Podając jako parametr wejściowy prawidłową liczbę funkcja zwróci wartość Q taką, że:

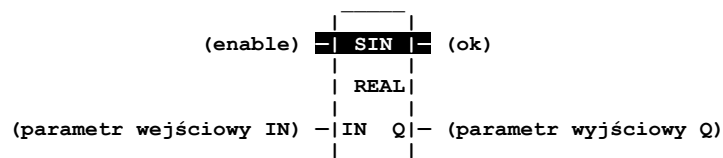
$$\text{ASIN (IN)} = \quad - \frac{\pi}{2} \leq Q \leq \frac{\pi}{2}$$

Funkcja ACOS\_REAL zwraca przez parametr Q wynik spełniający podaną poniżej zależność:

$$\text{ACOS (IN)} = \quad 0 \leq Q \leq \pi$$

Funkcja ATAN dopuszcza szeroki zakres wartości wejściowych, a mianowicie  $-\infty \leq IN \leq +\infty$ . Po wprowadzeniu poprawnego parametru IN, funkcja ATAN\_REAL zwraca przez parametr Q wynik spełniający podaną poniżej zależność:

$$\text{ATAN (IN)} = \quad - \frac{\pi}{2} \leq Q \leq \frac{\pi}{2}$$



### Uwaga

Funkcje trygonometryczne dostępne są wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 i CPU37x.

## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
IN	Wartość stała lub adres zmiennej, będącej parametrem wejściowym funkcji.
ok	Sygnal wyjściowy, wysyłany gdy działanie zostało poprawnie wykonane, jego wynik mieści się w dopuszczalnym przedziale wartości, parametr wejściowy ma odpowiednią wartość i jest liczbą.
Q	Wynik działania - wartość obliczona przez funkcje trygonometryczną dla parametru wejściowego IN.

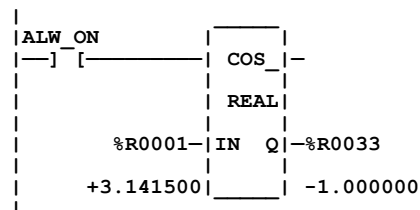
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN								•	•	•	•	
ok	•											•
Q								•	•	•		

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład

W zamieszczonym poniżej przykładzie obliczany jest cosinus dla wartości podanej w %R0001, a wynik zwracany jest poprzez zmienną %R0033.



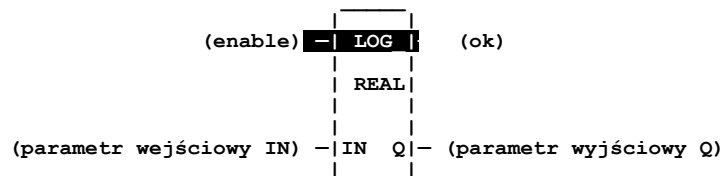


## Funkcje logarytmiczne/wykładnicze (LOG, LN, EXP, EXPT)

Funkcje LOG, LN i EXP posiadają dwa parametry wejściowe i dwa parametry wyjściowe. Doprowadzenie sygnału do tej funkcji powoduje wykonanie odpowiedniej operacji logarytmicznej/wykładniczej na parametrze wejściowym typu INT i zapisanie wyniku do parametru wyjściowego Q

- Dla funkcji LOG, parametr wyjściowy Q jest równy logarytmowi o podstawie dziesiętnej z liczby IN.
- Dla funkcji LN, parametr wyjściowy Q jest równy logarytmowi naturalnemu z liczby IN.
- Dla funkcji EXP, liczba  $e$  podnoszona jest do potęgi zadanej parametrem IN, a wynik zapisywany jest do parametru Q. (UWAGA:  $e$  jest stałą wykorzystywaną w obliczeniach logarytmicznych. W przybliżeniu równą 2,71828).
- Dla funkcji EXPT wartość parametru wejściowego I1 jest podnoszony do potęgi określonej za pomocą parametru I2 a wynik jest zapisywany do parametru Q.

Jeżeli parametr IN jest liczbą i ma wartość dodatnią, na wyjście OK przesyłany jest sygnał.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
IN	Wartość rzeczywista, będącej parametrem wejściowym funkcji.
ok	Sygnał wyjściowy, wysyłany gdy działanie zostało poprawnie wykonane, jego wynik mieści się w dopuszczalnym przedziale wartości, parametr wejściowy ma odpowiednią wartość i jest liczbą dodatnią.
Q	Wynik działania - wartość obliczona przez funkcje logarytmiczną/wykładniczą dla parametru wejściowego IN.

### Uwaga

Funkcje LOG, LN, EXP i EXPT dostępne są wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich jednostkach centralnych CPU352 i CPU37x.

### Uwaga

Jeżeli parametr wejściowy IN funkcji EXP ma wartość nieskończenie małą, funkcja ta zwraca wartość 0, zgodnie z oczekiwaniem. W jednostce centralnej CPU352, funkcja *nie* przesyła sygnału wyjściowego. W sterownikach serii 90-30 funkcja przewodzi sygnał, pomimo że wyjście ma wartość 0. ( $-\infty$  z dzielenia wartości przez zero). W oprogramowaniu Logicmaster jest przedstawiane jako **–OVERFLOW**).

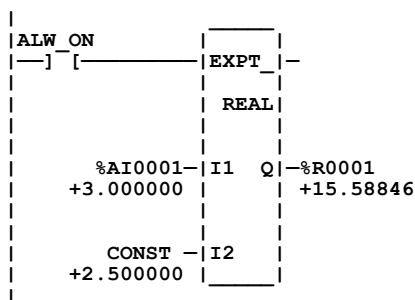
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN*								•	•	•	•	
ok	•											•
Q								•	•	•		
I1*								•	•	•	•	
I2*								•	•	•	•	

- \* W przypadku funkcji EXPT, parametr wejściowy In jest zastępowany przez parametry wejściowe I1 i I2.
- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład

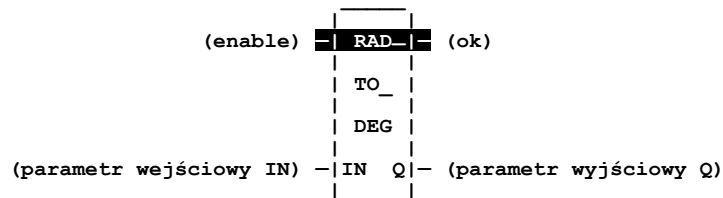
W zamieszczonym poniżej przykładzie wynik 15,58846 z podniesienia wartości %AI0001, 3000000 do potęgi +2,500000 jest zapisywany w %R0001.



## Konwersja miar kąta (RAD, DEG)

Po doprowadzeniu sygnału do tej funkcji, dla liczby podanej jako parametr wejściowy IN wykonywana jest odpowiednia konwersja wartości kąta (RAD\_TO\_DEG lub DEG\_TO\_RAD), tzn. z wartości w radianach na wartość w stopniach lub odwrotnie, a wynik zapisywany jest w wyjściu Q.

Jeżeli parametr IN jest liczbą, na wyjście OK przesyłany jest sygnał.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
IN	Wartość rzeczywista, będącej parametrem wejściowym funkcji.
ok	Sygnał wyjściowy, wysyłany gdy wynik działania mieści się w dopuszczalnym przedziale wartości i parametr IN jest liczbą.
Q	Wynik działania - wartość parametru wejściowego IN wyrażona w innych jednostkach miary kąta.

### Uwaga

Funkcje do konwersji miary kąta w radianach dostępne są wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 i CPU37x.

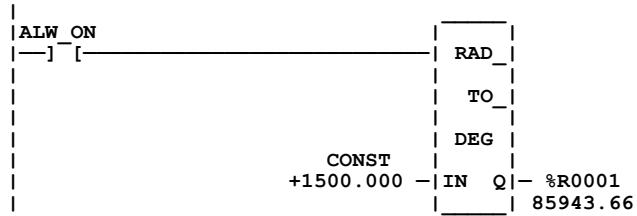
### Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN								•	•	•	•	
ok	•											•
Q								•	•	•		

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

### Przykład

W zamieszczonym poniżej przykładzie stała +1500 jest zamieniana na wartość w stopniach, a wynik przypisywany jest do zmiennej %R0001.



# Rozdział 7

## *Funkcje relacji*

Funkcje relacji stosowane są do porównywania dwóch wartości. W punkcie tym opisano następujące relacje:

Oznaczenie skrótowe	Funkcja	Opis	Strona
EQ	Równy	Sprawdzenie czy dwie liczby są równe.	7-2
NE	Różny	Sprawdzenie czy dwie liczby mają różne wartości.	7-2
GT	Większy	Sprawdzenie czy jedna liczba jest większa od drugiej.	7-2
GE	Większy lub równy	Sprawdzenie czy jedna liczba jest większa lub równa drugiej.	7-2
LT	Mniejszy	Sprawdzenie czy jedna liczba jest mniejsza od drugiej.	7-2
LE	Mniejszy lub równy	Sprawdzenie czy jedna liczba jest mniejsza lub równa drugiej.	7-2
RANGE	Przedział	Sprawdzenie czy liczba mieści się w zadanym przedziale (funkcja dostępna w jednostkach centralnych wersja 4.5 lub wyższe).	7-4

## Standardowe funkcje relacji (EQ, NE, GT, GE, LT, LE)

Doprowadzenie sygnału do sterującego powoduje porównanie parametru wejściowego I1 z parametrem wejściowym I2, parametry wejściowe muszą być tego typu. Relacje matematyczne wykonują operacje na następujących typach danych:

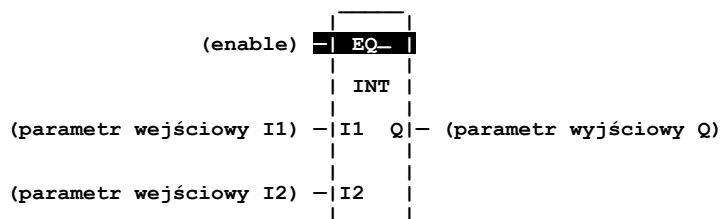
Typ	Opis
INT	Liczba całkowita ze znakiem (16 bitowa)
DINT	Liczba całkowita podwójnej precyzji ze znakiem (32 bitowa).
REAL	Liczba rzeczywista (nie dostępna z funkcją RANGE)

### Uwaga

Operacje na liczbach rzeczywistych można realizować wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x. Wartość bitu systemowego %S0020 jest ustawiana na 1 w prawidłowego wykonania funkcji na danych typu REAL. Jeżeli parametr wejściowy ma wartość NaN, bit ten jest zerowany. Funkcja RANGE nie pracuje z danymi typu REAL.

Domyślny typ danych to liczba całkowita ze znakiem. W celu porównania liczb całkowitych ze znakiem, liczb całkowitych podwójnej precyzji ze znakiem lub liczb rzeczywistych należy wybrać innych typ danych po wybraniu relacji. W celu porównania danych innych typów lub danych o różnych typach, należy je uprzednio przekonwertować za pomocą odpowiedniej funkcji (proszę porównać z Rozdziałem 11 "Funkcje konwersji").

Jeżeli parametry wejściowe I1 i I2 spełniają daną relację, wartość parametru wyjściowego Q jest ustawiana na 1; w innym przypadku jest ustawiona na 0.



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
I1	Wartość stała lub adres zmiennej, będącej pierwszym parametrem sprawdzanej relacji. (Parametr I1 znajduje się z lewej strony działania matematycznego, jak na przykład w $I1 < I2$ ).
I2	Wartość stała lub adres zmiennej, będącej drugim parametrem sprawdzanej relacji. (Parametr I2 znajduje się z prawej strony działania matematycznego, jak na przykład w $I1 < I2$ ).
Q	Sygnał wyjściowy, wysyłany jeżeli parametry I1 i I2 spełniają daną relację.

### Uwaga

I1 i I2 muszą być poprawnymi liczbami.

## Rozszerzony opis

Funkcja	Opis
Równy	Po doprowadzeniu sygnału do wejścia enable, jeżeli wartość I1 jest równa wartości I2, na wyjście Q wysyłany jest sygnał.
Różny	Po doprowadzeniu sygnału do wejścia enable, jeżeli wartość I1 NIE jest równa wartości I2, na wyjście Q wysyłany jest sygnał.
Większy	Po doprowadzeniu sygnału do wejścia enable, jeżeli wartość I1 jest większa od wartości I2, na wyjście Q wysyłany jest sygnał.
Większy lub równy	Po doprowadzeniu sygnału do wejścia enable, jeżeli wartość I1 jest większa lub równa wartości I2, na wyjście Q wysyłany jest sygnał.
Mniejszy	Po doprowadzeniu sygnału do wejścia enable, jeżeli wartość I1 jest mniejsza od wartości I2, na wyjście Q wysyłany jest sygnał.
Mniejszy lub równy	Po doprowadzeniu sygnału do wejścia enable, jeżeli wartość I1 jest mniejsza lub równa wartości I2, na wyjście Q wysyłany jest sygnał.

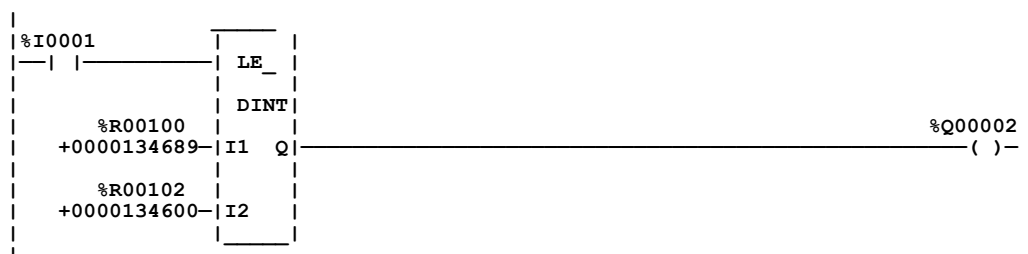
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
Q	•											•

- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu INT. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu DINT i REAL.
- † Symbol ten oznacza, że wartości stałych są ograniczone do przedziału od +32 767 do –32 768 w przypadku danych typu DINT i programowania w Logicmaster. Wykorzystując oprogramowanie VersaPro obsługiwany jest cały zakres danych typu DINT.

## Przykład

W poniższym przykładzie, dwie liczby typu DINT %R00100/101 i %R00102/103 są porównywane przy każdej zmianie wartości zmiennej %I0001 na 1. Jeśli liczba na wejściu I1 jest mniejsza lub równa liczbie na wejściu I2, cewka %Q00002 zostanie załączona. Cewka %Q00002 jest wyłączana jeśli liczba I1 jest większa niż liczba I2.



## RANGE (INT, DINT, WORD)

Funkcja RANGE służy do sprawdzania, czy dana liczba mieści się w przedziale wyznaczonym przez dwie inne liczby.

### Uwaga

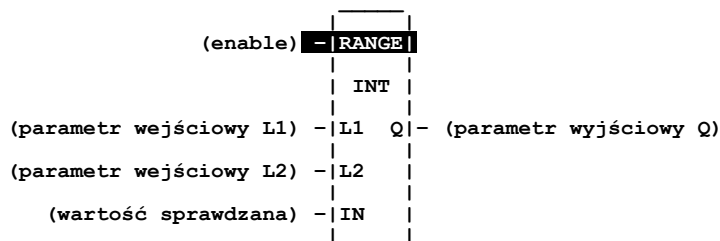
Funkcja ta dostępna jest wyłącznie w jednostkach centralnych wersja 4.41 i wyższe.

Funkcja RANGE obsługuje następujące typy danych (dane typu REAL nie są obsługiwane):

Typ	Opis
INT	Liczba całkowita ze znakiem (16 bitowa)
DINT	Liczba całkowita podwójnej precyzji ze znakiem (32 bitowa).
WORD	Dane typu Word (Słowo).

Domyślnym typem danych (wybieranym automatycznie przez oprogramowanie) jest INT, jednakże może on zostać zmieniony po dokonaniu wyboru bloku funkcyjnego i wprowadzeniu go do szczebla drabiny logicznej programu sterującego. Więcej informacji o typach danych podano w Rozdziale 2, części 2 "Organizacja programu. Typy danych i typy zmiennych."

Po doprowadzeniu sygnału do tego bloku funkcyjnego, następuje sprawdzenie, czy parametr wejściowy In mieści się w przedziale określonym przez wartości graniczne L1 i L2. Jeżeli jego wartość mieści się w przedziale domkniętym wyznaczonym przez L1 i L2, na wyjście Q przesyłany jest sygnał wyjściowy. W przeciwnym wypadku, sygnał wyjściowy nie jest przesyłany.



### Uwaga

Parametry L1 i L2 określają granice przedziału. Nie jest sprawdzane, który z tych parametrów określa lewą stronę przedziału, a który prawą. Tak więc jeżeli żądany przedział wynosi od 0 do 100, wartość 0 powinna być przypisana do L1, a wartość 100 do L2.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
L1	Lewy koniec przedziału.
L2	Prawy koniec przedziału.
IN	Wartość stała lub adres zmiennej, która ma się mieścić w określonym przedziale.
Q	Sygnal wyjściowy, wysyłany jeżeli parametr IN znajduje się wewnątrz przedziału domkniętego, określonego przez L1 i L2.

## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
L1		o	o	o	o		o	•	•	•	•‡	
L2		o	o	o	o		o	•	•	•	•‡	
IN		o	o	o	o		o	•	•	•		
Q	•											•

- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu INT lub Word. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu DINT.
- ‡ oznacza, że dla operacji wykonywanych na zmiennych typu DINT wartości stałe są ograniczone do przedziału od -32 768 do +32 767.

## Przykład 1

W zamieszczonym poniżej przykładzie, sprawdzane jest, czy wartość %AI0001 mieści się w przedziale określonym przez stałe 0 i 100.

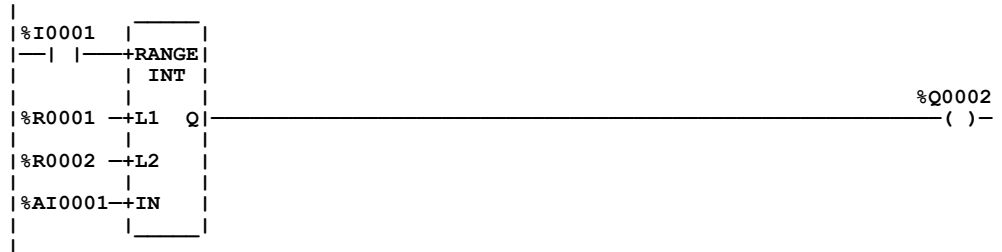


Tablica prawdy dla przykładu 1

Stan przekaźnika Enable %I0001	Wartość stałej L1	Wartość stałej L2	Wartość IN %AI0001	Stan Q %Q0001
1	100	0	< 0	0
1	100	0	0 — 100	1
1	100	0	> 100	0
0	100	0	Dowolna wartość	0

## Przykład 2

W zamieszczonym poniżej przykładzie, sprawdzane jest, czy wartość %AI0001 mieści się w przedziale określonym przez dwie wartości zapisane w rejestrach.



Tablica prawdy dla przykładu 2

Stan przełącznika Enable %I0001	Wartość L1 %R0001	Wartość L2 %R0002	Wartość IN %AI0001	Stan Q %Q0001
1	500	0	< 0	0
1	500	0	0 — 500	1
1	500	0	> 500	0
0	500	0	Dowolna wartość	0

Bloki funkcyjne z tej grupy wykonują operacje logiczne na ciągach bitów. Funkcje AND, OR, XOR i NOT wykonują operacje na pojedynczym słowie. Pozostałe funkcje z tej grupy mogą wykonywać działania na ciągu słów, długość takiego ciągu nie może przekraczać 256 słów. Wszystkie funkcje do operacji bitowych wymagają danych typu WORD.

Dane są wprawdzie pogrupowane w 16-bitowe słowa, lecz traktowane jako nieprzerwany ciąg bitów, z pierwszym bitem pierwszego słowa stanowiącym bit najbardziej znaczący (MSB), Ostatni bit ostatniego słowa stanowi bit najbardziej znaczący (MSB). Przykładowo, jeśli funkcja ma wykonać operację na trzech słowach o adresie początkowym %R0100, to wykona ją na 48 kolejnych bitach, traktowanych w sposób pokazany na poniższym schemacie.

%R0100	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	← bit 1 (LSB)
%R0101	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	
%R0102	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	
	↑																
	(MSB)																

### Uwaga

Zachodzące na siebie zakresy adresów parametrów wejściowych i wyjściowych w przypadku funkcji operujących na ciągach wielu słów bitowych mogą być przyczyną niespodziewanych rezultatów działania funkcji.

W rozdziale tym opisano następujące funkcje do operacji bitowych:

Oznaczenie skrótowe	Funkcja	Opis	Strona
AND	Logiczne AND	Jeżeli zarówno bit parametru wejściowego I1 jak i odpowiadający mu bit parametru wejściowego I2 mają wartość 1, wartość odpowiedniego bitu parametru wyjściowego Q jest ustawiana na 1.	8-3
OR	Logiczne OR	Jeżeli bit parametru wejściowego I1 ma wartość 1 i/lub odpowiadający mu bit parametru wejściowego I2 ma wartość 1, wartość odpowiedniego bitu parametru wyjściowego Q jest ustawiana na 1.	8-3
XOR	Alternatywa wyłączająca	Jeżeli bit parametru wejściowego I1 i odpowiadający mu bit parametru wejściowego I2 mają różne wartości, wartość odpowiedniego bitu parametru wyjściowego Q jest ustawiana na 1.	8-5
NOT	Negacja logiczna	Każdy bit w parametrze wyjściowym Q jest ustawiany na wartość przeciwną w stosunku do wartości odpowiedniego bitu parametru I1.	8-6
SHL	Przesunięcie w lewo	Przesunięcie wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w lewo, o wyszczególniona liczbę miejsc.	8-8
SHR	Przesunięcie w prawo	Przesunięcie wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w prawo, o wyszczególniona liczbę miejsc.	8-8
ROL	Przesunięcie w lewo w obiegu zamkniętym	Przesunięcie wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w lewo, o wyszczególniona liczbę miejsc, przy czym najbardziej znaczące bity (z lewej strony słowa) "wypchnięte" ze słowa bitowego zostają wpisane na puste miejsca z prawej strony słowa.	8-10
ROR	Przesunięcie w prawo w obiegu zamkniętym	Przesunięcie wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w prawo, o wyszczególniona liczbę miejsc, przy czym najmniej znaczące bity (z prawej strony słowa) "wypchnięte" ze słowa bitowego zostają wpisane na puste miejsca z lewej strony słowa.	8-57
BTST	Sprawdzanie wartości pojedynczego bitu	Sprawdzanie wartości (0 lub 1) jednego z bitów słowa bitowego.	8-12
BSET	Ustawianie wartości pojedynczego bitu na 1	Ustawianie wartości danego bitu w słowie bitowym na 1.	8-14
BCLR	Ustawianie wartości pojedynczego bitu na 0	Ustawienie wartości określonego bitu w ciągu bitowym na 0.	8-14
BPOS	Lokalizowanie pierwszego bitu o wartości 1	Przeszukiwanie słowa bitowego (lub ciągu słów) do napotkania pierwszego bitu o wartości równej 1.	8-16
MSKCMP	Porównanie z maskowaniem	Porównanie kolejnych bitów dwóch ciągów bitów, z możliwością maskowania wybranych bitów (funkcja dostępna w jednostkach centralnych 4.5 lub wyższych).	8-18

## AND i OR (WORD)

Za każdym razem, po doprowadzeniu sygnału, funkcje AND lub OR porównują każdy bit parametru I1 z odpowiednim bitem parametru I2, począwszy od najmniej znaczących bitów.

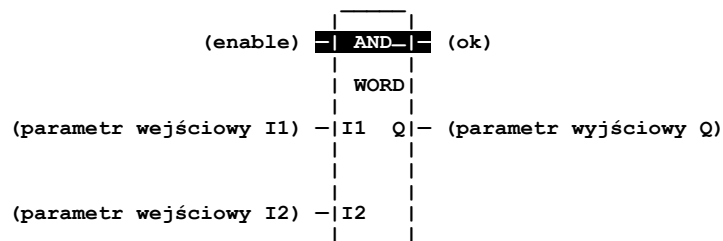
W przypadku funkcji AND, wartość każdego bitu parametru wyjściowego Q jest ustawiana na 1, jeśli jeden lub obydwa odpowiednie bity pierwszego i drugiego parametru wejściowego (tzn. słów I1 oraz I2) mają wartość 1.

Funkcja AND jest użyteczna do tworzenia masek, umożliwiających zmianę stanu (np. wyzerowanie) wyszczególnionych bitów pamięci lub wyzerowanie zadanego obszaru pamięci poprzez koniunkcję ze słowem zawierającym same zera. Ciągi bitów I1 i I2 mogą zajmować ten sam obszar pamięci.

W przypadku funkcji OR, wartość każdego bitu parametru wyjściowego Q jest ustawiana na 1, jeśli jeden lub obydwa odpowiednie bity pierwszego i drugiego parametru wejściowego (tzn. słów I1 oraz I2) mają wartość 1.

Funkcja OR jest stosowana np. do sterowania stanem wielu wyjść za pomocą jednej struktury logicznej. Blok funkcyjny OR jest równoważny układowi dwóch przełączników połączonych równolegle, zwielowokrotnionego tyle razy, ile bitów zawierają porównywane słowa. Może on być wykorzystywany do bezpośredniego sterowania lampkami sygnalizacyjnymi w zależności od stanów wejść lub ustanowienia wyższego priorytetu błyskania nad zapaleniem.

Sygnal wyjściowy jest przesyłany zawsze wtedy, gdy do bloku dopływa sygnał wejściowy.



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
I1	Wartość stała lub adres zmiennej, stanowiącej pierwsze słowo, na którym ma być wykonywana operacja logiczna.
I2	Wartość stała lub adres zmiennej, stanowiącej drugie słowo, na którym ma być wykonywana operacja logiczna.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
Q	Wynik działania.

## Dopuszczalne typy zmiennych

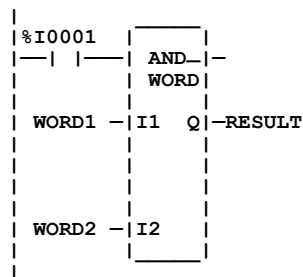
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

† można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

## Przykład

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %I0001 na 1 powoduje wykonanie operacji logicznej na dwóch parametrach o nazwach pomocniczych Word1 i Word2. Wynik działania funkcji AND zapisywany jest do parametru wyjściowego RESULT.



WORD1 (I1)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
WORD2 (I2)	1	1	0	1	1	1	0	0	0	0	0	0	1	1	1	1
RESULT (Q)	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0

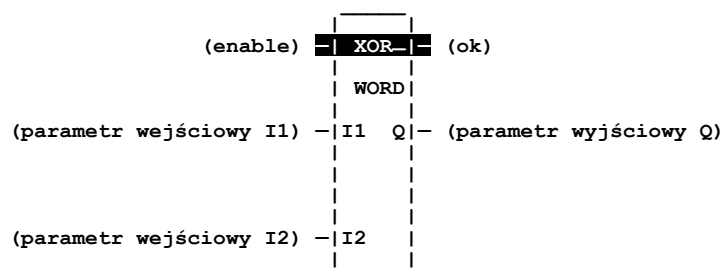
## XOR (WORD)

Funkcja XOR umożliwia porównanie poszczególnych bitów w słowach wejściowych I1 i I2. Wynikiem operacji jest bit o wartości 1, jeśli odpowiadające sobie bity mają różne wartości.

Funkcja XOR może być wykorzystana do porównania dwóch ciągów bitów lub do przełączania grupy bitów, z ustawianiem wartości 1 w co drugim cyklu pracy sterownika.

Po uaktywnieniu bloku funkcyjnego, wykonywane jest porównywanie bitów słowa I1 z odpowiednimi bitami słowa I2, rozpoczynając od najmniej znaczących bitów w każdym ze słów. Dla każdego sprawdzanych bitów, jeżeli jeden z nich posiada wartość logiczną 1, wtedy umieszczana jest 1 logiczna w odpowiednim miejscu słowa bitowego Q. Funkcja XOR przesyła sygnał zawsze wtedy, gdy do bloku dopływa sygnał wejściowy.

Jeśli słowa I2 oraz Q mają ten sam adres początkowy, bit słowa I1 o wartości 1 spowoduje migotanie (cykliczną zmianę wartości pomiędzy 1 a 0) odpowiedniego bitu słowa I2, zgodnie z częstotliwością wykonywania cykli pracy przez sterownik. Migotanie z mniejszą częstotliwością można wywołać, podając sygnał na wejście bloku funkcyjnego z częstotliwością równą podwójnej żądanej częstotliwości migotania (należy wykorzystywać styki przekaźników monostabilnych lub z obwodów samoczynnie zerujących się).



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
I1	Wartość stała lub adres zmiennej będącej pierwszym parametrem wykonywanej funkcji XOR.
I2	Wartość stała lub adres zmiennej będącej drugim parametrem wykonywanej funkcji XOR.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
Q	Wynik operacji logicznej XOR na parametrach I1 i I2.

## Dopuszczalne typy zmiennych

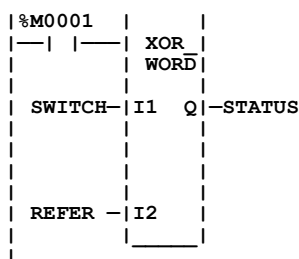
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

† Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

## Przykład obwodu alarmowego

W poniższym przykładzie przy każdym załączeniu cewki %M0001, 16-bitowe słowo „SWITCH” jest porównywane ze słowem odniesienia „REFER”. Słowo SWITCH jest grupą bitów przedstawiającą stan styków alarmowych. Słowo REFER reprezentuje stan normalny lub nie alarmowy tych bitów. Jeśli stan jakiegokolwiek bitu w słowie SWITCH jest inny niż stan odpowiedniego bitu słowa REFER, odpowiadający im bit w słowie wyjściowym Q zostanie ustawiony na 1. W warunkach normalnych (nie alarmowych) wartość słowa STATUS będzie równa zero.



Pozycja bitu	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
I1 (SWITCH)	0	1	0	1	1	1	1	0	1	1	0	0	0	0	0	0
I2 (REFER)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
Q (STATUS)	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0

Dane słowa STATUS mogą być wykorzystywane jako wejście funkcji NE (nie równy), która może porównać słowo STATUS z zerem. Jeśli STATUS nie jest równy zero, funkcja NE załącza wyjście informujące o obecności alarmu.

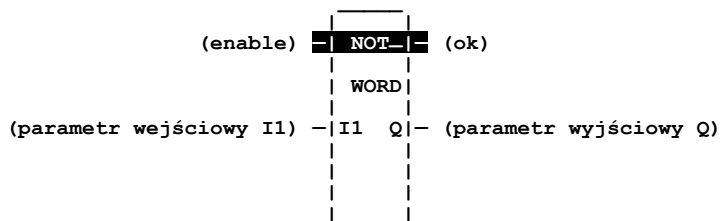
Bity o stanie logicznym 1 w słowie STATUS mogą zostać zidentyfikowane za pomocą funkcji BPOS (Bit Position), która może przeszukać słowo STATUS i określić położenie pierwszego bitu o stanie logicznym 1. W powyższym przykładzie funkcja BPOS zwróciłaby wartość 4, wskazując że czwarty bit ma wartość logiczną 1. Aby przeszukać kolejne bity należałoby zapisać wartość 4, po czym wyzerować bit 4 z pomocą funkcji BCLR (Bit clear) i powtórzyć przeszukiwanie słowa funkcją BPOS w celu odnalezienia kolejnego bitu 1 (9 w przykładzie). Proces ten może być powtarzany do momentu znalezienia wszystkich nie zerowych bitów. Funkcje BPOS oraz BCLR zostaną omówione w dalszej części tego rozdziału.



## NOT (WORD)

Funkcja negacji NOT jest wykorzystywana do zmiany stanu każdego bitu słowa, które jest parametrem wejściowym I1 bloku funkcyjnego, na przeciwny.

Funkcja NOT jest wykonywana i przepuszcza sygnał w każdym cyklu pracy, w którym zostanie uaktywniona.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji logicznej.
I1	Wartość stała lub adres zmiennej będącej parametrem wykonywanej funkcji NOT.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
Q	Wynik działania - negacja słowa I1.

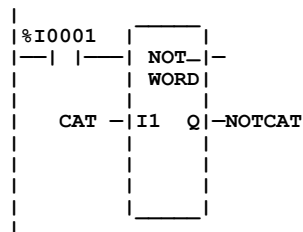
### Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
I1		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- † Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

### Przykład

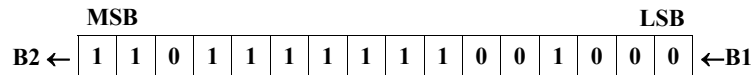
W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %I0001 na 1 powoduje ustawienie wszystkich bitów zmiennej NOTCAT na wartość przeciwną w stosunku do odpowiednich bitów zmiennej CAT.



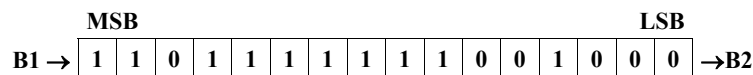
CAT	1	1	0	1	0	1	0	0	1	1	0	1	0	0	0	1
NOTCAT	0	0	1	0	1	0	1	1	0	0	1	0	1	1	1	0

## SHL i SHR (WORD)

Funkcję SHL można wykorzystać do przesunięcia wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w lewo, o wyszczególnioną liczbę miejsc. Wyższe bity (z lewej strony słowa) zostają "wypchnięte" ze słowa bitowego. Na puste miejsca zostają wpisane zadane wartości.



Funkcję SHR można wykorzystać do przesunięcia wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w prawo, o wyszczególnioną liczbę miejsc. Niższe bity (z prawej strony słowa) zostają "wypchnięte" ze słowa bitowego. Na puste miejsca zostają wpisane zadane wartości.



Każda z powyższych funkcji może wykonywać przesunięcie w zakresie od 1 do 256 słów bitowych.

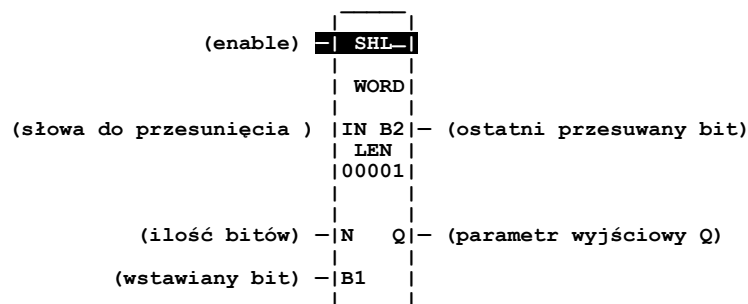
Jeżeli liczba miejsc, o jaką mają być przesunięte bity danego słowa (N) jest większa od liczby bitów zawartych w zadanym ciągu (LEN)\*16 lub jeżeli liczba ta jest równa 0, do ciągu wyjściowego (Q) kopiowany jest parametr wejściowy IN oraz bit wejściowy kopiowany jest do parametru wyjściowego B2, zawierającego wartość ostatniego bitu, który wyszedł poza zakres słowa po dokonaniu operacji przesunięcia. Jeżeli liczba miejsc, o jaką mają być przesunięte bity danego słowa jest równa zero, przesunięcie nie zostaje wykonane, ciąg wejściowy jest kopiowany do ciągu wyjściowego, a bit wejściowy (B1) jest kopiowany do parametru wyjściowego.

Bity przesuwane na początek słowa są określone przez parametr wejściowy B1. Jeżeli liczba miejsc, o którą mają być przesunięte bity danego słowa jest większa od 1, każdy z bitów ma ustawianą taką samą wartość (0 lub 1). Wejście B1 może być kontrolowane przez:

- Styk ALW\_ON (%S07), który ciągle utrzymuje B1 w stanie logicznym 1.
- Styk ALW\_OFF (%S07), który ciągle utrzymuje B1 w stanie logicznym 0.
- Styk z wewnętrznej cewki takiej jak %M lub %Q pozwalające użytkownikowi na zmianę wartości.
- Styk %I, pozwalający na zmianę wartości.

Funkcje SHL i SHR zawsze przesyłają sygnał wyjściowy, o ile liczba miejsc, o jaką mają być przesunięte bity danego słowa jest większa od zera.

Wynik, uzyskany po wykonaniu operacji przesunięcia jest kopiowany w miejsce wyszczególnione przez parametr Q. W celu uzyskania przesunięcia ciągu wejściowego parametr wyjściowy Q musi wykorzystywać ten sam obszar pamięci co parametr wejściowy IN. Funkcje SHL/SHR są wykonywane w każdym cyklu, w którym są uaktywnione. W parametrze wyjściowym B2 jest przetrzymywana wartość ostatniego przesuwanego bitu; na przykład, jeśli przesunięciu uległy cztery bity, B2 będzie zawierało wartość czwartego przesuwanego bitu.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie przesunięcia.
IN	Adres pierwszego słowa, którego bity mają zostać przesunięte.
N	Liczba miejsc (pozycji bitów), o które mają zostać przesunięte bity danego słowa.
B1	Wartość bitu (0 lub 1), które mają zostać wstawione w puste miejsca słowa, powstałe po przesunięciu jego zawartości.
B2	Wartość ostatniego bitu (0 lub 1), który został odrzucony.
Q	Adres pierwszego słowa ciągu słów, otrzymanego po przesunięciu bitów słowa adresowanego przez parametr IN.
LEN	Liczba słów bitowych (1-256) obszaru, w obrębie którego ma zostać dokonane przesunięcie bitów.

## Dopuszczalne typy zmiennych

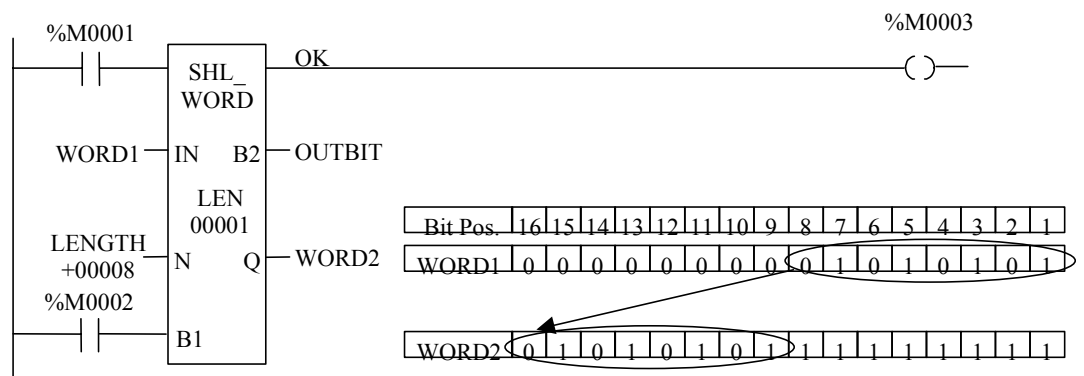
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•	•	•	•	•	•		
N		•	•	•	•		•	•	•	•	•	
B1	•											
B2	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

† Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

## Przykład

W poniższym przykładzie, załączenie wejścia %M0001, powoduje kopiowanie bitów słowa z wejścia IN (o nazwie WORD1) przez funkcję SHL. Później w skopiowanych bitach następuje przesunięcie w lewo o 8 pozycji (określone przez wartość N). Bity z pozycji 9-16 są odrzucane, a bity które znajdowały się na pozycjach 1-8 teraz zajmują miejsca 9-16. Pozycje bitów 1-8, które zostały "puste" po przesunięciu bitów 1-8, są wypełniane jedynkami, ponieważ w tym przykładzie styk %M0002 jest zwarty powodując zrównanie wejścia B1 z logiczną jedynką. W końcu słowa przesunięte/wypełnione są zapisywane pod adresem WORD2 na wyjściu Q. Pierwotne słowo WORD1 na wejściu IN nie zostaje zmienione. Wyjście B2 jest równe zero dopóki ostatnim odrzuconym była logiczne zero (bit znajdujący się na 9 pozycji), a cewka %M0003 pozostaje załączona ponieważ funkcja pracuje poprawnie.



## ROL i ROR (WORD)

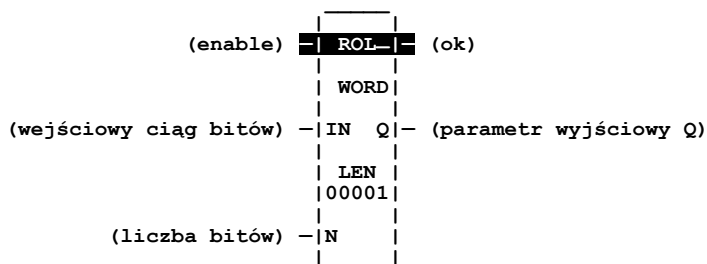
Funkcję ROL (przesunięcie bitowe w obiegu zamkniętym w lewo) można wykorzystać do przesunięcia wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w lewo, o wyszczególnioną liczbę miejsc. Najbardziej znaczące bity (z lewej strony słowa), "wypchnięte" ze słowa bitowego zostają wpisane na puste miejsca z prawej strony słowa.

Funkcję ROR (przesunięcie bitowe w obiegu zamkniętym w prawo) można wykorzystać do przesunięcia wszystkich bitów jednego słowa bitowego lub ciągu kilku słów bitowych w prawo, o wyszczególnioną liczbę miejsc. Najmniej znaczące bity (z prawej strony słowa), "wypchnięte" ze słowa bitowego, zostają wpisane na puste miejsca z lewej strony słowa.

Każda z powyższych funkcji może wykonywać przesunięcie w zakresie od 1 do 256 słów bitowych.

Jeżeli liczba miejsc (N) o jaką mają być przesunięte bity danego słowa jest równa Q, następuje przesunięcie słowa wejściowego (IN) do parametru wyjściowego (Q) i na wyjściu pojawia się sygnał potwierdzenia. W przeciwnym przypadku nie następuje wysyłanie sygnału potwierdzenia OK.

Wynik przesunięcia jest dostępny na wyjściu Q. W celu uzyskania przesunięcia ciągu wejściowego parametr wyjściowy Q musi wykorzystywać ten sam obszar pamięci co parametr wejściowy IN. Cały przesuwany ciąg jest zapisywany przy każdym podaniu zasilania.



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie rotacji bitów.
IN	Adres pierwszego słowa, którego bity mają zostać przesunięte.
N	Liczba miejsc (pozycji bitów), o które mają zostać przesunięte bity danego słowa.
ok	Sygnał OK jest załączany jeśli funkcja jest uaktywniona, liczba miejsc, o które ma nastąpić przesunięcie jest większa od zera, a mniejsza od długości słowa.
Q	Adres pierwszego słowa lub ciągu słów, otrzymanego po rotacji bitów słowa adresowanego przez parametr IN.
LEN	Liczba słów bitowych (1-256) obszaru, w obrębie którego ma zostać dokonane przesunięcie bitów.



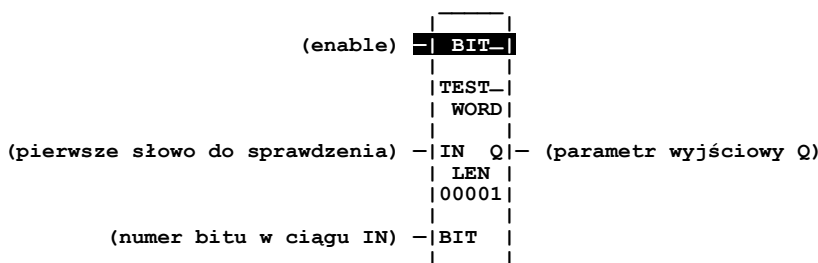
## BTST (WORD)

Funkcja Bit Test (Sprawdzanie wartości bitu) jest stosowana do określenia wartości (0 lub 1) jednego z bitów słowa bitowego. Wynik działania funkcji jest dostępny na wyjściu Q.

Jego wartość jest ustawiana zgodnie z wartością wyszczególnionego bitu sprawdzanego słowa. Parametr BIT może być również zmienną, co umożliwia sprawdzenie różnych bitów danego ciągu bitowego za pomocą tego samego bloku funkcyjnego. Jeśli wartość parametru BIT przekracza dopuszczalny zakres, wartość parametru wyjściowego jest ustawiana na 0.

Zakres:  $1 \leq \text{BIT} \leq (16 * \text{LEN})$

Funkcja ta może dokonać sprawdzenia wartości bitu w zakresie od 1 do 256 słów bitowych, czyli z ciągu liczącego maksymalnie  $16 * 256$  bitów.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie sprawdzania wartości.
IN	Adres pierwszego słowa ciągu słów, na którym ma zostać wykonana operacja.
BIT	Numer bitu słowa IN, którego wartość ma zostać określona. Zakres wartości to $(1 \leq \text{BIT} \leq (16 * \text{LEN}))$ .
Q	Wartość parametru Q zostaje ustawiona na 1, gdy wartość testowanego bitu również wynosi 1.
LEN	Liczba słów bitowych ciągu słów, z którego wybierany jest bit do sprawdzania wartości (maksymalnie 256).

### Uwaga

**W przypadku korzystania z funkcji Bit Test, Bit Set, Bit Clear lub Bit Pos**, bity numerowane są od 1 do 16, a **NIE** od 0 do 15

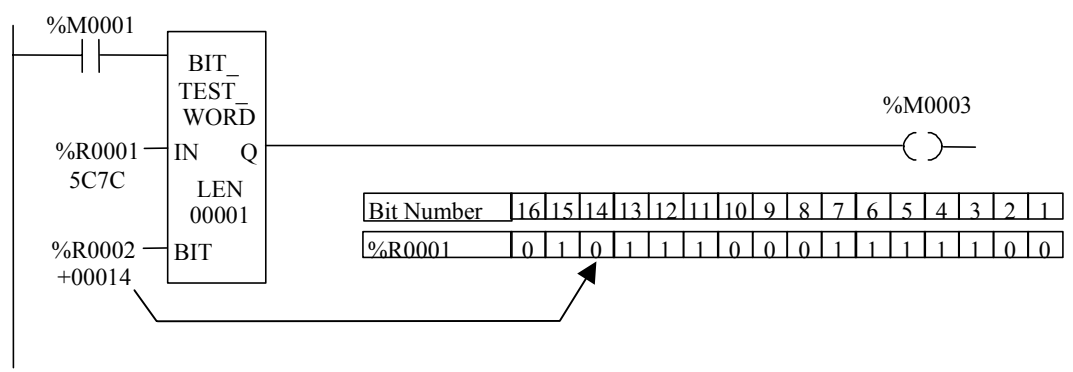
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•	•	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
Q	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład

W poniższym przykładzie załączenie styku %M0001 powoduje sprawdzenie bitu 14 w słowie %R0001 (bit 14 jest określony przez wartość w %R0002). Dopóki bit 14 będzie miał stan logiczny zero wartość w %R0002 (5C7C), wyjście nie zostanie załączone. Należy zauważyć, że funkcja może pracować tylko na danych typu WORD, dlatego adres wejścia IN będzie w oprogramowaniu Logicmaster wyświetlany w formacie szesnastkowym. Jednak wartość BIT będzie wyświetlana jako liczba całkowita niezależnie od tego czy jest stałą czy nie.

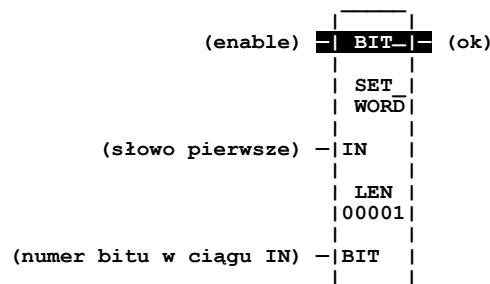


## BSET i BCLR (WORD)

Funkcja Bit Set (BSET) jest wykorzystywana do ustawiania wartości określonego bitu na 1 w ciągu bitów. Funkcja Bit Clear (BCLR) jest wykorzystywana do zerowania określonego bitu w ciągu bitów.

Każde doprowadzenie sygnału powoduje ustawienie wartości określonego bitu na 1, w przypadku funkcji BSET lub na 0, w przypadku funkcji BCLR. Parametr BIT może być również zmienną, co umożliwia ustawienie różnych bitów danego słowa bitowego za pomocą tego samego bloku funkcyjnego.

Funkcja ta może dokonać sprawdzenia wartości bitu w zakresie od 1 do 256 słów bitowych, czyli z ciągu liczącego maksymalnie  $16 \cdot 256$  bitów. Jeśli wyszczególniony numer bitu (BIT) przekracza dopuszczalny zakres ( $1 \leq \text{BIT} \leq [16 \cdot \text{LEN}]$ ), sygnał wyjściowy nie jest przesyłany. Wartość zmiennej ok ustawiana jest wtedy na 0. Na przykład, jeśli LEN ma wartość 1, wtedy długość sprawdzanego ciągu jest 16. Jeśli w tym przypadku liczbą na wejściu BIT byłaby 17 lub większa, wtedy była by ona poza zakresem, więc nie zostałyby załączone wyjście ok.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji.
IN	Adres pierwszego słowa w ciągu słów, na którym ma zostać wykonana operacja.
BIT	Numer bitu słowa IN, którego wartość ma zostać ustawiona na 1 lub 0. Zakres wartości: ( $1 \leq \text{BIT} \leq (16 \cdot \text{LEN})$ ).
ok	Sygnał ok pojawiający się po uaktywnieniu bloku, gdy wartość BIT jest poprawna.
LEN	liczba słów w ciągu słów, których adres jest określony w IN.

### Uwaga

**W przypadku korzystania z funkcji Bit Test, Bit Set, Bit Clear lub Bit Pos, bity numerowane są od 1 do 16, a NIE od 0 do 15**



## Dopuszczalne typy zmiennych

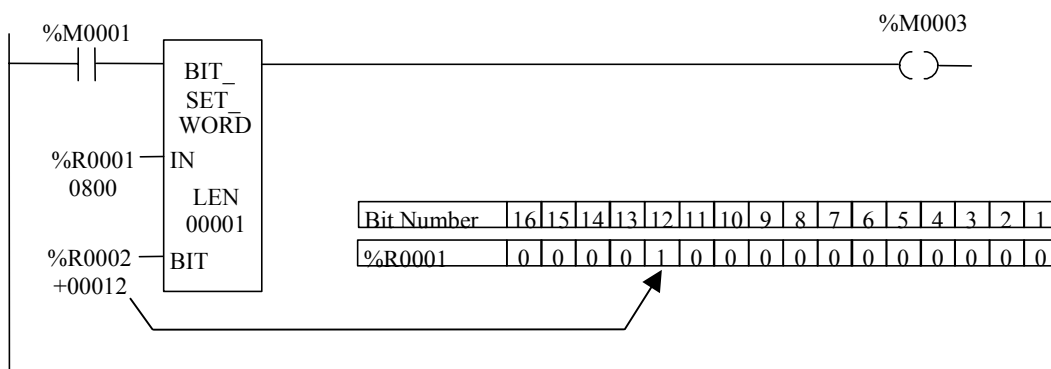
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•	†	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
ok	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- † Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

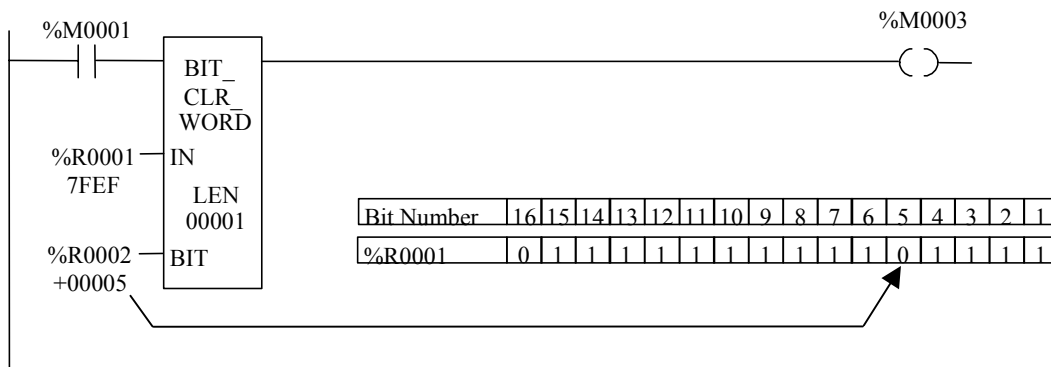
## Przykłady

Należy zauważyć, że funkcje Bit Set i Bit Clear mogą pracować tylko na danych typu WORD, dlatego adres wejścia IN będzie w oprogramowaniu LogiMaster wyświetlany w formacie szesnastkowym. Jednak wartość BIT będzie wyświetlana jako liczba całkowita niezależnie od tego czy jest stałą czy nie.

W poniższym przykładzie, załączenie %M0001 powoduje ustawienie bitu 12 (określonego przez BIT) w rejestrze %R0001 (na wejściu IN) jest ustawiana na 1.



W poniższym przykładzie, załączenie %M0001 powoduje ustawienie bitu 5 (określonego przez BIT) w rejestrze %R0001 (na wejściu IN) jest ustawiana na 0.



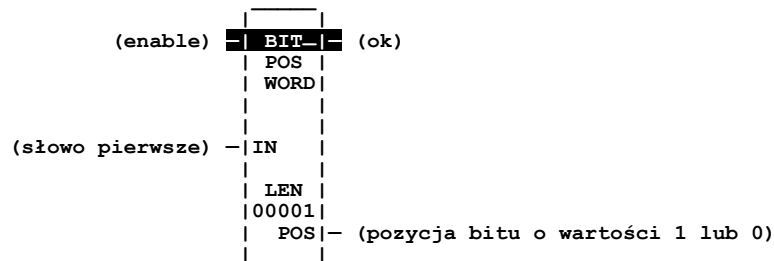
## BPOS (WORD)

Funkcja Bit Position (BPOS) jest wykorzystywana do zlokalizowania bitu o wartości 1 w słowie.

W każdym cyklu po uaktywnieniu funkcja przeszukuje ciąg bitów rozpoczynając od parametru IN. Gdy funkcja przerwie wyszukiwanie to oznacza, że albo został odnaleziony bit o wartości równej 1, albo został już przeszukany cały ciąg.

Pozycja zlokalizowanego bitu jest udostępniana przez parametr POS. Jeśli wszystkie bity w przeszukiwanym ciągu słów posiadają wartość 0, wartość tego parametru jest ustawiana na 0.

Funkcja ta może dokonać sprawdzenia wartości bitu w zakresie od 1 do 256 słów bitowych, czyli z ciągu liczącego maksymalnie 16\*256 bitów. Sygnał wyjściowy jest przesyłany zawsze wtedy, gdy do bloku dopływa sygnał wejściowy.



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji.
IN	Adres pierwszego słowa ciągu słów, na którym ma zostać wykonana operacja.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
POS	Pozycja pierwszego znalezionej niezerowego bitu przeszukiwanego słowa lub 0 w przypadku, gdy w przeszukiwanym obszarze występują tylko bity zerowe.
LEN	Liczba słów bitowych ciągu słów, z którego wybierany jest bit, którego wartość ma zostać zmieniona (maksymalnie 256).

### Uwaga

**W przypadku korzystania z funkcji Bit Test, Bit Set, Bit Clear lub Bit Pos, bity numerowane są od 1 do 16, a NIE od 0 do 15**

## Dopuszczalne typy zmiennych

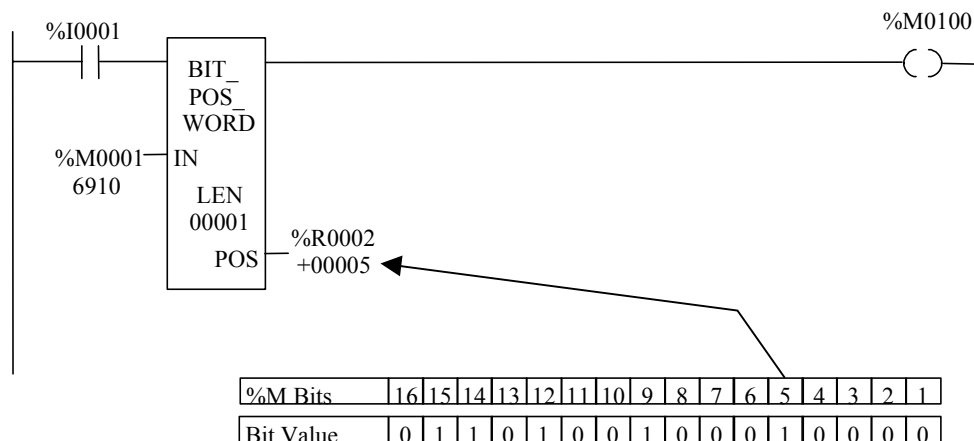
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•	•	•	•	•	•		
POS		•	•	•	•		•	•	•	•		
ok	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład

Należy zauważyć, że funkcja Bit Position może pracować tylko na danych typu WORD, dlatego adres wejścia IN będzie w oprogramowaniu Logimaster wyświetlany w formacie szesnastkowym. Jednak wartość POS będzie wyświetlana jako liczba całkowita. Oprogramowanie Logimaster wyświetla pierwsze 16 bitów w formacie szesnastkowym.

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %I0001 na 1 powoduje rozpoczęcie przeszukiwania ciągu bitów, począwszy od adresu wskazywanego przez %M0001, aż do momentu, kiedy odnaleziony zostanie bit o wartości 1 lub przeszukany zostanie cały ciąg bitów. Przekaznika %Q0001 jest ustawiana na 1. Jeśli zostanie znaleziony bit o wartości 1 jego położenie zostanie zapisane w rejestrze %R0002; w przeciwnym przypadku w %R0002 zostanie zapisane 0. W poniższym przykładzie, bit 5 jest pierwszym bitem o wartości 1 napotkanym w czasie przeszukiwania (przeszukiwanie rozpoczyna się od bitu 1), dlatego w rejestrze %R0002 jest zapisana 5.



## MSKCMP (WORD, DWORD)

Funkcja MSKCMP (*dostępna w jednostkach centralnych wersja 4.41 i nowszych*) służy do porównywania kolejnych bitów dwóch ciągów bitów, z możliwością maskowania wybranych bitów. Długość ciągu słów 16-bitowych (funkcja MSKCMP dla danych typu word) lub 32-bitowych (funkcja MSKCMP dla danych typu double-word) określa parametr LEN.

Po uaktywnieniu funkcja porównuje bity pierwszego ciągu z odpowiednimi bitami drugiego ciągu. Porównywanie trwa do czasu, aż zostanie wykryta różnica lub do osiągnięcia ostatniego bitu. Funkcja jest wykonywana w każdym cyklu pracy, w którym jest uaktywniona, dlatego też w wielu przypadkach wykorzystywane są styki monostabilne na wejściu enable.

Parametr wejściowy BIT wykorzystywany jest do zapamiętywania numeru bitu, od którego należy rozpocząć następne szukanie (0 oznacza pierwszy bit ciągu). Parametr wyjściowy BN wykorzystywany jest do zapamiętywania numeru ostatnio porównywanego bitu (1 oznacza pierwszy bit ciągu). Jeśli BN i BIT zostanie przypisany ten sam adres, kolejna operacja porównywania rozpocznie się na kolejnej pozycji po wykrytej niezgodności stanu bitów.

Jeżeli następne szukanie ma być rozpoczęte w innym miejscu, można wprowadzić inne wartości dla BIT i BN. Jeżeli wartość parametru BIT jest większa od długości ciągu, przed rozpoczęciem następnego porównywania wartość tego parametru ustawiana jest na 0.

### Wszystkie bity w parametrach I1 i I2 są takie same

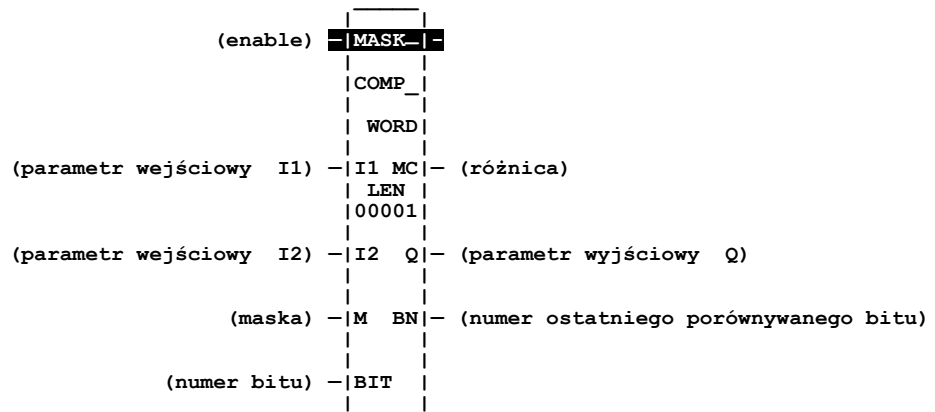
Jeśli wszystkie bity w obydwu porównywanych ciągach są identyczne, wyjście MC jest ustawiane na 0, a wyjście BN na najwyższy numer bitu w porównywanych ciągach. Następnie porównywanie zostaje przerwane. Następne wywołanie MSKCMP, BN spowoduje jej wyzerowanie.

### Znaleziona została różnica

Gdy dwa porównywane bity różnią się, funkcja sprawdza stan odpowiedniego bitu w ciągu bitów M (maska). Jeśli bit maski posiada wartość 1, niezgodność w porównywaniu bitów jest ignorowana, porównywanie jest kontynuowane do czasu znalezienia następnych dwóch różniących się bitów lub do końca słowa bitowego.

Jeśli bit maski posiada wartość 0, blok funkcyjny wykonuje następujące operacje:

1. Ustawia stan odpowiedniego bitu maski M na 1.
2. Ustawia wyjście MC (wykryta niezgodność) na 1.
3. Aktualizuje ciąg bitów Q tak, aby był zgodny ze zaktualizowaną maską.
4. Ustawia wartość parametru BN na numer bitu, dla którego wykryto niezgodność.
5. Przerzywa porównywanie.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Gdy do bloku funkcyjnego dopływa sygnał, wykonywana jest operacja logiczna.
I1	Adres pierwszego ciągu porównanych bitów.
I2	Adres drugiego ciągu porównanych bitów.
M	Adres maski.
BIT	Adres zmiennej (rejstru), w którym przechowywany jest numer bitu, od którego powinna się zacząć kolejna operacja porównywania.
MC	Jest ustawiana na 1 na jeden skan jeśli wykryto niezgodność porównywanych bitów. Można wykorzystać przekaznik z pamięcią jeśli pożądane jest „zapisanie” stanu wyjścia dłużej niż jeden skan.
Q	Kopia maski.
BN	Numer ostatniego porównywanego bitu.
LEN	Liczba słów bitowych ciągu słów, z którego wybierany jest bit, którego wartość ma zostać zmieniona (maksymalnie 256).

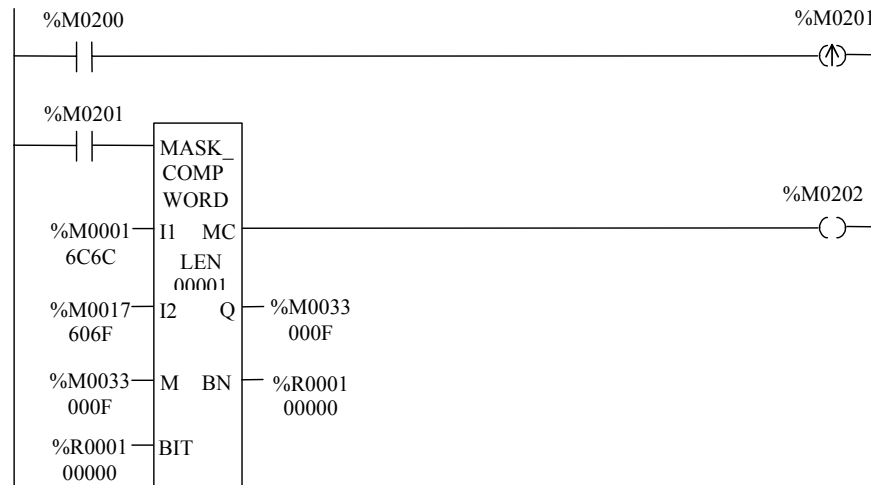
## Dopuszczalne typy zmiennych

Parametr	sygnal	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
I1		o	o	o	o	o	o	•	•	•		
I2		o	o	o	o	o	o	•	•	•		
M		o	o	o	o	o†	o	•	•	•		
BIT		•	•	•	•	•	•	•	•	•	•	
LEN											•‡	
MC	•											•
Q		o	o	o	o	o†	o	•	•	•		
BN		•	•	•	•	•	•	•	•	•		

- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Dopuszczalne jest stosowanie wyłącznie zmiennych typu WORD, nie można stosować zmiennych typu DWORD.
- † Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.
- ‡ Maksymalna wartość stałej może wynosić 4095 dla danych typu WORD i 2047 dla danych typu DWORD.

## Przykład 1 – instrukcja MSKCMP

W momencie zwarcia styku %M0200, styk %M0201 zostaje zwarty na czas jednego cyklu uaktywniając blok MSKCMP. Zmienne %M0001 do %M0016 (I1) są porównywane ze zmiennymi %M0017 do %M0032 (I2). Maska określana jest poprzez zmienne %M0033 do %M0048. Wartość podana przez %R0001 (BIT) określa numer bitu, od którego ma być rozpoczęte porównywanie dwóch ciągów wejściowych I1 i I2.



### Stan przed pierwszym wykonaniem funkcji MSKCMP

Zawartość wejścia odniesienia przed wykonaniem funkcji MSKCMP jest następująca:

%M Bits	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Input 1(I1)	0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0

%M Bits	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
Input 2(I2)	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1

%M Bits	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
Mask (M/Q)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

BIT/BN (%R0001) = 0

MC (%M0202) = OFF

## Stan po pierwszym wykonaniu funkcji MSKCMP

Poniższa tabela pokazuje zawartość maski odniesienia (MASK(M/Q)) po jednokrotnym wykonaniu funkcji MSKCMP. (I1 i I2 cały czas posiadają wartości jak pokazano powyżej). Po stwierdzeniu niezgodności w dziewiątym bicie, ustawiany jest na 1 dziewiąty bit (%M0041) w masce (MASK), BIT/BN zawiera wartość 9, a wyjście MC jest załączane na jeden cykl. Chociaż bity na pierwszym i drugim miejscu nie są równe nie powodują one niezgodności ponieważ bity maski dla tych pozycji są równe 1.

%M Bits	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
Mask (M/Q)	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1

BIT/BN (%R0001) = 9

MC (%M0202) = ON (na jeden skan)

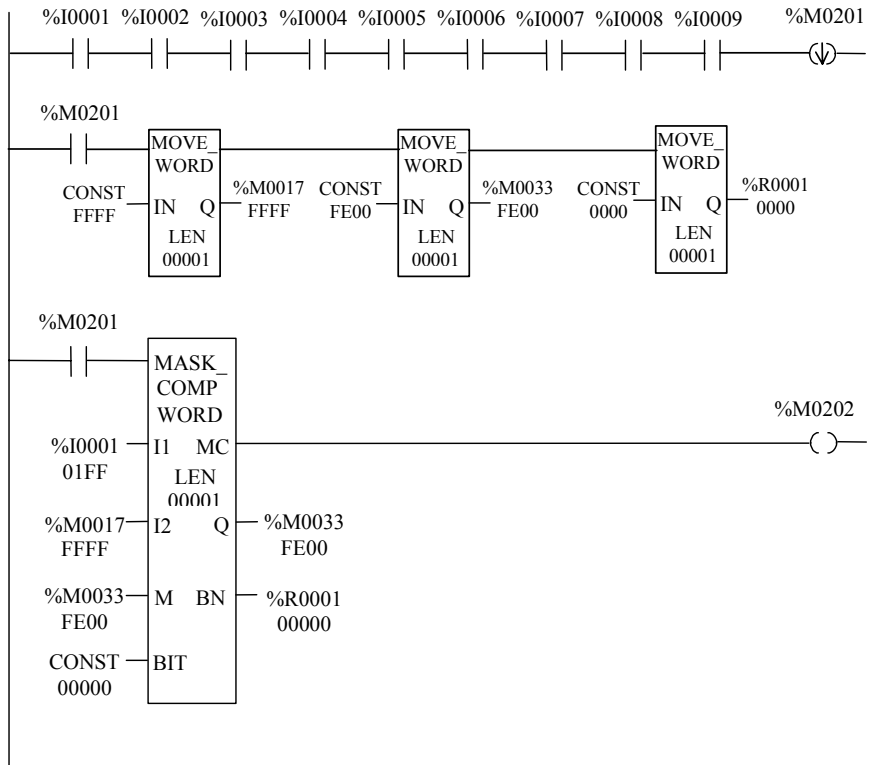
## Przykład 2 – Detekcja błędu z użyciem maski

Problemy z nieciągłością mogą być trudne do wykrycia i usunięcia. Przykładem mogą być przełączniki połączone szeregowo załączające przekaźnik alarmowy. W normalnych warunkach wszystkie przełączniki są zwarte a przekaźnik błędu zasilony (połączenie odporne na uszkodzenia). Po wystąpieniu błędu jeden z przełączników zostaje otwarty, a sygnał z przekaźnika alarmowego zanika. Jeśli styk, na którym wystąpił błąd pozostaje otwarty, można w łatwy sposób określić który z przełączników został uszkodzony. Jednak czasami styk zostaje rozarty tylko na chwilę po czym zostaje zwarty. Powoduje to zwolnienie przekaźnika na moment i wyłączenie procesu. Po powtórnym zwarceniu styku, wszystko może wydawać się normalnie.

Poniżej przedstawiono układ rozwiązujący ten problem, układ „zatrząskuje błędy” zapisując w rejestrach numer otwartego styku. W pierwszym szczeblu znajdują się styki odpowiadające wyłącznikom przyłączonym do modułu wejść (%I1 - %I9), załączają one przekaźnik uaktywniany zboczem opadającym sygnału %M0021.

W drugim szczeblu następuje inicjujący blok MSKCMP do przechwytywania błędu. Pierwszy blok Move zapisuje wszystkie stany logiczne do wejścia I2 bloku MSKCMP. Drugi blok Move zapisuje wartość 1 bitom od 10 do 16 w masce (dlatego te bity są ignorowane), pierwszych dziewięć bitów porównywanego ciągu (MSKCMP pracuje na całym ciągu bitów) jest niezbędnych dla przełączników %I0001 - %I0009. Trzeci blok Move zeruje rejestr wyjściowy %R0001, co powoduje że jest możliwe raportowanie późniejszych błędów.

W czasie normalnej pracy, pierwsze dziewięć bitów na wejściu MSKCMP posiadają stan logiczny 1 dopóki przełączniki są zwarte. Wejście I2 jest załączone dopóki występują warunki normalne którymi wyłączniki wejściowe są porównywane. Maskę 1 w bitach 10-16 posiada 1 ponieważ bity te nie są wykorzystywane dopóki jest dziewięć wyłączników. Po otwarciu wyłącznika, styk %M0201 zostaje zwarty na czas jednego cyklu. Powoduje to uaktywnienie drugiego i trzeciego szczebla. Blok MSKCMP porównuje stan wyłączników ze stanem wejścia I2, identyfikując otwarty wyłącznik (posiadający stan logiczny 0) i zapisując jego numer do wyjścia BN (%R0001). Bity są numerowane od 1 do 9 rozpoczynając od %I1. Na przykład jeśli %I4 zostanie otwarty rejestr %R0001 będzie zawierał wartość 4. Należy zwrócić uwagę, że w tym układzie jeśli wszystkie wyłączniki zostaną otwarte i zwarte powtórnie, przekaźnik %M0201 zostanie zwolniony i złączony, jednak w rejestrze %R0001 zostanie zapisany numer wyłącznika, który został rozarty. Jednak, jeśli wyłącznik zostanie powtórnie otwarty np. operator maszyny wciśnie wyłącznik bezpieczeństwa, porównywanie z maską zostanie ponownie uaktywnione i zapisany zostanie numer ostatnio otwartego wyłącznika w rejestrze %R0001. Oznacza to nie powinno się dotyczyć żadnego wyposażenia dopóki nie zostanie sprawdzona wartość w rejestrze %R0001. Jeśli nie wydaje się to praktyczne należy zastosować kolejną instrukcję Move.





# Rozdział 9

## Funkcje do operacji na danych

Funkcje z tej grupy umożliwiają przemieszczanie danych. W rozdziale tym opisano następujące funkcje do przesyłania danych.

Oznaczenie skrótowe	Funkcja	Opis	Strona
MOVE	Przemieszczenie	Kopiowanie danych, traktowanych jako pojedyncze bity. Maksymalna, dopuszczalna długość wynosi 256 słów, za wyjątkiem MOVE_BIT, dla której wielkość ta wynosi 256 bitów. Dane mogą być przemieszczane do zmiennych różnych typów, bez uprzedniej konwersji.	9-2
BLKMOV	Przemieszczanie bloku	Kopiowanie grupy siedmiu stałych wartości do określonego obszaru pamięci sterownika. Stałe te są podawane jako parametru wejściowe.	9-5
BLKCLR	Zerowanie fragmentu pamięci	Zerowanie określonego bloku pamięci sterownika. Funkcja ta może zostać wykorzystana do zerowania pamięci zmiennych dyskretnych (%I, %Q, %M, %G, lub %T) i pamięci zmiennych rejestrowych (%R, %AI i %AQ). Maksymalna, dopuszczalna długość wynosi 256 bitów.	9-7
SHFR	Rejestr przemieszczający	Wstawienie jednego lub większej liczby słów w określone miejsce pamięci. Maksymalna, dopuszczalna długość wynosi 256 bitów.	9-8
BITSEQ	Przemieszczanie jedynek	Przemieszczanie sekwencji bitów w ciągu bitów. Maksymalna, dopuszczalna długość wynosi 256 bitów.	9-11
COMMREQ	Inicjalizacja komunikacji z jednym z modułów sterownika	Funkcja ta pozwala na nawiązanie komunikacji sterownika z jednym z wyspecjalizowanych modułów dodatkowych, jak np. z modułem komunikacyjnym Genius lub z modułem programowalnego koprocesora.	9-15

## MOVE (BIT, INT, WORD, REAL)

Funkcja MOVE służy do przemieszczenia danych (jako pojedynczych bitów) z określonego miejsca pamięci w inne. Ponieważ dane są przesyłane jako bity, nowy adres nie musi odpowiadać temu samemu typowi danych jak adres oryginalny.

Funkcja MOVE posiada dwa parametry wejściowe i dwa parametry wyjściowe. Gdy do funkcja jest uaktywniona, dane są kopiowane bit po bicie z miejsca pamięci określonego przez parametr IN w miejsce określone przez parametr Q. Jeśli dane przesyłane są pomiędzy obszarami przypisanymi zmiennym dyskretnym (np. z obszaru pamięci przypisanego zmiennym typu %I do obszaru przypisanego zmiennym %T), bity towarzyszące, sterujące wymuszoną zmianą wartości tych zmiennych, są również przesyłane. Dane oryginalne nie ulegają zmianie, chyba że zakresy adresów parametrów wejściowego i wyjściowego pokrywają się.

W przypadku typu BIT należy również uwzględnić inny aspekt. Jeżeli ciąg danych typu BIT określony przy pomocy parametru Q nie zawiera wszystkich bitów bajtu, bity towarzyszące powiązane z tym bajtem (nie mieszczące się w tym ciągu) są ustawiane na wartość 0 po doprowadzeniu sygnału do funkcji MOVE\_BIT.

Parametr wejściowy IN może wyrażać adres zmiennej lub stanowić wartość stałą. Jeżeli parametrem jest wartość stała, w miejscu określonym przez parametr wyjściowy Q zapisywana jest wartość stała. Przykładowo jeśli parametrem wejściowym IN jest stała równa 4, a parametr LEN ma wartość 1, to jest ona zapisywana pod adresem wskazywanym przez parametr Q. Jeśli długość jest większa od 1 a wartość stała jest określona, to wtedy jest ona zapisywana w miejscu wskazywanym przez parametr wyjściowy Q oraz w następnych adresach aż do uzyskania zadanej długości. Przykładowo jeżeli parametrem wejściowym jest stała 9, długość wynosi 4, w pamięci, pod adresem wskazywanym przez Q oraz w trzech dalszych pozycjach zostanie zapisana wartość 9.

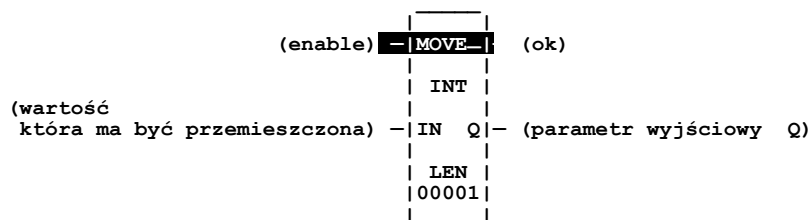
Parametr LEN określa liczbę:

- Słów, które mają być przemieszczone za pomocą funkcji MOVE\_INT i MOVE\_WORD.
- Bitów, które mają być przemieszczone za pomocą funkcji MOVE\_BIT.
- Liczb rzeczywistych, które mają być przemieszczone za pomocą funkcji MOVE\_REAL.

### Uwaga

Operacje na liczbach rzeczywistych można realizować wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x.

Sygnal wyjściowy jest przesyłany zawsze wtedy, gdy do bloku dopływa sygnał wejściowy.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie przemieszczenia.
IN	Wartość stała lub adres zmiennej, która ma zostać przemieszczona. W przypadku funkcji MOVE_BIT, można podać dowolną zmienną dyskretną, bez względu na ilość zajmowanych bitów. Jednak, oprogramowaniu Logicmaster wartości 16 bitowe są wyświetlane rozpoczynając od określonego adresu.
ok	Sygnal wyjściowy, pojawiający się po doplynięciu do bloku funkcyjnego sygnału wejściowego.
Q	Po dokonaniu przemieszczenia, wartość określona przez parametr IN jest kopiowana w miejsce określone przez parametr Q. W przypadku funkcji MOVE_BIT, można podać dowolną zmienną dyskretną, bez względu na ilość zajmowanych bitów. Jednak, oprogramowaniu Logicmaster wartości 16 bitowe są wyświetlane rozpoczynając od określonego adresu.
LEN	Liczba słów bitowych lub bitów, które mają zostać przemieszczone. Dla funkcji MOVE_INT oraz MOVE_WORD parametr LEN musi zawierać się pomiędzy 1 a 256. Dla funkcji MOVE_BIT, gdy parametr IN określa wartość stałą, parametr LEN musi zawierać się pomiędzy 1 a 16, w przeciwnym wypadku, parametr LEN musi przyjmować wartość z zakresu 1 do 256.

### Uwaga

W jednostkach centralnych 351, 352, 36x i jednostkach centralnych 37x funkcje MOVE\_INT i MOVE\_WORD nie dopuszczają możliwości nakładania się adresów IN i Q, jeżeli adres IN jest mniejszy od adresu Q. Przykładowo, w przypadku wartości:

IN=%R0001, Q=%R0004, LEN=5 (słów), zawartości wskazywane przez %R0007 i %R0008 będą nieokreślone; nie mniej jednak, dla wartości: Q=%R0001, IN=%R0004, LEN=5 (słów) otrzymany zostanie poprawny wynik.

Należy również zwrócić uwagę, że możliwość wykonywania działań na liczbach rzeczywistych, a więc możliwość korzystania z funkcji MOVE\_REAL, jest dostępna wyłącznie w jednostkach centralnych serii 35x i 36x (wersja 9 i nowsze) oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x.

## Dopuszczalne typy zmiennych

Parametr	sygnal	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•	o	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	o†	•	•	•	•		

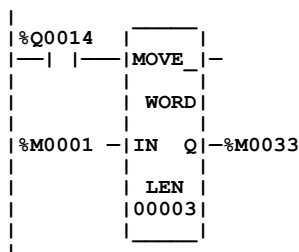
**Uwaga:** W przypadku danych REAL, dopuszczalne jest stosowanie wyłącznie zmiennych typu %R, %AI i %AQ.

- Dla danych typu BIT, INT i WORD, zmienne tego typu mogą być wykorzystywane jako dany parametr bloku funkcyjnego. W funkcji MOVE\_BIT, wartości zmiennych dyskretnych %I, %Q, %M i %T nie muszą być podawane z uwzględnieniem podziału na bajty. Zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu BIT lub WORD. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu INT.

† Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

## Przykład 1 - Nakładanie się adresów (tylko CPU 311-341)

Po ustawieniu wartości zmiennej %Q0014 na 1, z pamięci o adresie %M0001 przekopiowanych zostanie 48 bitów, do pamięci o adresie %M0033. Pomimo, że obszar pamięci do którego wykonywane jest kopiowanie pokrywa się z 16 bitami wejściowymi, operacja wykonywana jest poprawnie (za wyjątkiem jednostek centralnych 351 i 352, zgodnie z przytoczoną wcześniej uwagą).



### Przed wykonaniem funkcji Move:

INPUT (%M0001 do %M0048)

1

%M0016	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0032	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0048	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

### Po wykonaniu funkcji Move:

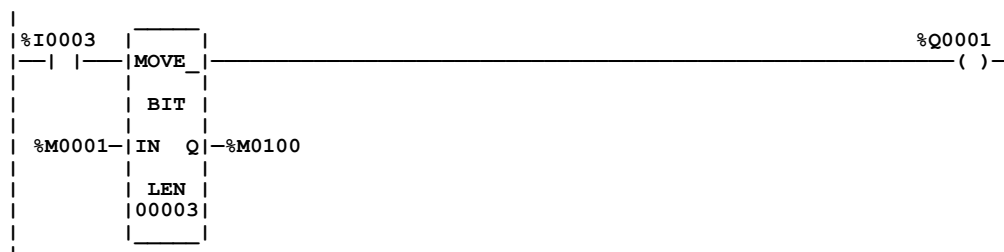
OUTPUT (%M0033 do %M0080)

33

%M0048	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0064	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0080	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

## Przykład 2 – wszystkie jednostki centralne

W zamieszczonym poniżej przykładzie, zmiana wartości zmiennej %I0001 na 1, powoduje przemieszczenie trzech bitów %M0001, %M0002 i %M0003 do odpowiednio %M0100, %M0101 i %M0102 oraz załączenie przekaźnika %Q0001.



## BLKMOV (INT, WORD, REAL)

Funkcja BLKMOV służy do skopiowania grupy siedmiu stałych wartości do określonego obszaru pamięci sterownika.

### Uwaga

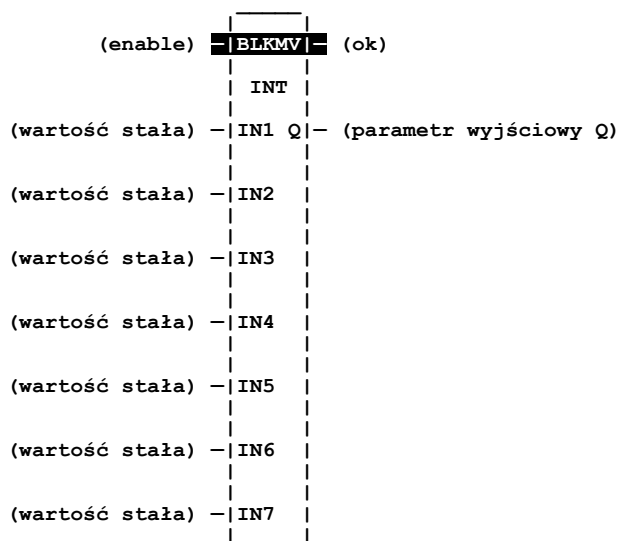
Operacje na liczbach rzeczywistych można realizować wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x.

Funkcja BLKMOV posiada osiem parametrów wejściowych i dwa parametry wyjściowe. Gdy do bloku funkcyjnego dopływa sygnał wejściowy, stałe wartości są kolejno kopiowane w miejsce zaadresowane przez parametr Q (tzn. pierwsza wartość IN1 jest kopiowana w miejsce określone przez Q, a następne sześć wartości w kolejne miejsca pamięci). Wyjście Q nie może być jednocześnie wejściem innego bloku funkcyjnego.

### Uwaga

Dla funkcji MOVE\_INT wartości parametrów IN1 - IN7 wyświetlane są jako liczby dziesiętne ze znakiem. Dla funkcji BLKMOV\_WORD parametry te są wyświetlane jako liczby heksadecymalne. Dla funkcji BLKMOV\_REAL parametry IN1-IN7 wyświetlane są jako liczby rzeczywiste.

Sygnał wyjściowy jest przesyłany zawsze wtedy, gdy blok jest uaktywniony.



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie przemieszczenia.
IN1—IN7	Wartości stałe, które mają zostać skopiowane.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
Q	Q określa miejsce w pamięci, w którym ma być umieszczona pierwsza z kopiowanych wartości. Do miejsca określonego przez parametr Q kopiowana jest wartość IN1.

## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN1 — IN7											•	
ok	•											•
Q		•	•	•	•	o†	•	•	•	•		

**Uwaga:** W przypadku danych REAL, dopuszczalne jest stosowanie wyłącznie zmiennych typu %R, %AI i %AQ.

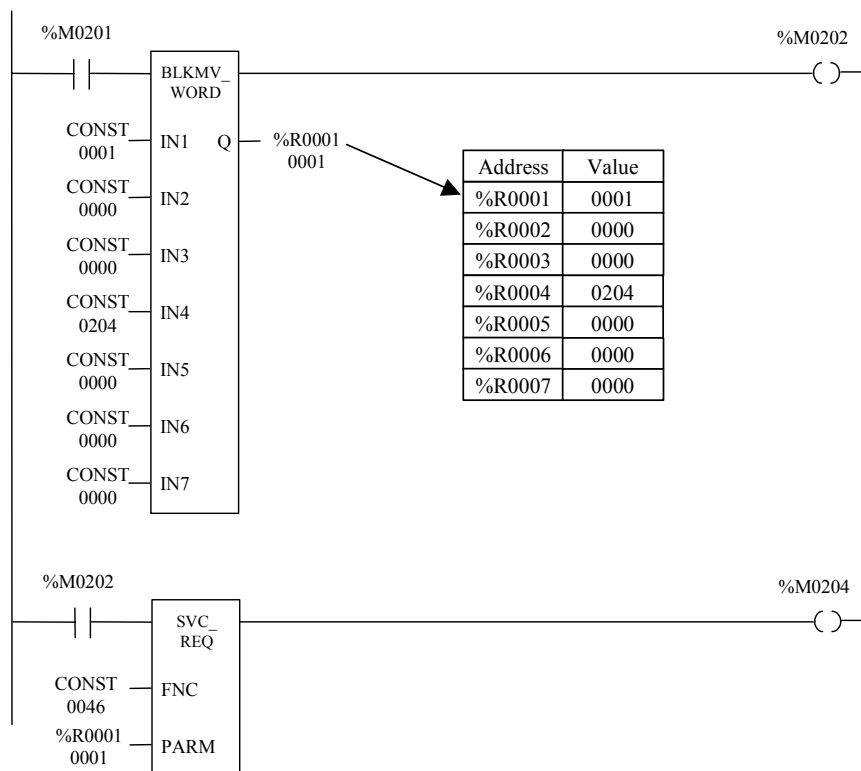
- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu WORD. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu INT i REAL.
- † Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

### Uwaga

Możliwość wykonywania operacji na liczbach rzeczywistych dostępna jest wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 i CPU37x. Tak więc z funkcji BLKMOV\_REAL można korzystać wyłącznie w tych jednostkach centralnych sterowników 90-30.

## Przykład

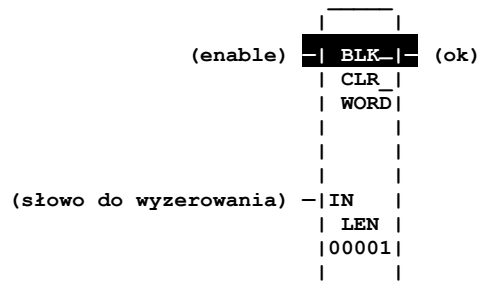
W poniższym przykładzie złączenie styku %M0201 powoduje kopiowanie wielkości z siedmiu wejść do pamięci %R0007 (określona przez Q) do %R0007. Jeśli bloku BLKMOV zostanie wykonany prawidłowo, załączone zostanie wyjście Ok, załączające %M0202. Załączenie przekaźnika %M0202 uaktywnia funkcję Service Request w następnym szczeblu, wykorzystującą rejestry %R0001 do %R0007 jako parametry. (Więcej informacji na temat funkcji Service Request zamieszczono w rozdziale 12).



## BLKCLR (WORD)

Funkcja BLKCLR służy do wyzerowania określonego bloku pamięci sterownika.

Funkcja BLKCLR posiada dwa parametry wejściowe i jeden parametr wyjściowy. Gdy do bloku funkcyjnego dopływa sygnał wejściowy, pod adresem rozpoczynającym się od zmiennej określonej przez parametr IN zapisywane są zera. Gdy dane mające być wyzerowane pochodzą ze zmiennych dyskretnych (%I, %Q, %M, %G, lub %T) to informacja o przemieszczeniu tych zmiennych jest także zerowana. Funkcja przesyła sygnał wyjściowy.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wyzerowanie zadanego obszaru pamięci.
IN	Adres pierwszego słowa pamięci, które ma zostać wyzerowane.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
LEN	Parametr LEN musi mieć wartość z zakresu 1 do 256 słów.

### Dopuszczalne typy zmiennych

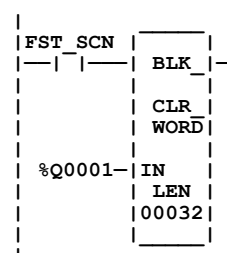
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•	•†	•	•	•	•		
ok	•											•

• Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

† Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

### Przykład

W zamieszczonym poniżej przykładzie, 32 słowa pamięci typu %Q, począwszy od adresu %Q0001 są zerowane podczas rozruchu sterownika.



## SHFR (BIT, WORD)

Blok funkcyjny SHFR służy do wstawienia jednego bitu (lub słowa bitowego) w określone miejsce w pamięci sterownika, przesunięcie zawartości obszaru pamięci rozpoczynającego się w tym miejscu (posiadającego określoną długość) o jeden bit (lub słowo bitowe) w lewo oraz przemieszczenie bitu (lub słowa bitowego), "wypchniętego" z ostatniego miejsca obszaru pamięci, w inne miejsce. Przykładowo, można wstawić jedno słowo do obszaru pamięci o długości pięciu słów. W wyniku takiej operacji, jedno słowo danych zostanie "wypchnięte" poza ten obszar pamięci.

### Uwaga

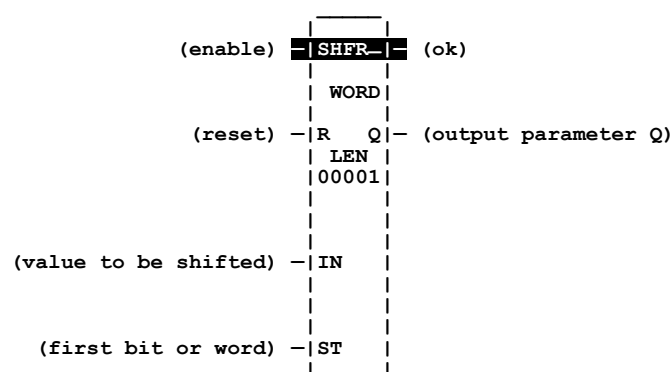
Pokrywanie się adresów parametrów wejściowych i wyjściowych w przypadku funkcji operujących na ciągach wielu słów bitowych mogą być przyczyną niespodziewanych rezultatów działania funkcji.

Funkcja SHFR posiada cztery parametry wejściowe i dwa parametry wyjściowe. Wejście zerujące R powoduje wyzerowanie wszystkich bitów lub słów w przemieszczanym obszarze pamięci i ma wyższy priorytet w stosunku do wejścia enable. Jeżeli aktywny jest sygnał reset obszaru pamięci, którego początek zawiera parametr ST (adres pierwszego bitu lub słowa bitowego rejestru) o długości LEN (długość rejestru w bitach lub słowach bitowych) jest wypełniany zerami.

Po doprowadzeniu do bloku funkcyjnego sygnału wejściowego, jeśli równocześnie na wejście reset nie jest podawany sygnał, każdy bit lub słowo bitowe rejestru przemieszczane są w lewo. Ostatni element w przesuwanym rejestrze jest przesuwany do wyjścia Q. Jeśli Q posiada unikalny adres, dane przesunięte z wyjścia Q są odrzucane. Jednak jeśli IN I Q posiadają ten sam adres, dane będą ponownie w przesuwanym rejestrze. Element o najwyższym adresie z przesuwanego rejestru IN jest przesuwany pustego elementu rozpoczynając od ST. Zawartość przesuwanego rejestru jest dostępna z poziomu programu sterującego gdyż jest przechowywana w adresowalnej pamięci.

Sygnał wyjściowy jest przesyłany zawsze wtedy, gdy do bloku dopływa sygnał wejściowy.

Funkcja będzie wykonywana jeden raz w czasie cyklu po uaktywnieniu; dlatego może być korzystne załączenie styku na wejściu enable przekaźnikiem monotabilnym jeśli jest pożądane wykonywanie bloku przy każdym zwarciu styku.





## Parametry

Parametr	Opis
enable	Gdy wejście enable jest załączone, a wejście R jest wyłączone, operacja jest wykonywana. Należy zwrócić uwagę, że blok SHFR będzie wykonywany jeden raz w czasie cyklu po uaktywnieniu.
R	Jeżeli na wejście R złączone, rejestr wskazywany przez ST wypełniany jest zerami.
IN	IN określa wartość, która ma zostać skopiowana do pierwszego bitu lub słowa bitowego rejestru przemieszczającego. Dla funkcji SHFR_BIT można tu wykorzystać dowolną zmienną dyskretną, bez uwzględniania podziału na bajty. Nie mniej jednak, w czasie pracy wyświetlanych jest 16 bitów, począwszy od zadanego adresu.
ST	ST określa adres pierwszego bitu lub słowa bitowego rejestru przemieszczającego. Dla funkcji SHFR_BIT można tu wykorzystać dowolną zmienną dyskretną, bez uwzględniania podziału na bajty. Nie mniej jednak, w czasie pracy wyświetlanych jest 16 bitów, począwszy od zadanego adresu.
ok	Wyjście ok jest złączane ilekroć wejście enable jest załączone, a wejście R wyłączone.
Q	Q określa adres, pod który ma zostać skopiowany bit lub słowo bitowe "wypchnięte" z rejestru przemieszczającego po wykonaniu operacji. Dla funkcji SHFR_BIT można tu wykorzystać dowolną zmienną dyskretną, bez uwzględniania podziału na bajty. Nie mniej jednak, w czasie pracy wyświetlanych jest 16 bitów, począwszy od zadanego adresu.
LEN	LEN określa ilość przemieszczanych rejestrów. Dla funkcji SHFR_WORD, parametr LEN musi mieć wartość z zakresu 1 do 256 słów. Dla funkcji SHFR_BIT, parametr LEN musi mieć wartość z zakresu 1 do 256 bitów.

## Dopuszczalne typy zmiennych

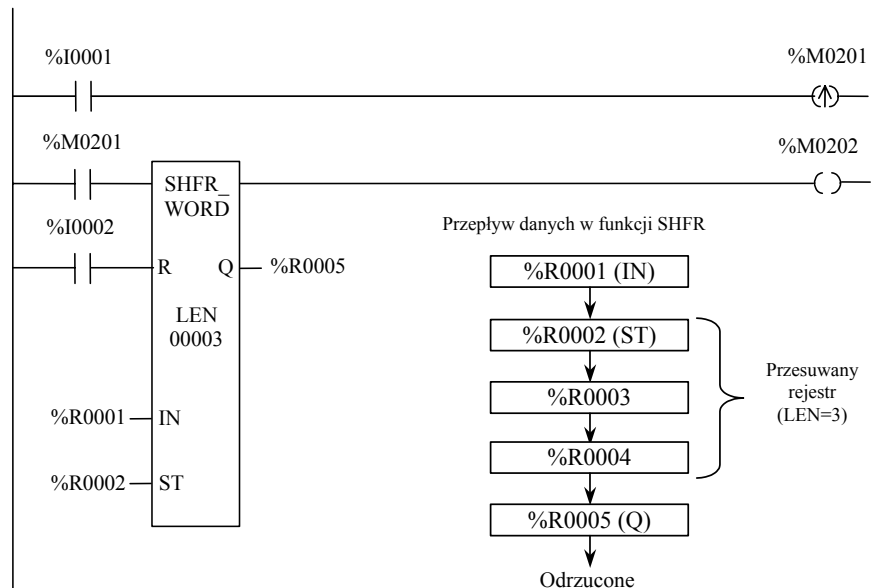
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
R	•											
IN		•	•	•	•	•	•	•	•	•	•	
ST		•	•	•	•	•†	•	•	•	•		
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Dla danych typu BIT i WORD, zmienne tego typu mogą być wykorzystywane jako dany parametr bloku funkcyjnego. W funkcji SHFR\_BIT, wartości zmiennych dyskretnych %I, %Q, %M i %T nie muszą być podawane z uwzględnieniem podziału na bajty.
- † Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

## Przykład 1

W zamieszczonym poniżej przykładzie, rejestr przemieszczający wykorzystuje trzy (LEN=3) lokacje pamięć od %R0002 do %R0004. Jeśli styk reset %I0002 jest załączony, następuje wyzerowanie przesuniętych słów.

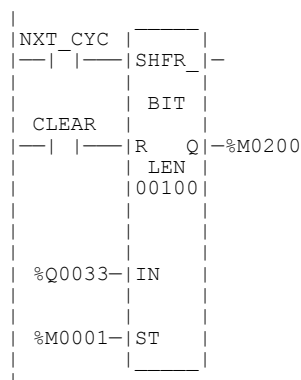
Zwarcie styku %I0001 powoduje załączenie na jeden skan styku %M0201 na wejściu enable bloku SHFR. Powoduje to przesuwanie danych z rejestru %R0004 w miejsce określone przez adres Q - %R0005 (dane znajdujące się w %R0005 są odrzucane). Dane z %R0003 są przesuwane do %R0004; dane z %R0002 – do %R0003, a dane z rejestru %R0001 (IN) do rejestru %R0002 (ST). Przepływ tych danych został przedstawiony na poniższym rysunku. Jeśli jest to wymagane to dane mogą być ponownie wprowadzone do rejestru wykorzystując ten sam adres IN i Q.



## Przykład 2

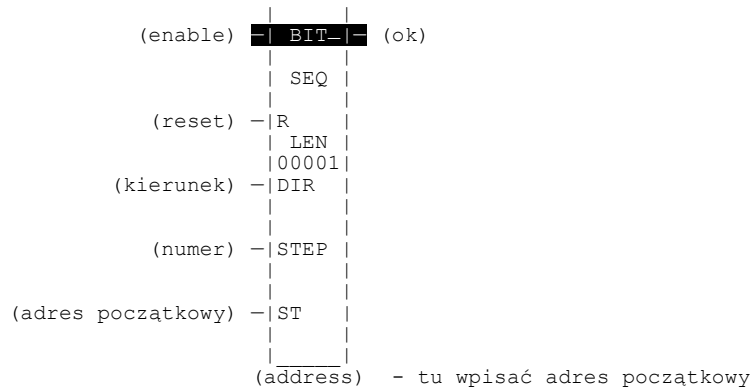
W tym przykładzie przesuwany rejestr jest typu BIT. Parametr LEN określa długość pamięci na 100 od %M0001 do %M0100. Jeżeli aktywny jest sygnał zerowania CLEAR, funkcja SHFR wypełnia zerami obszar pamięci ograniczony adresami %M0001 do %M0100.

Jeżeli wartość zmiennej NXT\_CYC (styk monostabilny) jest równa 1, a sygnał CLEAR nie jest doprowadzany funkcja SHFR przemieszcza dane obszaru od %M0001 do %M0100. Bit %Q0033 jest przesuwany do bitu %M0001, natomiast bit "wypchnięty" jest zapisywany do %M0200. Poprzednia wartość Q jest odrzucana.



## BITSEQ (BIT)

Funkcja Bit Sequencer (BITSEQ) jest wykorzystywana do przemieszczania pojedynczego bitu o wartości logicznej 1 sekwencyjnie w pewnym obszarze. Jeśli bit jest przesunięty na koniec tablicy zostanie przemieszczony na drugi koniec tej tablicy w czasie następnego przesuwania, a późniejsze przemieszczenia będą realizowane z tego miejsca. Funkcja BITSEQ posiada pięć parametrów wejściowych i jeden parametr wyjściowy.



### Uaktywnienie

Wejście enable jest załączane zboczem narastającym sygnału wykonując przemieszczenie, kolejne przemieszczenie będzie możliwe do wykonania podaniu na wejście enable kolejnego sygnału narastającego. Dlatego niezbędne jest styków z przekaźników uaktywnianych narastającym zboczem sygnału.

### Wejście zerujące - R

Podanie sygnału na to wejście powoduje wyłączenie bloku.

Podanie sygnału na wejście zerujące R ma większy priorytet od podania sygnału na wejście EN i zawsze powoduje przejście do stanu początkowego bloku. Jeżeli wejście R jest aktywne wartość bieżącego bitu jest ustawiana na wartość określoną przez parametr STEP a pozostałym bitom przypisywana jest wartość 0. Jeśli nie określono wartość STEP (STEP = 0) wartość 1 jest przypisywana pierwszemu bitowi, pozostałe bity są ustawiane na 0.

Gdy na wejście "enable" podany zostanie sygnał wejściowy i jednocześnie nie jest on podawany na wejście zerujące R, bit posiadający do tej pory wartość 1 zostaje wyzerowany. Numer bieżącego bitu jest inkrementowany lub dekrementowany, w zależności od wartości parametru DIR. Wartość bitu o zmienionym numerze zostaje ustawiona na 1.

### Wejście STEP

- Jeśli podczas działania funkcji bieżący numer bitu o wartości równej 1 jest zwiększany i przekroczy dopuszczalny zakres ( $1 \leq \text{bieżący numer} \leq \text{LEN}$ ), numer ten jest ustawiany na 1.
- Jeśli podczas działania funkcji bieżący numer bitu o wartości równej 1 jest zmniejszany i przekroczy dopuszczalny zakres ( $1 \leq \text{bieżący numer} \leq \text{LEN}$ ), numer ten jest ustawiany na LEN.

Parametr ST jest parametrem opcjonalnym. Jeżeli nie jest wykorzystywany, funkcja BITSEQ pracuje zgodnie z przedstawionym powyżej opisem, żadne bity nie są jednak ustawiane czy zerowane. W zasadzie, funkcja BITSEQ w takim przypadku powoduje cykliczne przemieszczanie bieżącego numeru bitu w dopuszczalnym zakresie.

## Wejście DIR

Kierunek przemieszczania bitu może być zmieniany przez załączenia i wyłączenia wejścia DIR. Jeśli wejście jest załączone bit jest przesuwany w prawo. W przypadku gdy wejście jest wyłączone bit jest przesuwany w lewo.

## Parametry ST (adres początkowy) i LEN (długość)

Wejście ST zawiera początkowy adres pamięci tablicy sekwensera. Długość tablicy w bitach jest określana parametrem LEN. Na przykład, jeśli parametrowi ST przypisano %M0001 a LEN ma wartość 16, funkcja będzie wykorzystywać bity od %M0001 do %M0016. Jeśli parametrowi ST przypisano pamięć typ %R, wtedy LEN będzie określać ile kolejnych bitów będzie wykorzystywać funkcja. Na przykład, jeśli parametrowi ST przypisano %R0004, a LEN ma wartość osiem, wtedy tylko osiem pierwszych bitów rejestru %R będzie wykorzystywane przez funkcje, ostatnich osiem bitów %R0004 będzie ignorowane przez funkcje.

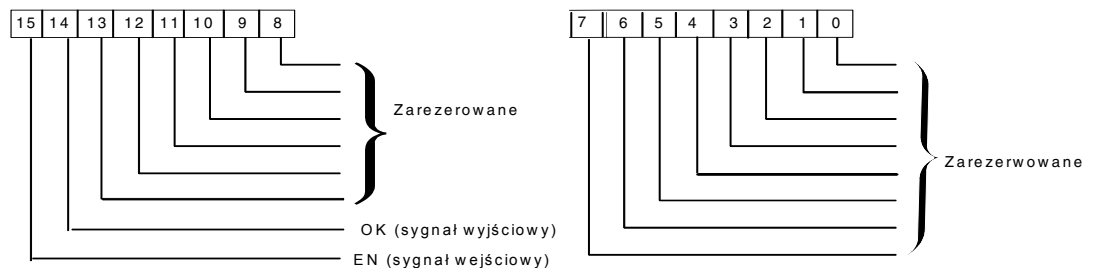
## Zapotrzebowanie na pamięć bloku funkcyjnego BIT\_SEQ

Każda funkcja BITSEQ wykorzystuje trzy słowa (rejstry) pamięci typu %R, do zapamiętywania następujących parametrów:

numer bieżącego bitu	słowo 1
długość obszaru pamięci (w bitach)	słowo 2
słowo sterujące	słowo 3

Używając oprogramowania LogiMaster w bloku funkcyjnym funkcji BITSEQ należy podać adres początkowy dla tych trzech słów (rejestrów), bezpośrednio pod symbolem graficznym (proszę porównać z przykładem zamieszczonym na następnej stronie).

Słowo sterujące zapamiętuje stan wejść i wyjść cyfrowych, powiązanych z danym blokiem funkcyjnym, zgodnie z przedstawionym poniżej formatem:



## Uwaga

Bity od 0 do 13 nie są wykorzystywane przez blok. Należy również zwrócić uwagę, że w parametrze STEP bity numerowane są od 1 do 16, a *nie* 0 do 15.

## Parametry

Parametr	Opis
adres	Adres pierwszego z trzech rejestrów, w których przechowywany jest bieżący numer bitu o wartości równej 1, długość wykorzystywanego obszaru pamięci oraz informacje o ostatnim stanie parametrów.
enable	Gdy do bloku funkcyjnego dopływa sygnał, a nie dopływał on w poprzednim cyklu, a jednocześnie na wejście R nie jest podawany sygnał, funkcja BITSEQ wykonuje przemieszczenie bitu o wartości równej 1 w górę lub w dół.
R	Podanie sygnału na wejście R powoduje ustawienie warunków początkowych, tzn. skopiowanie bieżącego numeru bitu o wartości 1 z parametru STEP (lub ustawienie tego numeru na 1 w przypadku braku parametru STEP), wyzerowanie obszaru pamięci przypisanego funkcji BITSEQ oraz ustawienie wartości bitu o numerze STEP na 1.
DIR	DIR określa kierunek przemieszczania bitu o wartości równej 1. Jeśli na wejście DIR podawany jest sygnał, bit przemieszczany jest w górę (bieżący numer rośnie). Jeśli sygnał nie jest podawany, bit przemieszczany jest w dół (bieżący numer maleje).
STEP	STEP jest początkowym numerem bitu o wartości równej 1.
ST	Adres pierwszego bitu obszaru pamięci, na którym działa funkcja BITSEQ.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
LEN	Parametr LEN musi mieć wartość z zakresu 1 do 256 bitów.

### Uwaga

W czasie sprawdzania wielokrotnego wykorzystania zmiennych w przełącznikach, dla funkcji BITSEQ sprawdzanych jest 16 bitów parametru ST, nawet jeżeli parametr LEN ma wartość mniejszą od 16.

## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
adres								•				
enable	•											
R	•											
DIR	•											
STEP		•	•	•	•		•	•	•	•	•	•
ST		•	•	•	•	•†	•	•	•	•		•
ok	•											•

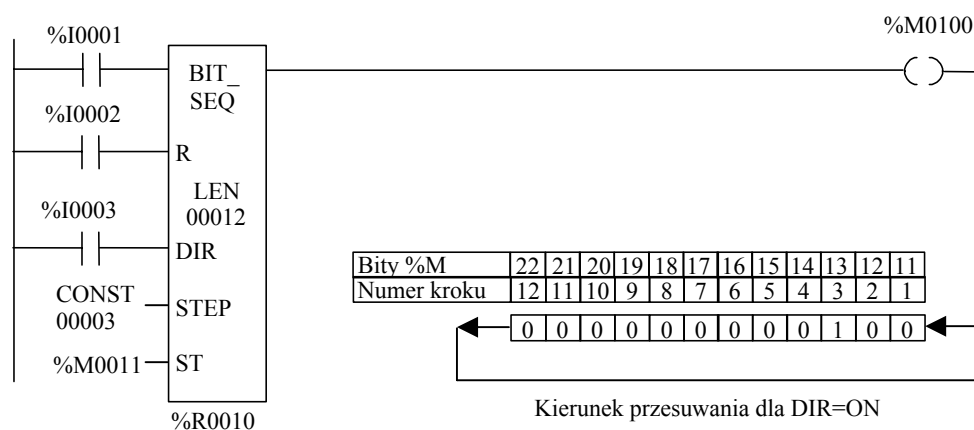
• Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

† Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

## Przykład

W zamieszczonym poniżej przykładu funkcja wykorzystuje bity od %M0011 (określone przez wejście ST) do %M0022 (LEN równa się 12). Blok trzech słów zapamiętany jest w rejestrach %R0010, %R0011 i %R0012. Ustawienie na zmiennej %I0002 (wejście R) powoduje wyzerowanie funkcji, co oznacza że bit trzeci (określony przez STEP) zostanie ustawiony, a wszystkie pozostałe bity zostaną wyzerowane.

Ustawienie zmiennej %I0001 (przy jednoczesnym wyzerowaniu zmiennej %I0002) na 1 powoduje że bit trzeci zostanie wyzerowany, a bit czwarty zostanie ustawiony jeśli DIR posiada stan logiczny 1, lub zostanie ustawiony bit drugi jeśli DIR posiada stan logiczny zero.



## COMMREQ

Funkcja COMMREQ służy do nawiązania przez sterownik komunikacji z jednym z wyspecjalizowanych modułów dodatkowych, jak np. z modułem komunikacyjnym GENIUS lub z modułem programowalnego koprocesora.

### Uwaga

Informacja podana w tym punkcie zapoznaje użytkownika z ogólnym formatem funkcji COMMREQ. Aby poprawnie zastosować funkcję COMMREQ dla każdego z modułów, należy posiadać dodatkowe informacje. Są one zawarte w dokumentacji technicznej modułów.

Funkcja COMMREQ posiada trzy parametry wejściowe i jeden parametr wyjściowy. Gdy do bloku funkcyjnego dopłynie sygnał wejściowy, do modułu wysyłany jest blok danych sterujących. Na początku bloku podawana jest zmienna określona przez parametr IN. Kasetę i gniazdo, w którym znajduje się inteligentny moduł są zadawane parametrem SYSID.

Funkcja COMMREQ może przesłać do wyszczególnionego modułu komunikat i czekać na odpowiedź lub też może przesłać komunikat i kontynuować wykonanie programu sterującego. Jeżeli w bloku sterującym określono, że program ma nie czekać na odpowiedź, zawartość bloku danych sterujących jest przesyłana do modułu docelowego, a jednostka centralna sterownika kontynuuje wykonywanie programu sterującego bez żadnych przerw. (Ignorowanie parametru określającego dopuszczalny czas oczekiwania). Jest to domyślny tryb **NOWAIT**.

Jeżeli w bloku sterującym określono, że program ma czekać na odpowiedź, zawartość bloku danych sterujących jest przesyłana do modułu docelowego, a jednostka centralna sterownika oczekuje na odpowiedź modułu. Maksymalny czas oczekiwania jednostki centralnej na odpowiedź modułu jest określony w bloku danych sterujących. Jeśli czas ten upłynie, a moduł nie odpowiada, wykonywanie programu sterującego jest kontynuowane. Jest to tryb **WAIT**.

Blok funkcyjny COMMREQ ustawia wartość parametru wyjściowego FT (błędne wykonanie funkcji) na 1, jeśli zachodzi jeden z następujących przypadków:

1. Pod wyszczególnionym adresem (SYSID) nie ma modułu.
2. Zadanie (TASK) wysłane modułowi nie może być przez ten moduł wykonane.
3. Długość danych w bloku wynosi 0.
4. Adres wskaźnika stanu urządzenia (część bloku danych sterujących) nie istnieje. Może to być spowodowane wybraniem zakresu pamięci przyporządkowanego innemu typowi zmiennych lub adresu przekraczającego zakres dozwolony dla zmiennych tego typu.

## Blok danych sterujących

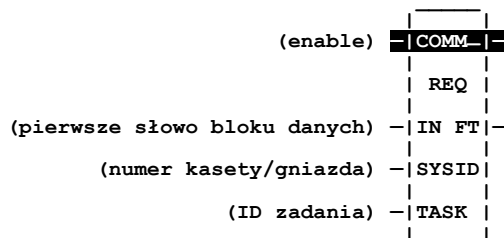
Blok danych sterujących dostarcza informacje do określonego inteligentnego modułu. Zawiera zarówno numer rozkazu do wykonania jak również przesyłane dane.

Adres początkowy tego bloku zawarty jest w parametrze IN bloku funkcyjnego. Może on wskazywać dowolny obszar pamięci zorientowanej rejestrowo (%R, %AI lub %AQ). Długość bloku danych sterujących jest zależny od typu modułu, do którego jest przesyłany oraz od długości przesyłanych danych.

Blok ten posiada następującą strukturę:

Długość (w słowach bitowych)	adres
Znacznik trybu pracy (WAIT lub NOWAIT)	adres + 1
Rodzaj pamięci, w której przechowywany jest status urządzenia	adres +2
Przesunięcie w pamięci	adres +3
Maksymalny czas oczekiwania na odpowiedź	adres +4
Maksymalny czas przeznaczony na komunikację	adres +5
Blok danych	adres +6 do adres +133

Informacje wymagane przez blok funkcyjny mogą być zadawane za pomocą odpowiednich funkcji takich jak Move.



## Parametry

Parametr	Opis
enable	Po uaktywnieniu tego wejścia wykonywane jest połączenie na czas jednego skanu. Jeśli nie jest pożądane wysyłanie polecenia COMMREQ wiele razy, wejście to powinno być uaktywniane stykiem monostabilnym.
IN	Określa adres pierwszego bitu słowa danych sterujących.
SYSID	Numer kasety (bardziej znaczący bajt) i numer gniazda (mniej znaczący bajt), w którym jest zainstalowany port przez który ma zostać nawiązana komunikacja.
TASK	Parametr TASK zawiera ID rozkazu dla docelowego modułu.
FT	Na wejście FT podawany jest sygnał, gdy jednostce centralnej nie udaje się nawiązać komunikacji z określonym modułem.

## Uwaga

W sterownikach serii 90-30 blok COMMREQ **nie** posiada wyjścia OK.

## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN								•	•	•		
SYSID		•	•	•	•		•	•	•	•	•	
TASK								•	•	•	•	
FT	•											•

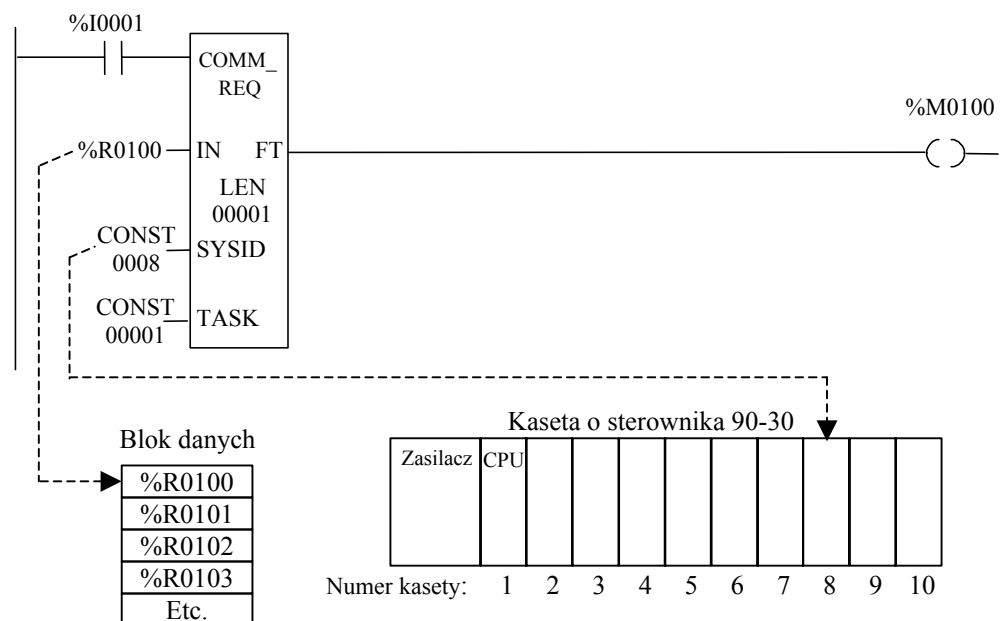
- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.



## Przykład

W poniższym przykładzie, uaktywnienie wejście %I0001, powoduje rozpoczęcie wysłania bloku danych sterujących (jego początek jest określony przez IN %R0100) zadania 1 (wejście TASK = 1) do modułu znajdującego się kasecie 0, gnieździe 8 (SYSID = 0008) sterownika. Jeśli w czasie wykonywania bloku CMMREQ wystąpi błąd, złączone zostanie wyjście FT załączające przekaźnik %M0100.

Należy zwrócić uwagę, że wejście IN określa adres początkowy bloku danych sterujących. Numer SYSID (hex) określa numer kasety i gniazda modułu docelowego, najbardziej znaczący bajt określa numer kasety, a bajt najmniej znaczący określa numer gniazda. Dlatego też w przykładzie tym SYSID równy 0008 oznacza kasetę 00, gniazdo 08. Kasetę o numerze zero oznacza kasetę główną lub kasetę z jednostką centralną, dlatego jeśli moduł docelowy znajduje się w kasecie rozszerzającej lub oddalonej najbardziej znaczący bajt parametru SYSID jest nie zerowy i odpowiada numerowi kasety, w której znajduje się moduł docelowy.





Do operacji tablicowych przeznaczone są następujące funkcje:

Oznaczenie skrótowe	Funkcja	Opis	Strona
ARRAY_MOVE	Kopiowanie danych	Kopiowanie określonej liczby danych z tablicy źródłowej do tablicy docelowej	10-1
SRCH_EQ	Szukanie wartości zadanej	Przeszukiwanie tablicy danych w celu znalezienia wartości równej wartości zadanej.	10-7
SRCH_NE	Szukanie wartości różnej	Przeszukiwanie tablicy danych w celu znalezienia wartości różnej od wartości zadanej.	10-7
SRCH_GT	Szukanie wartości większej	Przeszukiwanie tablicy danych w celu znalezienia wartości większej od wartości zadanej.	10-7
SRCH_GE	Szukanie wartości większej lub równej	Przeszukiwanie tablicy danych w celu znalezienia wartości większej lub równej wartości zadanej.	10-7
SRCH_LT	Szukanie wartości mniejszej	Przeszukiwanie tablicy danych w celu znalezienia wartości mniejszej od wartości zadanej.	10-7
SRCH_LE	Szukanie wartości mniejszej lub równej	Przeszukiwanie tablicy danych w celu znalezienia wartości mniejszej lub równej wartości zadanej.	10-7

Maksymalna długość ciągu, będącego parametrem tych funkcji wynosi 32 767 bajtów lub słów, albo 262 136 bitów (operacje na bitach mogą być wykonywane wyłącznie w przypadku funkcji ARRAY\_MOVE).

Funkcje te wykonują operacje na następujących typach danych:

Typ	Opis
INT	Liczba całkowita ze znakiem (16 bitowa)
DINT	Liczba całkowita podwójnej precyzji ze znakiem (32 bitowa).
BIT *	Ciąg bitów.
BYTE	Dane typu bajt.
WORD	Dane typu Word (Słowo).

\* Wyłącznie dla funkcji ARRAY\_MOVE.

Domyślny typ danych to liczba całkowita ze znakiem. Typ danych można zmienić po wprowadzeniu bloku funkcyjnego do szczebla drabiny logicznej programu sterującego. W celu porównania danych innych typów lub danych o różnych typach, należy je uprzednio przekonwertować do jednego z wymienionych typów danych za pomocą odpowiedniej funkcji (proszę porównać z Rozdziałem 11 "Funkcje konwersji").

## ARRAY\_MOVE (INT, DINT, BIT, BYTE, WORD)

### Definiowanie tablic i elementów danych

Do celów rozważań założono, że **tablica** jest zgrupowaną ciągłą przestrzenią adresową taką jak %R0100 do %R0120. **Element danych** jest zapisany w jednostce tablicy pamięci. Na przykład, jeśli tablica jest typu BIT, wtedy każdy element danych jest przechowywany w pojedynczym bicie pamięci, takim jak %M0001 (lub mógłby być pojedynczym bitem w pamięci typu rejestrowego). Jeśli tablica jest typu Word, wtedy każdy element jest zapisany w 16-bitowym słowie pamięci, takiej jak %R0100 (lub mógłby być 16 kolejnymi bitami). Patrz "Dopuszczalne typy zmiennych".

### Numery indeksu

Każdy element tablicy posiada numer odniesienia zwany **indeksem**, jest on automatycznie przypisywany przez sterownik. Numer indeksu określa położenie elementu w tablicy. Elementy są numerowane w porządku rosnącym rozpoczynając od najmniejszych adresów pamięci w tablicy, któremu jest przypisywany numer jeden.

Na przykład, poniższa tablica typu Word rejestr %R0105 posiada numer jeden. Tablica posiada 10 elementów numerowanych od 1 do 10.

Adres	Indeks
%R0105	1
%R0106	2
%R0107	3
%R0108	4
%R0109	5
%R0110	6
%R0111	7
%R0112	8
%R0113	9
%R0114	10

### Instrukcja Move Array

Funkcja Array Move umożliwia skopiowanie określonej liczby elementów z tablicy źródłowej do tablicy docelowej. Każdy element w tablicy odwołuje się do odpowiedniej zmiennej. Instrukcja Array Move pozwala na uwzględnienie względnych lokacji w czasie przesuwania pomiędzy tablicami. Na przykład, trzy kolejne elementy rozpoczynając od elementu o indeksie 5 w tablicy źródłowej mogą zostać przesunięte do tablicy docelowej w trzy kolejne miejsca rozpoczynające się od elementu o indeksie 7.

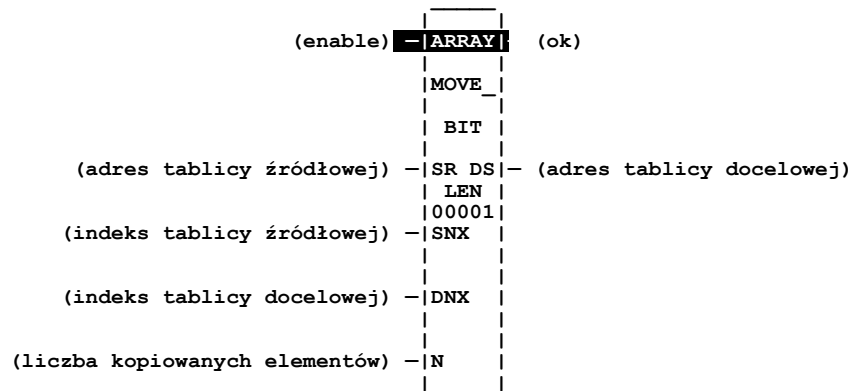
Funkcja ARRAY\_MOVE posiada pięć parametrów wejściowych i dwa parametry wyjściowe. Po uaktywnieniu, funkcja ta kopiuje N elementów ze źródłowej tablicy danych, począwszy od elementu indeksowanego przez parametr SNX. Elementy są zapisywane w tablicy wyjściowej rozpoczynając od lokacji indeksowanej przez DNX. Parametr LEN ilość elementów uzupełniających każdą tablicę.

Gdy dla funkcji ARRAY\_MOVE\_BIT (operującej na bitach) adres początkowy tablicy źródłowej i/lub docelowej leży w obszarze pamięci zorientowanym rejestrowo, pierwszym bitem tablicy jest najmniej znaczący bit wyszczególnionego rejestru. W oprogramowaniu Logicmaster wyświetlana wartość zawiera 16 bitów, bez względu na długość tablicy.

Wskaźniki instrukcji ARRAY\_MOVE liczone są od 1. Przy korzystaniu z funkcji ARRAY\_MOVE nie można odwoływać się do elementów położonych poza tablicą źródłową lub docelową (określonymi poprzez adres początkowy i długość).

Funkcja przesyła sygnał wyjściowy, chyba że wystąpi jeden z następujących przypadków:

- Do bloku funkcyjnego nie dopływa sygnał wejściowy.
- $(N + SNX - 1)$  jest większe od LEN. Wzór ten jest wykorzystywany przez sterownik do sprawdzenia czy nie nastąpiło przesunięcie elementów z poza tablicy źródłowej.
- $(N + DNX - 1)$  jest większe od LEN. Wzór ten jest wykorzystywany przez sterownik do sprawdzenia czy nie nastąpiło przesunięcie poza tablicę docelową.
- SNX lub DXN = 0.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji.
SR	Adres początkowy tablicy źródłowej. W przypadku funkcji ARRAY_MOVE_BIT, można podać dowolną zmienną dyskretną, bez uwzględniania podziału na bajty. Nie mniej jednak, w czasie pracy z programem Logicmaster wyświetlanych jest 16 bitów, począwszy od zadanego adresu.
SNX	SNX określa numer pierwszego elementu w tablicy źródłowej.
DNX	SNX określa numer pierwszego elementu w tablicy docelowej.
N	Określa ilość danych do przesunięcia.
ok	Sygnal wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego.
DS	Adres początkowy tablicy docelowej. W przypadku funkcji ARRAY_MOVE_BIT, można podać dowolną zmienną dyskretną, bez uwzględniania podziału na bajty. Nie mniej jednak, w czasie pracy wyświetlanych jest 16 bitów, począwszy od zadanego adresu.
LEN	Liczba elementów tworzących zarówno tablicę źródłową jak i docelową.

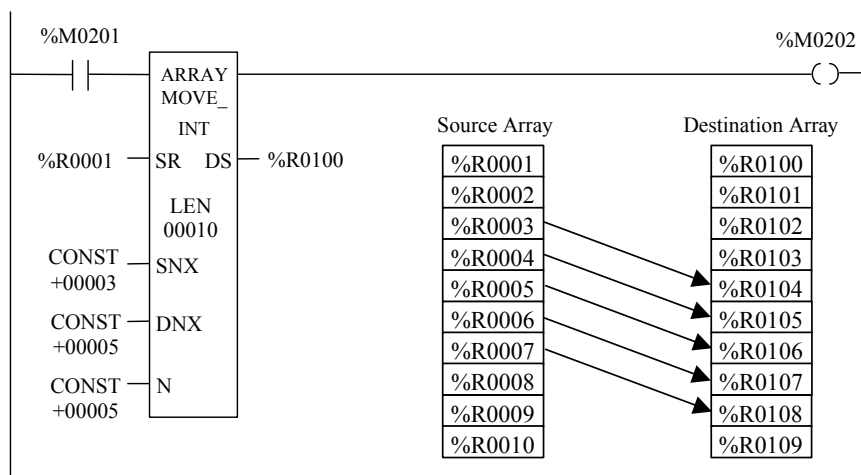
## Dopuszczalne typy zmiennych

Parametr	sygnal	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
SR		o	o	o	o	Δ†	o	•	•	•		
SNX		•	•	•	•		•	•	•	•	•	
DNX		•	•	•	•		•	•	•	•	•	
N		•	•	•	•		•	•	•	•	•	
ok	•											•
DS		o	o	o	o	†	o	•	•	•		

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego. W funkcji ARRAY\_MOVE\_BIT, wartości zmiennych dyskretnych %I, %Q, %M i %T nie muszą być podawane z uwzględnieniem podziału na bajty.
- o Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu INT, bit, byte lub Word. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu DINT.
- Δ Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu bit, byte lub Word. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu INT lub DINT.
- † Można stosować wyłącznie zmienne systemowe typu %SA, %SB i %SC. Nie można korzystać ze zmiennych typu %S.

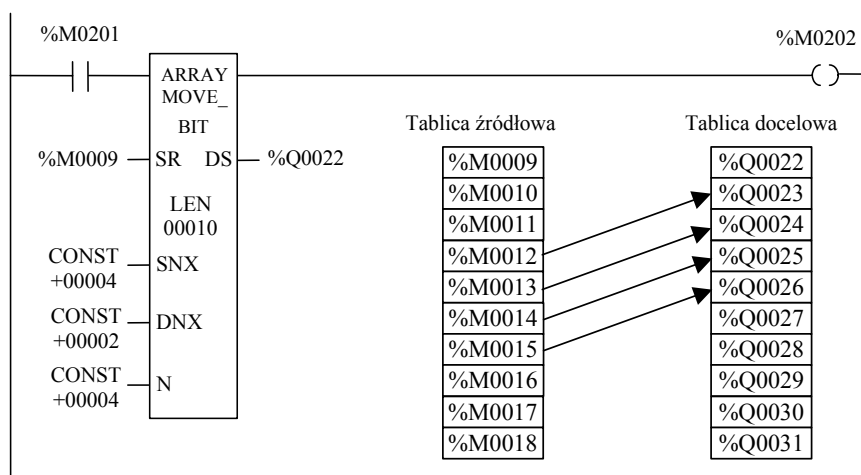
### Przykład 1

W przykładzie tym zarówno tablica źródłowa jak i docelowa posiadają 10 elementów (INT). Ich adresy początkowe są określone przez parametry SR i DS. Załączenie styku %M0201 powoduje kopiowanie pięciu elementów (N=5) z tablicy wejściowej do tablicy docelowej. Kopiowanych jest pięć elementów z tablicy źródłowej rozpoczynając od elementu o indeksie 3 (SNX=3). Kopiowane dane są umieszczane w tablicy docelowej począwszy od elementu o indeksie 5 (DNX=5). W związku z tym zawartość rejestrów %R0003 do %R0007 w tablicy źródłowej jest kopiowana do rejestrów %R0104 do %R0108 w tablicy docelowej.



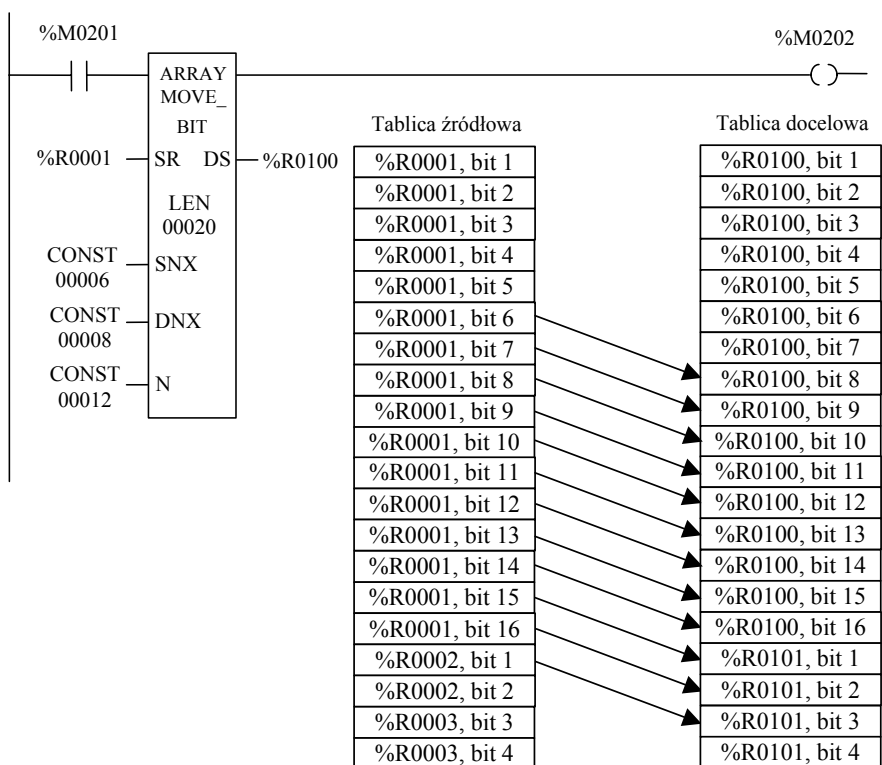
### Przykład 2

W przykładzie tym zarówno tablica źródłowa jak i docelowa posiadają 10 elementów (BIT). Ich adresy początkowe są określone przez parametry SR i DS. Załączenie styku %M0201 powoduje kopiowanie czterech elementów (N=4) z tablicy wejściowej do tablicy docelowej. Kopiowanych jest pięć elementów z tablicy źródłowej rozpoczynając od elementu o indeksie 4 (SNX=4). Kopiowane dane są umieszczane w tablicy docelowej począwszy od elementu o indeksie 2 (DNX=2). W związku z tym %M0012 do %M0015 w tablicy źródłowej są kopiowane od %Q0023 do %Q0026 w tablicy docelowej.



### Przykład 3

W przykładzie tym zarówno tablica źródłowa jak i docelowa posiadają 20 elementów (BIT). Ich adresy początkowe są określone przez parametry SR i DS. Załączenie styku %M0201 powoduje kopiowanie 12 elementów (N=12) z tablicy wejściowej do tablicy docelowej. Kopiowanych jest 12 elementów z tablicy źródłowej rozpoczynając od elementu o indeksie 6 (SNX=6). Kopiowane dane są umieszczane w tablicy docelowej począwszy od elementu o indeksie 8 (DNX=8). W związku z tym bity od 6 w %R0001 oraz pierwszy bit w %R0002 tablicy źródłowej są kopiowane do 8 bitów w %R0104 oraz 3 bitów w %R0108 tablicy docelowej.





## Funkcje do szukania danych

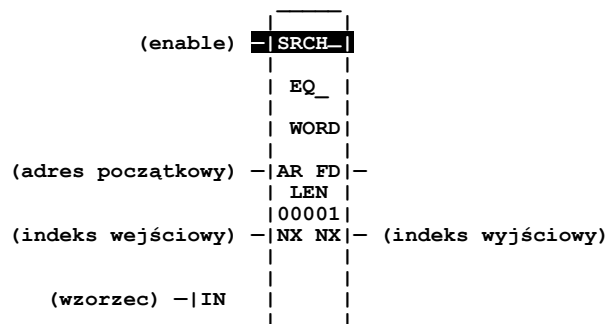
Funkcje te umożliwiają przeszukiwanie tablicy w celu znalezienia wszystkich wartości, spełniającej określone kryterium.

Oznaczenie skrótowe	Funkcja	Opis
SRCH_EQ	Szukanie wartości zadanej	Przeszukiwanie tablicy danych w celu znalezienia wartości równej wartości zadanej.
SRCH_NE	Szukanie wartości różnej	Przeszukiwanie tablicy danych w celu znalezienia wartości różnej od wartości zadanej.
SRCH_GT	Szukanie wartości większej	Przeszukiwanie tablicy danych w celu znalezienia wartości większej od wartości zadanej.
SRCH_GE	Szukanie wartości większej lub równej	Przeszukiwanie tablicy danych w celu znalezienia wartości większej lub równej wartości zadanej.
SRCH_LT	Szukanie wartości mniejszej	Przeszukiwanie tablicy danych w celu znalezienia wartości mniejszej od wartości zadanej.
SRCH_LE	Szukanie wartości mniejszej lub równej	Przeszukiwanie tablicy danych w celu znalezienia wartości mniejszej lub równej wartości zadanej.

Każda z tych funkcji posiada cztery parametry wejściowe i dwa parametry wyjściowe. Po dopłynięciu do bloku funkcyjnego sygnału, następuje rozpoczęcie przeszukiwania tablicy, począwszy od adresu "AR + NX" (początek tablicy plus indeks pierwszego elementu danych podlegających porównaniu, czyli parametr wejściowy NX).

Poszukiwanie jest kontynuowane do momentu znalezienia pierwszego elementu spełniającego zadany warunek lub do dojścia do końca tablicy. Jeśli poszukiwany element zostanie znaleziony, wartość parametru wyjściowego FD zostanie ustawiona na 1, a na wyjście oznaczone NX skopiowany zostanie indeks elementu, spełniającego warunek poszukiwań. Jeśli w przeszukiwanym obszarze nie zostanie znaleziony element spełniający warunek poszukiwań, wartości obydwu parametrów wyjściowych FD i NX są ustawiane na 0.

Poprawne wartości dla wejścia NX to od 0 do LEN — 1. Aby rozpocząć wyszukiwanie od pierwszego elementu NX powinno być ustawione na wartość 0. Podczas wykonywania operacji przeszukiwania tablicy wartość ta jest zwiększana o 1. Tak więc wartości parametru wyjściowego NX mogą zawierać się w przedziale od 1 do LEN. Jeśli wartość parametru wejściowego NX wychodzi poza zakres (jest mniejsza od 0 lub większa lub równa od LEN), jest ona ustawiana samoczynnie na 0.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie operacji.
AR	Adres początkowy tablicy danych, która ma zostać przeszukana.
NX wejściowe	Indeks pierwszego z elementów tablicy podlegającej przeszukaniu.
IN	Obiekt poszukiwań.
NX wyjściowe	Oznacza indeks znalezionej elementu.
FD	Wyjście załączane w momencie znalezienia obiektu.
LEN	Liczba elementów tworzących tablicę podlegającą przeszukiwaniu. Wartość ta musi mieścić się w zakresie od 1 do 32767 bitów lub słów.

## Dopuszczalne typy zmiennych

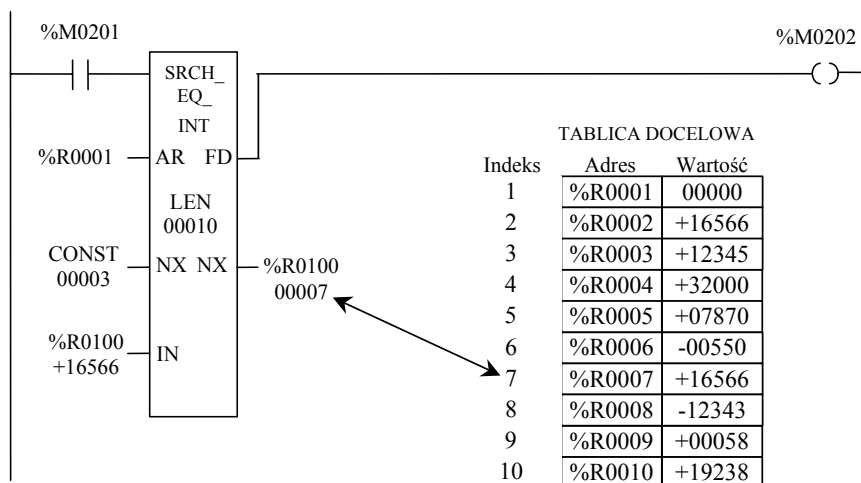
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
AR		o	o	o	o	Δ	o	•	•	•		
NX in		•	•	•	•		•	•	•	•	•	
IN		o	o	o	o	Δ	o	•	•	•	•	
NX out		•	•	•	•		•	•	•	•		
FD	•											•

- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu INT, byte lub Word. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu DINT.
- Δ Symbol ten oznacza, że zmienna danego typu może być wykorzystana wyłącznie jako parametr bloku funkcyjnego wykonującego działanie na danych typu byte lub Word. Zmiennych tych nie można wykorzystywać w bloku operującym na danych typu INT lub DINT.

### Przykład 1

Funkcja SRCH\_EQ (INT) przeszukuje blok pamięci od %R0001 (AR) do %R0010 (LEN=10.) Wartość poszukiwana jest określona na wejściu IN = +16566. Wejście NX = 3 oznacza że przeszukiwanie rozpocznie się na czwartym elemencie, ponieważ wartość NX jest inkrementowana o 1.

Załączenie styku %M0201, powoduje przeszukiwanie przez funkcje SRCH\_EQ tablicy, rozpoczynając od elementu o indeksie 4 aż do momentu znalezienia wartości +16566. Wartość tę posiada %R0007 (indeks 7) w związku z tym na wyjściu NX zostanie zapisana wartość 7 w rejestrze %R0100. Załączone zostanie również wyjście FD oznaczające znalezienie określonego obiektu w tablicy. Należy zwrócić uwagę, że również rejestr %R0002 ma poszukiwaną wartość +16566, ale nie został on uwzględniony w czasie przeszukiwania gdyż parametr NX posiada wartość 3 oznaczającą, że przeszukiwanie rozpocznie się na czwartym elemencie %R0004.



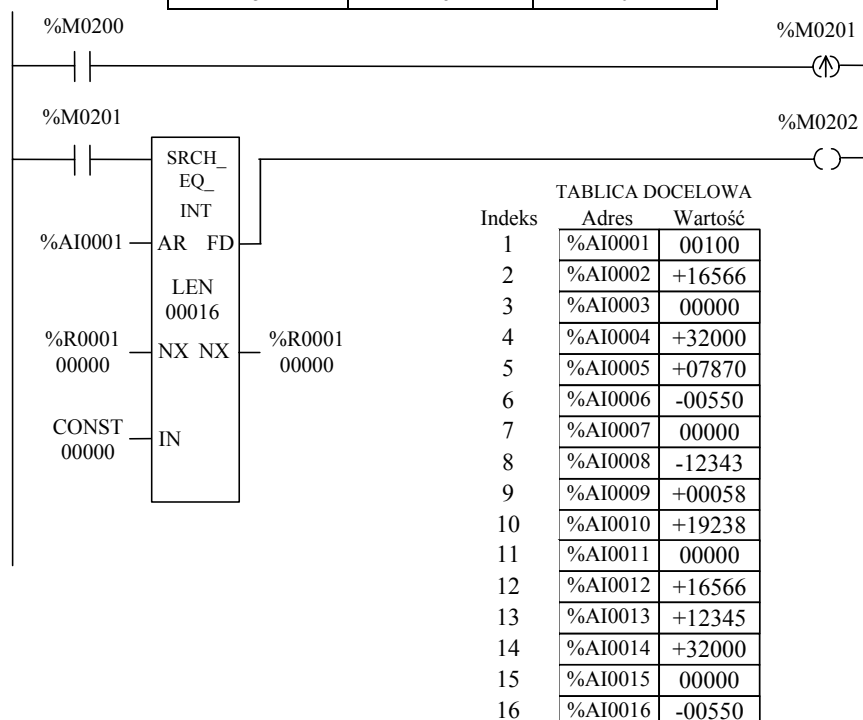
## Przykład 2

W poniższym przykładzie tablica danych rozpoczyna się o obszarze pamięci o adresie %AI0001 (określone przez AR), a kończy na %AI00016 (LEN=16). Wartość poszukiwana określona przez parametr IN wynosi +16566. Parametr NX równy 0 oznacza, że przeszukiwanie rozpocznie się od pierwszego elementu w tablicy, ponieważ wartość NX jest zwiększana o jeden w czasie wykonywania funkcji.

Pierwsze zwarcie styku %M0200 powoduje rozpoczęcie przeszukiwania tablicy po raz pierwszy rozpoczynając od pierwszego elementu aż do znalezienia wartości równej 00000. Wartość tę posiada element %AI0003 o indeksie 3, w związku z tym wartość ta zostanie zapisana na wyjściu i wejściu NX posiadają one adres %R0001. Załączone zostanie również wyjście FD oznaczające znalezienie określonego obiektu w tablicy.

Zwarcie styku %M0200 po raz drugi spowoduje przeszukiwanie tablicy od elementu o indeksie 4 gdyż wejście NX posiadające teraz wartość 3 jest inkrementowane o jeden po uaktywnieniu bloku. Poszukiwana wartość 00000 zostanie znaleziona w elemencie o indeksie 7 %AI0007, co oznacza że w rejestrze %R0001 zostanie zapisana wartość 7. Sytuacja będzie się powtarzać pięć razy. Jeśli nie zostanie znaleziona poszukiwana wartość, w rejestrze %R0001 zostanie zapisana wartość 0, co będzie oznaczać że następne przeszukiwanie zostanie rozpoczęte od pierwszego elementu tablicy.

Nr przeszukiwania	Element od którego rozpocznie się przeszukiwanie	Wynik przeszukiwania (%R0001)
1	1	3
2	4	7
3	8	11
4	12	15
5	16	0



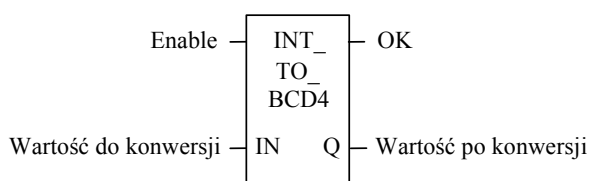
Funkcje te umożliwiają konwersję danych. Konieczność taka wynika z faktu, że wiele bloków funkcyjnych musi operować na danych tego samego typu (jak np. funkcje matematyczne). W punkcie tym opisano następujące funkcje do konwersji:

Oznaczenie skrótowe	Funkcja	Opis	Strona
BCD-4	Konwersja na dane typu BCD	Konwersja danych typu INT na kod BCD.	11-2
INT	Konwersja na dane typu INT	Konwersja danych typu BCD-4 na dane typu INT.	11-3
DINT	Konwersja na dane typu DINT	Konwersja danych typu REAL na dane typu DINT.	11-5
REAL	Konwersja na dane typu REAL	Konwersja danych typu INT, DINT, BCD-4 lub WORD na dane typu REAL.	11-7
WORD	Konwersja na dane typu WORD	Konwersja danych typu REAL na dane typu WORD	11-9
TRUN	Odrzucanie części dziesiętnej	Zaokrąglenie liczby rzeczywistej poprzez odrzucenie części dziesiętnej.	11-11

## —>BCD-4 (INT)

Blok funkcyjny > BCD-4 stosowany jest do zastąpienia danych zapisanych jako liczba całkowita ze znakiem (typ INT) przez równoważną liczbę zapisaną w kodzie BCD-4 (liczba dziesiętna zakodowana w układzie dwójkowym). Dane wejściowe nie ulegają zmianie wskutek działania tej funkcji. Konwersja danych na system zapisu BCD może być przeprowadzona np. w celu podłączenia diodowego wyświetlacza cyfr sterowanego w kodzie BCD lub w celu przesłania kodów sterujących do niektórych urządzeń zewnętrznych, jak np. do licznika wysokiej częstotliwości.

Gdy do bloku funkcyjnego dopływa sygnał wejściowy, zostaje dokonana konwersja zadanej poprzez parametr IN wartości, a jej wynik jest dostępny poprzez parametr Q. Sygnał wyjściowy jest przesyłany, gdy do bloku funkcyjnego dopłyne sygnał wejściowy, chyba że po dokonaniu konwersji wartość parametru Q przekracza zakres od 0 do 9999.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie konwersji.
IN	Wartość stała lub adres zmiennej, której wartość podlega konwersji na system zapisu BCD-4.
ok	Sygnał wyjściowy, wysyłany po poprawnym wykonaniu funkcji.
Q	Wartość parametru wejściowego IN po dokonaniu konwersji.

### Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•		•	•	•	•	•	
ok	•											•
Q		•	•	•	•		•	•	•	•		

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

### Przykład

W zamieszczonym poniżej przykładzie, po każdej zmianie wartości zmiennej %I0002 na 1, jeżeli nie wystąpiły żadne błędy, parametry wejściowe z obszaru %M0017 do %M0032 są konwertowane na liczby w kodzie BCD, a wynik zapisywany jest w pamięci ograniczonej adresami %Q0033 do %Q0048. Przekaznik %M0032 jest załączany po prawidłowym wykonaniu konwersji.



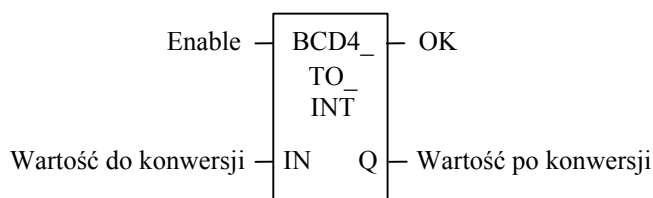
## —>INT (BCD-4, REAL)

Blok funkcyjny > INT stosowany jest do konwersji danych typu BCD-4 lub REAL na równoważną liczbę całkowitą za znakiem (INT). Dane wejściowe nie ulegają zmianie wskutek działania tej funkcji.

### Uwaga

The REAL data type is only available on 35x and 36x series CPUs, Release 9 or later, and on all releases of CPU352 and CPU37x. Operacje na liczbach rzeczywistych można realizować wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x.

Gdy do bloku funkcyjnego dopływa sygnał wejściowy, zostaje dokonana konwersja zadanej poprzez parametr IN wartości, a jej wynik jest dostępny poprzez parametr Q. Sygnał wyjściowy jest przesyłany, gdy do bloku funkcyjnego dopłyne sygnał wejściowy, chyba że po dokonaniu konwersji wartość parametru Q przekracza dozwolony zakres.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie konwersji.
IN	Wartość stała albo adres zmiennej typu BCD lub REAL, której wartość podlega konwersji na typ INT.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego i dokonania konwersji bez wystąpienia przekroczenia dozwolonego zakresu wartości.
Q	Wartość parametru wejściowego IN po dokonaniu konwersji.

### Dopuszczalne typy zmiennych

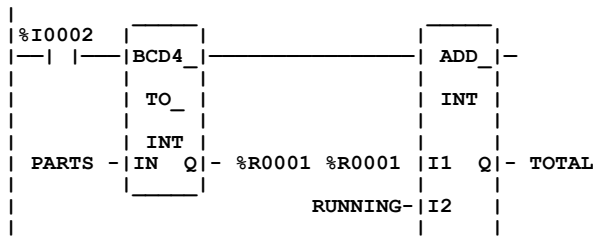
Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		•	•	•	•		•	•	•	•	•	
ok	•											•
Q		•	•	•	•		•	•	•	•		

**Uwaga:** W przypadku danych REAL, dopuszczalne jest stosowanie wyłącznie zmiennych typu %R, %AI i %AQ.

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

### Przykład 1 - konwersja danych typu BCD4 na INT

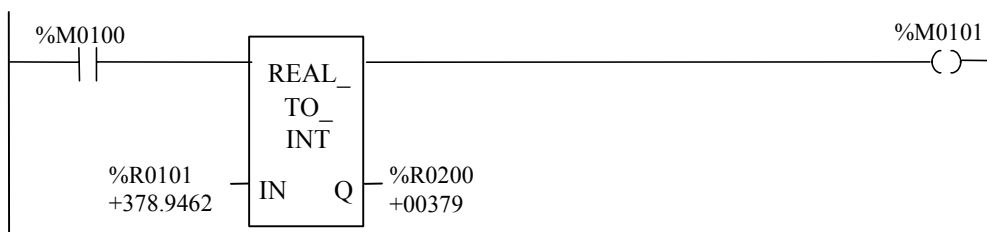
W poniższym przykładzie złączenie wejścia %I0002, powoduje konwersję danych BCD4 (PARTS) na dane typu INT zapisane w rejestrze %R0001. Następnie blok ADD dodaje wartości z zapisane w %R0001 i RUNNING. Suma tych dwóch liczb jest zwracana przez funkcję ADD jako parametr TOTAL.



### Przykład 2 – konwersja danych REAL na INT

W przykładzie tym wartość rzeczywista %R0101 jest zamienia na wartość typu INT %R0200. Zwarcie styku %M0100 powoduje wykonanie konwersji. Należy zwrócić uwagę, że w czasie konwersji liczba jest zaokrąglana do najbliższej liczby całkowitej. Jeśli część dziesiętna jest większa niż 0,5 liczba zostanie zaokrąglona w górę. Natomiast zostanie odrzucona część dziesiętna jest mniejsza niż 0,5. W poniższym przykładzie wartość rzeczywista 378,9462 zostanie zaokrąglona do 379.

Jeśli nie jest wymagane zaokrąglanie, należy wykorzystać funkcję REAL\_TRUN\_INT, która w czasie konwersji odrzuci część dziesiętną liczby rzeczywistej niezależnie od jej wartości.





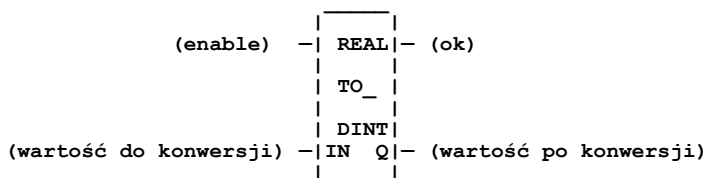
## —>DINT (REAL)

Funkcja ta umożliwi konwersję danych typu REAL na dane typu DINT. Dane wejściowe nie ulegają zmianie wskutek działania tej funkcji.

### Uwaga

Operacje na liczbach rzeczywistych można realizować wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 oraz CPU37x.

Gdy do bloku funkcyjnego dopływa sygnał wejściowy, zostaje dokonana konwersja zadanej wartości, a jej wynik jest dostępny poprzez parametr Q. Sygnał wyjściowy jest przesyłany, gdy do bloku funkcyjnego dopłyne sygnał wejściowy, chyba że po dokonaniu konwersji wartość parametru Q przekracza dozwolony zakres.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie konwersji.
IN	Wartość stała lub adres zmiennej, której wartość podlega konwersji typ DINT.
ok	Sygnał wyjściowy, pojawiający się po dopłynięciu do bloku funkcyjnego sygnału wejściowego i dokonaniu konwersji, o ile liczba rzeczywista mieści się w dozwolonym zakresie wartości.
Q	Wartość parametru wejściowego IN po dokonaniu konwersji.

### Uwaga

W przypadku konwersji danych typu REAL na dane typu DINT w wyniku może zostać otrzymana liczba o mniejszej dokładności, ponieważ dane typu REAL posiadają 24 bity znaczące.

### Dopuszczalne typy zmiennych

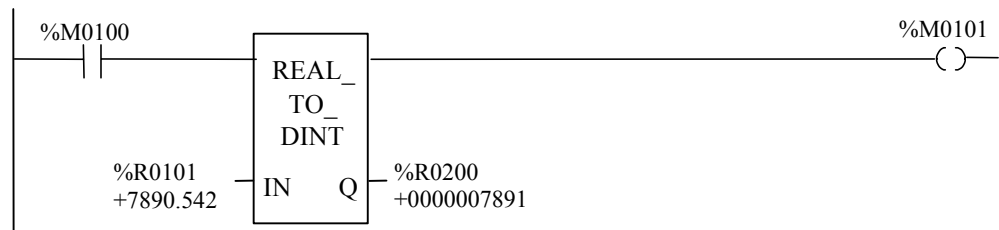
Parametr	sygnal	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		o	o	o	o		o	•	•	•	•	
ok	•											•
Q								•	•	•		

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %M0100 na 1 powoduje konwersję parametru wejściowego %R0101 na liczbę całkowitą podwójnej precyzji ze znakiem i zapisanie wyniku w parametrze wyjściowym %R0200. Należy zwrócić uwagę, że w czasie konwersji liczba jest zaokrąglana do najbliższej liczby całkowitej. Jeśli część dziesiętna jest większa niż 0,5 liczba zostanie zaokrąglona w górę. Natomiast zostanie odrzucona część dziesiętna jest mniejsza niż 0,5. W poniższym przykładzie wartość rzeczywista 7890,542 zostanie zaokrąglona do 7891.

Jeśli nie jest wymagane zaokrąglanie, należy wykorzystać funkcję REAL\_TRUNC\_DINT, która w czasie konwersji odrzuci część dziesiętną liczby rzeczywistej niezależnie od jej wartości.



## —>REAL (INT, DINT, BCD-4, WORD)

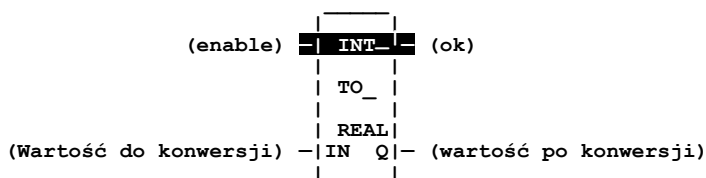
Blok funkcyjny > REAL stosowany jest do konwersji danych typu INT, DINT, BCD-4 lub WORD na równoważną liczbę rzeczywistą (REAL). Dane wejściowe nie ulegają zmianie wskutek działania tej funkcji.

Gdy do bloku funkcyjnego dopływa sygnał wejściowy, zostaje dokonana konwersja zadanej poprzez parametr IN wartości, a jej wynik jest dostępny poprzez parametr Q. Sygnał wyjściowy jest przesyłany, gdy do bloku funkcyjnego dopłyne sygnał wejściowy, chyba że po dokonaniu konwersji wartość parametru Q przekracza zakres od 0 do FFFFh.

W przypadku konwersji danych typu DINT na dane typu REAL w wyniku może zostać otrzymana liczba o mniejszej dokładności, ponieważ liczba bitów znaczących zostaje zredukowana do 24.

### Uwaga

Funkcja ta jest dostępna wyłącznie w jednostkach centralnych 35x i 36x, wersja 9 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 i CPU37x.



### Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie konwersji.
IN	Wartość stała lub adres zmiennej, której wartość podlega konwersji na typ REAL.
ok	Sygnał wyjściowy, wysyłany po poprawnym wykonaniu funkcji.
Q	Wartość parametru wejściowego IN po dokonaniu konwersji.

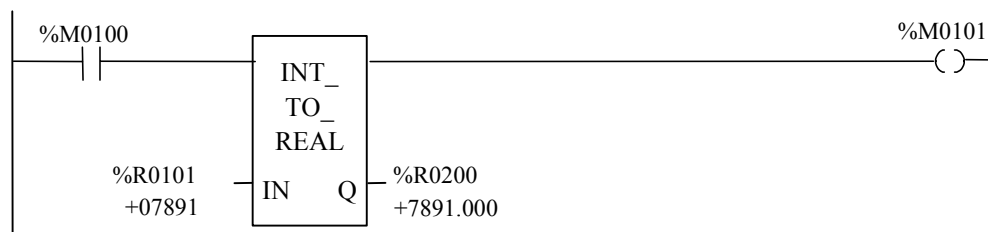
### Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN		o	o	o	o		o	•	•	•	•	
ok	•											•
Q								•	•	•		

- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Zmienna danego typu nie może być zastosowana w przypadku konwersji typu DINT na REAL.

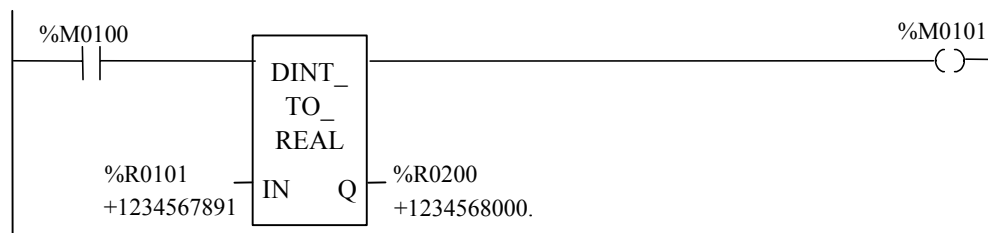
### Przykład 1 – konwersja danych typu INT na REAL

W poniższym przykładzie parametrem wejściowym jest liczba typu INT +07891. Wartość rzeczywista +7891,000 po konwersji jest zapisywana w rejestrze %R0200.



### Przykład 2 – konwersja danych typu DINT na REAL

W poniższym przykładzie, parametrem wejściowym IN jest liczba całkowita podwójnej precyzji +1234568000. Należy zwrócić uwagę, że liczby tego typu mają 10 najbardziej znaczących miejsc podczas gdy liczby typu REAL – 7. W związku z tym w czasie konwersji liczba całkowita jest zaokrąglana do 7 znaczących miejsc. W przykładzie tym cztery ostatnie znaki (7891) liczby całkowitej podwójnej precyzji są zaokrąglane do 8000 w liczbie rzeczywistej.



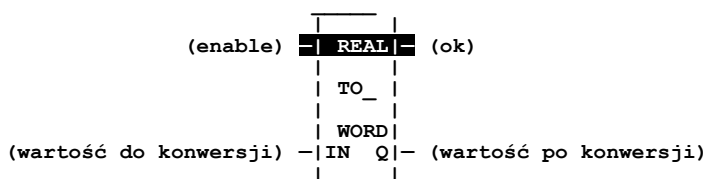
## —>WORD (REAL)

Blok funkcyjny > WORD stosowany jest do konwersji danych typu REAL na równoważną liczbę typu WORD. Dane wejściowe nie ulegają zmianie wskutek działania tej funkcji.

### Uwaga

Funkcja ta dostępna jest wyłącznie w jednostkach centralnych serii 35x i 36x.

Gdy do bloku funkcyjnego dopływa sygnał wejściowy, zostaje dokonana konwersja zadanej poprzez parametr IN wartości, a jej wynik jest dostępny poprzez parametr Q. Sygnał wyjściowy jest przesyłany, gdy do bloku funkcyjnego dopłyne sygnał wejściowy, chyba że po dokonaniu konwersji wartość parametru Q przekracza zakres od 0 do FFFFh.



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie konwersji.
IN	Wartość stała lub adres zmiennej, której wartość podlega konwersji na typ WORD.
ok	Sygnał wyjściowy, wysyłany po poprawnym wykonaniu funkcji.
Q	Wartość parametru wejściowego IN po dokonaniu konwersji.

## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN								•	•	•	•	
ok	•											•
Q		•	•	•	•		•	•	•	•		

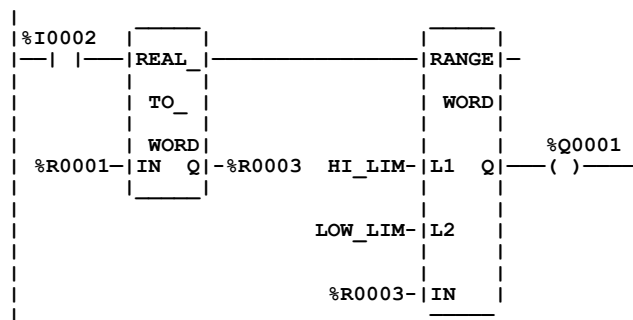
- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład – konwersji danych typu REAL na WORD

Ponieważ funkcja RANGE nie może operować na danych typu REAL, wartość rzeczywista z rejestru %R0001 jest w pierwszym rzędzie konwertowana na wartość typu WORD (%R0003), która następnie jest wprowadzana do bloku RANGE WORD.

W poniższej tabeli zamieszczono wartości na wejściach i wyjściach funkcji.

Pozycja	Wartość lub stan
%R0001	15767.83
%R0003	3A89h (15,768 dziesiętnie)
HI LIM	4E20h (20,000 dziesiętnie)
LOW LIM	2710h (10,000 dziesiętnie)
Q1	ON



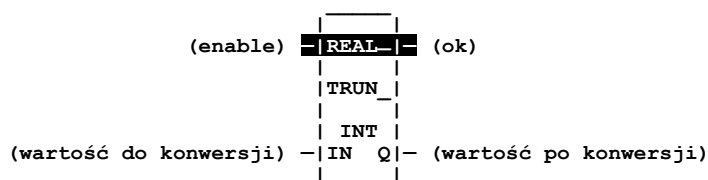
## TRUN (INT, DINT)

Funkcja ta umożliwia zaokrąglenie liczby rzeczywistej poprzez odrzucenie części dziesiętnej. W czasie konwersji wszystkie liczby po prawej stronie przecinka są odrzucane. Liczba wejściowa nie ulega zmianie wskutek działania tej funkcji.

### Uwaga

Operacje na liczbach rzeczywistych mogą realizować wyłącznie jednostki centralne 35x i 36x (wersja 9 i nowsze oraz wszystkie wersje jednostek centralnych CPU352) oraz 37x, tak więc funkcja TRUN nie jest obsługiwana przez pozostałe jednostki centralne serii 90-30.

Gdy do bloku funkcyjnego dopływa sygnał wejściowy, zostaje dokonana konwersja zadanej przez parametr IN wartości, a jej wynik jest dostępny poprzez parametr Q. W jednostce centralnej 352 sygnał wyjściowy jest przesyłany, gdy do bloku funkcyjnego dopłyne sygnał wejściowy, chyba że po dokonaniu konwersji wartość parametru Q przekracza dozwolony zakres lub parametr IN nie jest wartością liczbową. W przypadku innych jednostek centralnych serii 35x i 36x/37x, funkcja *nie przesyła* sygnału wyjściowego.



## Parametry

Parametr	Opis
Enable	Sygnał wejściowy. Doprowadzenie tego sygnału do bloku funkcyjnego powoduje wykonanie konwersji.
IN	Wartość stała lub adres zmiennej, której wartość ma zostać zaokrąglona.
Ok	Sygnał wyjściowy, wysyłany gdy działanie zostało poprawnie wykonane, jego wynik mieści się w dopuszczalnym przedziale wartości a parametr wejściowy IN jest liczbą.
Q	Wartość zaokrąglonego parametru wejściowego INT lub DINT po dokonaniu konwersji.

### Uwaga

W przypadku konwersji danych typu REAL na dane typu DINT w wyniku może zostać otrzymana liczba o mniejszej dokładności, ponieważ dane typu REAL posiadają 24 bity znaczące.

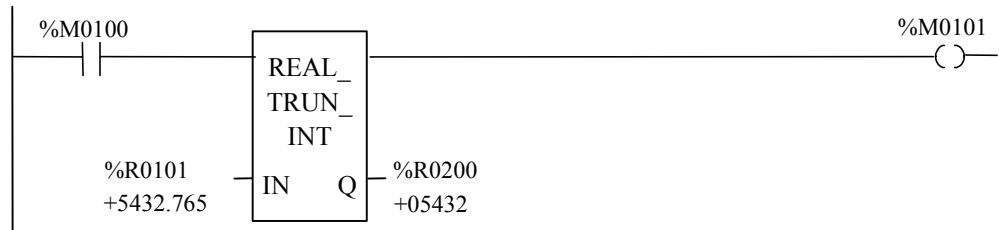
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
IN								•	•	•	•	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Symbol ten oznacza, że zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.
- o Wyłącznie w przypadku zaokrąglania danych typu REAL do danych typu INT.

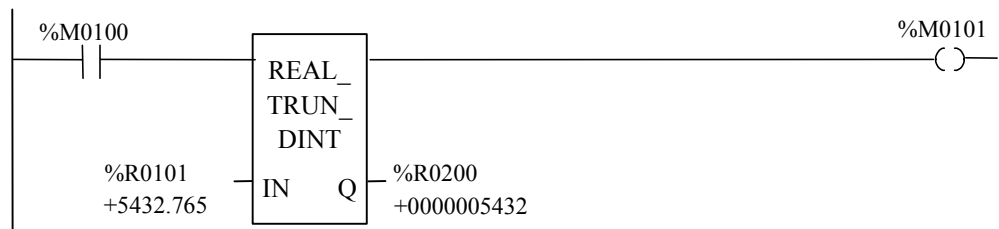
### Przykład 1 – Konwersja liczby typu REAL na INT (CPU352)

W przykładzie tym wartość z rejestru %R0101 jest zaokrąglona przez odrzucenie części dziesiętnej, a otrzymana liczba całkowita +05432 jest zapisywana w rejestrze %R0200. W przypadku jednostki centralnej 352 załączony został by przełącznik %M0101 informujący o prawidłowym wykonaniu operacji. W przypadku innych jednostek centralnych serii 35x i 36x/37x nie zostanie załączony sygnał OK.



### Przykład 2 – Konwersja liczby typu REAL na DINT (CPU352)

W przykładzie tym wartość z rejestru %R0101 jest zaokrąglona przez odrzucenie części dziesiętnej, a otrzymana liczba całkowita +0000005432 jest zapisywana w rejestrze %R0200. W przypadku jednostki centralnej 352 załączony został by przełącznik %M0101 informujący o prawidłowym wykonaniu operacji. W przypadku innych jednostek centralnych serii 35x i 36x/37x nie zostanie załączony sygnał OK.





# Rozdział 12

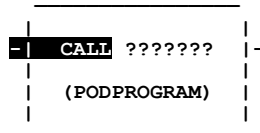
## Funkcje sterujące

W niniejszym rozdziale opisano funkcje sterujące wykonywaniem programu, umożliwiające zmianę sposobu jego działania lub wprowadzenie pewnych ograniczeń.. Cykle pracy sterownika opisano w Rozdziale 2, części "Cykl pracy sterownika".

Funkcja	Opis	Strona
CALL	Wywołanie podprogramu w danym miejscu programu sterującego.	12-2
DOIO	Natychmiastowe uaktualnienie stanu wybranych wejść lub wyjść na okres jednego cyklu pracy sterownika. (Funkcja ta obsługuje wszystkie wejścia i wyjścia modułu, wyszczególnione jako parametry jej wywołania. Nie jest możliwe uaktualnienie wybranych (częściowe) wejść i wyjść modułu. Opcjonalnie można umieścić kopię obsługiwanych wejść i wyjść w pamięci wewnętrznej, a nie w standardowej pamięci wejść dyskretnych.	12-2
SER	Blok funkcyjny SER (Sequential Event Recorder) przeznaczony jest do rejestrowania serii próbek. Sposób wykonywania tego bloku funkcyjnego uzależniony jest od parametrów, definiowanych przez użytkownika w bloku danych sterujących.	12-8
END	Koniec programu. Program wykonywany jest od pierwszego szczebla drabiny logicznej aż do ostatniego szczebla, jeżeli jednak napotkana zostanie instrukcja END, wykonanie programu zostaje bezwarunkowo przerwane. Funkcja END jest użyteczna podczas uruchamiania programu, nie jest jednak dozwolone korzystanie z niej przy programowaniu w języku SFC (proszę porównać z uwagą na stronie 12-8).	12-23
MCR i MCRN	Przełącznik sterujący wykonaniem fragmentu programu. Blok funkcyjny MCR wyznacza początek fragmentu programu sterującego, w którym wszystkie szczeble zostaną wykonane bez dopływu sygnału sterującego. Koniec takiego fragmentu wyznaczony jest przez odpowiedni blok funkcyjny ENDMCR. Oprogramowanie Logicmaster 90-30/20/Micro udostępnia dwa rodzaje funkcji MCR: z możliwością zagnieżdżenia (MCRN) i bez możliwości zagnieżdżenia (MCR).	12-23
ENDMCR i ENDMCRN	Instrukcja ta odwołuje wcześniejszą instrukcję MCR (MCRN). Oprogramowanie Logicmaster 90-30/20/Micro udostępnia dwa rodzaje funkcji ENDMCR: z możliwością zagnieżdżenia (ENDMCRN) i bez możliwości zagnieżdżenia (ENDMCR).	12-30
JUMP i JUMPN	Przejdźcie do innego miejsca w programie sterującym (oznaczonego instrukcją LABEL, proszę porównać z zamieszczoną poniżej uwagą). Oprogramowanie Logicmaster 90-30/Micro udostępnia dwa rodzaje funkcji JUMP: bez możliwości pokrywania się zakresów działania (JUMP) i z możliwością pokrywania się zakresów działania (JUMPN).	12-31
LABEL i LABELN	Miejsce docelowe dla instrukcji skoku (JUMP). Oprogramowanie Logicmaster 90-30/Micro udostępnia dwa rodzaje funkcji LABEL: bez możliwości pokrywania się zakresów działania (LABEL) i z możliwością pokrywania się zakresów działania (LABELN).	12-33
COMMENT	Wstawienie komentarza (objaśnienia danego szczebla programu sterującego). Tekst komentarza można wprowadzić oraz wyświetlić na ekranie komputera w celu dokonania w nim ewentualnych zmian po zaakceptowaniu szczebla z wstawionym blokiem funkcyjnym COMMENT, poprzez ustawienie kursora w miejscu, w którym znajduje się blok i naciśnięcie klawisza F10 (Zoom).	12-34
SVCREQ	Wywołanie funkcji specjalnej sterownika. (Proszę porównać z listą bloków funkcyjnych SVCREQ, zamieszczono na stronie 12-35.)	12-34
PID	Dwa regulatory proporcjonalno- całkowo- różniczkowe: <ul style="list-style-type: none"> <li>• Standardowy regulator PID ISA (PIDISA).</li> <li>• Regulator PID o niezależnych wyrazach (PIDIND).</li> </ul>	12-70

# CALL

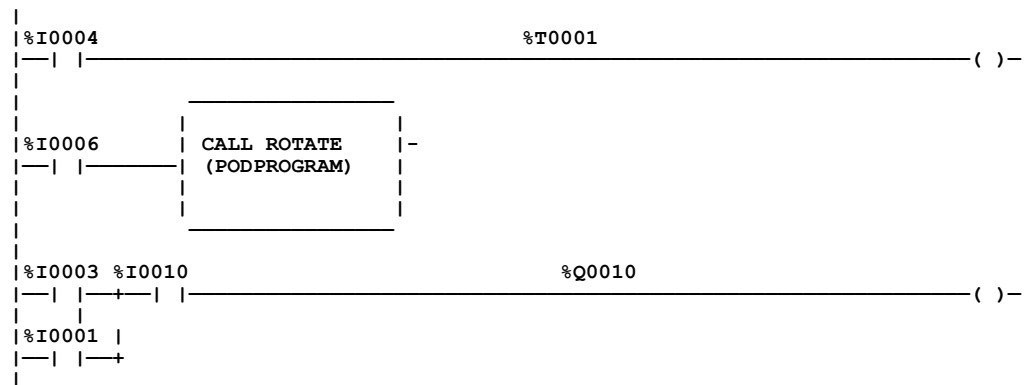
Blok funkcyjny CALL umożliwia wywołanie podprogramu w danym miejscu programu sterującego.



Gdy do bloku funkcyjnego CALL dopływa sygnał wejściowy, powoduje on natychmiastowe wywołanie i wykonanie żadanego podprogramu. Po zakończeniu wykonywania podprogramu, sygnał powraca do szczebla następującego bezpośrednio po bloku CALL.

## Przykład

W przykładzie tym po załączeniu styku %I0006 instrukcja CALL wywołuje podprogram ROTATE. (Należy zwrócić uwagę że przed wprowadzeniem funkcji CALL należy zadeklarować podprogram, który będzie wywoływany). Ustawienie kursora wewnątrz instrukcji CALL i naciśnięcie klawisza **F10** powoduje przejście do podprogramu. Po wywołaniu podprogram zostanie wykonany po czym sygnał powróci do szczebla programu poniżej funkcji CALL. W przykładzie tym podprogram jest wywoływany w drugim szczeblu, więc po jego wykonaniu wykonywanie programu zostanie wznowione w trzecim szczeblu.



### Uwaga

Sterowniki Micro nie udostępniają możliwości korzystania z podprogramów. Nie można więc, w przypadku tych sterowników, korzystać z instrukcji CALL.

## DOIO

Blok funkcyjny DOIO stosowany jest w celu natychmiastowego uaktualnienia stanu wybranych wejść lub wyjść podczas wykonywania programu sterującego, w czasie trwania cyklu pracy sterownika. Funkcja ta umożliwia dodatkowe uaktualnienie wybranych wejść/wyjść, poza normalną obsługą wejść i wyjść, wykonywaną odpowiednio na początku i przy końcu każdego z cykli pracy sterownika. W normalnych warunkach tabela stanu wejść jest aktualizowana w czasie ich odczytywania i nie jest nadpisywana aż do następnego cyklu pracy. Tabela stanu wyjść jest aktualizowana w czasie wykonywania części logicznej programu, ale moduły nie są aktualizowane aż nie zakończy się wykonywanie części logicznej programu. Wykorzystując funkcję DO I/O można aktualizować stan tabeli wejść modułów wyjść w czasie wykonywania części logicznej programu. Właściwość ta pozwala na szybsze odczytywanie zmian stanu wejść i zapisywanie wyjść niż wynikałoby to z cyklu pracy sterownika. W rozdziale 2 zamieszczono więcej informacji na temat cyklu pracy sterownika.

W przypadku wyszczególnienia jako parametrów bloku funkcyjnego zmiennych przypisanych wejściom sterownika, funkcja DOIO umożliwia wykorzystanie przez program sterujący najświeższych wartości fizycznych wejść sterownika. Jeśli jako parametry bloku funkcyjnego podane są zmienne przypisane wyjściom sterownika, funkcja DOIO wykorzystuje najświeższe wartości zapisane w pamięci sterownika przez inne elementy logiczne programu sterującego. Wejścia i wyjścia są obsługiwane narastająco dla całych modułów wejść/wyjść, jeżeli to jest konieczne, sterownik uaktualnia zmienne w trakcie wykonywania funkcji.

### Praca z modułami wejść

Funkcja DOIO posiada cztery parametry wejściowe i jeden parametr wyjściowy. Gdy do bloku funkcyjnego dopływa sygnał wejściowy i za pomocą parametrów bloku wyszczególnione są wejścia, których stan ma zostać uaktualniony, przejrzany zostaje zbiór wejść o adresie początkowym ST i adresie końcowym END. Dalsze działanie funkcji uzależnione jest od tego, czy w bloku funkcyjnym wykorzystywany jest parametr ALT (adres). Jeśli podano parametr ALT, uaktualnione zostają wartości zmiennych, rozpoczynając od określonego za pomocą parametru ALT adresu (może to być na przykład obszar pamięci przypisany zmiennym typu %M lub %R), a standardowa pamięć sygnałów wejść dyskretnych nie jest uaktualniana. Typ parametru ALT musi być taki sam jak typ aktualizowanej zmiennej. Jeśli jako parametry ST i END występują zmienne dyskretne, parametr ALT musi być również zmienną dyskretną. Jeśli parametr ALT nie występuje, uaktualniona zostaje odpowiednia tabela wejść.

### Praca z modułami wyjść

Gdy do bloku funkcyjnego dopływa sygnał wejściowy i za pomocą parametrów bloku wyszczególnione są wyjścia, których stan ma zostać uaktualniony, do modułów wyjść przesłane zostają wartości zmiennych przypisanych blokowi wyjść o adresie początkowym ST i adresie końcowym END. Jeśli na wyjścia sterownika mają zostać przesłane wartości z innego obszaru pamięci niż %Q czy %AQ, początkowy adres tego obszaru musi zostać podany za pomocą parametru ALT. Wartości z tego obszaru są następnie przesyłane do zbioru wyjść adresowanego przez parametry ST i END.

Funkcja DOIO jest wykonywana do momentu, aż wszystkie wyszczególnione wejścia lub wyjścia zostaną obsługane. Po wykonaniu tej funkcji program sterujący przechodzi do kolejnej instrukcji.

### Praca z modułami dodatkowymi

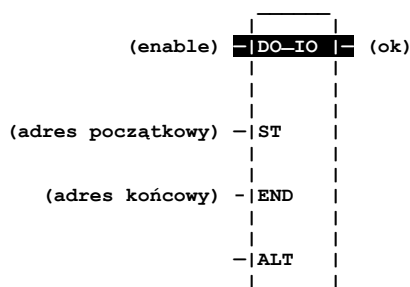
Jeżeli wyszczególnione zmienne obejmują moduły opcjonalne (HSC, APM, itp.), obsługane zostaną wszystkie dane wejściowe (%I i %AI) i wszystkie dane wyjściowe (%Q i %AQ) dla tego modułu PAGE sens. Parametr ALT jest ignorowany w czasie aktualizacji modułów opcjonalnych. Także, jeśli wymagane jest wykorzystanie funkcji z modułem Enhanced GCM (IC693CMM302) należy spełnić poniższe wymagania.

### Uwaga

Tylko jednostki centralne w wersji 9 lub nowszej obsługują funkcję DOIO z modułami Enhanced GCM (IC693CMM302).

Sygnal wyjściowy jest przesyłany zawsze wtedy, gdy do bloku dopływa sygnał wejściowy, o ile nie wystąpi żaden z następujących przypadków:

- Nie wszystkie zmienne wyszczególnionego typu, z zadanego obszaru pamięci, są fizycznie zrealizowane (np. brakuje modułu wejść lub wyjść, którym przypisane są zmienne wyszczególnione jako parametry bloku funkcyjnego).
- Jednostka centralna sterownika nie jest w stanie zrealizować instrukcji.
- Obszar pamięci wyszczególniony za pomocą parametrów bloku zawiera moduły, których brak został zgłoszony w postaci komunikatu alarmowego "Brak modułu wejść/wyjść".



### Parametry

Parametr	Opis
enable	Po uaktywnieniu wykonywana jest operacja uaktualnienia stanu wejść/wyjść sterownika.
ST	Adres początkowy grupy wejść/wyjść, które mają być uaktualnione.
END	Adres końcowy grupy wejść/wyjść, które mają być uaktualnione.
ALT	W przypadku uaktualniania wejść, parametr ten zawiera adres początkowy gdzie zapisywany jest stan skowanych wejść. W przypadku uaktualniania wyjść, parametr ALT określa adres obszaru w pamięci sterownika, z którego poszczególne wartości mają zostać skopiowane na odpowiadające im fizyczne wyjścia sterownika. W jednostkach centralnych model 331 i późniejszych, parametr ALT może wpływać na prędkość wykonywania bloku funkcyjnego DOIO (porównać z zamieszczoną poniżej Uwagą, oraz punktem poświęconym rozszerzonej funkcji DO/IO, dostępnej w jednostkach centralnych model 331 i późniejszych). <b>Jeśli funkcja ALT nie jest wykorzystywana, wejście to należy pozostawić puste, wprowadzenie wartości może spowodować błędy zegara alarmowego.</b>
ok	Sygnal wyjściowy wysyłany po pomyślnym wykonaniu funkcji.

### Uwaga

Rozszerzona funkcja DOIO jest dostępna w jednostkach centralnych 311 i późniejszych. W rozszerzonej funkcji DOIO, parametr ALT może być wykorzystywany do wpisywania numeru gniazda w kasecie głównej z pojedynczym modułem wejść lub wyjść. Wykonanie bloku funkcyjnego DOIO będzie trwać 80 mikrosekund, a nie 236 mikrosekund, jak to ma miejsce w przypadku nie wykorzystywania parametru ALT. Nie ma żadnego mechanizmu do sprawdzania pokrywania się adresów zmiennych lub do badania niezgodności typów modułów. Bardziej szczegółowe informacje na temat rozszerzonej funkcji DOIO opisano w dalszej części rozdziału.

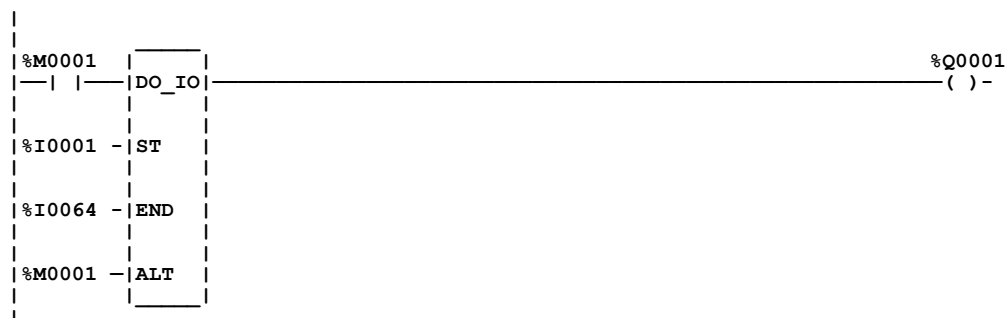
### Dopuszczalne typy zmiennych

Parametr	sygnal	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
ST		•	•						•	•		
END		•	•						•	•		
ALT		•	•	•	•		•	•	•	•		•
ok	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Przykład 1 - Uaktualniania stanu wejść

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %M0001 na 1, powoduje odczytanie zmiennych od %I0001 (ST) do %I0064 (END) oraz ustawienie wartości zmiennej %Q0001 na 1. Kopia odczytanych wejść umieszczona zostanie w pamięci wewnętrznej, w obszarze o adresach od %M0001 do %M0064. Ponieważ na wejście ALT podawany jest zmienny adres, tablica stanu wejść %I nie jest uaktualniana przez funkcję DOIO. Takie zastosowanie funkcji pozwala na porównanie bieżących wartości wejść z wartościami wejść odczytanymi na początku cyklu.



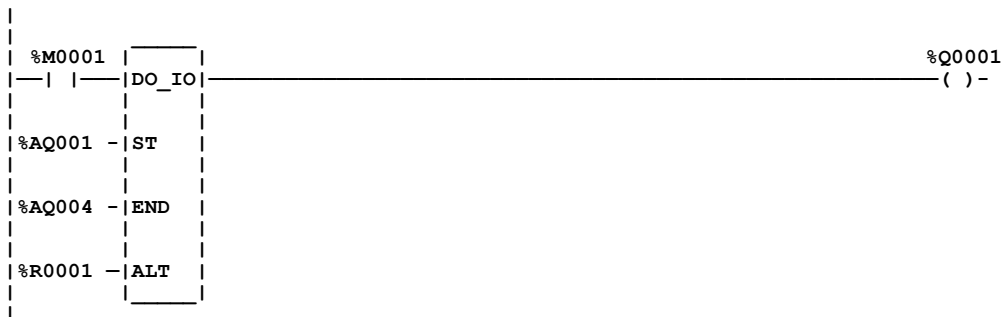
## Przykład 2 - Uaktualniania stanu wejść

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %M0001 na 1, powoduje odczytanie zmiennych od %I0001 (ST) do %I0064 (END) oraz ustawienie wartości zmiennej %Q0001 na 1. Ponieważ na wejście ALT nie dopływa sygnał, wartości skanowanych wejść zostaną wykorzystane przez funkcję DOIO do uaktualnienia tabeli stanu wejść (zmiennie od %I0001 do %I0064). Takie zastosowanie funkcji pozwala na wielokrotne odczytywanie i ustawianie stanu wejść w czasie wykonywania programu sterującego. Należy zwrócić uwagę, że jeśli wejście ALT nie jest wykorzystywane powinno zostać nie podłączone. Podanie na wejście ALT stanu logicznego 0 spowoduje błędy zegara alarmowego.



### Przykład 1 - Uaktualniania stanu wyjść

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %M0001 na 1 powoduje zapisanie wartości zmiennych %R0001 do %R0004 do kanałów wyjść analogowych %AQ001 do %AQ004 oraz ustawienie wartości zmiennej %Q0001 na 1. Ponieważ na wejście ALT podano zmienną %R0001, wartości %AQ001 do %AQ004 nie zostaną zapisane przez funkcję w modułach wyjść.



### Przykład 2 - Uaktualniania stanu wyjść

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %M0001 na 1 powoduje zapisanie wartości zmiennych %AQ0001 do %AQ0004 do kanałów wyjść analogowych %AQ001 do %AQ004 oraz ustawienie wartości zmiennej %Q0001 na 1. Funkcja DOIO uaktualnia stan wyjść analogowych, ponieważ na wejście ALT nie podano żadnego sygnału. Należy zwrócić uwagę, że jeśli wejście ALT nie jest wykorzystywane powinno zostać nie podłączone. Podanie na wejście ALT stanu logicznego 0 spowoduje błędy zegara alarmowego sterownika.



## Rozszerzona funkcja DO I/O dla jednostek centralnych 331 i nowszych

### Ostrzeżenie

Programy, w których wykorzystywana jest rozszerzona funkcja DOIO nie powinna być uruchamiane na oprogramowaniu Logicmaster 90-30/20 w wersji wcześniejsze niż 4.01.

Rozszerzona funkcja DO I/O (DOIO) dostępna jest w jednostkach centralnych wersja 4.30 lub późniejsze, modelu 331 i nowszych. Ta rozszerzona funkcja DOIO może być wykorzystywana wyłącznie dla modułów z pojedynczym modulem wejść/wyjść, 8 punktowym, 16 punktowym lub 32 punktowym.

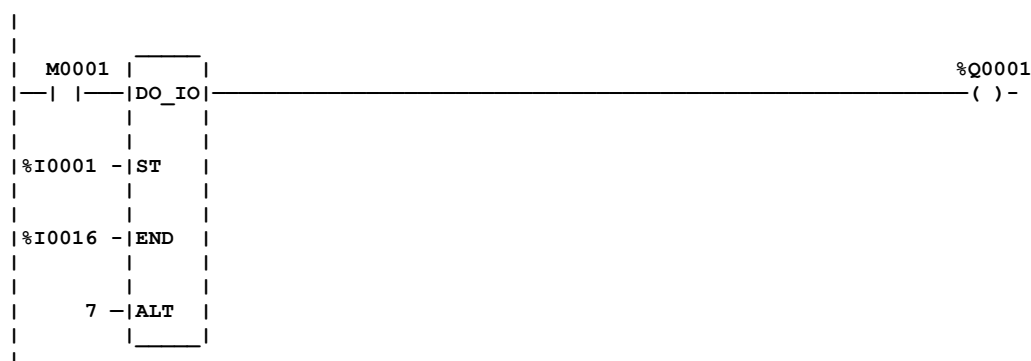
Przy pomocy parametru ALT identyfikowane jest gniazdo w kasecie głównej, w której zainstalowany jest moduł. Na przykład podanie na wejście ALT wartość 2 spowoduje, że moduł w 2 gnieździe będzie modulem docelowym. Parametry ST i END określają zakres wykorzystywanej pamięci.

### Uwaga

Rozszerzona funkcja DOIO sprawdza stan modułu docelowego pod kątem błędów.

Rozszerzona funkcja DOIO może być wykorzystana wyłącznie do obsługi modułów zainstalowanych w kasecie głównej. Z tego powodu, parametr ALT musi mieć wartość od 2 do 5 dla kaset z 5 gniazdami lub 2 do 10 dla kaset z 10 gniazdami.

Zmienna zawierająca adres początkowy (ST) i końcowy (END) muszą być typu %I lub %Q. Określają one pierwszy i ostatni adres, dla których jest skonfigurowany moduł. Przykładowo, jeżeli moduł z 16 wejściami, przypisanymi zmiennym %I0001 do %I0016, zainstalowany jest w 7 gnieździe kasety podstawowej z 10 gniazdami, parametr ST musi być równy %I0001, parametr END %I0016, a parametr ALT musi mieć wartość równą 10 (tzn. 0AH heksadecymalnie), zgodnie z pokazanym poniżej przykładem.



W zamieszczonej poniżej tabeli porównano czasy wykonania normalnego bloku funkcyjnego DOIO dla modułu z 8, 16 lub 32 wejściami/ wyjściami, z czasami wykonania dla rozszerzonej funkcji DOIO.

Moduł	Czas wykonywania normalnego bloku funkcyjnego DOIO	Czas wykonywania normalnego bloku funkcyjnego DOIO
Moduł wejść dyskretnych, 8 punktowy	224 mikrosekundy	67 mikrosekundy
Moduł wyjść dyskretnych, 8 punktowy	208 mikrosekund	48 mikrosekund
Moduł z 16 wejściami dyskretnymi	224 mikrosekundy	68 mikrosekundy
Moduł z 16 wyjściami dyskretnymi	211 mikrosekund	47 mikrosekund
Moduł wejść dyskretnych, 32 punktowy	247 mikrosekundy	91 mikrosekundy
Moduł wyjść dyskretnych, 32 punktowy	226 mikrosekund	50 mikrosekund

# SER

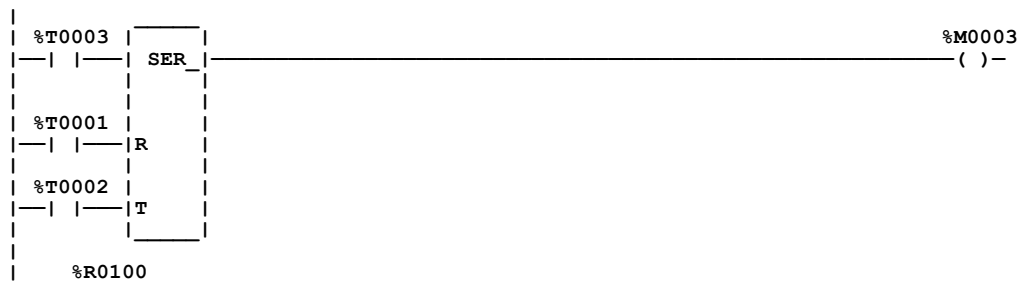
**Funkcja ta może być wykorzystywana w jednostkach centralnych 35x lub 36x w wersji 9.00 lub nowszej oraz we wszystkich wersjach jednostek centralnych 37x.**

- Funkcja SER zbiera serie próbek z modułów dyskretnych. Po doprowadzeniu sygnału do wejścia Enable, blok funkcyjny SER odczytuje próbkę, składającą się z maksymalnie 32 bitów kolejnych bądź nie.
- Blok SER może zapisywać do 1024 próbek każda do 32 bitów.
- Jeżeli natomiast blok funkcyjny SER zostanie umieszczony w podprogramie wywoływanym okresowo, częstotliwość wykonywania zależy od częstotliwości wywoływania tego podprogramu.
- Czas jest zapisywany wyłącznie dla próbki rejestrowanej w momencie wyzwolenia (próbka rejestrowana). Czas dla próbki rejestrowanej może być podany w formacie BCD (maksymalna częstotliwość wykonywania 1 s) lub w formacie POSIX (maksymalna częstotliwość wykonywania 10 ms). Czas jest zapisywany wyłącznie dla próbki zarejestrowanej w momencie doprowadzenia sygnału wyzwolenia. Blok funkcyjny SER nie rejestruje czasów dla innych próbek.
- Blok funkcyjny SER może pracować w następujących trybach: Rejestrowanie próbek do sygnału wyzwolenia (pre-trigger), Rejestrowanie próbek do zebrania ich określonej liczby po sygnale wyzwolenia (mid-trigger) lub w trybie Rejestrowanie do zebrania określonej liczby próbek (post-trigger). (Proszę porównać z informacjami na stronie 12-14.)
- Przebieg pracy bloku funkcyjnego SER zależy od zawartości bloku danych sterujących, który można utworzyć za pomocą funkcji Block Move (BLKMOV). (Proszę porównać z informacjami na stronie 12-9.)
- Funkcja SER może skanować moduł wejściowy przy każdym jej uaktywnieniu. Pozwala to na upewnienie się, że przechwytywane dane są najnowsze.

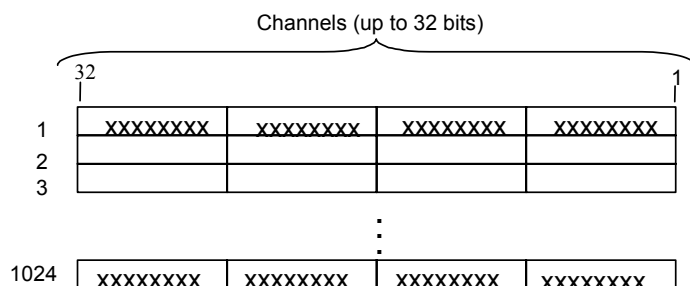
### Uwaga

Nie jest obsługiwana synchronizacja pomiędzy sterownikami korzystającymi z bloków funkcyjnych SER.

Funkcja SER posiada trzy wejściowa i jedno wyjście. enable, reset (R) i wyzwalacz (T).



Jak pokazano na poniższym rysunku kanały 8, 13, 24 lub 32 mogą zostać skonfigurowane z każdym kanałem reprezentującym punkty wejść dyskretnych. Można określić do 1024 próbek.





## Parametry

Parametr	Opis
enable	Po doprowadzeniu sygnału wejściowego, jeżeli nie jest doprowadzany sygnał zerowania, blok funkcyjny SER odczytuje jedną próbkę ze wszystkich skonfigurowanych kanałów.
R	Sygnał zerowania. Po jego doprowadzeniu, funkcja SER jest zerowana, bez względu na wartość sygnału wejściowego. Powoduje wyzerowanie parametrów Sample Buffer , Trigger Sample Offset , Trigger time i Current Sample Offset . Blok funkcyjny będzie pozostawał w stanie zerowania, aż do momentu, kiedy sygnał zerowania przestanie być doprowadzany. W stanie zerowania, na wyjście OK nie jest wysyłany sygnał. Zaprzeszanie doprowadzania sygnału do wejścia R powoduje wznowienie próbkowania.
T	Jeżeli wybrany został tryb Trigger Input Mode i blok funkcyjny SER jest aktywny, doprowadzenie sygnału na wejście T powoduje przełączenie bloku SER do trybu wyzwolenia. W trybie tym zapisywana jest próbka (tzw. próbka rejestrowana), indeks tej próbki oraz czas jej pobrania.  W momencie wyzwolenia próbka zostaje zarejestrowana bez względu na liczbę pobranych do tej pory próbek. Po wyzwoleniu, rejestrator nadal kontynuuje pobieranie próbek, aż do momentu pobrania określonej liczby (określonej parametrem Number of Samples After Trigger ), kiedy to następuje zatrzymanie pobierania próbek, do momentu doprowadzenia sygnału na wejście R.  Sygnał wyzwolenia jest ignorowany po wybraniu trybu Rejestrowanie do wypełnienia bufora.  Szczegółowe informacje odnośnie konfigurowania parametru Trigger Mode podano w punkcie “Blok danych sterujących”, na stronie 12-9.
Adres początkowy	Adres początkowy bloku danych sterujących, składającego się z 78 rejestrów. Blok danych sterujących zawiera parametry określające sposób wykonywania bloku funkcyjnego SER. Szczegółowe informacje podano w punkcie “Blok danych sterujących”, na stronie 12-9
ok	Sygnał wyjściowy, pojawiający się po spełnieniu wszystkich warunków do wyzwolenia (określonych przez parametr Trigger Mode) oraz po zebraniu wszystkich próbek. Sygnał ten jest wysyłany bez względu na stan sygnału wejściowego, aż do momentu doprowadzenia sygnału na wejście zerujące.

## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
Control Block								•				
R	•											
T	•											
ok	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Blok danych sterujących

Blok sterujący funkcji SER jest tablicą złożoną z 78 elementów typu Word, definiującą sposób zbierania danych oraz działanie mechanizmu wyzwiania dla tej funkcji. W jednym programie sterującym może występować wyłącznie jeden blok funkcyjny SER, który może być powiązany z każdym z poleceń i blokiem danych.

W celu skonfigurowania parametrów bloku funkcyjnego SER należy postępować zgodnie z zamieszczoną poniżej procedurą:

1. Ustawić wartość elementów tablicy, zgodnie z zamieszczoną poniżej tabelą. Wartości rejestrów można zainicjować za pomocą funkcji kopiowania bloków pamięci lub zainicjować wartości w tabeli rejestrów, a następnie zapamiętać tę tabelę, przed wywołaniem funkcji SER.
2. Wprowadzić funkcję SER do szczebla programu sterującego.

### Uwaga

Jeżeli użytkownik chce próbować x kanałów, gdzie x jest różne od 8, 16, 24 ale mniejsze od 32, musi wybrać liczbę kanałów większą od x i będącą wielokrotnością 8, a dla pozostałych, niewykorzystywanych kanałów należy wprowadzić pusty opis kanałów. Dla pustego opisu kanałów, parametr Segment Selector ma wartość 0xFFh, parametr długość musi być równy licznie niewykorzystywanych kanałów, a indeks wynosi 0.

Rejestr	Parametr	Opis
0 (pierwsze słowo)	Status	Zmienna przeznaczona wyłącznie do odczytu, której zadaniem jest informowanie o bieżącym statusie funkcji SER. Dodatkowe informacje przekazywane są za pomocą słowa Status Extra Data (Słowo 1). <b>Uwaga:</b> Jeżeli w tablicy Control Block wykryty zostanie błąd, wartość parametru Status zostanie ustawiona na 6, na wyjście OK przestanie być wysyłany sygnał i nie będą realizowane żadne działania. Parametr status może przyjmować między innymi następujące wartości: 0 = Zerowanie 1 = Funkcja nieaktywna 2 = Funkcja aktywna 3 = Wyzwolenie 4 = Zakończenie działania 5 = Brak pamięci 6 = Błędny parametr
1	Status Extra Data	Zmienna przeznaczona wyłącznie do odczytu, udostępniająca dodatkowe informacje o statusie funkcji SER. Szczegółowe informacje odnośnie konfigurowania tego parametru podano w punkcie "Słowo Status Extra Data" na stronie 12-12
2	Trigger mode	Parametr ten definiuje warunki, których spełnienie jest wymagane, aby blok funkcyjny SER przeszedł do stanu wyzwolenia. Parametr ten może przyjmować następujące wartości: 0 = Trigger Input mode (Wyzwalanie po doprowadzeniu sygnału do wejścia T) 1 = Full Buffer mode (Rejestrowanie do wypełnienia bufora) W trybie Trigger Input Mode, jeżeli blok funkcyjny SER jest aktywny, doprowadzenie sygnału na wejście T powoduje wygenerowanie czasu dla próbki rejestrowanej. Próbkowanie jest kontynuowane do momentu zebrania określonej liczby próbek (parametr Number of Samples After Trigger). Po zebraniu odpowiedniej liczby próbek, określonej tym parametrem, następuje wysłanie sygnału. W trybie Rejestrowanie do wypełnienia bufora ignorowane jest doprowadzenie sygnału do wejścia T. Jeżeli blok funkcyjny SER jest aktywny, pobieranie próbek jest kontynuowane do momentu wypełnienia bufora próbek. Po zebraniu odpowiedniej liczby próbek, określonej tym parametrem, następuje wysłanie sygnału wyjściowego OK. Rozmiar bufora jest określany za pomocą parametru Number of Samples.
3	Trigger Time Format:	Format wyświetlania czasu rejestrowania, w którym nastąpiło wyzwolenie funkcji SER. W przypadku wyświetlacza BCD, parametr ten należy ustawić na 0, natomiast dla wyświetlacza POSIX, parametr ten należy ustawić na 1 (szczegółowe informacje podano na stronie 12-16).
4—7	Zarezerwowane	Słowa 4 do 7 są słowa zarezerwowanymi, ich wartość powinna zostać ustawiona na zero (0).

Rejestr	Parametr	Opis
8	Number of Channels (Liczba kanałów) (bitów na próbkę)	Liczba próbkowanych bitów danych, kopiowanych do bufora próbek, po każdym wykonaniu bloku funkcyjnego SER. Dopuszczalnymi wartościami są 8, 16, 24 lub 32 bity. Wszystkie nie wykorzystane muszą zostać puste. (See Words 14—77.) (Proszę porównać ze słowami 14—77).
9	Number of Samples	Rozmiar bufora próbek w bajtach. Zakres dopuszczalnych wartości wynosi od 1 do 1024 próbek. (Rzeczywista wielkość bufora w bitach jest równa (Number of Samples)*(Number of Channels ).
10	Number of Samples After Trigger (Liczba próbek rejestrowanych po doprowadzeniu sygnału wyzwolenia)	Liczba próbek zapamiętywanych w buforze próbek po wyzwoleniu bloku funkcyjnego SER. Parametr ten może mieć wartość z zakresu od 0 do Number of Samples . Jest on wykorzystywany wyłącznie jeżeli parametr Trigger Mode ma wartość Trigger Input (0).
11	Input Module Slot (Gniazdo modułu wejść)	Określa położenie modułu w kasecie głównej, z którego będą zbierane dane. Podanie 0 powoduje, że żaden moduł nie będzie skanowany. Wartości odczytane z modułu wejść są zapamiętywane lokalnie, bez zmieniania wartości zmiennych przypisanych do tego modułu. W celu zapisania wartości odczytanych z modułu wejść do bufora próbek w bloku danych należy podać opis kanału . Jeżeli moduł jest nie zainstalowany lub uszkodzony, w momencie czytania wartości te zostaną ustawione na zero. W przypadku wystąpienia takiego błędu, nie zostanie on zarejestrowany w tabeli uszkodzeń, jego zasygnalizowanie zostanie pozostawione dla skanera wejść i wyjść.
12	Data Block Segment Selector (Typ pamięci)	Typ danych przypisanych przez użytkownika dla Bloku danych . Przykładowo, jeżeli blok danych ma rozpoczynać się pod adresem %R, należy wprowadzić wartość 08. Poprawnymi wartościami tego parametru są: %R (08h), %AI (0Ah), %AQ (0Ch). Szczegółowe informacje o bloku danych podano na stronie 12-13.
13	Data Block Offset:	Adres początkowy Bloku danych. Parametr ten jest liczony od zera. Przykładowo, jeżeli blok danych ma rozpoczynać się pod adresem %R0100, należy wprowadzić wartość 99. Użytkownik ponosi odpowiedzialność za przypisanie odpowiednio dużego obszaru pamięci dla całego bloku danych.
14—77	Channel Descriptions	Położenie zmiennej (Segment Selector, Length i Offset) związanej z danym kanałem. Można zgromadzić od 1 do 32 opisów kanałów, w zależności od liczby próbkowanych kanałów oraz od długości danych. Dane zwracane są w kolejności, zdefiniowanej w następnym punkcie.
	Channel Segment Selector/Length	Wartość ta wprowadzana jest w systemie heksadecymalnym, słowo to określa zarówno wybrany przez użytkownika typ danych jak i długość danych (w bitach). MSB = Wybór segmentu. LSB = Długość danych. Długość danych jest szczególnie użyteczna przy odczycie próbek z sąsiadujących ze sobą punktów. Parametr Segment Selector może wskazywać na dowolny obszar pamięci typu dyskretnego: %I (46h), %Q (48h), %M (4Ch), %T (4Ah), %G (56h), %S (54h), %SA (4Eh), %SB (50h), %SC (52h), typ pusty (FFh) i wybrany moduł wejściowy Module Selector(00h).
		Parametr Length może przyjmować wartość z zakresu 1—32, przy ograniczeniu, że suma wszystkich długości nie może być większa od parametru Number of Channels. Wprowadzenie długości większej od 1 pozwala na skonfigurowanie większej liczby sąsiadujących ze sobą kanałów za pomocą jednego opisu kanału.
	Channel Offset	Wartość tego parametru wprowadzana jest w systemie heksadecymalnym, słowo to określa przesunięcie parametru BIT dla typu danych lub modułu wejściowego określonego parametrem Segment Selector. Przesunięcie to jest liczone od zera. Zakres wartości tego parametru zależy od parametru Segment Selector (typ danych i długość). Przesunięcie określa położenie w tabeli danych lub w module wejściowym, z którego pobierane są próbki.

## Słowo Status Extra Data

Słowo Status Extra Data (Słowo 1 w bloku danych sterujących ) zawiera dodatkowe informacje odnośnie statusu (słowo 0) bloku funkcyjnego SER.

Wartość	Stan	Opis
0	Reset State	Doprowadzenie sygnału na wejście zerowania R powoduje wyzerowanie parametrów Sample Buffer, Trigger Sample Offset, Trigger time i Current Sample Offset. Nie jest wysyłany sygnał na wyjście logiczne. Jeżeli sygnał zerowania przestanie być doprowadzany, funkcja przejdzie do stanu <b>Inactive State</b> . Parametr Status Extra Data nie ma w takiej sytuacji żadnego znaczenia i jego wartość zostaje ustawione na zero.
1	Inactive	Stan pomiędzy stanem zerowania (Reset State) a stanem aktywnego bloku funkcyjnego (Active State). W stanie tym nie są realizowane żadne działania. Blok funkcyjny SER nie wysyła sygnału wyjściowego. Po doprowadzeniu sygnału wejściowego, status zmieniany jest na Active State.
2	Active	Stan po doprowadzeniu sygnału wejściowego, jeżeli blok funkcyjny SER nie jest wyzerowany, nie stwierdzono błędów oraz nie nastąpiło wyzwolenie. W stanie tym, każde wykonanie tego bloku funkcyjnego powoduje zarejestrowanie jednej próbki. Nie jest wysyłany sygnał na wyjście logiczne. Monitorowane są warunki wymagane do wyzwolenia (określone za pomocą parametru Trigger Mode), a w przypadku ich spełnienia funkcja przechodzi w stan wyzwolenia (Triggered State). Jeżeli liczba pobranych próbek jest większa od liczby określonej za pomocą parametru "Number of Samples", wartość parametru Status Extra Data jest równa 0x01, w przeciwnym wypadku jest równa 0x00.
3	Triggered	Status przyjmowany po spełnieniu warunków określonych parametrem Trigger Mode. Pobieranie dodatkowych próbek zależy od trybu wyzwalania oraz od wartości parametrów. Nie jest wysyłany sygnał na wyjście logiczne. Po pobraniu wszystkich próbek, funkcja przechodzi w stan zakończenia działania (Complete State). Jeżeli liczba pobranych próbek jest większa od liczby określonej za pomocą parametru "Number of Samples", wartość parametru Status Extra Data jest równa 0x01, w przeciwnym wypadku jest równa 0x00.
4	Complete	Blok funkcyjny SER przechodzi do tego stanu po zakończeniu pobierania wszystkich próbek. Wysyłany jest sygnał wyjściowy. Dopuszczalne jest przejście wyłącznie do stanu zerowania. Jeżeli liczba pobranych próbek jest większa od liczby określonej za pomocą parametru "Number of Samples", wartość parametru Status Extra Data jest równa 0x01, w przeciwnym wypadku jest równa 0x00.
5	Overrun Error	Blok funkcyjny SER przechodzi do tego stanu, jeżeli Blok danych/ Blok danych sterujących przekroczył zakres przewidzianego dla niego typu pamięci. Nie jest wysyłany sygnał na wyjście logiczne. Dopuszczalne jest przejście wyłącznie do stanu zerowania. Parametr Status Extra Data nie ma w takiej sytuacji żadnego znaczenia i jego wartość zostaje ustawione na zero.
6	Parameter Error	Wykrycie błędów w bloku danych sterujących lub w innych parametrach bloku funkcyjnego SER. Nie jest wysyłany sygnał na wyjście logiczne. Dopuszczalne jest przejście wyłącznie do stanu zerowania. Parametr Status Extra Data zawiera pozycję w bloku danych sterujących, określając tym samym położenie błędnego parametru.
7	Status Error	Nieprawidłowa wartość parametru Status . Nie jest wysyłany sygnał na wyjście logiczne. Dopuszczalne jest przejście wyłącznie do stanu zerowania. Nieprawidłowa wartość jest zapisywana w Bloku sterującym w miejscu określonym parametrem Status Extra Data.

## Blok danych funkcji SER

Blok danych SER zawiera bufor próbek, indeksy próbek oraz informacje o wyzwoleniu. Informacje te są zapisywane przez jednostkę centralną, są one przeznaczone wyłącznie do odczytu. Użytkownik jest odpowiedzialny za przypisanie dostatecznie dużego obszaru pamięci dla Bloku danych. Blok ten ma następującą strukturę:

Pozycja*	Opis parametru
0	Numer indeksu dla bieżącej próbki. Parametr ten podaje indeks ostatnio zarejestrowanej próbki. Indeks ten liczony jest od zera. Zakres dopuszczalnych wartości wynosi od -1 do 1023. Indeks rejestru zawierającego próbkę = (Liczba bajtów na próbkę)* (Indeks) + (Rejestr początkowy bufora próbek). <b>Uwaga:</b> Wartość ta zostaje określona dopiero w momencie wyzwolenia. W momencie zerowania funkcji SER (poprzez doprowadzenie sygnału na wejście R), zostaje ona ustawiona na -1.
1	Numer indeksu dla próbki rejestrowanej. Miejsce, w którym zostanie zapisana próbka w momencie wyzwolenia. Indeks ten liczony jest od zera. Zakres dopuszczalnych wartości wynosi od 0 do 1023. Indeks rejestru zawierającego próbkę = (Liczba bajtów na próbkę)* (Indeks) + (Rejestr początkowy bufora próbek). <b>Uwaga:</b> Wartość ta zostaje określona dopiero w momencie wyzwolenia, w związku, z czym nie należy z niej korzystać przed wyzwoleniem. W momencie zerowania funkcji SER za pośrednictwem sygnału zerującego, zostaje ona ustawiona na 0.
2 do 5	Czas wyzwolenia: Jest to czas wyzwolenia, zapisywany według wskazań zegara czasu bieżącego sterownika, w którym nastąpiło spełnienie warunków wyzwolenia. Wartość czasu wyświetlana jest (domyślnie) w formacie BCD, ale istnieje również możliwość wyświetlania w formacie POSIX. Format wyświetlania określany jest w bloku danych sterujących za pomocą parametru <i>Trigger Time Format</i> . Doprowadzenie sygnału zerującego powoduje ustawienie tej wartości na zero.
6 do końca bufora.	Bufor próbek. Obszar w pamięci, w którym przechowywane są próbki danych. Obszar ten jest zerowany po doprowadzeniu sygnału zerowania. Rozmiar bufora próbek zależy od liczby kanałów oraz wielkości jednej próbki. Jest to bufor o kołowej strukturze - po zapisaniu na ostatniej pozycji, następna próbka zostanie zapisana w pierwszym rejestrze. Adres końca bufora próbek = $5 + ((\text{liczba próbek do pobrania}) * (\text{liczba kanałów do próbkowania} / 8) + 1 / 2)$

\*Kolumna 1 podaje adres względny liczony od adresu początkowego, zdefiniowanego za pomocą parametrów Data Block Segment Selector (Słowo 12) i Data Block Offset (Słowo 13) w Bloku danych sterujących funkcji SER.

## Uwagi

Odczytywane próbki są umieszczane w tablicy o strukturze kołowej. Sygnał wyjściowy jest wysyłany po odczytaniu skonfigurowanej liczby próbek. Sygnał wyjściowy można wykorzystać w celu odnotowania czasu pobrania ostatniej próbki lub w celu rozpoczęcia zbierania dodatkowych próbek. (Proszę porównać z punktem "Tryby próbkowania").

Blok funkcyjny Ser musi zostać wyzerowany (poprzez doprowadzenie sygnału zerowania) przed rozpoczęciem próbkowania. Wyzerowanie powoduje zainicjowanie obszaru pamięci dla bloku danych. Jeżeli blok funkcyjny SER nie zostanie wyzerowany, będzie nadal korzystał z tego samego bloku danych na, wskutek czego indeks bieżącej próbki oraz dane zapisane w bloku danych będą niepoprawne.

Wartości w Bloku sterującym funkcji SER są uaktualniane przy każdym wykonaniu tej funkcji, w stanie zerowania, wyzwolenia lub, jeżeli funkcja jest aktywna. Jeżeli użytkownik zmieni jeden z parametrów konfiguracyjnych Bloku danych sterujących w czasie wykonywania programu, zmiana zostanie uwzględniona przy następnym wywołaniu bloku funkcyjnego SER, powiązane go z tym Blokiem danych sterujących. W przypadku wystąpienia błędu, praca zostanie zatrzymana a blok funkcyjny SER przechodzi do odpowiedniego stanu, w zależności od rodzaju napotkanego błędu. W celu wznowienia próbkowania, użytkownik musi usunąć przyczynę błędu i wyzerować blok funkcyjny (poprzez doprowadzenie sygnału do wejścia zerowania).

Jeżeli próbki odczytywane są z modułu wejść, sterownik nie sprawdza, czy jest to moduł wejść dyskretnych oraz czy Opisy kanałów tego modułu mają długości i pozycje dostosowane do wielkości modułu. Użytkownik jest odpowiedzialny za wprowadzenie poprawnych parametrów próbkowania dla modułu wejść. Moduł wejściowy może mieć wiele opisów kanałów, ale przy każdym wykonaniu bloku funkcyjnego będzie on nadal obsługiwany wyłącznie jeden raz.

Blok funkcyjny SER może być wykorzystywany w normalnym bloku programu sterującego lub w okresowo wywoływanych podprogramach. W przypadku umieszczenia w normalnym bloku programu sterującego, częstotliwość wykonywania bloku funkcyjnego SER zależy od czasu trwania cyklu pracy sterownika, a więc od ilości i rodzaju aktywnych w danym cyklu bloków funkcyjnych. W przypadku umieszczenia go w podprogramie wywoływanym przez przerwanie, blok SER może być wywoływany nawet, co 1ms, z bardzo niewielkimi odchyleniami od tej wartości.

Wykonywanie bloku funkcyjnego SER, co 1ms może spowodować zajęcie prawie 50% zasobów jednostki centralnej. W podprogramie wykonywanym, co 1ms nie należy umieszczać więcej niż dwóch bloków funkcyjnych SER.

## Tryby próbkowania

Tryb próbkowania jest określany przez parametr Trigger Mode (Słowo 2 w bloku danych sterujących) i przez parametr Number of Samples After Trigger (Słowo 10). Interpretacja zawartości bufora próbek zależna jest od wartości tych parametrów.

W poniższej tabeli podsumowano tryby próbkowania.

Tryb	Słowo 2	Słowo 10
Pre Trigger	0	0
Mid Trigger	0	Od 1 do (numer próbki -1)
Post Trigger	0	Równe ilości próbek (określone w słowie 9)
Full Buffer	1	Słowo 10 i sygnał z wejścia trigger jest ignorowane.

## Próbkowanie sterowane sygnałem wyzwiania

W celu skonfigurowania trybu: Rejestrowanie próbek do sygnału wyzwolenia, Rejestrowanie próbek do zebrania ich określonej liczby po sygnale wyzwolenia lub w trybie Rejestrowanie do zebrania określonej liczby próbek należy parametr Trigger Mode (słowo 2) ustawić na 0. Tryb próbkowania jest sterowany za pomocą parametru Number of Samples After Trigger (Słowo 10). We wszystkich przypadkach próbkowanie jest rozpoczynane po doprowadzeniu sygnału wejściowego. Po doprowadzeniu sygnału na wejście wyzwiania (T), próbkowanie jest kontynuowane do momentu zebrania odpowiedniej liczby próbek (określonej za pomocą parametru Number of Samples After Trigger). Po zakończeniu próbkowania następuje wysłanie sygnału wyjściowego.

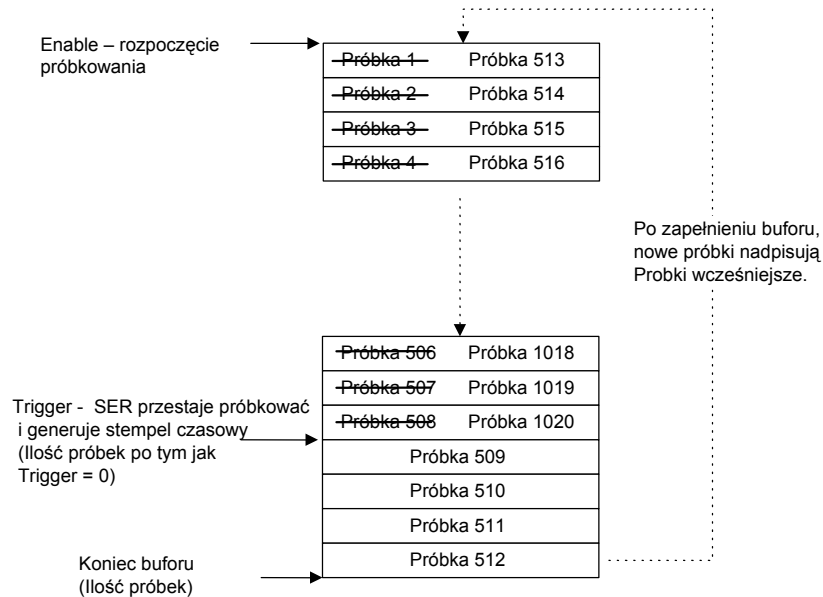
Jeżeli przed zebraniem liczby próbek określonej parametrem Number of Samples After Trigger, dotychczas zebrana liczba próbek jest większa od wartości parametru Number of Samples (Słowo 9), bufor zostanie "przewinięty", tzn. blok funkcyjny rozpoczyna zapisywanie na jego początku, w miejsce próbek zebranych wcześniej.

Zmiana stanu sygnału wyzwiania z 0 na 1 powoduje zapisanie czasu wyzwolenia do określonego miejsca w bloku danych.

### Pre Trigger

#### Zbieranie próbek do momentu wykrycia sygnału wyzwolenia.

W celu skonfigurowania tego trybu, należy w Słowie 10 zapisać wartość 0, co spowoduje, że po doprowadzeniu sygnału wyzwolenia, próbkowanie zostanie zatrzymane i w określonym miejscu w bloku danych (słowa o indeksach 2-5) zapisany zostanie Czas rejestrowania. (Wszystkie próbki zebrane są przed wyzwoleniem).

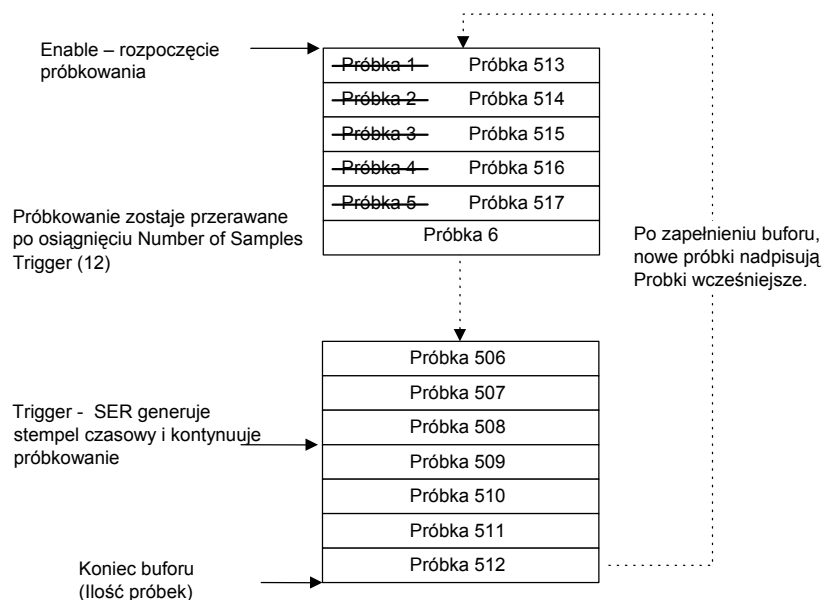


Rysunek 12-1. Przykład próbkowania w trybie Rejestrowanie próbek do sygnału wyzwolenia (dla 512 próbek)

### Mid Trigger

Zbieranie próbek jest kontynuowane do momentu zebrania określonej liczby próbek, zadanej parametrem Number of Samples After Trigger.

W celu skonfigurowania tego trybu, należy w Słowie 10 zapisać wartość pomiędzy 1 a (ilość próbek – 1). Po doprowadzeniu sygnału na wejście wyzwalań (T), próbkowanie jest kontynuowane do momentu zebrania określonej liczby próbek. W poniższym przykładzie parametr Number of Samples After Trigger ustawiono na 12. Po zakończeniu próbkowania w buforze znajdować się będzie 500 próbek pre-trigger i 12 próbek post-trigger.

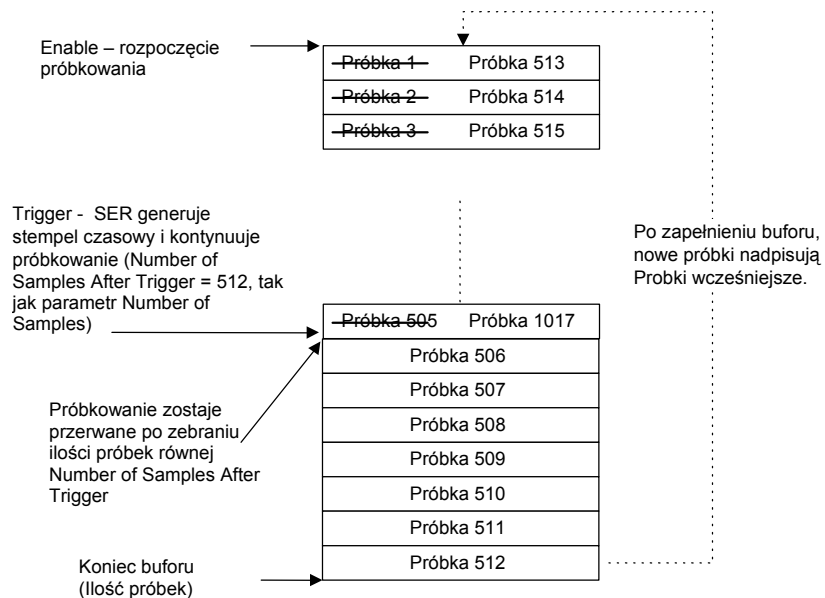


Rysunek 12-2. Przykład próbkowania do zebrania określonej ilości próbek po sygnale wyzwolenia (dla 512 próbek)

### Post Trigger

Próbkowanie jest kontynuowane przez cały czas do momentu zebrania liczby próbek określonej parametrem Number of Samples.

W celu skonfigurowania tego trybu, należy w Słowie 10 zapisać wartość równą wartości parametru Number of Samples (Word 9). Po doprowadzeniu sygnału na wejście wyzwalania (T), próbkowanie jest kontynuowane do momentu zebrania określonej liczby próbek. (Uwaga: Wszystkie próbki zebrane są po wyzwoleniu).



Rysunek 12-3. Przykład próbkowania do zebrania określonej ilości próbek po sygnale wyzwolenia (dla 512 próbek)

### Tryb Full Buffer (Rejestrowanie do wypełnienia bufora)

Jeżeli parametr Trigger Mode zostanie ustawiony na 1, ignorowany jest parametr Number of Samples After Trigger (Słowo 10), a sygnał wyzwalania nie ma wpływu na sposób wykonywania bloku funkcyjnego SER. Jeżeli blok funkcyjny SER jest aktywny, pobieranie próbek jest kontynuowane do momentu zebrania liczby próbek określonej za pomocą parametru Number of Samples (Słowo 9). Po wypełnieniu bufora następuje zatrzymanie próbkowania, wygenerowanie czasu rejestrowania oraz wysłanie sygnału wyjściowego.



## Formaty stępli czasowych funkcji SER

Format BCD		
Nr słowa w bloku danych	Zawartość (bajt wyższy/bajt niższy)	Sugerowany format podglądu
Słowo 2	Miesiąc/rok	Hex (MMYY)
Słowo 3	Godzina/dzień miesiąca	Hex (HHDD)
Słowo 4	Sekundy/minuty	Hex (SSMM)
Słowo 5	Nie wykorzystywane	Same zera

Format POSIX		
Nr słowa w bloku danych	Zawartość	Sugerowany format podglądu
Słowa 2 i 3	Liczba sekund począwszy od 1 stycznia 1970	Dint
Słowa 4 i 5	Ilość nano sekund następnej sekundy	Dint

### Przykład

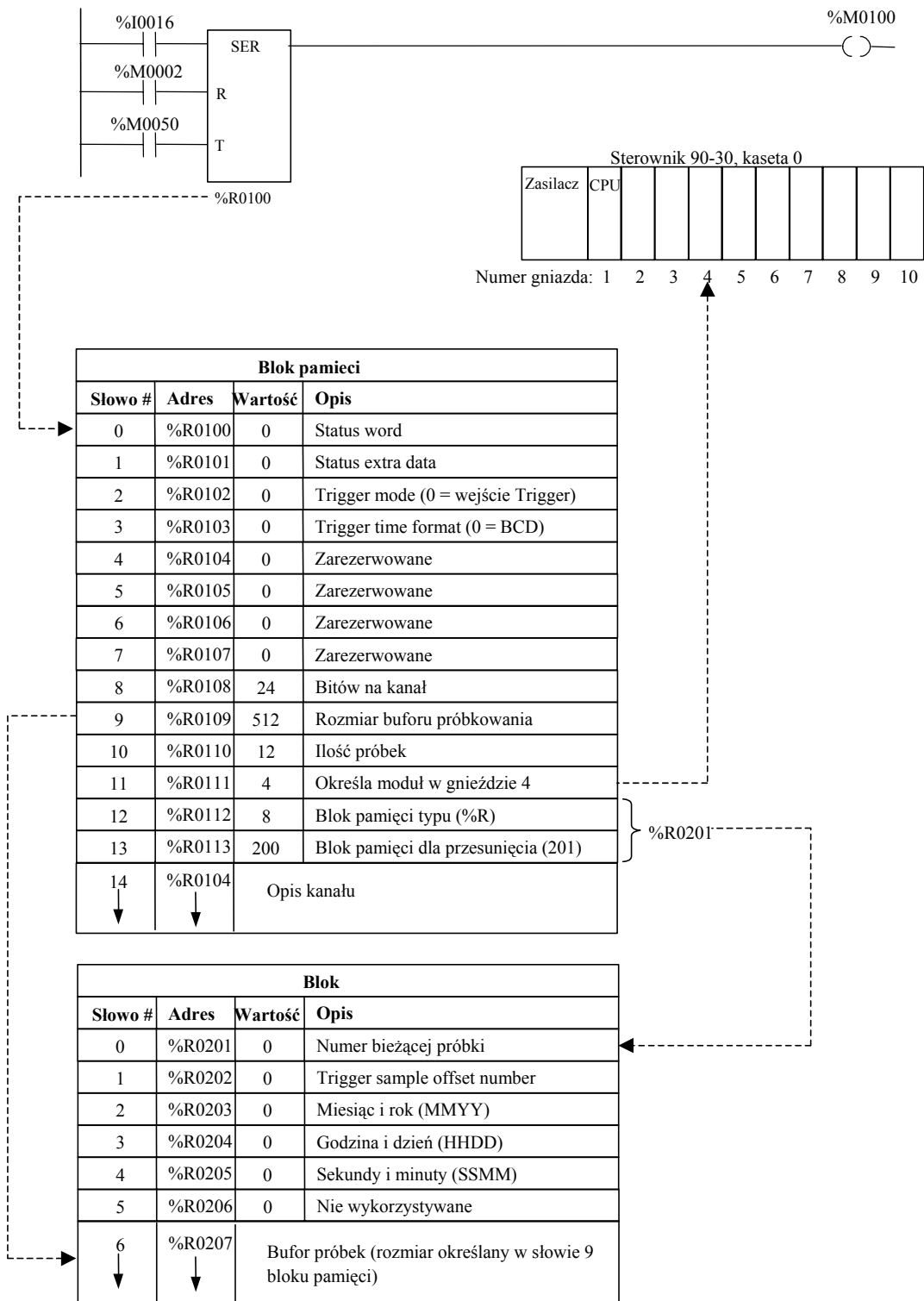
Poniższe dwie tabelach pokazują sposób, w jaki może zostać przedstawiona data 3 listopada 1998 8:34:05.010 w formacie BCD i POSIX, w bloku danych rozpoczynającym się od %R0201 (słowo 0).

3 listopada 1998 8:34:05.010 w formacie BCD		
Rejestr	Parametr	Wartość (heksadecymalna)
%R0203	Miesiąc/rok	1198
%R0204	Godzina/dzień miesiąca	0803
%R0205	Sekundy/minuty	0534
%R0206	Nie wykorzystywany	0000

3 listopada 1998 8:34:05.010 w formacie POSIX			
Rejestr	Parametr	Wartość (dziesiętna)	Wartość (heksadecymalna)
%R0203/R0204	Sekundy	910,082,045	363EBFFD
%R0205/R0206	Nano sekundy	010,000,000	00989680

### Przykłady

Poniżej przedstawiono powiązania pomiędzy programem logicznym, blokiem pamięci a modułem wejściowym sterownika. Wartości wpisane do Bloku danych sterujących przedstawiono w tabeli 12-1.



## Przykład bloku danych sterujących

W przykładzie tym 16 punktowy moduł wejść dyskretnych został wpięty w gniazdo 4 kasety głównej (0). Z modułu pobrano 572 próbki (512 + 60). Sygnał jest doprowadzany do wejścia Enable, a nie jest doprowadzany do wejść Reset (R) i Trigger (T).

Tabela 12-1. Blok danych sterujących dla przykładowej funkcji SER

Słowo	Rejestr	Parametr	Wartość (dziesiętna)	Wartość (heksadecymalna)	Opis
0	%R0100	Status	2	0002	Blok funkcyjny SER znajduje się w stanie aktywnym. Oznacza to, że blok funkcyjny jest normalnie wykonywany, a próbka pobierana jest za każdym razem, kiedy w programie sterującym zostanie napotkany ten blok funkcyjny.
1	%R0101	Status Extra Data	1	0001	Słowo Extra Status Data sygnalizuje, że pobrano więcej niż 512 próbek, a więc nastąpiło, co najmniej jedno przesunięcie w buforze próbek o strukturze kołowej.
2	%R0102	Trigger mode	0	0000	Blok funkcyjny SER przejdzie do stanu wyzwolenia po doprowadzeniu sygnału do wejścia T.
3	%R0103	Trigger Time Format	0	0000	0=BCD
4	%R0104	Zarezerwowane	0	0000	Parametry zarezerwowane mają zawsze wartość 0.
5	%R0105	Zarezerwowane	0	0000	
6	%R0106	Zarezerwowane	0	0000	
7	%R0107	Zarezerwowane	0	0000	
8	%R0108	# of channels	24	0018	Każda próbka zawiera 24 bity (3 bajty) danych.
9	%R0109	# of samples to be taken	512	0200	Wielkość bufora próbek wynosi 512. Należy zwrócić uwagę, że liczba buforów jest równa $512 \times (24/8) = 1536$ bajtów lub 768 słów. (Długość próbki określono w słowie 8 na 3 bajty).
10	%R0110	# of samples after trigger	12	000C	Po każdym wyzwoleniu zapisywanych jest 12 próbek.
11	%R0111	Input module slot	4	0004	Dane będą czytane z modułu wejść zainstalowanego w kasecie 0, gnieździe 4.
12	%R0112	Data Block Segment Selector	8	0008	Segment danych wynosi 0x08(%R).
13	%R0113	Data Block Offset	200	00C8	Indeks wynosi 200, a więc adres początkowy bloku danych to %R0201. Indeks jest liczony od zera, ale tablica rejestrów rozpoczyna się od %R0001. Z tego powodu, punkt początkowy bloku danych jest równy %R0001 + 200 = %R0201.

Ciąg dalszy na następnej stronie

Słowo	Rejestr	Parametr	Wartość (dziesiętna)	Wartość (heksadecymalna)	Opis
<b>Opis kanałów</b>		Pozostałe słowa zawierają Opisy kanałów. W niniejszym przykładzie zdefiniowano 6 opisów kanałów.			
14	%R0114	Set. Sel. : Length	17921	4601	Opis kanłu 1: Dla pierwszego kanału wybrano Segment %I o długości 1 i indeksie 0. Oznacza to %I0001.
15	%R0115	Offset	0	0000	
16	%R0116	Seg. Sel. : Length	-253	FF03	Opis kanłu 2: Dla drugiego kanału wybrano segment zerowy o długości 3 i adresie względnym 0. Powoduje to pominięcie ("zignorowanie") kanałów 2 - 4. Wartość próbki dla tych kanałów będzie zawsze równa zero.
17	%R0117	Offset	0	0000	
18	%R0118	Seg. Sel. : Length	3	0003	Opis kanłu 3: Wybrano moduł wejść o długości 3 i adresie względnym 12. Powoduje to bezpośrednie pobieranie próbek z modułu wejść. W tym opisie kanału wybrano dla kanałów 5 - 7 wartości elementów 13, 14 i 15 modułu wejść.
19	%R0119	Offset	12	0012	
20	%R0120	Seg. Sel. : Length	18434	4802	Opis kanłu 4: W opisie czwartego kanału wybrano Segment %Q o długości 2 i adresie względnym 8. Tak więc kanałom 8 i 9 przypisane są %Q0009 i %Q0010.
21	%R0121	Offset	8	0008	
22	%R0122	Seg. Sel. : Length	8	0008	Opis kanłu 5: Dla piątego kanału wybrano ponownie moduł wejść. Posiada on długość 8 i adresie względnym 0. Spowoduje to, że wartości punktów 1 do 8 modułu wejść będą umieszczane w kanałach 10 - 17.
23	%R0123	Offset	0	0000	
24	%R0124	Seg. Sel. : Length	-249	FF07	Opis kanłu 6: Dla szóstego kanału wybrany segment zerowy. Posiada on długość 7 i adresie względnym 0. Ten zerowy opis kanału powoduje, że kanały 18 - 24 będą wypełniane zerami. Ten ostatni opis kanału wymagany jest do uzupełnienia bufora próbek do 24 bitów. Ponieważ skonfigurowano tutaj wszystkie 24 kanały, nie są potrzebne jakiegokolwiek inne opisy kanałów.
25	%R0125	Offset	0	0000	

#### Konfiguracja kanałów dla powyższego przykładu

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
U	U	U	U	U	U	U	U	N	N	N	N	N	N	N	C8	C7	C6	C5	C4	C3	C2	C1	%Q 10	%Q 09	C15	C14	C13	N	N	N	%I 01

U = Nie wykorzystany, N = Null, C oznacza skonfigurowany kanał modułu wejść (na przykład, C0 = 1 punkt wejść, C15 = 16 punkt wejść)

## Przykładowa zawartość próbki

W zamieszczonej poniżej tabeli zestawiono wartości zapisane dla pojedynczej próbki, w oparciu o opisy kanałów zamieszczone w bloku danych sterujących. Bazują one na przykładowym ekranie zamieszczonym na następnej stronie. Należy zwrócić uwagę, że w przykładzie tym bity 1 – 16 są znajdując się w %R00207 a bity 17 – 24 są częścią %R00208.

Tabela 12-2. Zawartość próbki dla przykładowego bloku funkcyjnego SER

Numer kanału	Zawartość kanału	Wartość
1	%I0001	1
2 - 4	Zera	000
5	Element 13 modułu wejść	1
6	Element 14 modułu wejść	1
7	Element 15 modułu wejść	1
8	%Q0009	0
9	%Q0010	0
10 - 17	Elementy 1 - 8 modułu wejść	100100010
18 - 24	Zera	0000000

## Blok danych dla przykładowego bloku danych sterujących

Zamieszczona poniżej tabela podaje format bloku danych, dla przedstawionego wcześniej, przykładowego bloku danych sterujących. Należy zwrócić uwagę, że blok ten rozpoczyna się w rejestrze 201, zgodnie z parametrami wprowadzonymi w bloku sterującym (Słowo 12 i Słowo 13 ).

Tabela 12-3. Blok danych dla przykładowej funkcji SER

Pozycja	Rejestr	Opis parametru	Wartość (dziesiętna)	Wartość (heksadecymalna)
0	%R0201	Pozycja dla bieżącej próbki:	59	003B
1	202	Pozycja dla rejestrowanej próbki:	0	0000
2 - 5	203 – 206	Czas zarejestrowania (BCD)	0 0 0 0	0000 0000 0000 0000
6 - 768	207 – 975	Bufor próbek.	dane dla próbki	dane dla próbek

Numer indeksu bieżącej próbki wynosi 59, co oznacza, że ostatnią z zapisywanych w buforze próbek jest próbka o numerze 59. Ponieważ każda próbka zajmuje 3 bajty, indeks wynosi  $59 \cdot 3 = 177$  bajtów lub inaczej, jest to starszy bajt 89-o rejestru. Ponieważ w tym momencie nie są spełnione warunki do wyzwolenia, zarejestrowana próbka i czas wyzwolenia są równe 0 oraz nie jest wysyłany sygnał wyjściowy. W buforze zapisanych jest 512 próbek, ostatnio zebrana próbka zajmuje pozycję 59, a najstarsza pozycję 60.

## Przykład zarejestrowanych danych

### Sprawdzanie przechwyconych danych

Poniższy ekran przedstawia stan po doprowadzeniu sygnału wyzwalającego. Blok danych sterujących rozpoczyna się w %R00100, a obszar danych w %R00201. Kursor jest ustawiony na %R00207.

Rejestr %R00207 jest pierwszym, w którym przetrzymywane są zmierzone dane wejściowe. Należy zwrócić uwagę, że wartość -21855 ma w tym momencie mniejsze znaczenie, jednak umieszczenie kursora na %R00207 spowoduje wyświetlenie tej wartości. Wykorzystując format binarny można określić stan bitów skonfigurowanych w części Channel Descriptions bloku sterującego.

Rejestry %R00203 do %R00205 zawierają czas i datę w formacie 24 godzinnym (15 maja 2001 godzina 16:06:57).

%R00207 w formacie binarnym

PROGRM	TABLES	STATUS	real	hex	bin	ascii	FOLDER	UTILTY	PRINT
1	int	dint							chgall
(R9) Format entered is not allowed for this reference type									
REGISTER									
		%R00207			00100010 01110001				
00100	+00004	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00110	+00012	+00512	+00024	+00000	+00000	+00000	+00000	+00000	+00001
00120	+18434	+00012	+00003	+00000	-00253	+00000	+17921	+00200	+00008
00130	+00000	+00000	+00000	+00000	+00000	+00000	-00249	+00000	+00008
00140	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00150	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00160	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00170	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00180	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00190	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00200	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000	+00000
00210	+08817	+00290	+28929	+08817	+00000	5706	1615	0501	+00174

ID:	RUN/OUT EN	1ms	SCAN	MONITOR	L4	ACC:	WRITE	LOGIC	LOGIC	EQUAL
C:\LM90\SER2				PRG: SER2						
REPLACE		%R00207	:	:	:	:	:	:	:	:

%R00207 pierwszy rejestr aktualnych danych	%R00205 sekundy i minuty (SSMM)	%R00204 godzina i dzień (HHDD)	%R00203 miesiąc i rok (MMYY)	%R00202 Trigger Sample Offset Number	%R00201 Current Sample Offset Number
--	--	---	------------------------------------	--	--

## END

Blok funkcyjny END przerywa wykonanie części logicznej programu sterującego. Program wykonywany jest począwszy od pierwszego szczebla drabiny logicznej aż do szczebla ostatniego lub do momentu napotkania funkcji END.

Instrukcja END powoduje bezwarunkowe przerwanie wykonywania programu. Po funkcji End nie można umieszczać dalszej części programu sterującego. Część programu sterującego, umieszczona po funkcji END, nie jest dalej wykonywana, w następnym cyklu program jest wykonywany od początku. Należy zwrócić uwagę, że w szczeblach poniżej instrukcji END wejścia będą wydawać się załączone lub wyłączone, ale wyjścia nie będą uaktualniane. Mimo, że w normalnych warunkach może być to problemem, jeśli nie jest oczywiste, że marker END poprzedza działające szczeble.

Funkcja END jest przydatna przy wykrywaniu i usuwaniu usterek gdyż pozwala na izolowanie części programu sterującego. Dzieje się tak przez niedopuszczenie do wykonania części programu następującego po markerze END.

W oprogramowaniu Logicmaster 90 domyślnie wstawiany jest marker wskazujący koniec części logicznej programu sterującego: [END OF PROGRAM LOGIC]. Element ten jest wykorzystywany, jeżeli w programie sterującym nie umieszczono funkcji END.

┌ [ END ]

## Przykład

W przykładzie tym styk %I0222 i przekaźnik %Q0017 oraz kolejne szczeble nie zostaną wykonane ponieważ wcześniej występuje instrukcja END.



### Uwaga

Użycie funkcji END w programie sterującym SFC lub w programie wywoływanym z poziomu SFC powoduje wygenerowanie w jednostkach centralnych w wersji 7 i późniejszych błędu "END Function Executed from SFC Action". (W jednostkach centralnych, wersjach wcześniejszych od wersji 7, funkcja ta pracuje nie prawidłowo, ale nie jest generowany błąd). Więcej informacji na temat tego błędu podano w Rozdziale 3, punkcie 2 pod hasłem "System Configuration Mismatch".

## MCRN/MCR

### Przegląd funkcji MCR i MCRN

Funkcja Master Control Relay (MCR/MCRN) musi być stosowana wraz z odpowiadającą jej funkcją End Master Control Relay (ENDMCR/ENDMCRN). Obydwie te funkcje muszą posiadać taką samą etykietę. Funkcje MCR/MCRN nie mogą być bezpośrednio podłączone do szyny zasilającej. Wszystkie szczeble programu sterującego, umieszczone pomiędzy instrukcją MCR/MCRN i odpowiadającą jej funkcją ENDMCR/ENDMCRN zostaną wykonane bez dopływu sygnału sterującego. Po napotkaniu funkcji ENDMCR/ENDMCRN wznawiane jest normalne wykonywanie programu sterującego. W odróżnieniu od instrukcji JUMP, instrukcja MCR/MCRN może być użyta tylko w kierunku do przodu programu. Instrukcje ENDMCR/ENDMCRN muszą znajdować się później w programie niż odpowiadające im instrukcje MCR/MCRN.

Wpływ instrukcji MCR/MCRN na wykonywanie programu:

- Przełączniki czasowe nie są inkrementowane lub dekrementowane. Przełączniki czasowe typu TMR są zerowane. Przełącznik czasowy ONDTR zostaje zatrzymany na wartości, którą posiadał przed uaktywnieniem instrukcji MCR/MCRN.
- Przepływ sygnału sterującego zostaje zatrzymany. Normalne wyjścia są zerowane; wyjścia zanegowane są ustawiane na 1.
- Funkcje nie uaktualniają swoich wyjść. Na przykład, instrukcja ADD nie przeprowadzi działania i nie załączy wyjścia, funkcja Move nie będzie kopiowała wartości z wejścia na wyjście, blok SHFR nie będzie przesuwac danych itp. Wartości rejestrów wyjściowych zostaną zamrożone.

#### Uwaga

Po doprowadzeniu sygnału do instrukcji MCR/MCRN, obejmowany przez nią fragment programu sterującego jest wykonywany, wyświetlany jest stan styków, ale nie są wysyłane sygnały wyjściowe. Jeżeli użytkownik nie ma świadomości, że dany fragment programu sterującego znajduje się w zasięgu instrukcji MCR/MCRN, może podejrzewać błędne wykonywanie programu sterującego. W celu zasygnalizowania, że dany szczebel programu sterującego znajduje się pod kontrolą funkcji MCR/MCRN, oprogramowanie Logicmaster wyświetla na ekranie podwójną szynę sygnałów. Podwójna szyna sygnałowa jest wyświetlana niezależnie od tego czy funkcja MCR/MCRN jest uaktywniona czy nie.

Oprogramowanie Logicmaster 90-30/Micro udostępnia dwa rodzaje funkcji MCR: bez możliwości pokrywania się zakresów działania (MCR) i z możliwością pokrywania się zakresów działania (MCRN).



## Kompatybilność z jednostkami centralnymi

Typ jednostki centralnej	Obsługiwane rodzaje
CPU311 – CPU341, wersja 1	Dozwolone jest sotoswanie wyłącznie formy nie zagnieżdżonej (MCR)
CPU311 – CPU341, wersja 2 i późniejsze	Dozwolone jest sotoswanie wyłącznie formy zagnieżdżonej (MCRN)
CPU 35x, 36x i 37x	Dozwolone jest sotoswanie wyłącznie formy zagnieżdżonej (MCRN)

### Możliwe problemy z kompatybilnością

Podczas konwertowania programu sterującego z CPU340 lub CPU341 do CPU35x/36x/37x w oprogramowaniu Logimaster możliwe jest wystąpienie błędu "Feature not Supported" (przekroczone poziomy zagnieżdżenia). Może się tak dziać w programie po konwersji do CPU35x./36x/37x jeśli występuje więcej niż osiem poziomów zagnieżdżenia funkcji MCRN.

Instrukcje MCRN są aktualnie funkcją w CPU340/341, co oznacza, że są wykonywane przez jednostkę centralną, a nie przez koprocessor logiczny (BCP). Poziom zagnieżdżenia dla bloku funkcyjnego został ograniczony do 256. Poziom taki jest znacznie większy od wykorzystywanego, na co dzień. W czasie projektowania jednostek centralnych 35x/36x/37x instrukcje MCRN zostały przeprogramowane do wykonywania przez koprocessor logiczny w celu podwyższenia możliwości samej jednostki centralnej (blok funkcyjny był wykonywany wolniej niż w koprocessorze). W koprocessorze logicznym BCP3 stosowanym w jednostkach centralnych 35x/36x/37x zaimplementowano osiem poziomów zagnieżdżenia. W związku z tym Logimaster 90 nałucza osiem poziomów zagnieżdżenia w czasie konwersji programu, a w przypadku gdy występuje więcej niż osiem poziomów zagnieżdżeń wydaje komunikat "Nesting Levels Exceeded".

Dlatego programy, w których występuje więcej niż osiem poziomów zagnieżdżeń po konwersji będą wymagały modyfikacji. Należy więc rozważyć zastosowanie funkcji JUMP.

### Zagnieżdżanie funkcji MCRN

Funkcja MCRN może być umieszczana w dowolnym miejscu w programie, pod warunkiem, że jest poprawnie zagnieżdżona wewnątrz innych funkcji MCRN oraz, że zakres jej działania nie pokrywa się z zakresem działania funkcji MCR i JUMP.

Instrukcja MCRN i odpowiadająca jej instrukcja ENDMCRN muszą być całkowicie zawarte wewnątrz innej pary instrukcji MCRN/ENDMCRN. Dopuszczalne jest zagnieżdżanie do 8 poziomów. Przykład podano na stronie 12-28.

Jednej instrukcji ENDMCRN może odpowiadać kilka instrukcji MCRN (za wyjątkiem jednostek centralnych serii 35x/36x i 37x, proszę porównać z zamieszczoną powyżej uwagą). Odpowiednie funkcje MCRN i ENDMCRN muszą mieć takie same nazwy. Jest to właściwość analogiczna do tej, jaką posiadają instrukcje JUMP/LABEL z możliwością zagnieżdżenia - jednej instrukcji LABEL może odpowiadać kilka instrukcji JUMP. Funkcje JUMP i MCR porównano w punkcie "Różnice pomiędzy funkcjami MCR i JUMP".

#### Uwaga

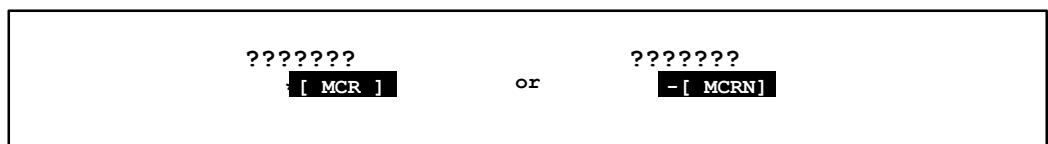
W jednostkach centralnych serii 35x, 36x i 37x każdej instrukcji ENDMCRN musi odpowiadać dokładnie jedna (1) instrukcja MCRN.

## Opis działania funkcji MCR

Na jeden blok funkcyjny MCR może przypadać tylko jeden blok funkcyjny ENDMCR. Zakres działania określony parami instrukcji MCR/ENDMCR nie może zawierać się wewnątrz zakresu działania innej pary instrukcji MCR/ENDMCR ani też pary JUMP/LABEL. Instrukcji MCR nie można umieszczać wewnątrz zakresu działania innej pary instrukcji MCR/ENDMCR lub pary instrukcji JUMP/LABEL.

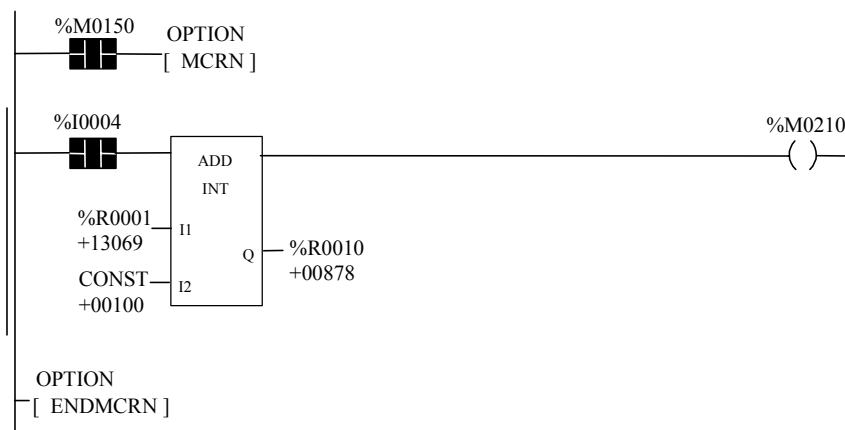
## Parametry

Obydwie formy instrukcji MCR posiadają takie same parametry. Są to: sygnał aktywujący EN typu bitowego oraz etykieta. Nazwa ta powinna również wystąpić ponownie przy odpowiednim bloku ENDMCR lub ENDMCRN. Bloki funkcyjne MCR oraz MCRN nie posiadają parametrów wyjściowych. Muszą być one ostatnimi instrukcjami w szczelbu.

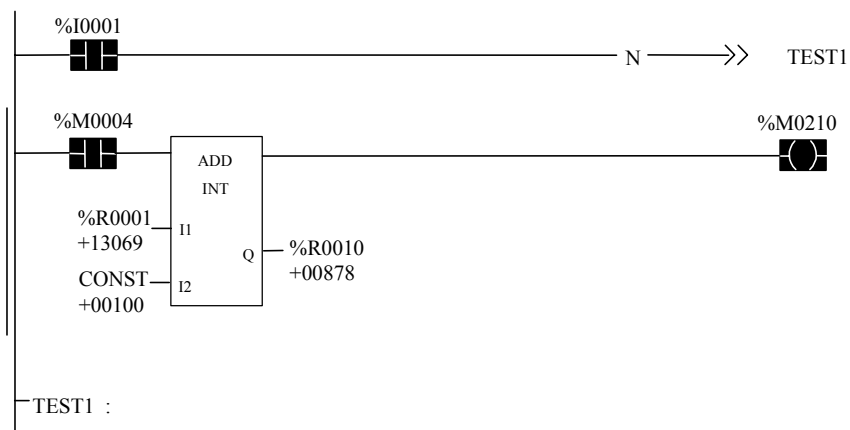


## Różnice pomiędzy funkcjami MCR/MCRN a JUMP

Bloki funkcyjne znajdujące się w zakresie działania funkcji MCR są wykonywane *bez dopływu sygnału sterującego*, wartość zmiennych przypisanych do przełączników *ustawiana jest na 0*. W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %M0150 na 1 powoduje aktywowanie funkcji MCRN. Jeżeli aktywna jest funkcja MCRN - nawet, jeżeli wartość zmiennej %I0004 jest równa 1 - blok funkcyjny ADD wykonywany jest *bez* doprowadzania sygnału sterującego (tzn. nie dodaje on wartości 100 do zmiennej %R0001), a zmienna %M0210 jest równa 0. Status styku takiego jak %I0004 i wartość rejestrów wejściowych takich jak %R0001 będą aktualizowane na ekranie, ale rejestry wyjściowe, które są pod kontrolą funkcji MCRN, takie jak %R0010 zostaną zatrzymane na ich bieżącej wartości.



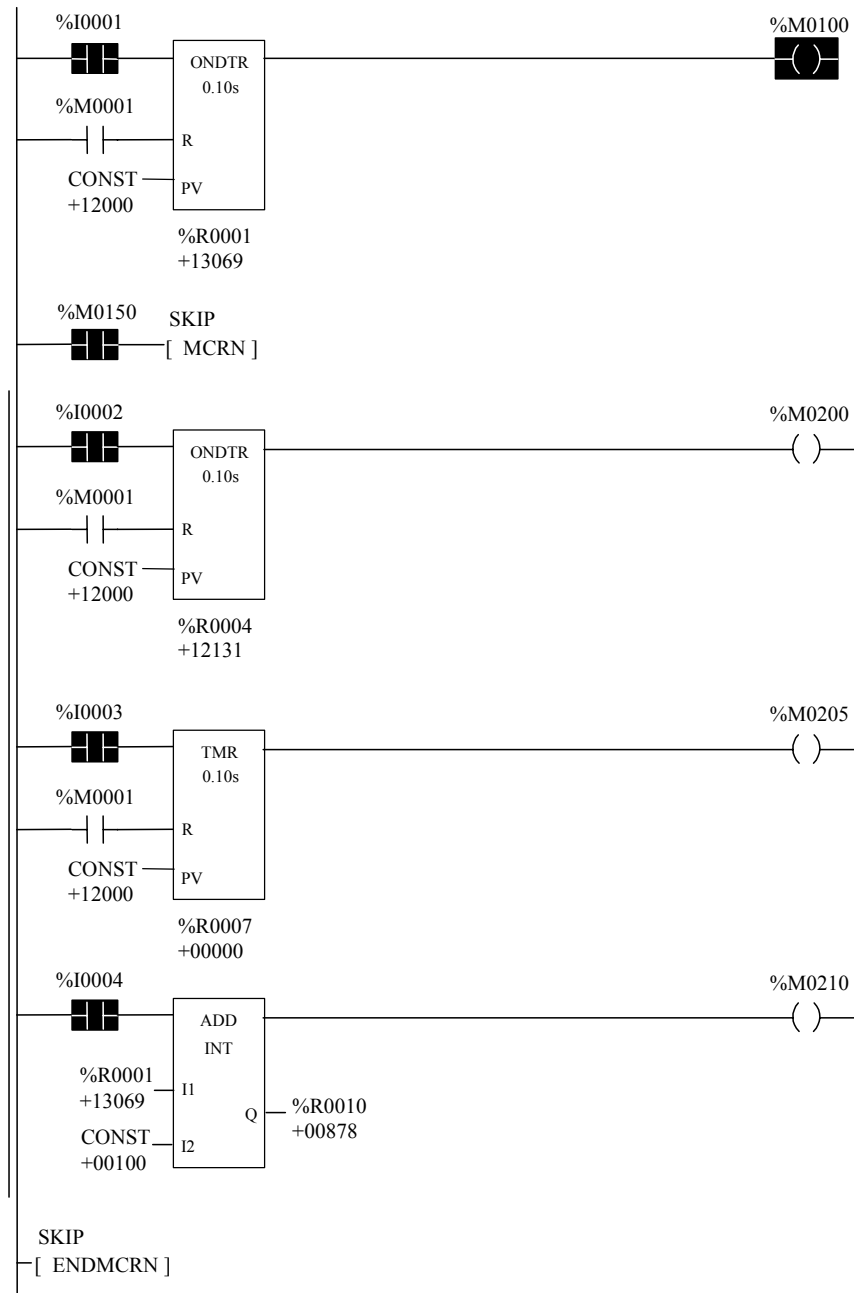
W przypadku funkcji JUMP, wszystkie bloki funkcyjne pomiędzy instrukcjami JUMP i LABEL *nie są* wykonywane oraz *nie jest zmieniana* wartość zmiennych przełącznikowych. W zamieszczonym poniżej przykładzie, zmiana wartości zmiennej %I0001 na 1 powoduje aktywowanie funkcji Jump o nazwie TEST1. Ponieważ bloki funkcyjne umieszczone pomiędzy instrukcjami JUMP i LABEL zostają pominięte, wartość zmiennej %M0210 nie ulega zmianie (tzn. jeżeli miała ona wartość 1, pozostawiana jest wartość 1, a jeżeli miała wartość 0, pozostawiana jest wartość 0). Status styku takiego jak %M0004 i wartość rejestrów wejściowych takich jak %R0001 będą aktualizowane na ekranie, ale rejestry wyjściowe, które są pod kontrolą funkcji JUMP, takie jak %R0010 zostaną zatrzymane na ich bieżącej wartości.





## Przykład 2

W tym przykładzie pierwszy szczebel drabiny logicznej jest wykonywany normalnie. Jednakże blok MCRN o nazwie SKIP kontroluje resztę szczebli, dlatego wyświetlana jest przy nich podwójna szyna sygnałowa. W pierwszym szczeblu kontrolowanym przez MCRN znajduje się przełącznik czasowy ONDTR, który nawet po zliczeniu odpowiedniej wartości nie załączy wyjścia (%M0200). W następnym szczeblu znajduje się blok TMR który został wyzerowany przez MCRN. Jego wartość zliczona (%R0007) jest zatrzymywana na wartości zero, wyjście (%M0205) ni zostaje załączone. W następnym szczeblu zostaje zamrożone wyjście bloku ADD (wyjście %R0010 nie jest sumą parametrów wejściowych) oraz nie zostaje załączone wyjście sygnalizujące wyoknanie operacji (%M0210). Należy zwrócić uwagę, że na ekranie status styków i wartości rejestrów wejściowych (takich jak %R0001) są aktualizowane.

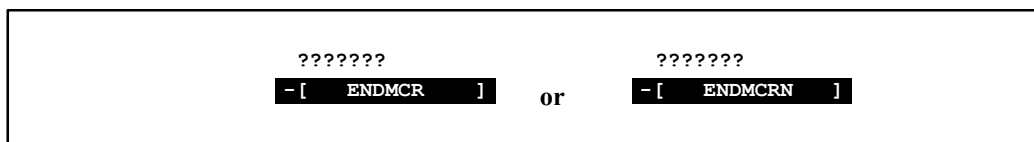


## ENDMCRN/ENDMCR

Blok funkcyjny ENDMCR/ENDMCRN przywraca normalne wykonywanie programu sterującego (z normalnym dopływem sygnału do szczebli programu). Jeżeli do bloku funkcyjnego MCR dopływa sygnał wejściowy, ENDMCR kończy działanie tego bloku. Jeżeli do bloku MCR nie dopływa sygnał wejściowy, wywołanie bloku funkcyjnego ENDMCR nie powoduje żadnego działania.

Oprogramowanie Logimaster 90-30/Micro udostępnia dwa rodzaje funkcji MCR: bez możliwości zawierania się w sobie, ani też częściowego pokrywania się zakresów działania przekaźników (funkcja MCR) i z możliwością pokrywania się zakresów działania przekaźników, pod pewnymi ograniczeniami (funkcja MCRN). Instrukcja ENDMCR może być stosowana wyłącznie dla instrukcji MCR (a nie MCRN). Instrukcja ENDMCRN może być natomiast stosowana wyłącznie dla instrukcji MCRN.

Pierwszym parametrem bloku funkcyjnego ENDMCR jest zanegowane wejście sygnału sterującego EN. Sygnał wejściowy musi być doprowadzany bezpośrednio z szyny sygnału, nie dopuszczalne jest stosowanie warunkowego wykonania. Drugim parametrem jest nazwa identyfikacyjna bloku, wiążąca go z odpowiednim blokiem (blokami) MCR. Bloki funkcyjne ENDMCR oraz ENDMCRN nie posiadają parametrów wyjściowych. Powinny one być jedynymi instrukcjami w swoich szczeblach.



### Przykład

W zamieszczonym poniżej przykładzie, instrukcja ENDMCR powoduje przerwanie działania funkcji MCR "CLEAR".

Przykład nie zagnieżdżonej instrukcji ENDMCR:

```
|  
| CLEAR  
|  
|- [ ENDMCR ]  
|
```

Przykład zagnieżdżonej instrukcji ENDMCRN:

```
|  
| CLEAR  
|  
|- [ ENDMCRN ]  
|
```

## JUMP

Blok funkcyjny JUMP powoduje pominięcie fragmentu części logicznej programu sterującego, zamieszczonego pomiędzy instrukcją JUMP a instrukcją LABEL (etykieta). Gdy do bloku funkcyjnego JUMP dopływa sygnał wejściowy, wszystkie przełączniki zawarte we wspomnianym obszarze zachowują swój pierwotny stan. (dotyczy to również takich elementów logicznych, jak przełączniki czasowe, liczniki, itp.).

Oprogramowanie Logicmaster 90 posiada dwa typy instrukcji skoku: bez możliwości zagnieżdżenia i z możliwością zagnieżdżenia. Pierwsza z nich, bez możliwości zagnieżdżenia, dostępna począwszy o wersji 1 jednostek centralnych 311-341, oznaczana jest symbolem, `—————>>LABEL01`, gdzie LABEL01 jest nazwą nie zagnieżdżonej instrukcji LABEL.

W przypadku instrukcji JUMP bez możliwości zagnieżdżenia, każdej instrukcji JUMP odpowiada dokładnie jedna instrukcja LABEL. Instrukcja JUMP może powodować przejście zarówno w kierunku do przodu, jak i do tyłu programu sterującego.

Zakres działania określony parami instrukcji MCR/ENDMCR nie może zawierać się wewnątrz zakresu działania innej pary instrukcji MCR/ENDMCR ani też pary JUMP/LABEL. Instrukcje JUMP bez możliwości zagnieżdżenia nie mogą być umieszczane wewnątrz zakresu działania innej pary instrukcji JUMP/LABEL lub pary instrukcji MCR/ENDMCR. Dodatkowo, para instrukcji MCR/MNDMCR lub inna para instrukcji JUMP/LABEL nie mogą znajdować się obrębie instrukcji JUMP/LABEL bez możliwości zagnieżdżenia.

### Uwaga

W wersji 1 sterowników centralnych serii 90-30 dostępna jest wyłącznie nie zagnieżdżana instrukcja JUMP. We wszystkich nowo opracowywanych aplikacjach zalecane jest stosowanie zagnieżdżanej instrukcji JUMP.

Należy również zwrócić uwagę, że jednostki centralne serii 35x/36x i 37x obsługują wyłącznie zagnieżdżane instrukcje JUMP.

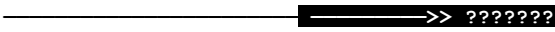
Zagnieżdżona forma instrukcji JUMP ma symbol `————N————>>LABEL01`, gdzie LABEL01 jest nazwą zagnieżdżonej instrukcji LABEL. Zagnieżdżona funkcja JUMP dostępna jest w wersji 2 i wszystkich wersjach późniejszych oprogramowania Logicmaster i oprogramowania systemowego sterowników 90-30/Micro.

Zagnieżdżona instrukcja JUMP może być umieszczana w dowolnym miejscu w programie, pod warunkiem, że jest poprawnie zagnieżdżona wewnątrz innych nie zagnieżdżonych i zagnieżdżonych instrukcji JUMP.

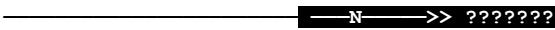
Pojedynczej, zagnieżdżanej instrukcji LABEL typu może odpowiadać kilka instrukcji JUMP tego samego typu. Zagnieżdżana instrukcja JUMP może powodować przejście zarówno w kierunku do przodu jak i do tyłu programu sterującego.

Obydwie formy instrukcji JUMP umieszczane są zawsze w kolumnie 9 i 10 bieżącego szczebla, po instrukcji JUMP w szczeblu nie mogą być umieszczane żadne inne elementy logiczne. Po napotkaniu takiej instrukcji, sygnał sterujący kierowany jest bezpośrednio do szczebla o zadanej etykiecie.

Nie zagnieżdżona instrukcja JUMP:



Zagnieżdżona instrukcja JUMP:

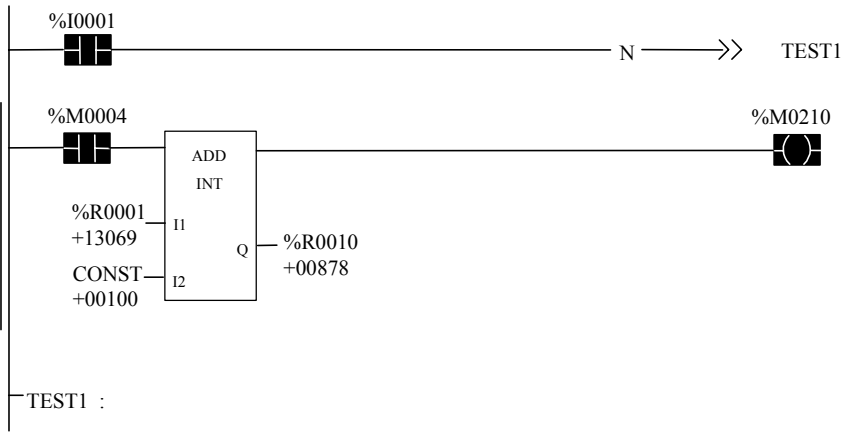


**Ostrzeżenie**

Aby uniknąć zapętlenia programu sterującego przy użyciu instrukcji JUMP wstecz musi być związany warunek logiczny.

**Przykłady**

W poniższym przykładzie załączenie styku %I0001 powoduje uaktywnienie funkcji JUMP o nazwie TEST1 i następuje przeniesienie sygnału sterującego do miejsca oznaczonego TEST1. Ponieważ bloki funkcyjne umieszczone pomiędzy instrukcjami JUMP i LABEL zostają pominięte, wartość zmiennej %M0210 nie ulega zmianie (tzn. jeżeli miała ona wartość 1, pozostawiana jest wartość 1, a jeżeli miała wartość 0, pozostawiana jest wartość 0). Status styku takiego jak %M0004 i wartości rejestrów wejściowych takich jak %R0001 będą aktualizowane na ekranie, ale rejestry wyjściowe, które są pod kontrolą funkcji JUMP, takie jak %R0010 zostaną zatrzymane na ich bieżącej wartości. Należy zwrócić uwagę że w miejscu na podwójną szynę sygnałową w obszarze objętym działaniem funkcji JUMP i LABEL.





# LABEL

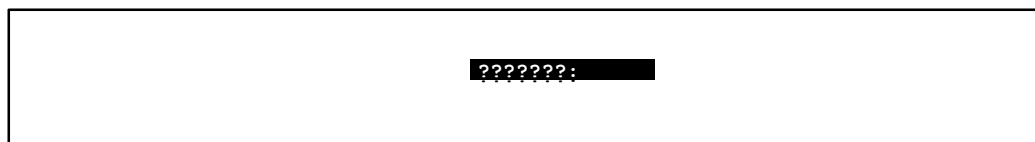
Blok funkcyjny LABEL (etykieta) określa miejsce docelowe skoku (wywołanego przez funkcję JUMP z tą samą etykietą). Powoduje on kontynuację normalnego wykonywania programu sterującego począwszy od tej etykiety.

W programie sterującym nie mogą wystąpić dwie takie same etykiety. Również nie może zostać wykonany program, w którym występują instrukcje JUMP lub LABEL, bez odpowiedniej pary.

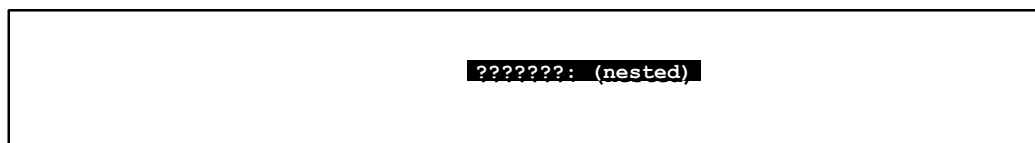
Oprogramowanie Logicmaster 90-30/Micro udostępnia dwa rodzaje funkcji etykiet: zagnieżdżane i nie zagnieżdżane. Na przykład forma nie zagnieżdżana, LABEL01:, może być stosowana wyłącznie z nie zagnieżdżaną instrukcją JUMP: —————>>LABEL01. Forma zagnieżdżana, LABEL01:, może być stosowana wyłącznie z zagnieżdżaną instrukcją JUMP: ———N——>>LABEL01.

Instrukcja LABEL(N) nie posiada żadnych parametrów wejściowych czy wyjściowych. Nie wolno umieszczać przed nią lub po niej żadnych innych bloków funkcyjnych.

Nie zagnieżdżona instrukcja LABEL:

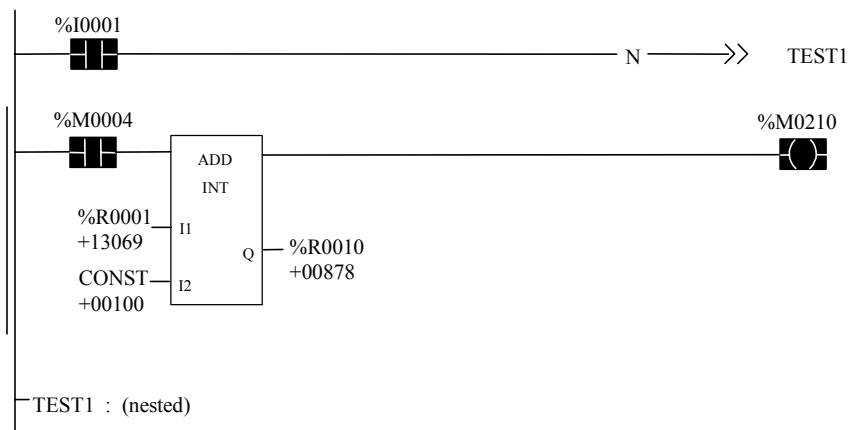


Zagnieżdżona instrukcja LABEL:



## Przykład

W poniższym przykładzie uaktywnienie funkcji JUMP TEST1 powoduje przesunięcie sygnału sterującego do szczelbli TEST1: (nested), co powoduje omińnięcie szczelbli pomiędzy instrukcją JUMP a LABEL.



## COMMENT

Blok funkcyjny COMMENT jest umożliwiał wstawianie w programie sterującym komentarza, uwag itp. Komentarze są przydatne, ponieważ dostarczają wiele przydatnych informacji w czasie diagnostyki programu. Dostarczają również wiele użytecznych informacji twórcy programu.

### Uwaga

Aby chronić pamięć sterownika adnotacje (komentarze, nazwy pomocnicze i opisy) nie są zapisywane w sterowniku. Dlatego, niezbędne jest posiadanie kopii programu sterującego (zawierającego adnotacje) by móc zobaczyć komentarze. Połączenie komputera ze sterownikiem spowoduje automatyczne odtworzenie połączeń pomiędzy adnotacjami a programem sterującym.

### Tworzenie standartowego komentarza

Komentarz może zawierać tekst o rozmiarach do 2048 znaków. W drabinie logicznej programu sterującego ma on następującą formę:

```
(* COMMENT *)
```

#### Tworzenie komentarza

1. Utworzyć nowy szczebel. Szczebel z komentarzem nie może zawierać żadnych elementów logicznych prócz instrukcji COMMENT.
2. Wstawić blok COMMENT, znajdujący się instrukcjach sterujących.
3. Zaakceptować szczebel wciskając klawisz Escape.
4. Przesunąć kursor na instrukcję (\*COMMENT\*) i wcisnąć klawisz F10 aby wpisać komentarz.
5. Wpisać tekst komentarza. Należy zwrócić uwagę, że wiersze nie są automatycznie zawijane. Zawijanie wierszy następuje po wciśnięciu klawisza Enter.
6. Po wpisaniu komentarza wcisnąć klawisz Escape, aby zapisać i zamknąć edytor komentarzy.

Tekst komentarza można wprowadzić oraz wyświetlić na ekranie komputera w celu dokonania w nim ewentualnych zmian po zaakceptowaniu szczebla z wstawionym blokiem funkcyjnym COMMENT, poprzez ustawienie kursora w jego pozycji i naciśnięcie klawisza F10 (Zoom). Szczebel z komentarzem może zostać również wydrukowany.

### Tworzenie długiego komentarza w celu wydrukownia

W oprogramowaniu Logicmaster dłuższe komentarze można również wprowadzać z pliku utworzonego za pomocą dowolnego edytora tekstu, pracującego w systemie MS-DOS.

1. Utworzyć komentarz (patrz wyżej):
  - A. Wprowadzić tekst komentarza, aż do miejsca, w którym dołączany będzie tekst znajdujący się w innym pliku.
  - B. W nowej linii wpisać \I (lub \i), wpisać ścieżkę dostępu do pliku z napisanym komentarzem, na przykład:

```
\I d:\text\commnt1
```

(Jeżeli zarówno plik jak i kartoteka programu znajdują się na tym samym dysku, nie jest konieczne wprowadzanie oznaczenia literowego dysku).

- C. Wcisnąć klawisz Escape aby zapisać tekst i zamknąć edytor komentarzy.
2. Uruchomić edytor tekstu i stworzyć plik tekstowy.
3. Zapisać tekst w formacie .txt nadając mu nazwę wprowadzoną w bloku comment i zapisać plik w miejscu określonym przez ścieżkę podaną w edytorze Comment.

## SVCREQ

Instrukcja Service Request jest instrukcją ogólnego przeznaczenia, która może wykonywać szeroki zakres funkcji specjalnych (usług), które nie są dostępne jako pojedyncze bloki funkcyjne. Blok funkcyjny SVCREQ służy do uruchomienia jednej ze specjalnych funkcji sterownika.

Tabela 12-4. Specjalne funkcje sterownika

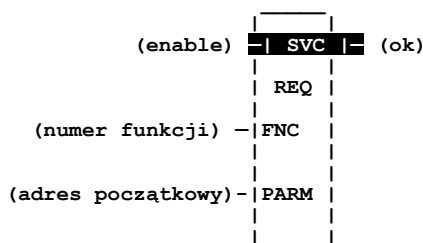
Funkcja	Opis
1	Zmiana/odczyt czasu trwania cyklu pracy sterownika w trybie ze stałym czasem trwania cyklu
2	Odczyt wartości z programatora
3	Zmiana trybu i czasu komunikacji z programatorem.
4	Zmiana trybu komunikacji systemowej oraz wartości przypisanej generatorowi sygnału prostokątnego.
6	Odczyt/zmiana liczby słów sumy kontrolnej programu sterującego.
7	Odczyt/zmiana wskazań zegara podtrzymującego aktualną datę i czas
8	Zerowanie zegara alarmowego.
9	Odczyt czasu trwania cyklu
10	Odczyt nazwy katalogu
11	Odczyt nazwy sterownika
12	Odczyt trybu pracy sterownika
13	Zatrzymanie sterownika na końcu bieżącego cyklu.
14	Wymazanie komunikatów z tabeli błędów działania sterownika i układów wejść/wyjść.
15	Odczyt ostatnio zarejestrowanego w odpowiedniej tabeli błędów
16	Odczyt wskazań zegara odmierzającego czas pracy sterownika
18	Kontrolę występowania wymuszonej zmiany wartości zmiennych wejściowych i wyjściowych.
23	Odczyt sumy kontrolnej programu sterującego i konfiguracji
24	Zerowanie inteligentnego modułu
26/30	Porównanie rzeczywistej konfiguracji modułów wejść/wyjść sterownika ze zdefiniowaną
29	Odczyt czasu trwania ostatniej przerwy w zasilaniu sterownika
45	Pominięcie obsługi wejść/wyjść w następnym cyklu. (Suspend I/O.)
46	Szybki dostęp do statusu magistrali komunikacyjnej
48	Ponowne uruchomienie po błędzie krytycznym automatycznego zerowania
49	Automatyczne zerowanie statystyk

## Podstawowe informacji o bloku funkcyjnym SVCREQ

Funkcja SVCREQ posiada trzy parametry wejściowe i jeden parametr wyjściowy. Gdy do bloku funkcyjnego dopływa sygnał wejściowy, uruchamiana jest funkcja sterownika o numerze identyfikacyjnym określonym za pomocą parametru FNC. Blok parametrów funkcji sterownika odczytywany jest z pamięci sterownika, z obszaru o początkowym adresie zadanym przez parametr PARM. Funkcja SVCREQ zawsze przesyła sygnał wyjściowy, chyba że podany zostanie niewłaściwy numer identyfikacyjny funkcji, błędne parametry lub adres wychodzący poza zakres pamięci sterownika. Ewentualne dodatkowe przyczyny niezadziałania funkcji opisano na kolejnych stronach podręcznika.

Adres zadany przez parametr PARM musi odnosić się do pamięci zorientowanej rejestrowo (%R, %AI lub %AQ). Adres ten oznacza początek bloku parametrów funkcji sterownika. Kolejne słowa 16-bitowe zawierają pozostałe parametry. Liczba parametrów uruchamianej funkcji sterownika zależy od jej typu (tzn. od numeru identyfikacyjnego FNC).

Obszar pamięci rozpoczynający się od adresu PARM zarezerwowany jest zarówno dla bloku parametrów wejściowych, jak i wyjściowych, przesyłanych jako wynik działania jednej ze specjalnych funkcji sterownika. Dane wejściowe i wyjściowe są więc przechowywane w tym samym miejscu.



## Parametry

Parametr	Opis
enable	Sygnał wejściowy. Gdy do bloku funkcyjnego dopływa sygnał, uruchamiana jest jedna ze specjalnych funkcji sterownika.
FNC	Każdy typ funkcji Service Request posiada unikalny numer, który musi zostać wprowadzony na wejściu FNC. FNC może być stałą jak również adresem zmiennej zawierającym numer funkcji.
PARM	Adres początkowy bloku parametrów, dla funkcji określonej za pomocą parametru FNC.
ok	Sygnał wyjściowy, wysyłany po poprawnym wykonaniu funkcji.

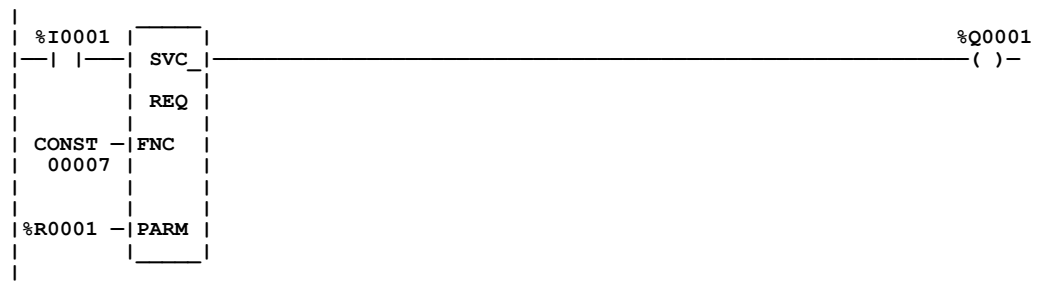
## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
FNC		•	•	•	•		•	•	•	•	•	
PARM		•	•	•	•		•	•	•	•		
ok	•											•

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

### Przykład

W zamieszczonym poniżej przykładzie, załączenie styku %I0001 powoduje uaktywnienie funkcji SVCREQ o numerze 7. Blok parametru rozpoczyna się na %R0001. W przypadku pomyślnego wykonania funkcji, wartość zmiennej przekaźnikowej %Q0001 jest ustawiana na 1.



## SVCREQ #1: Zmiana/odczyt czasu trwania cyklu pracy sterownika w trybie ze stałym czasem trwania cyklu

Funkcja SVCREQ #1, dostępna począwszy od jednostek centralnych 90-30, wersja 8,0, pozwala na:

- Wyłączenie trybu ze stałym czasem trwania cyklu (**CONSTANT SWEEP**).
- Włączenie trybu ze stałym czasem trwania cyklu (**CONSTANT SWEEP**), z pozostawieniem poprzedniej wartości czasu trwania cyklu pracy sterownika.
- Włączenie trybu ze stałym czasem trwania cyklu (**CONSTANT SWEEP**), z w prowadzeniem nowej wartości dla generatora sygnału prostokątnego.
- Wprowadzenie nowej wartości czasu trwania cyklu pracy sterownika.
- Sprawdzenie, czy sterownik pracuje w trybie **CONSTANT SWEEP** oraz na odczyt wartości czasu trwania cyklu pracy sterownika.

### Uwaga

Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #1 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Blok parametrów posiada długość dwóch słów.

W celu wyłączenia trybu **CONSTANT SWEEP**, wprowadzić funkcję SVCREQ #1 z następującym blokiem parametrów:

0	adres
ignorowany	adres + 1

W celu aktywowania trybu **CONSTANT SWEEP**, wprowadzić funkcję SVCREQ #1 z następującym blokiem parametrów:

1	adres
0 lub wartość czasu trwania cyklu programu	adres + 1

### Uwaga

Jeżeli zachodzi potrzeba wprowadzenia nowej wartości czasu trwania cyklu programu, wartość tę należy wprowadzić w drugim słowie. Jeżeli długość cyklu programu ma pozostać bez zmian, w drugim słowie należy wprowadzić 0. Jeżeli wartość czasu trwania cyklu pracy sterownika nie została do tej pory określona, wprowadzenie wartości 0 spowoduje ustawienie wyjścia OK na 0.

W celu dokonania zmiany czasu trwania cyklu programu **bez** wybierania trybu ze stałym czasem trwania cyklu, należy wprowadzić funkcję SVCREQ #1 z następującym blokiem parametrów:

2	adres
nowa wartość cyklu programu	adres + 1

To read the current timer state and value without changing either, enter SVCREQ function #1 with this parameter block: W celu odczytania bieżącego czasu trwania czasu cyklu, bez wprowadzania jakichkolwiek zmian, funkcja SVCREQ #1 powinna być wywołana z następującym blokiem parametrów:

3	adres
ignorowany	adres + 1

## Uwaga

Jeżeli funkcja SVCREQ #1 wywołana zostanie z blokiem parametrów, pokazanym na poprzedniej stronie, jednostki centralne w wersji 8 i późniejszych, zwrócą wartość 0 dla trybu standardowego i 1 dla trybu ze stałym czasem trwania cyklu. Nie należy mylić tych wartości z wartościami parametrów wejściowych, opisanych na następnym stronie.

Funkcja zostanie pomyślnie wykonana, o ile nie zaistnieje jedna z podanych poniżej sytuacji:

1. Jako parametr określający żądane działanie do wykonania wprowadzono liczbę inną niż 0, 1, 2 lub 3.

0	Wyłączenie trybu ze stałym czasem trwania cyklu ( <b>CONSTANT SWEEP</b> ).
1	Włączenie trybu ze stałym czasem trwania cyklu ( <b>CONSTANT SWEEP</b> ).
2	Wprowadzenie nowej wartości czasu trwania cyklu pracy sterownika.
3	Sprawdzenie, czy sterownik pracuje w trybie <b>CONSTANT SWEEP</b> , oraz odczyt wartości generatora sygnału prostokątnego. (Proszę porównać z zamieszczoną powyżej Uwagą).

2. Wartość czasu trwania cyklu pracy sterownika jest większa od 2550 ms (2.55 sek).
3. Tryb ze stałym czasem trwania cyklu jest włączany jeżeli nie jest wprowadzona wartość czasu trwania cyklu pracy sterownika lub też jeśli wprowadzono dla niego wartość 0.

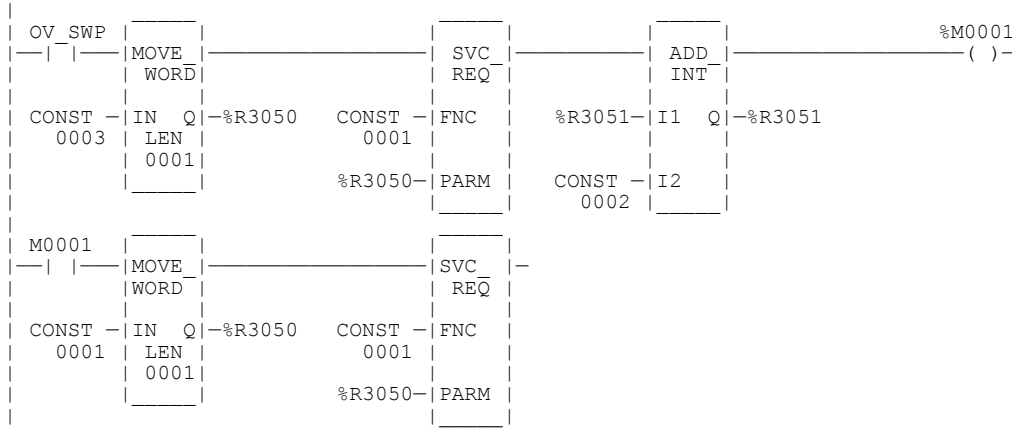
Po wykonaniu, funkcja ta zwraca w tym samym bloku czas trwania cyklu oraz wprowadzoną dla niego wartość.

0 = wyłączony	adres
1 = włączony	
bieżąca wartość czasu trwania cyklu	adres + 1

Jeżeli słowo o adresie adres +1 ma wartość heksadecymalną FFFF, oznacza to, że nie wprowadzono żadnej wartości czasu trwania cyklu pracy sterownika.

### Przykład

Zamieszczony poniżej przykład zawiera fragment programu sterującego. Jeżeli zmienna OV\_SWP jest równa 1, następuje odczyt ustawionego stałego czasu trwania cyklu, następnie odczytana wartość jest zwiększana o dwie milisekundy i przesyłana z powrotem do sterownika. Blok parametrów przechowywany jest w pamięci lokalnej o adresie początkowym %R3050. Ponieważ funkcje MOVE i ADD wymagają trzech poziomych połączeń styków, w przykładzie wykorzystano wewnętrzny przełącznik %M0001 do przechowywania pomyślnego wyniku pierwszego szczebla. W każdym cyklu, w którym wartość zmiennej OV\_SWP jest równa 0, zmienna %M0001 jest także równa 0.





## SVCREQ #2: Odczyt wartości z programatora

Funkcja SVCREQ #2 pozwala na odczyt wartości czasu trwania fazy komunikacji z programatorem, fazy komunikacji systemowej.

### Uwaga

Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #2 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Istnieją trzy tryby komunikacji:

Nazwa trybu	Wartość	Opis
Limited Mode (Komunikacja ograniczona czasowo)	0	Czas trwania komunikacji jest ograniczony do wartości domyślnej lub wartości zdefiniowanej za pomocą SVCREQ #3 dla komunikacji z programatorem, albo SVCREQ #4 dla komunikacji systemowej. Komunikacja jest zakończona, jeżeli zrealizowane zostały wszystkie żądania.
Constant Mode (Stały czas trwania komunikacji)	1	Dla każdego z rodzajów komunikacji ustawiany jest tryb Realizacji wszystkich żądań ( <b>RUN TO COMPLETION</b> ), a sterownik przełącza się pomiędzy poszczególnymi fazami komunikacji na czas równy sumie odpowiednich wartości dla każdego rodzaju komunikacji. Jeżeli dla jednego typu komunikacji wybrano tryb stałego czasu trwania ( <b>CONSTANT</b> ), tryb ten jest automatycznie wybierany dla pozostałych dwóch typów komunikacji. Jeżeli sterownik pracuje w trybie <b>CONSTANT WINDOW</b> i czas trwania dla jednego z typów komunikacji nie został określony za pomocą odpowiedniej funkcji SVCREQ, do obliczenia czasu trwania komunikacji wykorzystywany jest czas domyślny dla danego rodzaju komunikacji.
Run to Completion (Realizacja wszystkich zadań)	2	Bez względu na fakt, czy stosowany jest domyślny czy też określony za pomocą odpowiedniej funkcji SVCREQ czas trwania komunikacji, komunikacja jest realizowana aż do momentu zrealizowania wszystkich żądań.

Po ustawieniu wartości czasu na zero, komunikacja jest przerywana.

Blok parametrów posiada długość trzech słów.

	Bajt wyższy	Bajt niższy:	
Komunikacja z programatorem	Tryb	Wartość w ms	adres
Komunikacja systemowa	Tryb	Wartość w ms	adres + 1
Zarezerwowane*	*Patrz uwaga	*Patrz uwaga	adres +2

\* Uwaga. słowo adres+2 jest wykorzystywane przez system. Zostaną zwrócone same zera.

Wszystkie parametry są parametrami wyjściowymi. Funkcja ta nie wymaga określania jakichkolwiek parametrów w momencie jej wywołania. Parametry wyjściowe dla obu rodzajów komunikacji podawane są w milisekundach.

## Przykład

W zamieszczonym poniżej przykładzie, po każdej zmianie wartości zmiennej %Q0102 na 1, sterownik zapisuje bieżące wartości czasu trwania dla trzech rodzajów komunikacji w bloku parametrów, umieszczonym pod adresem %R0102. Dodatkowe przykłady, obrazujące zastosowanie funkcji SVCREQ #2 zamieszczono w opisach trzech następnych funkcji SVCREQ.

```
| %Q0102 | _____ |  
|—| |—| | SVC |  
|   |   | | REQ |  
|-----|-----|  
| CONST —| FNC |  
| 0002   | |  
|-----|-----|  
| %R5010—| PARM |  
|-----|-----|
```

## SVCREQ #3: Zmiana trybu i czasu trwania komunikacji z programatorem

Funkcja SVCREQ #3 pozwala na zmianę trybu komunikacji z programatorem oraz wartości czasu okna komunikacyjnego z programatorem. Zmiany zostaną uwzględnione w cyklu następującym po cyklu, w którym wywołana została ta funkcja.

### Uwaga

Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #3 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Funkcja SVCREQ #3 wysyła sygnał wyjściowy o ile został wybrany tryb 0 (Limited), 1 (Constant) lub 2 (Run-to-Completion).

Blok parametrów posiada długość jednego słowa.

W celu wyłączenia komunikacji z programatorem, wprowadzić funkcję SVCREQ #3 z następującym blokiem parametrów:

Bajt wyższy	Bajt niższy:	
0	0	adres

W celu włączenia komunikacji z programatorem, wprowadzić funkcję SVCREQ #3 z następującym blokiem parametrów:

Bajt wyższy	Bajt niższy:	
Tryb	Wartość z przedziału 1 do 255 ms	adres



## SVCREQ #4: Zmiana trybu komunikacji systemowej oraz czasu trwania komunikacji systemowej

Funkcja SVCREQ #4 pozwala na zmianę trybu komunikacji systemowej oraz czasu przeznaczanego na komunikację systemową. Zmiany zostaną uwzględnione w cyklu następującym po cyklu, w którym wywołana została ta funkcja.

### Uwaga

Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #4 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Funkcja SVCREQ #4 wysyła sygnał wyjściowy o ile został wybrany tryb 0 (Limited), 1 (Constant) lub 2 (Run-to-Completion).

Blok parametrów posiada długość jednego słowa.

W celu wyłączenia komunikacji systemowej, wprowadzić funkcję SVCREQ #4 z następującym blokiem parametrów:

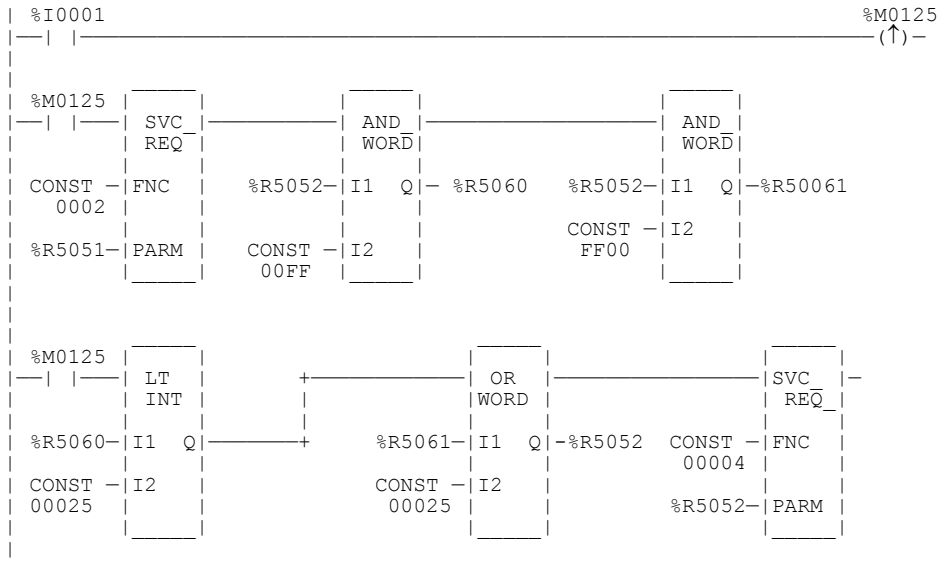
Bajt wyższy	Bajt niższy:	
0	0	adres

W celu włączenia komunikacji systemowej, wprowadzić funkcję SVCREQ #4 z następującym blokiem parametrów:

Bajt wyższy	Bajt niższy:	
Tryb	Wartość z przedziału 1 do 255 ms	adres

### Przykład

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %M0125 na 1 powoduje odczytanie trybu i wartości czasu dla komunikacji systemowej. Jeżeli wartość czasu jest większa lub równa 25 ms, nie ulega ona zmianie. Jeżeli jest ona mniejsza niż 25 ms, zostanie zmieniona na 25 ms. W każdym przypadku wykonanie szeregu prowadzi do włączenia komunikacji. Adres początku bloku parametrów dla wszystkich trzech rodzajów komunikacji to %R5051. Ponieważ tryb i wartość czasu dla komunikacji systemowej określone są przez drugą wartość w bloku parametrów, zwracanym przez funkcję SVCREQ #2, czas komunikacji systemowej zawarty jest w niższym bajcie zmiennej %R5052.



## SVCREQ #6: Odczyt/zmiana liczby słów sumy kontrolnej programu sterującego

Funkcja SVCREQ z parametrem FNC o wartości 6 pozwala na:

- odczyt bieżącej liczby słów sumy kontrolnej programu sterującego,
- zmianę liczby słów sumy kontrolnej.

Funkcja ta zostanie poprawnie wykonana, pod warunkiem, że jako parametr określający żądane do wykonania działanie wprowadzona zostanie liczba 0 lub 1 (proszę porównać z zamieszczonymi poniżej informacjami).

Długość bloku parametrów dla tej funkcji wynosi 2 słowa.

### W celu odczytania bieżącej liczby słów sumy kontrolnej programu sterującego:

Wywołać funkcję SVCREQ #6 z następującym blokiem parametrów:

0	adres
ignorowany	adres + 1

Po wykonaniu, funkcja ta zwraca w tym drugim słowie bloku parametrów wartość sumy kontrolnej. Funkcja odczytu nie ma określonego zakresu wartości, zwracana jest liczba słów, dla których obliczana jest obecnie suma kontrolna.

0	adres
liczba słów sumy kontrolnej	adres + 1

### W celu ustawienia liczby słów do obliczania sumy kontrolnej:

Wywołać funkcję SVCREQ #6 z następującym blokiem parametrów:

1	adres
nowa liczba słów (0 - 32)	adres + 1

Po wprowadzeniu wartości 1, sterownik ustawia liczbę słów do obliczania sumy kontrolnej na wartość podaną jako drugie słowo bloku parametrów. Dla wszystkich jednostek centralnych sterowników 90-30 wartość drugiego słowa może zawierać się w granicach 0 – 32. Wartość z poza tego zakresu powoduje generowanie błędu. Dla sterowników serii 90-20 wartość ta wynosi 0 lub 4.

### Uwaga

Funkcja ta nie jest dostępna w sterownikach Micro.





## SVCREQ #7: Odczyt/zmiana wskazań zegara czasu rzeczywistego

Funkcja SVCREQ #7 pozwala na odczyt/zmianę wskazań zegara sterownika, podtrzymującego aktualną datę i czas.

### Uwaga

Funkcja ta dostępna jest wyłącznie w jednostkach centralnych 90-30, modele 331 i wyższe, w jednostkach centralnych 28-o punktowych PAGE 28-point series Micro PLC sterowników serii 90-30 (tzn. IC693UDR005, IC693UAA007 i IC693UDR010) oraz w jednostkach centralnych 23 punktowych sterowników serii 90 (IC693UAL006).

Funkcja zostanie pomyślnie wykonana, o ile nie zaistnieje jedna z podanych poniżej sytuacji:

1. Wartość pierwszego słowa bloku parametrów jest różna od 0 i od 1 (proszę porównać z opisem poniżej).
2. Wartość drugiego słowa bloku parametrów jest różna od 1 i od 3, co wskazuje na niemożliwy do zastosowania format zapisu.
3. Dane są zapisane w formacie innym niż wyszczególniony za pomocą drugiego słowa bloku parametrów.
4. Zostanie wprowadzona zła data taka jak 02/29/01, która określa rok 2001 jako przestępny (rok 2001 nie jest przestępny).

Dla funkcji daty/czasu, długość bloku parametrów zależy od formatu, w którym zapisane są dane. Kod zapisu BCD (liczby dziesiętne kodowane binarnie) wymaga 6 słów, a kod ASCII wymaga 12 słów.

0 = odczyt bieżącego czasu i daty	adres
1 = ustawienie nowego czasu i daty	
1 = dane w kodzie BCD	adres + 1
3 = dane w kodzie ASCII	
dane	adres od parametr + 2 do końca bloku parametrów

Wartość 1 słowa decyduje, czy funkcja ma czytać, czy też zmieniać wartości.

0 = **czytanie**  
1 = **zmiana**

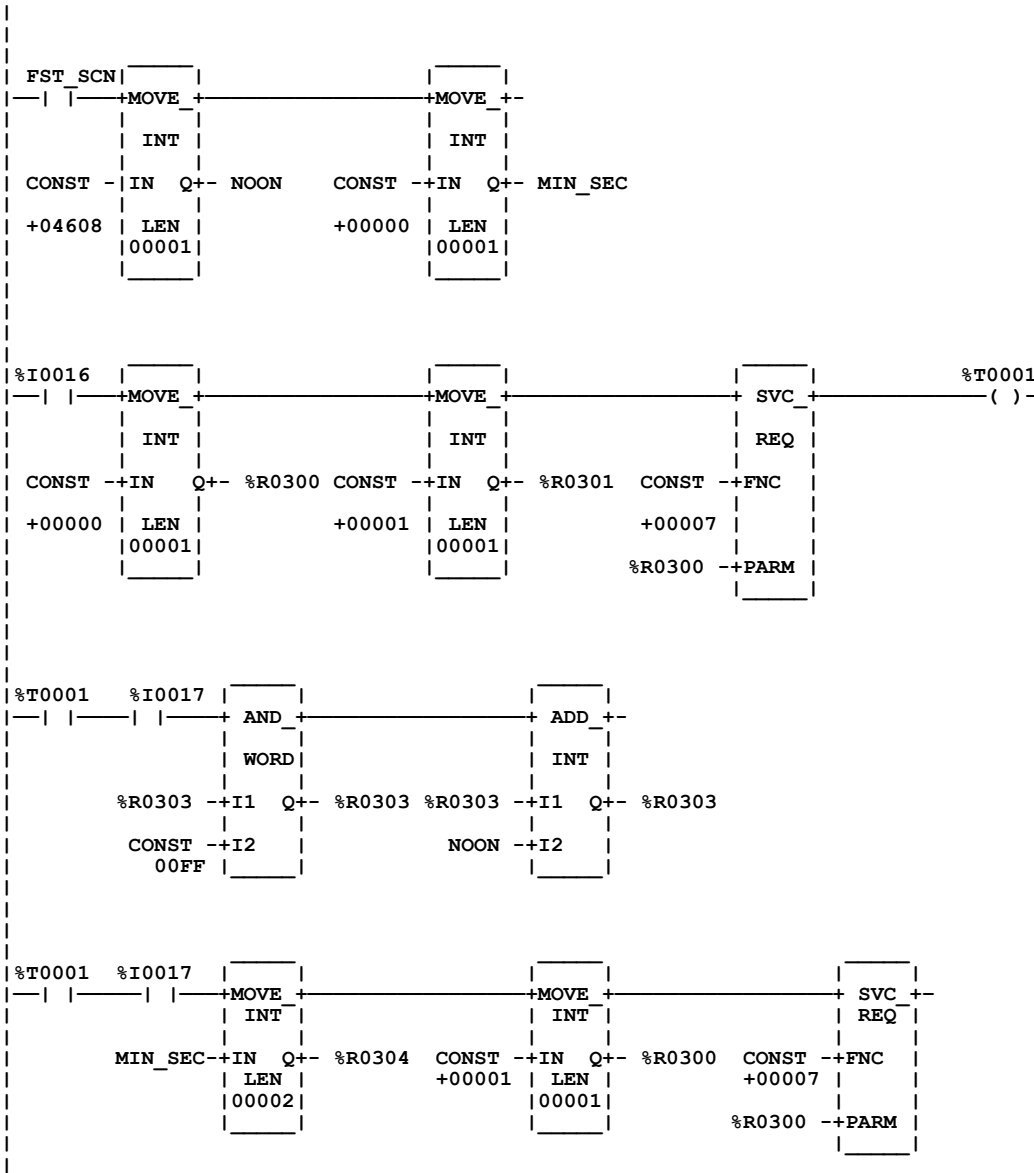
Drugie słowo określa format danych:

1 = **kod BCD**  
3 = **kod ASCII ze spacjami i znakami dwukropka (:)**

Słowa od trzeciego do końca bloku danych zawierają wartości zwrócone przez funkcję w przypadku odczytu (wartość pierwszego słowa bloku parametrów jest równa 0) lub zadane przez użytkownika w przypadku ustawiania nowego czasu i daty (wartość pierwszego słowa bloku parametrów jest równa 1). W obydwu przypadkach, format słów z danymi jest taki sam. W czasie czytania daty i czasu, słowa (adres+2) do (adres + 8) bloku parametrów są ignorowane na wejściu.

**Przykład**

W zamieszczonym poniżej przykładzie tworzony jest blok parametrów, który ma za zadanie odczytanie bieżącej daty i czasu, a następnie ustawienie wskazań zegara na godzinę 12:00:00 w formacie BCD. Blok parametrów umieszczony jest w obszarze danych globalnych %R0300. Tablica NOON została wcześniej zadeklarowana w programie, zawiera ona wartości 12, 0 i 0. (Tablica NOON musi zawierać również dane zapisane pod adresem %R0300.) Format zapisu wymaga zarezerwowania sześciu sąsiadujących komórek pamięci na blok parametrów.



## Zawartość bloku danych

Blok danych zawiera dokładną datę i czas, zapisane wg podanych poniżej reguł: Dla obydwu formatów zapisu danych:

- Godziny są zapisywane w konwencji 24-godzinnej;
- Dni tygodnia zapisywane są w konwencji numerycznej:

Wartość	Dzień tygodnia
1	Poniedziałek
2	Wtorek
3	Środa
4	Czwartek
5	Piątek
6	Sobota
7	Niedziela

## Zmiana/odczyt czasu w formacie BCD:

W formacie wykorzystującym kod zapisu BCD poszczególne elementy daty i czasu zajmują po jednym bajcie. Format taki wymaga zastosowania sześciu słów. Ostatni bajt szóstego słowa nie jest używany. Podczas ustawiania daty i czasu bajt ten jest ignorowany, natomiast przy odczytywaniu bieżącej daty i czasu bajt ten jest wypełniany zerami (tzn. ma wartość 00 w kodzie BCD).

Bajt wyższy		Bajt niższy:		
1 = zmiana lub		0 = odczyt		adres
1				adres + 1
miesiąc		rok		adres + 2
godziny		dzień miesiąca		adres + 3
sekundy		minuty		adres + 4
(puste pola)		dzień tygodnia		adres + 5

Przykładowy, wyjściowy blok parametrów:  
Odczyt daty i czasu w formacie BCD  
(Niedziela, 3 lipca 1988, godz. 14:45:30)

0	
1	
07	88
14	03
30	45
00	01

## W celu zmiany/odczytu daty i czasu w formacie ASCII ze spacjami i znakami dwukropka

W formacie tym każda cyfra daty i czasu zapisywane są w kodzie ASCII. Dodatkowo, spacje i dwukropki włączone są do bloku danych, aby umożliwić przesłanie bloku danych w niezmienionej formie na ekran lub drukarkę. Format taki wymaga zastosowania 12 słów.

Bajt wyższy	Bajt niższy:	
1 = zmiana lub	0 = odczyt	adres
3		adres + 1
rok	rok	adres + 2
miesiąc	(spacja)	adres + 3
(spacja)	miesiąc	adres + 4
dzień miesiąca	dzień miesiąca	adres + 5
godziny	(spacja)	adres + 6
:	godziny	adres + 7
minuty	minuty	adres + 8
sekundy	:	adres + 9
(spacja)	sekundy	adres + 10
dzień tygodnia	dzień tygodnia	adres + 11

Przykładowy, wyjściowy blok parametrów:  
Odczyt daty i czasu w formacie ASCII  
(Poniedziałek, 2 października 1989, 23:13:00)

0	
3	
39	38
31	20
20	30
32	30
32	20
3A	33
33	31
30	3A
20	30
32	30

## SVCREQ #8: Zerowanie zegara alarmowego

Funkcja SVCREQ #8 pozwala na wyzerowanie zegara alarmowego w czasie trwania cyklu.

### Uwaga

Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #8 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Po upływie nastawionego na tym zegarze czasu, sterownik jest resetowany. Funkcja ta pozwala na zmianę wskazań tego zegara w czasie czasochłonnych zadań (na przykład w okresie oczekiwania na odpowiedź z systemu komunikacji).

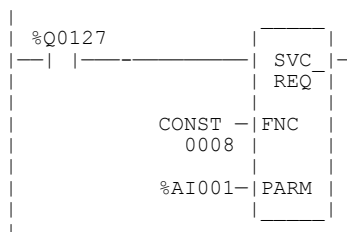
### Ostrzeżenie

**Należy upewnić się, że wznowienie pracy zegara alarmowego nie będzie miało ujemnego wpływu na sterowany proces.**

Funkcja ta nie posiada bloku parametrów, pomimo tego, pakiet programistyczny wymaga wprowadzenia wartości parametru PARM.. Należy wprowadzić odpowiednią wartość, nie będzie ona wykorzystywana.

### Przykład

W zamieszczonym poniżej przykładzie złączenie styku %Q0127 powoduje wyzerowanie zegara alarmowego.



## SVCREQ #9: Odczyt czasu trwania cyklu

Funkcja SVCREQ #9 pozwala na odczyt czasu w milisekundach, który upłynął od momentu rozpoczęcia cyklu. Dane są zapisywane w formacie 16 bitowych słów.

### Uwaga

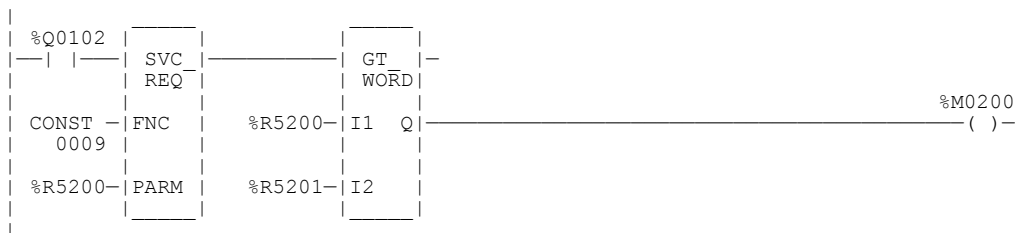
Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #9 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Blok parametrów pełni wyłącznie funkcję wyjściowego bloku, ma on długość jednego słowa.

czas od rozpoczęcia cyklu    adres

### Przykład

W zamieszczonym poniżej przykładzie, czas, który upłynął od momentu rozpoczęcia cyklu zapisywany jest do zmiennej %R5200. Jeżeli jego wartość jest większa od wartości zmiennej %R5201, następuje ustawienie wartości wewnętrznego przełącznika %M0200 na 1.



## SVCREQ #10: Odczyt nazwy folderu

Funkcja SVCREQ #10 pozwala na odczyt wykorzystywanego w danym momencie folderu.

### Uwaga

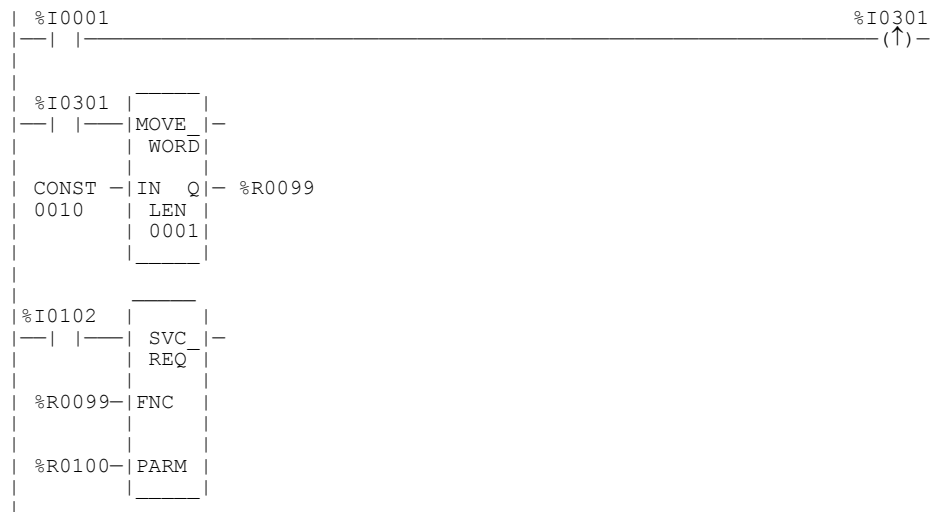
Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #10 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Blok parametrów wyjściowych posiada długość czterech słów. Zwraca on osiem znaków ASCII, ostatni z nich jest pusty (00h). Jeżeli nazwa programu jest krótsza od siedmiu znaków, na końcu dołączane są znaki puste.

Bajt niższy:	Bajt wyższy	
znak 1	znak 2	adres
znak 3	znak 4	adres + 1
znak 5	znak 6	adres +2
znak 7	00	adres +3

### Przykład

W zamieszczonym poniżej przykładzie, zmiana wartości zmiennej %I0301, powoduje zapisanie wartości 10 do rejestru %R0099. Wartość ta jest numerem identyfikacyjnym funkcji SVCREQ #10 do odczytu nazwy folderu. W kolejnym szczelbku załączenie styku %I0102, powoduje zczytanie nazwy folderu i zapisanie go w bloku czterech słów danych rozpoczynających się od %R0100 (określone przez parametr PARM).



## SVCREQ #11: Odczyt identyfikatora sterownika

Funkcja SVCREQ #11 pozwala na odczytanie nazwy sterownika serii 90, wykonującego program.

### Uwaga

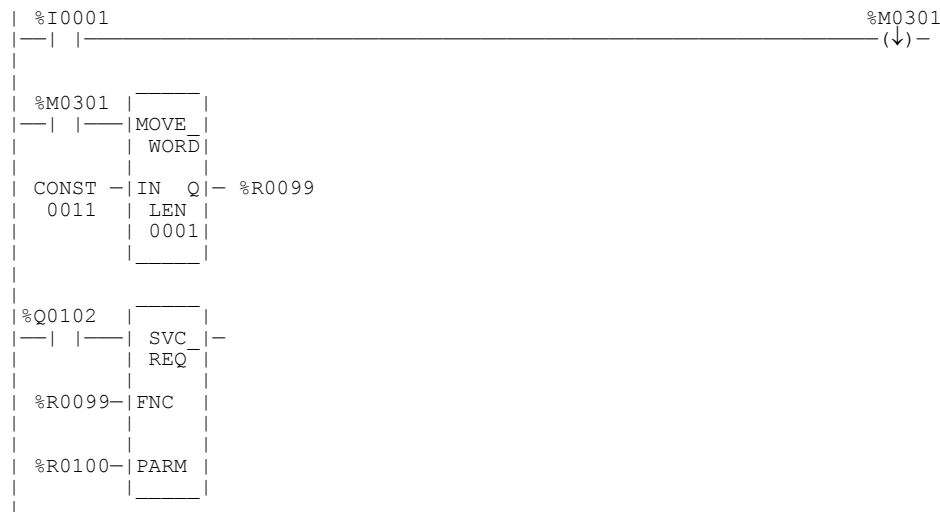
Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #11 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Blok parametrów wyjściowych posiada długość czterech słów. Zwraca on osiem znaków ASCII, ostatni z nich jest pusty (00h). Jeżeli długość identyfikatora sterownika jest krótsza od siedmiu znaków, na końcu dołączane są znaki puste.

Bajt niższy:	Bajt wyższy	
znak 1	znak 2	adres
znak 3	znak 4	adres + 1
znak 5	znak 6	adres +2
znak 7	00	adres +3

### Przykład

W zamieszczonym poniżej przykładzie, zmiana wartości zmiennej %I0001 na 0, powoduje zapisanie wartości 11 do rejestru %R0099. Wartość ta jest numerem identyfikacyjnym funkcji SVCREQ #11 do odczytu nazwy sterownika. W kolejnym szczeblu załączenie styku %Q0102, powoduje zczytanie ID sterownika i zapisanie go w bloku czterech słów danych rozpoczynających się od %R0100 (określone przez parametr PARM).





## SVCREQ #12: Odczyt trybu pracy sterownika

Funkcja SVCREQ #12 pozwala na odczytanie bieżącego statusu trybu pracy (RUN) jednostki centralnej sterownika.

### Uwaga

Spośród jednostek centralnych opisywanych w niniejszym podręczniku, funkcja SVCREQ #12 dostępna jest wyłącznie w serii 90-30, począwszy od wersji 8.

Blok parametrów pełni wyłącznie funkcję wyjściowego bloku, ma on długość jednego słowa. Możliwe są tylko dwa poprawne wyniki działania funkcji:

1 = praca/wyjścia nieaktywne	adres
2 = praca/wyjścia aktywne	

### Przykład

W zamieszczonym poniżej przykładzie załączenie styku %I0102 powoduje odczyt trybu pracy sterownika i zapisanie go w rejestrze %R402. Jeśli sterownik będzie znajdował się w trybie praca/wyjścia nie aktywne, w rejestrze %R402 zapisana zostanie wartość 1. Jeśli w trybie praca/wyjścia aktywne, zapisana zostanie wartość 2.

```

| %I0102 | _____ |
|-----|-----| SVC |
|         |         | REQ |
|-----|-----|
| CONST  | FNC  |
| 0012   |      |
|         |      |
| %R402  | PARM |
|         |      |
|-----|-----|

```





## SVCREQ #15: Odczyt ostatnio zarejestrowanego komunikatu o błędzie działania

Funkcja SVCREQ z parametrem FNC o wartości 15 służy do odczytania ostatniego komunikatu o błędzie w działaniu sterownika lub układów wejść/wyjść, zarejestrowanego w odpowiedniej tablicy. Funkcja ta wysyła sygnał wejściowy pod warunkiem, że jako parametr określający żądane do wykonania działanie wprowadzona zostanie liczba 0 lub 1 (proszę porównać z zamieszczonymi poniżej informacjami). (Dodatkowe informacje o tabelach błędów podano w Rozdziale 3 "Błędy działania sterownika - opis i ich usuwanie").

Funkcja ta posiada blok parametrów o długości 22 słów. Wejściowy blok danych ma następujący format:

0 = Odczyt ostatnio zarejestrowanego błędu w tabeli błędów działania sterownika	adres
1 = Odczyt ostatnio zarejestrowanego błędu w tabeli błędów działania układów wejścia/wyjścia	

Format danych wyjściowych zapisywanych w bloku parametrów funkcji, pod adresami od "PARM + 1" do "PARM + 21" zależy od tablicy, z której czytany jest komunikat.

### Format bloku parametrów wyjściowych dla tabeli błędów działania sterownika

Bajt niższy:	Bajt wyższy	
0		
wskaźnik długości bloku szczegółowych informacji o błędzie		adres + 1
pole wolne		adres +2
Adres wystąpienia błędu w działaniu sterownika		adres +3
		adres +4
Przynależność błędu do określonej grupy błędów oraz waga błędu i związana z nią reakcja systemu na jego wystąpienie		adres +5
kod błędu		adres +6
		adres +7
		adres +8
		adres +9
		adres +10
		adres +11
blok szczegółowych informacji o błędzie		adres +12
		adres +13
		adres +14
		adres +15
		adres +16
		adres +17
		adres +18
		adres +19
Czas i data wystąpienia błędu		adres +20
		adres +21

### Format bloku parametrów wyjściowych dla tabeli błędów działania układów wejść/wyjść

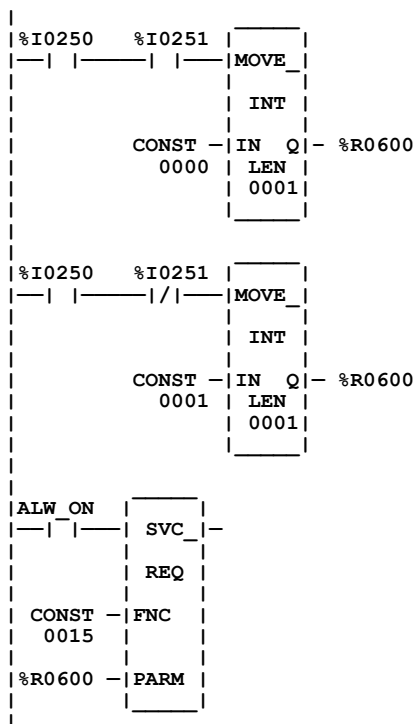
Bajt niższy:	Bajt wyższy	
1		
wskaźnik długości bloku szczegółowych informacji o błędzie		
Adres wystąpienia błędu		
Adres wystąpienia błędu w działaniu układów wejść/wyjść		
Przynależność błędu do określonej grupy błędów oraz waga błędu i związana z nią reakcja systemu na jego wystąpienie		
kategoria błędu	typ błędu	
opis błędu		
blok szczegółowych informacji o błędzie		
Czas i data wystąpienia błędu		

Wskaźnik długości bloku szczegółowych informacji o błędzie (wartość słowa bloku parametrów funkcji o adresie PARM + 1) definiuje ilość szczegółowych danych opisujących błąd. Może to być:

Tabela błędów działania sterownika:	wartość 00 dla 8 bajtów (blok krótki)
	wartość 01 dla 24 bajtów (blok długi)
Tabela błędów działania układów wejść/wyjść	wartość 02 dla 5 bajtów (blok krótki)
	03 = 21 bajtów (blok długi)

### Przykład 1

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennych %I0250 i %I0250 na 1 powoduje odczyt ostatnio zarejestrowanego komunikatu o błędzie z tabeli błędów działania sterownika i zapisanie go do bloku parametrów. Jeżeli zmienna %I0251 ma wartość 0 a zmienna i %I0250 ma wartość 1, odczytywany jest ostatnio zarejestrowany komunikat o błędzie z tabeli błędów działania układów wejść/wyjść, a następnie zapisywany do bloku parametrów. Blok parametrów umieszczony jest pod adresem %R0600.



## Przykład 2

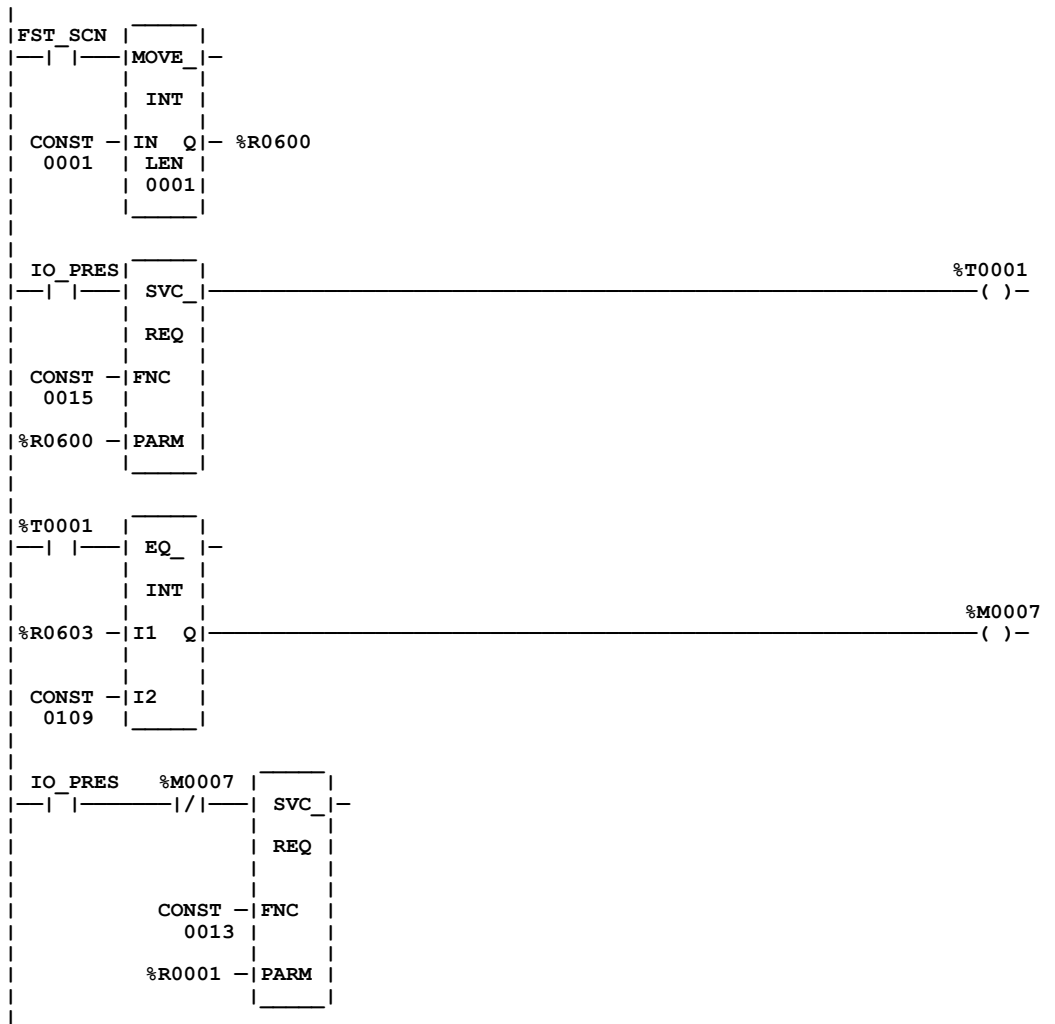
W następnym przykładzie, w przypadku wystąpienia jakiegokolwiek błędu działania modułu wejść i wyjść, za wyjątkiem wystąpienia błędów w modułach umieszczonych w gniazdach 1 i 9 kasety 0, sterownik jest zatrzymywany. Jeśli błąd wystąpi w którymś z tych modułów będzie działał. W pierwszym cyklu ustawiany jest parametr określający typ tabeli błędów działania. Jeżeli zmienna przypisana stykowi IO\_PRES ma wartość 1, informuje to o wprowadzeniu nowej pozycji do tabeli błędów działania układów Wejść/Wyjść. Sterownik ustawia na 1 wartość styku otwartego w pierwszym cyklu, realizowanym po wprowadzeniu pozycji do tabeli błędów. Jeżeli błędy są rejestrowane w dwóch kolejnych cyklach, wartość zmiennej przypisanej stykowi otwartemu jest ustawiana w dwóch dalszych cyklach.

W przykładzie tym wykorzystywany jest blok parametrów umieszczony pod adresem %R0600. Po wykonaniu funkcji SVCREQ, czwarte słowo bloku parametrów zawiera numer modułu i gniazda, w którym wystąpił błąd.

1		%R0600
wskaźnik długości bloku szczegółowych informacji o błędzie		%R0601
Adres wystąpienia błędu		%R0602
numer kasety	numer gniazda	%R0603
Nr szyny wejść/wyjść	adres szyny	%R0604
adres punktu		%R0605

### Blok szczegółowych informacji o błędzie

W programie tym, bloki EQ\_INT porównują adresy kasety/gniazda zapisane w tabeli z stałymi heksadecymalnymi. Wartość wewnętrznego przełącznika %M0007 jest ustawiana na 1, jeżeli kasetę/gniazdo dla których sygnalizowany jest błąd spełniają kryteria podane powyżej. Jeżeli wartość zmiennej %M0007 jest równa 1, styki otwarte są rozwierane, co zapobiega zatrzymaniu sterownika. Z drugiej strony, jeżeli wartość zmiennej %M0007 jest równa 0, informuje to o wystąpieniu błędu w innym module, styki otwarte są zwierane i następuje zatrzymanie sterownika.



## SVCREQ #16: Zegar odmierzający czas pracy sterownika

Funkcja SVCREQ z parametrem FNC o wartości 16 może dokonać odczytu bieżącego stanu zegara odmierzającego czas pracy sterownika. Zegar ten odmierza w sekundach czas, który upłynął od momentu ostatniego włączenia zasilania sterownika. Zakres wskazań tego zegara wynosi ok. 100 lat.

Blok parametrów funkcji stanowią jedynie parametry wyjściowe. Blok parametrów posiada długość trzech słów.

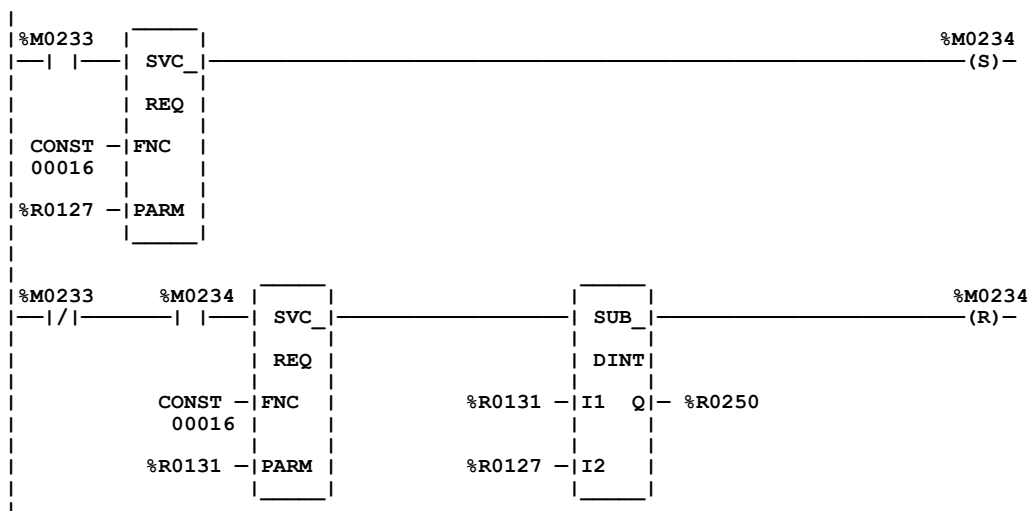
Liczba sekund, które upłynęły od momentu włączenia zasilania sterownika (w pierwszym słowie bloku danych)	adres
Liczba sekund, które upłynęły od momentu włączenia zasilania sterownika (w drugim słowie bloku danych)	adres + 1
Liczba setnych części odmierzanej aktualnie sekundy	adres +2

Pierwsze dwa słowa podają czas w sekundach. Ostatnie słowo zawiera liczbę setnych części odmierzanej aktualnie sekundy.

### Przykład

W zamieszczonym poniżej przykładzie, każda zmiana wartości wewnętrznego przełącznika %M0233 z 0 na 1 powoduje odczytanie i zapisanie do zmiennej %M0234 wskazań tego zegara. Zmiana na 0 powoduje powtórne odczytanie tej wartości. Różnica pomiędzy tymi wartościami jest następnie zapisywana do pamięci pod adresem %R0250.

W pierwszym odczycie dane są odczytywane z obszaru określonego adresem %R0127, a drugim z obszaru określonego adresem %R0131. W czasie obliczania ignorowana jest liczba setnych części sekundy oraz fakt, że typ DINT jest w rzeczywistości liczbą całkowitą ze znakiem. Uzyskiwany jest prawidłowy wynik obliczeń, o ile czas od momentu włączenia nie przekroczył 50 lat.





## SVCREQ #18: Kontrola występowania wymuszeń zmiany wartości zmiennych wejściowych i wyjściowych

Funkcja SVCREQ z parametrem FNC o wartości 18 służy do sprawdzenia, czy występują zmienne, których wartość została wymuszona i zablokowana (OVERRIDE).

### Uwaga

Funkcja ta dostępna jest *wyłącznie* jednostkach centralnych 331 i wyższych.

Długość bloku parametrów dla tej funkcji wynosi 1 słowo. Jest to jedynie parametr wejściowy.

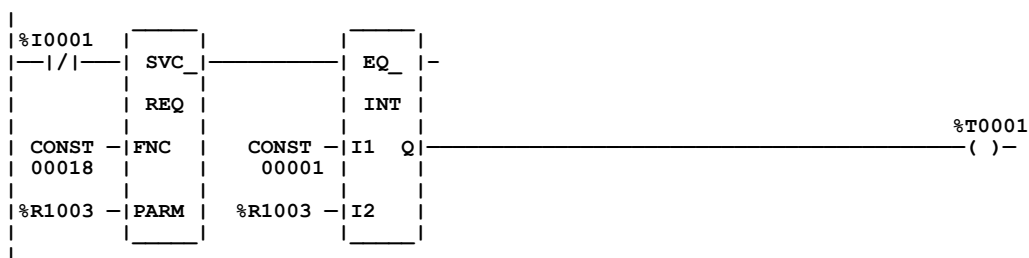
0 = brak zmiennych, których wartość została wymuszona i zablokowana. 1 = występowanie zmiennych, których wartość została wymuszona i zablokowana.	adres
--	-------

### Uwaga

Funkcja SVCREQ #18 podaje informacje wyłącznie dla zmiennych typu %I i %Q.

### Przykład

W zamieszczonym poniżej przykładzie odczytywany jest i zapisywany do zmiennej %R1003 status wymuszenia i zablokowania wartości zmiennych. Jeżeli występują zmienne, których wartość jest wymuszona i zablokowana, wartość zmiennej wyjściowej %T0001 ustawiana jest na 1.



## SVCREQ #23: Odczyt sumy kontrolnej programu sterującego i konfiguracji

Funkcja SVCREQ z parametrem FNC o wartości 23 służy do odczytu sumy kontrolnej programu sterującego i konfiguracji. Wyjście bloku funkcyjnego jest zawsze ustawiane na 1 po uaktywnieniu funkcji, a blok informacji wyjściowych rozpoczyna się pod adresem podanym jako parametr PARM bloku.

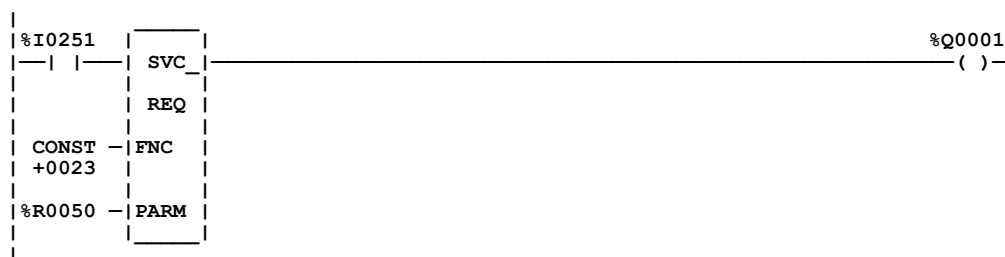
W czasie zapisywania programu (konfiguracji) w trybie **RUN**, sumy kontrolne mogą nie być uaktualnione do czasu zakończenia operacji. W celu określenia, czy sumy kontrolne są ważne przewidziano dwa wskaźniki, umieszczone na samym początku parametrów wyjściowych bloku.

Blok parametrów wyjściowych o długości 12 słów posiada następujący format:

Suma kontrolna programu sterującego (0 = błędna, 1 = poprawna)	adres
Suma kontrolna konfiguracji (0 = błędna, 1 = poprawna)	adres + 1
Liczba bloków programu (włącznie z blokiem głównym MAIN)	adres +2
Rozmiar programu sterującego w bajtach (2 słowa)	adres +3
Suma kontrolna programu sterującego	adres +5
32-bitowa suma kontrolna typu CRC programu sterującego	adres +6
Rozmiar danych konfiguracyjnych w bajtach	adres +8
Suma kontrolna konfiguracji	adres +9
32-bitowa suma kontrolna typu CRC dla konfiguracji (2 słowa)	adres +10

### Przykład

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej wejściowej %I0251 na 1 powoduje zapisanie do bloku parametrów sumy kontrolnej programu sterującego oraz ustawienie zmiennej przekaźnika %Q0001 na 1. Blok parametrów umieszczony jest pod adresem %R0050.





## SVCREQ #26/30: Porównanie rzeczywistej konfiguracji modułów wejść/ wyjść sterownika ze zdefiniowaną

Funkcja SVCREQ z parametrem FNC o wartości 26 lub 30 (obydwa parametry realizują to samo zadanie) służy do sprawdzenia, jakie moduły są obecne aktualnie w sterowniku i porównania rzeczywistej konfiguracji ze zdefiniowaną, generując alarmy związane z dodaniem i odjęciem modułu oraz niezgodnością konfiguracji. Funkcja zapisuje błędy w tablicy błędów sterownika i w tablicy błędów wejść/wyjść.

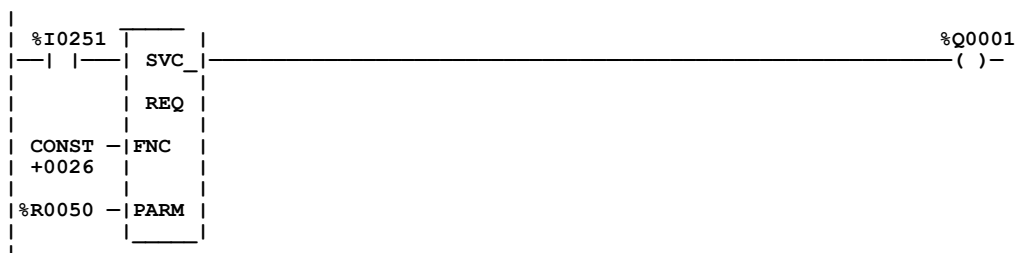
Blok SVCREQ #26/30 nie posiada parametrów i zawsze przesyła sygnał na swoje wyjście.

### Uwaga

Czas wykonania tej funkcji SVCREQ zależy od liczby zarejestrowanych błędów. Z tego powodu, czas wykonania funkcji SVCREQ ulega wydłużeniu w przypadku, gdy błędy zarejestrowane zostały w większej liczbie modułów.

### Przykład

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej %I0251 na 1 powoduje sprawdzenie faktycznie zainstalowanych modułów i porównanie ich z wprowadzoną konfiguracją. Po zakończeniu wykonywania funkcji SVCREQ, wartość zmiennej wyjściowej %Q0001 jest ustawiana na 1.



### Uwaga

Funkcja ta nie jest dostępna w sterownikach Micro.

## SVCREQ #29: Zegar czasu trwania przerwy w zasilaniu sterownika

Funkcja SVCREQ z parametrem FNC o wartości 29 umożliwia odczyt czasu trwania ostatniej przerwy w zasilaniu sterownika. Wyjście bloku funkcyjnego jest zawsze ustawiane na 1, a blok informacji wyjściowych rozpoczyna się pod adresem podanym jako parametr PARM bloku.

### Uwaga

Funkcja ta dostępna jest wyłącznie w jednostkach centralnych 331 i wyższych.

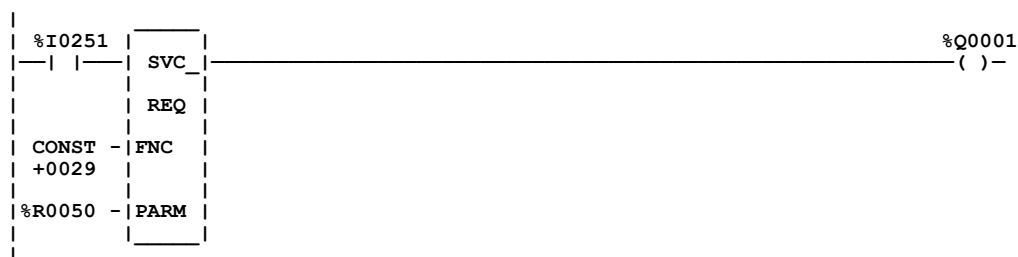
Blok parametrów funkcji stanowią jedynie parametry wyjściowe. Blok parametrów posiada długość trzech słów.

Czas trwania przerwy w zasilaniu w sekundach (w drugim słowie bloku danych)	adres
Czas trwania przerwy w zasilaniu w sekundach (wyższy rząd wielkości)	adres + 1
Pozostały czas w 100- $\mu$ s okresach	adres +2

Pierwsze dwa słowa podają czas w sekundach, który upłynął od momentu wyłączenia zasilania. Ostatnie słowo jest to pozostały czas, podawany w 100 mikrosekundowych okresach. Gdy sterownik nie może prawidłowo oszacować czasu, czas zostanie ustawiony na zero. Zdarza się tak przy włączaniu sterownika z wciśniętymi klawiszami CLR M/T na programatorze ręcznym. Może to również wystąpić po zadziałaniu zegara alarmowego.

### Przykład

W zamieszczonym poniżej przykładzie, każda zmiana wartości zmiennej wejściowej %I0251 na 1 powoduje zapisanie do bloku parametrów czasu, który upłynął od momentu wyłączenia zasilania oraz ustawienie zmiennej przekaźnika %Q0001 na 1. Blok parametrów umieszczony jest pod adresem %R0050.



## SVCREQ #45: Pominięcie obsługi wejść/wyjść w następnym cyklu

Blok funkcyjny SVC\_REQ 45 pozwala na pominięcie obsługi wejść i wyjść w następnym cyklu pracy sterownika. Jakakolwiek zmiana zmiennych wyjść w cyklu, w którym wykonywany jest blok SVC\_REQ 45 nie zostaje odzwierciedlona na wyjściach fizycznego modułu. Jakakolwiek zmiana danych o stanie wejść fizycznych modułu nie jest odzwierciedlana w odpowiednich zmiennych wejść, w cyklu następującym po cyklu, w którym wywołano blok funkcyjny SVC\_REQ 45.

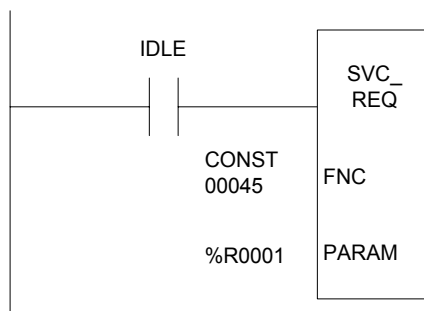
Funkcja ta nie posiada bloku parametrów.

### Uwaga

Blok funkcyjny DOIO nie działa z funkcją SVCREQ #45. Funkcja DOIO będzie działać w tym samym programie, w którym występuje instrukcja SVCREQ #45.

### Przykład

W zamieszczonym poniżej przykładzie, doprowadzenie sygnał do styku "Idle" powoduje pominięcie obsługi wejść/wyjść w następnym cyklu.



## SVCREQ #46: Szybki dostęp do statusu komunikacji przez szynę komunikacyjną

Funkcja ta umożliwia szybkie w porównaniu z normalną komunikacją przesyłanie kilku bitów do/z jednego lub więcej inteligentnych modułów przez magistralę komunikacyjną. Przyspieszenie komunikacji jest osiągnięte przez ograniczenie ilości danych i ilości odpowiedzi.

Funkcja SVCREQ #46 pozwala na wywołanie jednej z następujących funkcji, zapewniających szybki dostęp do szyny komunikacyjnej.

- Odczytanie słowa dodatkowych danych o statusie jednego z inteligentnych modułów.
- Zapisanie słowa dodatkowych danych o statusie jednego z inteligentnych modułów.
- Odczyt/ Zapis: Odczyt słowa dodatkowych danych o statusie jednego z inteligentnych modułów i zapisanie wartości z zakresu 0-15 do tego samego modułu, w czasie jednej operacji.

### Uwagi:

Obecnie, funkcja ta obsługiwana jest wyłącznie przez moduł DSM (Cyfrowy kontroler ruchu), wersja 312.

Bloki funkcyjne COMM\_REQ i DO\_IO nie powinny być wywoływane dla danego modułu (modułów) w cyklu, w którym wywoływana jest funkcja do zapisu danych, ponieważ może to prowadzić do utraty zapisywanych danych.

Dwa bloki funkcyjne zapisujące do modułu (zapis lub zapis/odczyt) nie powinny być wywoływane w tym samym cyklu, ponieważ może to prowadzić do utraty danych, które są zapisywane jako pierwsze.

Service Request jest również nazywane "SNAP".

Funkcja ta posiada blok parametrów o zmiennej długości, opisany poniżej: Pierwsze słowo bloku parametrów określa funkcję, która ma zostać wykonana i ma następujący format:

1 = Odczyt dodatkowych danych	adres (słowo 1)
2 = Zapis dodatkowych danych	
3 = Odczyt/zapis dodatkowych danych	

## Odczyt dodatkowych danych o statusie (Funkcja #1)

Funkcja ta odczytuje dodatkowe dane o statusie dla każdego z modułów określonych na liście w bloku parametrów, a następnie zapisuje te dane do bloku parametrów. Blok ten wymaga (N+4) słów pamięci zmiennych, gdzie N jest liczbą modułów, do których zapisywane będą dane.

Zamieszczona poniżej tabela jest pomocna przy interpretowaniu wartości wyjściowych:

**Tabela 12-6. Parametry bloku dla funkcji odczytu dodatkowych danych**

Pozycja	Pole	Opis
Adres	Funkcja	1 = odczyt dodatkowych danych o statusie
Adres +1	Kod błędu	Jeżeli stwierdzony zostanie brak modułu lub jeżeli jest on nieodpowiedni, albo nie pracuje, funkcja umieszcza kod błędu. Szczegółowe informacje podano w punkcie "Kody błędów", na stronie 12-75.
Adres +2	Kaseta & gniazdo błędu	Numer kasety i gniazda, określający miejsce wystąpienia błędu.
Adres +3	Pierwsza kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a GG jest numerem gniazda) i-ego modułu, z którego odczytywane są dane
Adres +4	Odczyt danych z pierwszego modułu	W tym miejscu umieszczane są dane odczytane z pierwszego modułu
Adres +5	Druga kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a GG jest numerem gniazda) 2-ego modułu, z którego odczytywane są dane
Adres +6	Odczyt danych z drugiego modułu	W tym miejscu umieszczane są dane odczytane z drugiego modułu
Adres + (I*2) + 1	i-ta kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a GG jest numerem gniazda) i-ego modułu, z którego odczytywane są dane
Adres + (I*2) + 2	Odczyt danych z i-ego modułu	W tym miejscu umieszczane są dane odczytane z i-ego modułu
Adres + (N*2) + 1	Ostatnia kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a GG jest numerem gniazda) ostatniego modułu, z którego odczytywane będą dane
Adres + (N*2) + 2	Odczyt danych z ostatniego modułu	W tym miejscu umieszczane są dane odczytane z ostatniego modułu
Adres + (N*2) + 3	Wskaźnik końca listy	Umieszczenie wartości zero informuje o zakończeniu listy modułów



## Zapis danych (Funkcja #2)

Funkcja ta zapisuje wartość z zakresu 0-15 z bloku parametrów do jednego lub większej liczby modułów, podanych na liście w bloku parametrów. Blok ten wymaga (N+4) słów pamięci zmiennych, gdzie N jest liczbą modułów, do których zapisywane będą dane.

Tabela 12-8. Parametry bloku dla funkcji zapisu danych

Pozycja	Pole	Opis
Adres	Funkcja	2 = Zapis danych
Adres +1	Kod błędu	Jeżeli stwierdzony zostanie brak modułu lub jeżeli jest on nieodpowiedni, albo nie pracuje, funkcja umieszcza kod błędu. Jeżeli funkcja zostanie wykonana bez błędów, ale dane nie zostaną poprawnie zapisane do któregośkolwiek z modułów, nie następuje zapisanie kodu błędu. Szczegółowe informacje podano w punkcie "Kody błędów", na stronie 12-75.
Adres +2	Kaseta & gniazdo błędu	Numer kasety i gniazda, określający miejsce wystąpienia błędu.
Adres +3	Pierwsza kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a G jest numerem gniazda) 1-ego modułu, do którego zapisywane będą dane
Adres +4	Zapis danych do pierwszego modułu	Zapis danych do pierwszego modułu
Adres +5	Druga kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a G jest numerem gniazda) 2-ego modułu, do którego zapisywane będą dane
Adres +6	Zapis danych do drugiego modułu	Zapis danych do drugiego modułu
Adres + (I*2) + 1	i-ta kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a G jest numerem gniazda) I-ego modułu, do którego zapisywane będą dane
Adres + (I*2) + 2	Zapis danych do i-ego modułu	Zapis danych do i-ego modułu
Adres + (N*2) + 1	Ostatnia kasetka & gniazdo	Numer kasety i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasety a G jest numerem gniazda) ostatniego modułu, do którego zapisywane będą dane
Adres + (N*2) + 2	Zapis danych do ostatniego modułu	Zapis danych do ostatniego modułu
Adres + (N*2) + 3	Wskaźnik końca listy	Umieszczenie wartości zero informuje o zakończeniu listy modułów

## Odczyt/ zapis danych (Funkcja #3)

Funkcja ta czyta słowo dodatkowych danych o statusie z modułu określonego w bloku parametrów, a następnie zapisuje wartość z zakresu 0 - 15 z bloku parametrów do tego modułu. Proces odczytu i zapisu powtarzany jest dla każdego modułu określonego na liście w bloku parametrów. Blok ten wymaga  $(N*4)+3$  słów pamięci zmiennych, gdzie N jest liczbą modułów, z którymi wymieniane będą dane.

Tabela 12-9. Parametry bloku dla funkcji odczyt/zapis danych

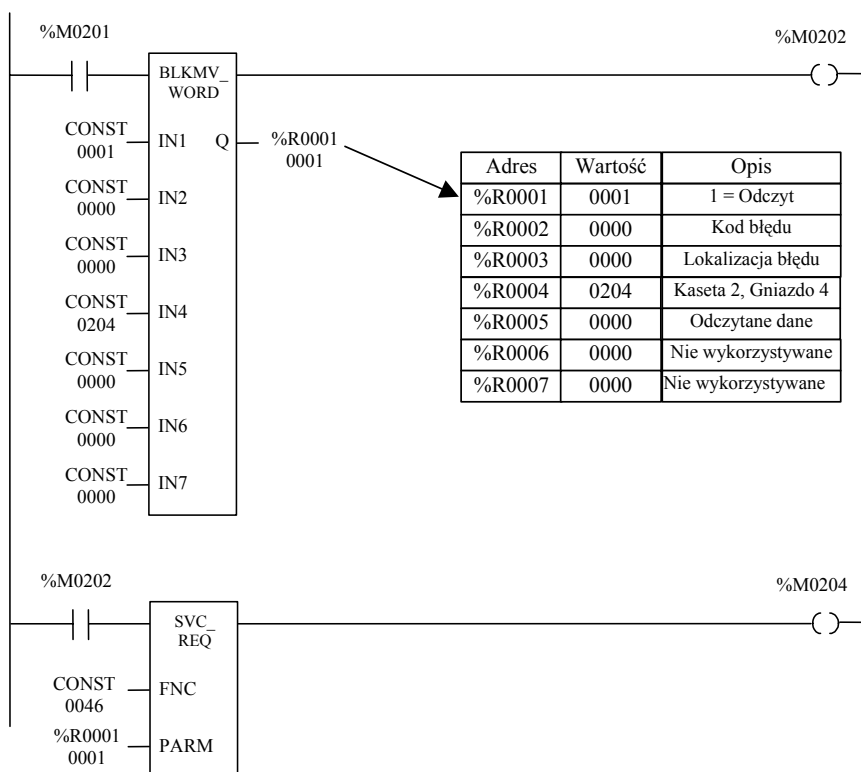
Pozycja	Pole	Opis
Adres	Funkcja	3 = odczyt/zapis
Adres +1	Kod błędu	Jeżeli stwierdzony zostanie brak modułu lub jeżeli jest on nieodpowiedni, albo nie pracuje, funkcja umieszcza kod błędu. Jeżeli funkcja zostanie wykonana bez błędów, ale dane nie zostaną poprawnie zapisane do któregośkolwiek z modułów, nie następuje zapisanie kodu błędu. Szczegółowe informacje podano w punkcie "Kody błędów", na stronie 12-75.
Adres +2	Kaseta & gniazdo błędu	Numer kasy i gniazda, określający miejsce wystąpienia błędu.
Adres +3	Pierwsza kaset & gniazdo	Numer kasy i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasy a GG jest numerem gniazda) 1-go modułu, z którym wymieniane będą dane
Adres +4	Odczyt danych z pierwszego modułu	W tym miejscu umieszczane są dane odczytane z pierwszego modułu
Adres +5	Zapis danych do pierwszego modułu	Zapis danych do pierwszego modułu
Adres +6	Druga kaset & gniazdo	Numer kasy i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasy a GG jest numerem gniazda) 2-go modułu, z którym wymieniane będą dane
Adres +7	Odczyt danych z drugiego modułu	W tym miejscu umieszczane są dane odczytane z drugiego modułu
Adres +8	Zapis danych do drugiego modułu	Zapis danych do drugiego modułu
Adres + $((I-1) * 3) + 3$	i-ta kaset & gniazdo	Numer kasy i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasy a GG jest numerem gniazda) I-ego modułu, z którym wymieniane będą dane
Adres + $((I-1) * 3) + 4$	Odczyt danych z i-ego modułu	W tym miejscu umieszczane są dane odczytane z i-ego modułu
Adres + $((I-1) * 3) + 5$	Zapis danych do i-ego modułu	Zapis danych do i-ego modułu
Adres + $((N-1) * 3) + 3$	Ostatnia kaset & gniazdo	Numer kasy i gniazda (w postaci liczby heksadecymalnej KKG, gdzie KK jest numerem kasy a GG jest numerem gniazda) ostatniego modułu, z którym wymieniane będą dane
Adres + $((N-1) * 3) + 4$	Odczyt danych z ostatniego modułu	W tym miejscu umieszczane są dane odczytane z ostatniego modułu
Adres + $((N-1) * 3) + 5$	Zapis danych do ostatniego modułu	Zapis danych do ostatniego modułu
Adres + $(N*3) + 3$	Wskaźnik końca listy	Umieszczenie wartości zero informuje o zakończeniu listy modułów

Tabela 12-10. Kody błędów

Wartość	Opis
1	Pomyślne zakończenie wykonywania bloku funkcyjnego.
-1	Brak modułu w określonym gnieździe.
-2	Niewłaściwy moduł -moduł w podanym gnieździe nie jest modułem wyspecjalizowanym lub nie obsługuje tej funkcjonalności.
-3	Moduł nie pracuje - moduł w podanym gnieździe nie komunikuje się w poprawny sposób z jednostką centralną.
-4	Błąd parzystości przy odczycie - przy odczycie danych z kasyty rozszerzającej lub zdalnej wystąpił błąd parzystości.
-5	W bloku parametrów podano niepoprawną funkcję.

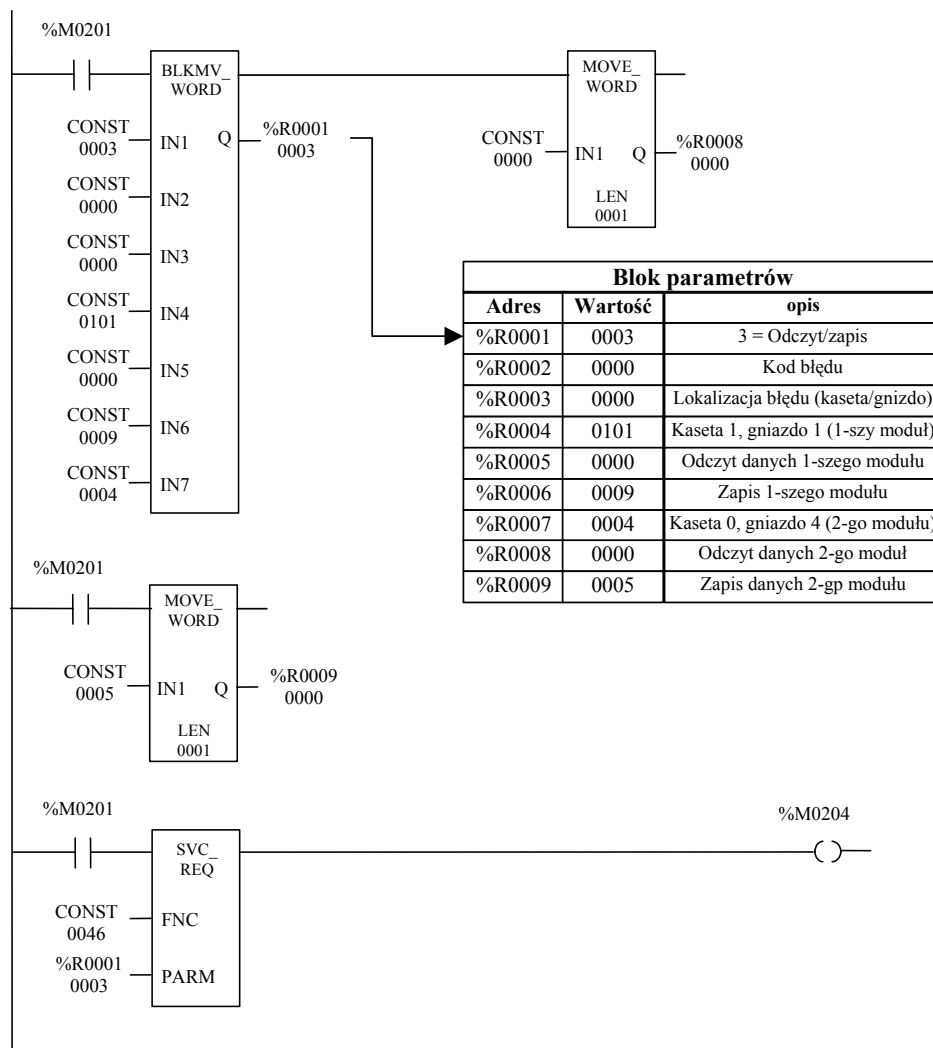
### Przykład 1

W zamieszczonym poniżej przykładzie odczytywane (%R0001) są dane dla jednego modułu, zainstalowanego w kasecie 2, gnieździe 4 (%R0004). Po pomyślnym wykonaniu bloku funkcyjnego, dane zapisywane są do %R0005. W przypadku wystąpienia błędu jego kod zostanie zapisany w rejestrze %R0002, a położenie (kasecja/gniazdo) modułu zostanie zapisany w rejestrze %R0003. Należy zwrócić uwagę, że jest to funkcja odczytu dla pojedynczego modułu, adres + 5 i adres + 6 nie jest wykorzystywany. Dlatego w odpowiednich obszarach pamięci %R0006 i %R0007 zapisane są zera z wejść funkcji IN6 i IN7 BLKMV. Jeśli moduł dodatkowy został by odczytany rejestry %R0006 i %R0007 zostały by wykorzystane właśnie dla tego modułu. Więcej informacji na temat funkcji odczytu zamieszczono w tabeli 12-5 we wcześniejszej części tego rozdziału.



## Przykład 2

W przykładzie tym bloki BLKMV i MOVE zapisują dane do bloku parametrów, którego początek stanowi rejestr %R0001 (wejście SVCREQ PARM). Uaktywnienie funkcji SVCREQ powoduje odczytanie dodatkowych informacji o statusie z modułu zainstalowanego w kasecie 0 gnieździe 4 oraz z modułu zainstalowanego w kasecie 1 gnieździe 1. Do pierwszego modułu zapisywana jest wartość 0005, a drugiego modułu – wartość 0009. (Należy zwrócić uwagę, że moduły nie muszą być wyszczególnione w bloku parametrów w kolejności gniazd). Dane odczytane z modułu zainstalowanego w kasecie 0, gnieździe 4 zostaną przekazane za pomocą zmiennej %R0008. Dane odczytane z modułu zainstalowanego w kasecie 1, gnieździe 1 zostaną przekazane za pomocą zmiennej %R0005.



## SVCREQ #48: Ponowne uruchomienie po błędzie krytycznym automatycznego zerowania

### Kompatybilność z funkcją SVCREQ 48

CPU – Funkcja Service Request jest obsługiwana przez jednostki centralne 331, 340, 341, 350, 36 i 37x w wersji 10 lub późniejszej.

Oprogramowanie – VersaPro w wersji 1.1 lub późniejszej Logicmaster nie obsługuje tej funkcji.

### Niebezpieczeństwo

**Nie należy wykorzystywać funkcji ponownego uruchomienia po wystąpieniu błędu krytycznego (ustawić parametr Ignore Fatal Faults na Disabled) w aplikacjach, w których automatyczne uruchomienie sterownika po wystąpieniu błędu może wywołać niepewne działanie sterowanych urządzeń. Na projektańce systemu spoczywa określenie czy dozwolne jest wykorzystywanie tej funkcji. Nie przestrzeganie tego zalecenia może spowodować okaleczenie lub śmierć obsługi i/lub zniszczenie urządzeń.**

### Opis

Ponowne uruchomienie po wystąpieniu błędu krytycznego pozwala na wznowienie normalnej pracy po wystąpieniu błędu krytycznego. Po wystąpieniu błędu krytycznego sterownik jest automatycznie zerowany, po czym wznawia normalną pracę. Błędy nie są kasowane, ale są traktowane jako błędy niekrytyczne. Sterownik przechodzi do trybu Run, nawet jeżeli po włączeniu zasilania nadal występują błędy krytyczne. W czasie konfigurowania parametrów sprzętowych jednostki centralnej należy ustawić parametr Fatal Faults (lub Fatal Fault Override).

Blok funkcyjny SVCREQ 48 pozwala na ustawienie maksymalnej liczby prób, które można podjąć w określonym czasie. W przypadku przekroczenia maksymalnej liczby prób w określonym czasie, sterownik jest przełączany do trybu STOP/FAULT. Po ustawieniu czasu na 0, jednostka centralna jest przełączona do trybu STOP/FAULT po wyczerpaniu maksymalnej liczby prób.

Jeżeli zasilanie zostanie wyłączone i włączone przez operatora, błędy krytyczne są ignorowane. Powoduje to zainicjalizowanie licznika prób oraz czasu. Sumaryczna liczba błędów krytycznych pozostaje bez zmian, natomiast sumaryczna liczba prób jest inkrementowana. Bit systemowy %S0021 jest ustawiany na 1 jeżeli próba zakończyła się pomyślnie i zachowuje tę wartość do momentu usunięcia wszystkich błędów krytycznych lub przejścia do trybu STOP/FAULT.

Tabela 12-11. Parametry bloku uruchomienie po błędzie krytycznym

Pozycja	Pole	Opis
Słowo 1	Status.	Patrz definicje statusu raportowania Program sterujący musi zapisywać do tego słowa wartość zero.
Słowo 2	Nieograniczone ponowienia	0 = Wyłączenie (Liczba prób jest podana w Słowie 3) 1 = Włączenie ( Słowa 3 & 4 są ignorowane.)
Słowo 3	Dopuszczalna liczba prób.	Zakres od 0 do 128 0 = Wyłączenie automatycznego kasowania 1 do 128 = Maksymalna liczba prób, które można podjąć w okresie czasu podanym w słowo 4.
Słowo 4	Czas podejmowania prób (minuty).	Zakres od 0 do 5940 minut (99 godzin) 0 = Brak ograniczeń czasowych na ilość ponowień ustawionych w słowie 3. Ponowne uruchomienie będzie możliwe po ostatnim dozwolonym ponowieniu. 1 do 5940 = Wyłączenie automatycznego kasowania jeżeli w podanym okresie czasu przekroczono maksymalną liczbę prób.

Tabela 12-12. Definicje statusu raportowania

Status	Opis	Uwagi	Przepl yw sygna lu
-5	Niepoprawna wartość czasu podejmowania prób	Zakres od 0 do 5940	Nie
-4	Ilość nieudanych ponowień	Zakres od 0 do 128	Nie
-3	Niepoprawna wartość nieograniczonej liczby prób	Polecenie musi mieć wartość 0 lub 1	Nie
-2	Konfiguracja nieaktywna	W czasie konfigurowania sprzętu należy zaznaczyć opcję Ignore Fatal Faults (Fatal Fault Override).	Nie
0	Brak działania	Polecenie nie wymaga zmian	Tak
1	Automatyczne kasowanie włączone	Poprawne polecenie pozwala na kasowanie po wystąpieniu błędu krytycznego.	Tak
2	Automatyczne kasowanie wyłączone	Poprawne polecenie wyłącza kasowanie po wystąpieniu błędu krytycznego. Polecenie ignorowania błędów krytycznych włączone.	

## SVCREQ 49 Automatyczne zerowanie statystyk

Funkcja Service Request 49 umożliwia dostęp do dwóch zmiennych, w których zapisana liczba wystąpień błędów krytycznych oraz ponowień. Zakres tych zmiennych wynosi od 0 do 65535. Nie są one nadpisywane po przekroczeniu maksymalnej wartości. (Service Request 48 jest wykorzystywana do konfigurowania maksymalnej liczby dozwolonych ponowień oraz czasu, w jakim mogą one występować).

**Tabela 12-13. Blok parametrów funkcji**

Słowo 1	Status.	Patrz definicje statusu raportowania Program sterujący musi zapisywać do tego słowa wartość zero.
Słowo 2	Polecenie	0 = raportuj całkowitą ilość wystąpień błędów krytycznych oraz ilość ponowień. 1 = rozpocznij zerowanie statystyk.
Słowo 3	Wartość zwrócona = całkowita liczba wystąpień błędów krytycznych i ponowień.	Program powinien inicjować zerowanie.
Słowo 4	Wartość zwrócona = całkowita liczba Auto Reset Retries	Program powinien inicjować zerowanie.

**Tabela 12-14. Definicje statusu raportowania**

Status	Opis	Uwagi	Przeływ sygnału
-2	Konfiguracja nieaktywna	W czasie konfigurowania sprzętu należy zaznaczyć opcję Ignore Fatal Faults (Fatal Fault Override).	Nie
-1	Niepoprawne polecenie	Polecenie musi mieć wartość 0 lub 1	Nie
1	Normalna praca	Poprawne polecenie	Tak

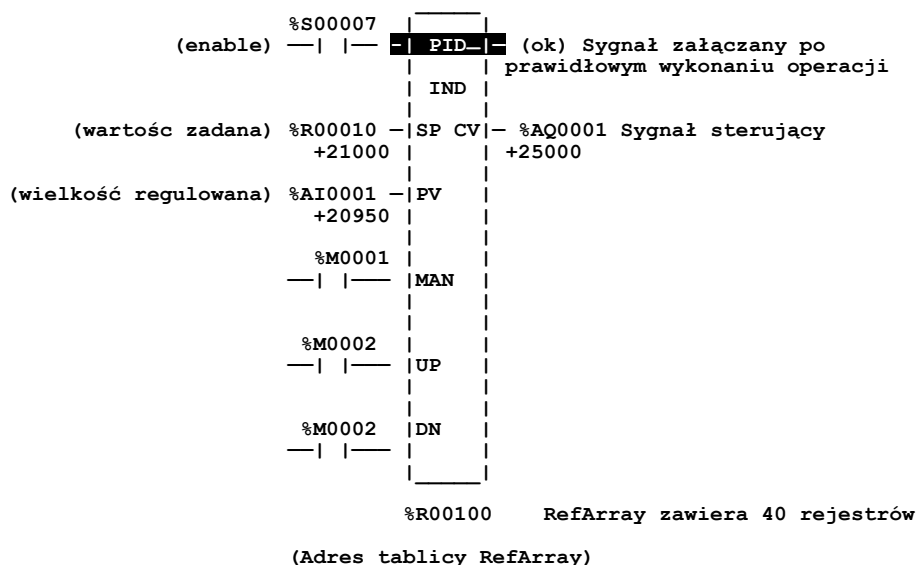
### Kompatybilność z funkcją SVCREQ 49

Funkcja Service Request jest obsługiwana przez jednostki centralne 331, 340, 341, 350, 36 i 37x w wersji 10 lub późniejszej.

# PID

blok proporcjonalno- całkowo- różniczkowy PID rozwiązuje algorytm sterowania zamkniętego układu regulacji. Blok ten na podstawie dwóch wartości: punktu ustalonego (wartości zadanej) i bieżącej wartości wielkości regulowanej oraz korzystając z zadanych parametrów bloku, oblicza taką wartość sygnału sterującego (nastawiającego), która spowoduje zmniejszenie uchybu, czyli odchylenia wielkości regulowanej od wartości zadanej (punktu pracy).

Blok funkcyjny PID wykorzystuje 40 rejestrów w pamięci sterownika, w których przechowuje zbiór parametrów regulatora. Wszystkie parametry są 16 bitowymi liczbami całkowitymi, co ma na celu zapewnienie kompatybilności z 16 bitowymi analogowymi wielkościami regulowanymi. Pozwala to na zastosowanie dla wartości regulowanych PV pamięci adresowanej przez %AI oraz pamięci adresowanej przez %AQ dla sygnału sterującego. Rysunek zamieszczony poniżej obrazuje typowe parametry funkcji PID:



Ponieważ wiele parametrów jest skalowanymi liczbami całkowitymi, 16 bitowymi, ich wartość musi być podawana w jednostkach PV lub w jednostkach bezwymiarowych PV, albo w jednostkach CV lub jednostkach bezwymiarowych CV. Przykładowo, parametr SP musi być przeskalowany do takiego samego zakresu jak parametr PV, ponieważ blok PID oblicza uchyb jako różnicę pomiędzy wartościami tych parametrów. Parametry wielkość regulowana PV i wielkość ustawiająca Cv mogą przyjmować wartości z zakresu -32000 lub 0 do 32000 z dopasowaniem do skalowania wielkości analogowych lub z zakresu, 0 do 10000 co pozwala na wyświetlanie wartości w formacie 0.00% do 100.00%. Wartości parametrów PV i CV nie muszą być tak samo skalowane, współczynniki skalowania są wtedy zawarte we współczynnikach wzmocnienia regulatora PID.

### Uwaga

Blok PID nie może być wykonywany częściej, niż co 10 milisekund. Jeżeli więc blok ten ma być wykonywany w każdym cyklu, a czas trwania cyklu jest krótszy od 10 milisekund, może to być powodem nie uzyskiwania żądanych wyników. W takim przypadku, blok PID zostanie wywołany dopiero po kilku cyklach, dokładnie po upływie 10 milisekund. Przykładowo, jeżeli cykl pracy jednostki centralnej sterownika wynosi 9 milisekund, blok funkcyjny jest wykonywany, co drugi cykl pracy jednostki centralnej, czyli co 18 ms.



## Parametry

Parametr	Opis
enable	Sygnal wejściowy. Gdy do bloku funkcyjnego dopływa sygnał, wykonywany jest algorytm PID (standardowy lub o niezależnych wyrazach).
SP	Wartość zadana wielkości regulowanej (punkt pracy regulatora). Po porównaniu parametrów wielkości regulowanej PV z wielkością zadaną SP blok PID dobiera tak sygnał sterujący CV, aby parametry PV i SP miały taką samą wartość (uchyb zerowy).
PV	Wielkość regulowana, której wartość odczytywana jest ze sterowanego procesu, deklarowana zwykle jako zmienna typu %AI.
MAN	Parametr MAN o wartości 1 powoduje przełączenie regulatora w ręczny tryb pracy ( <b>MANUAL</b> ). Wartość tego parametru równa 0 powoduje przełączenie regulatora w tryb sterowania automatycznego.
UP	Parametr mający znaczenie tylko w ręcznym trybie pracy*. Wartość tego parametru równa 1 powoduje zwiększenie wartości sygnału nastawiającego, wartość 0 nie wywołuje żadnego działania.
DN	Parametr mający znaczenie tylko w ręcznym trybie pracy*. Wartość tego parametru równa 1 powoduje zmniejszenie wartości sygnału nastawiającego, wartość 0 nie wywołuje żadnego działania.
Adres tablicy RefArray	Adres pierwszego z rejestrów, w których przechowywane są wewnętrzne parametry regulatora (parametry użytkownika i parametry wewnętrzne). Obszar ten zajmuje 40 rejestrów pamięci typu %R, które nie mogą być wykorzystywane w innym celu.
ok	Sygnal wyjściowy, wysyłany po poprawnym wykonaniu funkcji. Jest wyłączony, jeśli występuje błąd(y).
CV	Sygnal sterujący (ustawiający) procesu, często jest to zmienna typu %AQ.

\* Inkrementowany (parametr UP) lub dekrementowany (parametr DN) o jeden (1) w jednym wywołaniu funkcji PID.

## Dopuszczalne typy zmiennych

Parametr	sygnał	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	stała	brak
enable	•											
SP		•	•	•	•		•	•	•	•	•	
PV		•	•	•	•		•	•	•	•		
MAN	•											
UP	•											
DN	•											
address								•				
ok	•											•
CV		•	•	•	•		•	•	•	•		

- Zmienna danego typu może być wykorzystana jako dany parametr bloku funkcyjnego.

## Blok parametrów funkcji PID

Poza 2 parametrami wejściowymi i 3 ręcznie sterowanymi stykami, blok PID korzysta z 13 parametrów zapisanych w tablicy RefArray. Wartość tych parametrów należy ustawić przed wywołaniem tego bloku. Pozostałe parametry są wykorzystywane przez sterownik i nie mogą być zmieniane. Zmienna %Ref w zamieszczonej poniżej tabeli jest adresem tablicy RefArray, pokazanej na schemacie bloku funkcyjnego PID. Liczba podana po znaku plus określa pozycję w tabeli. Przykładowo, jeżeli adres początkowy tabeli RefArray jest równy %R100, zmienna %R113 zawiera wartość sygnału sterującego w ręcznym trybie sterowania.

**Tabela 12-15. Zestawienie parametrów bloku funkcyjnego PID**

Rejestr	Parametr	Jednostki program.	Zakres wartości
%Ref+0000	Loop Number (Numer układu regulacji)	Integer	0 do 255 (wyłącznie na potrzeby informowania użytkownika)
%Ref+0001	Algorithm (Rodzaj algorytmu)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0002	Sample Period (Okres próbkowania)	10 ms	0 (każdy cykl) do 65535 (10.9 min). Dla sterowników 90-30 należy wprowadzić wartość co najmniej 1 (proszę porównać z uwagą na stronie 4-171).
%Ref+0003	Dead Band (Górna granica strefy nieczułości)	Jednostki bezwymiarowe PV	0 do 32000 (wartości ujemne nie dozwolone)
%Ref+0004	Dead Band (Dolna granica strefy nieczułości)	Jednostki bezwymiarowe PV	-32000 do 0 (wartości dodatnie nie dozwolone)
%Ref+0005	Proportional Gain –Kp (Współczynnik wzmocnienia proporcjonalnego Kp)	0.01 CV%/PV%	0 do 327.67 %/%
%Ref+0006	Derivative Gain–Kd (Współczynnik wzmocnienia różniczkowego Kd)	0.01 sekundy	0 do 327.67 sek.
%Ref+0007	Integral Rate–Ki (Współczynnik wzmocnienia całkowego Ki)	powtórzenia/ 1000 sek.	0 do 32.767 powtórzeń/sek.
%Ref+0008	CV Bias/Output Offset (Przesunięcie punktu pracy)	Jednostki bezwymiarowe parametru CV	-32000 do 32000 (dodawana do wyjścia bloku całkowującego)
%Ref+0009	Upper Clamp (Górna granica wartości sygnału ustawiającego)	Jednostki bezwymiarowe parametru CV	-32000 do 32000 (>%Ref+10) wartości granicznych sygnału wyjściowego
%Ref+0010	Lower Clamp (Dolna granica wartości sygnału sterującego)	Jednostki bezwymiarowe parametru CV	-32000 do 32000 (<%Ref+09) wartości granicznych sygnału wyjściowego
%Ref+0011	Minimum Slew Time (Minimalny czas narastania sygnału ustawiającego)	Sekundy/pełne przemieszczenie	0 (brak) do 32000 sek. do przesunięcia o 32000 CV
%Ref+0012	Config Word- Parametr konfiguracyjny	5 młodszych bitów	Bity 0 do 2 informują o sposobie obliczania uchybu, polaryzacji sygnału ustawiającego CV i sposobie różniczkowania.
%Ref+0013	Manual Command (Sygnał sterujący w trybie ręcznym)	Jednostki bezwymiarowe parametru CV	Nadaje za parametrem CV w trybie automatycznym lub określa wartość CV w trybie ręcznym
%Ref+0014	Control Word (Słowo sterujące)	Zapisywany przez sterownik, o ile bit 1 jest ustawiony	Jeżeli wartość bitu 1 wynosi 0, wartość ta zapisywana jest przez sterownik. Proszę porównać z punktem "Szczegółowy opis parametrów bloku funkcyjnego PID" na stronie 12-85.
%Ref+0015	Internal SP (Punkt pracy regulatora SP)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0016	Internal CV (Sygnał ustawiający CV)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana

Tabela 12-15. Zestawienie parametrów bloku funkcyjnego PID (kontynuacja)

Rejestr	Parametr	Jednostki program.	Zakres wartości
%Ref+0017	Internal PV (Wartość wielkości regulowanej PV)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0018	Wyjście	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0019	Diff Term Storage (Dane robocze dotyczące bloku różniczkującego)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0020 i %Ref+0021	Int Term Storage (Dane robocze dotyczące bloku całkującego)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0022	Slew Term Storage (Dane robocze dotyczące prędkości narastania sygnału ustawiającego)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0023	Clock (Zegar wewnętrzny)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0024	(czas który upłynął od momentu ostatniego wykonania algorytmu)	~	
%Ref+0025			
%Ref+0026	Y Remainder Storage (Rejestr do przechowywania wartości Y - wewnętrzna zmienna sterownika)	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowana
%Ref+0027	Lower Range for SP, PV (Dolna granica zakresu wartości parametrów SP i PV)	Jednostki bezwymiarowe PV	-32000 do 32000 (>%Ref+28) do wyświetlania
%Ref+0028	Upper Range for SP, PV (Górna granica zakresu wartości parametrów SP i PV)	Jednostki bezwymiarowe PV	-32000 do 32000 (<%Ref+27) do wyświetlania
%Ref+0029 +•	Zarezerwowane do użytku wewnętrznego	Brak	Nie konfigurowana
%Ref+0034			
%Ref+0035 •	Zarezerwowane do użytku wewnętrznego	Brak	Nie konfigurowana
%Ref+0039			

W sterownikach 90-30 tablica RefArray musi być przechowywana w obszarze pamięci adresowanej przez %R. Należy zwrócić uwagę, że każdy blok funkcyjny PID musi korzystać z innego obszaru pamięci, nawet jeżeli wszystkich 13 parametrów jest takich samych. Jest to spowodowane przez fakt, że pozostałe rejestry tablicy wykorzystywane są do pamiętania danych wewnętrznych bloku funkcyjnego PID. Należy upewnić się, że tablica mieści się w całości w pamięci.

W celu skonfigurowania parametrów użytkownika należy zaznaczyć funkcję PID, a następnie wcisnąć **F10** w celu wyświetlenia obszaru ekranu zawierającego parametry użytkownika, a następnie wtedy wejść za pomocą klawiszy kursora przejść do odpowiednich pól i wprowadzić żądane wartości. Jako wartość domyślną można w większości przypadków wprowadzić 0, za wyjątkiem parametru Górna granica wartości sygnału ustawiającego, który musi być większy od parametru dolna granica wartości sygnału ustawiającego, co jest warunkiem wykonania bloku PID. Należy zwrócić uwagę, że blok PID **nie** przekazuje sygnału wyjściowego w przypadku wykrycia błędu w parametrach użytkownika, przez co, w czasie modyfikowania wartości, zalecane jest monitorowanie jego statusu za pomocą chwilowego przekaźnika (%T).

Po ustawieniu odpowiednich wartości bloku funkcyjnego PID powinny one zostać zdefiniowane jako stałe za pomocą funkcji, BLKMOV, dzięki czemu będzie można jest powtórnie załadować, jeżeli wymagane będzie wprowadzenie domyślnych wartości użytkownika dla tego bloku.

## Opis działania bloku funkcyjnego PID

Gdy do wejścia "enable" bloku funkcyjnego dopływa sygnał wejściowy, a jednocześnie do wejścia MAN sygnał nie dopływa (wartość parametru MAN wynosi 0), realizowany jest algorytm PID. Czas, który upłynął od ostatniego wykonania bloku PID porównywany jest z zaprogramowanym okresem próbkowania (impulsowania) ( $\%Ref + 2$ ). Jeśli czas ten osiągnął wartość większą lub równą okresowi próbkowania, wykonywany jest algorytm PID (rozwiązywane jest równanie zamkniętego układu regulacji), dla którego podstawą czasu jest okres, jaki upłynął od ostatniego wykonania bloku PID, a nie zaprogramowany okres próbkowania. W trybie automatycznym, wartość sygnału sterującego przypisywana jest do parametru Sygnał sterujący w trybie ręcznym o adresie  $\%Ref + 13$ .

Gdy blok funkcyjny PID pracuje w trybie ręcznym (do wejścia MAN dopływa sygnał), wartość sygnału ustawiającego jest utrzymywana na poziomie wartości zapisanej w rejestrze o adresie " $\%Ref + 13$ ", która z kolei może być zwiększana lub zmniejszana za pomocą parametrów wejściowych UP i DN bloku. W celu szybkiej zmiany wyjściowego sygnału ustawiającego, możliwe jest również dodanie lub odjęcie dowolnej wartości w jednostkach sygnału ustawiającego bezpośrednio do/ od parametru Sygnał sterujący w trybie ręcznym ( $\%Ref + 13$ ).

Wartość sygnału ustawiającego CV ograniczana jest w bloku PID za pomocą parametrów: Górna granica wartości sygnału ustawiającego i Dolna granica wartości sygnału ustawiającego. W przypadku wprowadzenia dodatniej wartości parametru Minimalny czas narastania sygnału ustawiającego, wartość ta jest wykorzystywana do ograniczenia prędkości zmian wartości sygnału ustawiającego CV. W przypadku przekroczenia amplitudy lub prędkości zmian sygnału ustawiającego CV, układ całkujący zostanie tak ustawiony, aby ustawiona została wartość graniczna tego parametru. W wyniku działania mechanizmu zapobiegającego przekroczeniu wartości dopuszczalnych (opisany na stronie 12-**Błąd! Nie zdefiniowano zakładki.**), jeżeli wartość sygnału ustawiającego CV jest większa (lub mniejsza) od wartości granicznych przez dłuższy okres czasu, zmiana znaku błędu powoduje szybkie odejście od wartości granicznej.

Mechanizm taki, przy nadążaniu parametru Sygnał sterujący w trybie ręcznym za wartością sygnału ustawiającego w trybie automatycznym oraz przy wprowadzaniu wartości sygnału ustawiającego CV w trybie ręcznym zapewnia możliwość płynnego przejścia pomiędzy trybami sterowania automatycznego a ręcznego. Parametry Górna i Dolna granica wartości sygnału ustawiającego oraz Minimalny czas narastania sygnału ustawiającego są nadal wykorzystywane w trybie ręcznym, a parametry wewnętrzne zapamiętane w bloku całkującym są uaktualniane. Oznacza to, że przejście do trybu ręcznego sterowania nie spowoduje przekroczenia maksymalnej prędkości narastania sygnału ustawiającego, określonej przy użyciu Minimalny czas narastania sygnału ustawiającego oraz nie zostaną przekroczone wartości graniczne, określone parametrami Górna i Dolna granica wartości sygnału ustawiającego.

### Uwaga

Ten sam blok funkcyjny PID nie może być wywoływany więcej niż jeden raz w ciągu jednego cyklu.

W poniższej tabeli zawarto dokładny opis parametrów przedstawionych w tabeli 12-3. Liczba w nawiasach po każdej nazwie parametru oznacza przesunięcie w tablicy RefArray.

Tabela 12-16. Szczegółowy opis parametrów bloku funkcyjnego PID

Parametr	Opis
Loop Number - Numer układu regulacji (00)	Liczba całkowita bez znaku, która pozwala na identyfikację układu regulacji poprzez jego numer. Numer ten wyświetlany jest pod adresem bloku parametrów podczas śledzenia wykonania programu sterującego przez sterownik za pomocą oprogramowania Logicmaster 90-30/Micro. Jest to parametr opcjonalny.
Algorithm - Rodzaj algorytmu PID (01)	Liczba całkowita bez znaku, której wartość ustawiana jest przez sterownik, w celu określenia, który z dwóch dostępnych algorytmów PID jest realizowany. Wartość parametru równa 1 oznacza standardowy algorytm ISA, natomiast wartość 2 oznacza algorytm o niezależnych wyrazach.
Sample Period - Okres Próbkowania (impulsowania) (02)	<p>Odstęp czasowy (mierzony w setnych częściach sekundy) pomiędzy dwoma kolejnymi wykonaniami bloków funkcyjnych PID. Przykładowo, jeżeli okres ten ma wynosi 100 milisekund, należy wprowadzić wartość 10. Wprowadzenie wartości 0 powoduje, że blok funkcyjny jest realizowany przy każdym doprowadzeniu sygnału sterującego.</p> <p>Algorytm PID realizowany jest wyłącznie pod warunkiem, że czas wskazywany przez zegar sterownika jest równy lub większy od czasu poprzedniej realizacji tego algorytmu + Okres próbkowania. Należy pamiętać, że sterowniki 90-30 nie pozwalają na wprowadzenie okresu próbkowania o wartości mniejszej od 10 milisekund (porównać z Uwagą na stronie 12-80}, w przypadku krótszych czasów realizacji cyklu, bloki funkcyjne PID będą ignorowane. Funkcja zapamiętuje czas, który upłynął od ostatniej realizacji algorytmu PID, z dokładnością do 100 mikrosekund. Jeżeli wartość tego parametru wynosi 0, algorytm PID jest realizowany za każdym razem, gdy do bloku funkcyjnego dopływa sygnał wyjściowy.</p>
Dead Band (+/-) - Górna i dolna granica strefy nieczułości (03/04)	Wartości całkowite ze znakiem, określające górną i dolną granice strefy nieczułości w jednostkach bezwymiarowych PV. Jeżeli nie jest wymagane określenie granicy strefy nieczułości wartości te muszą być ustawione na 0. Jeżeli uchyb PID ( $SP - PV$ ) lub ( $PV - SP$ ) zawiera się powyżej wartości (-) i poniżej wartość (+), algorytm PID jest realizowany z uchybem o wartości 0. Jeżeli uchyb nie ma wartości zerowej, wartość (+) musi być większa od 0 i wartość (-) musi być mniejsza od 0 gdyż blok nie będzie funkcjonował. Należy pozostawić wartość 0 tych parametrów do momentu ustawienia i dostrojenia regulatora PID. Później można wprowadzić inne wartości tych parametrów w celu uniknięcia małych zmian wyjściowego sygnału ustawiającego CV, wynikłych z małych wartości błędów, mogących być powodem ewentualnego przyspieszonego zużycia mechanicznego członu wykonawczego.
Proportional Gain-Kp - Współczynnik wzmocnienia proporcjonalnego o Kp (05)	Wartość całkowita, oznaczana symbolem Kc w wersji ISA, określająca zmianę sygnału ustawiającego CV odpowiadającą zmianie uchybu o 100 jednostek bezwymiarowych PV. Parametr ten wyświetlany jest jako 0.00 %/%, z domyślnie przyjmowanymi dwoma miejscami dziesiętnymi. Przykładowo parametr Kp równy 450 będzie wyświetlany jako 4.50, a powodowane przez niego zmiany sygnału nastawiającego będzie wynosić $Kp * Uchyb / 100$ lub $450 * Uchyb / 100$ . Ogólnie ujmując, pierwsza nastawa jest zmieniana przy strojeniu regulatora.
Derivative Gain-Kd- Czas różniczkowania Kd (06)	Wartość całkowita określająca czas różniczkowania w setnych częściach sekundy. Wprowadzana w jednostkach 10 milisekund, a wyświetlana w formacie 0.00 sekund, z 2 miejscami na część dziesiętną. Przykładowo, jeżeli parametr Kd ma wartość 120, będzie wyświetlany jako 1.20 sek. a powodowana przez niego zmiany sygnału wyjściowego bloku PID będzie wynosić $Kd * \text{przyrost uchybu} / \text{przyrost czasu}$ , czyli $120 * 4 / 3$ , jeżeli uchyb zmieni się o 4 jednostki bezwymiarowe PV w ciągu 30 milisekund. Kd może zostać wykorzystany do przyspieszenia wolnej odpowiedzi regulatora, ale jest bardzo wrażliwy na zakłócanie wielkości regulowanej PV.
Integral Rate Gain-Ki- Współczynnik wzmocnienia członu całkującego - Ki(07)	Wartość całkowita ze znakiem określająca częstotliwość całkowania, jeżeli Uchyb ma stałą wartość równą 1 jednostce bezwymiarowej PV. Przykładowo, jeżeli parametr Ki ma wartość 1400, wyświetlany będzie jako 1.400 powtórzeń/sek., a powodowana przez niego zmiany sygnału wyjściowego bloku PID wynosi $Ki * Uchyb * dt$ , czyli $1.400 * 20 * 50 / 1000$ , dla Uchybu o wartości 20 jednostek bezwymiarowych wartości regulowanej PV i czasu trwania cyklu sterownika 50 milisekund (Okres próbkowania równy 0). Parametr Ki jest zwykle drugą nastawą ustawianą przy strojeniu regulatora, po parametrze Kp.
CV Bias/Output Offset- Przesunięcie punktu pracy (08)	Wartość całkowita w jednostkach bezwymiarowych CV, dodawana do sygnału wyjściowego bloku PID, przed zastosowaniem wartości granicznych prędkości zmian i amplitudy. Parametr ten można wykorzystać do ustawienia niezerowej wartości sygnału ustawiającego CV, jeżeli stosowany jest wyłącznie współczynnik wzmocnienia proporcjonalnego lub do sterowania z oddziaływaniem ("feed forward control").

Tabela 12-16. Zestawienie parametrów bloku funkcyjnego PID (kontynuacja)

Parametr	Opis
CV Upper and Lower Clamps- Górna i dolna granica wartości sygnału ustawiającego (09/10)	Wartości całkowite ze znakiem, określające największą i najmniejszą wartość sygnału ustawiającego. Jeżeli wartości tych parametrów nie zostaną wprowadzone lub jeżeli wartość granicy górnej jest mniejsza od granicy dolnej, blok funkcyjny PID nie zostanie wykonany. Wartości tych granic są zwykle wyznaczane w oparciu o ograniczenia fizyczne sygnału ustawiającego CV. Są one również wykorzystywane do graficznej reprezentacji sygnału ustawiającego CV, przy użyciu oprogramowania LM90. Blok funkcyjny PID monitoruje wartość wewnętrzną bloku całkowitego, zabezpieczając tym samym przed przekroczeniem wartości dopuszczalnych.
Minimum Slew Time - Minimalny czas narastania sygnału ustawiającego (11)	Wartość całkowita dodatnia, definiująca minimalną liczbę sekund, wymaganą do przejścia sygnału ustawiającego od wartości 0 do 100% lub 32000 jednostek bezwymiarowych CV. Jest to odwrotność dopuszczalnej prędkości zmian sygnału ustawiającego CV. Jeżeli parametr ten ma wartość dodatnią, wielkość zmiany sygnału CV w jednostkach bezwymiarowych CV nie może przekroczyć 32000 razy przyrost czasu (w sekundach) podzielony przez minimalny czas narastania sygnału ustawiającego. Przykładowo, jeżeli okres próbkowania wynosi 2.5 sekundy, a minimalny czas narastania sygnału ustawiającego jest równy 500 sekund, sygnał ustawiający Cv nie może się zmienić o więcej niż $32000 \cdot 2.5 / 500$ , czyli o 160 jednostek bezwymiarowych sygnału CV w ciągu jednego obliczania bloku PID. Podobnie jak w przypadku wartości granicznych sygnału ustawiającego, dostępny jest mechanizm zapobiegający przekroczeniu wartości dopuszczalnych, zmieniający wartość bloku całkowitego w przypadku przekroczenia wartości granicznych sygnału ustawiającego CV. Jeżeli Minimalny czas narastania sygnału ustawiającego jest równy 0, nie ma żadnego ograniczenia, co do prędkości narastania. Wartość 0 należy wprowadzać w czasie dostrajania bloku PID.
Config Word- Parametr konfiguracyjny	<p>5 młodszych bitów tego słowa wykorzystywanych jest do modyfikowania trzech standardowych parametrów regulatora PID. Bit zerowy modyfikuje sposób obliczania uchybu ze standardowego (SP - PV) na (PV - SP), a więc odwraca znak tego uchybu. Jest to wykorzystywane w układach sterowania, gdzie wzrost wielkości regulowanej powinien powodować spadek sygnału ustawiającego CV. Po ustawieniu na 1 wartości pierwszego bitu, zmieniana jest polaryzacja sygnału ustawiającego CV, parametr ten będzie więc miał wartość ujemną, a nie dodatnią. Ustawienie na 1 wartości trzeciego bitu zmienia działanie mechanizmu zapobiegającego przekraczaniu wartości granicznych.</p> <p>Poniżej przedstawiono znaczenie młodszych 5 bitów parametru konfiguracyjnego:</p> <p><b>Bit 0</b> = Sposób obliczania uchybu. Gdy wartość logiczna tego bitu wynosi 0, uchyb obliczany jest według wzoru SP-PV. Gdy wartość logiczna tego bitu wynosi 1, uchyb obliczany jest według zależności PV-SP.</p> <p><b>Bit 1</b> = Polaryzacja sygnału ustawiającego CV. Gdy wartość tego bitu wynosi 0, parametr wyjściowy CV stanowi bezpośredni wynik obliczeń algorytmu PID. Obliczenia PID. Gdy wartość tego bitu wynosi 1, sygnał Cv posiada polaryzację przeciwną do obliczonej według algorytmu PID.</p> <p><b>Bit 2</b> = Sposób różniczkowania wielkości regulowanej PV. Jeżeli wartość tego bitu wynosi 0, różniczkowany jest uchyb. Jeżeli wartość tego bitu wynosi 1, różniczkowany jest sygnał PV.</p> <p><b>Bit 3</b> = Działania podejmowane po przekroczeniu strefy nieczułości. Po ustawieniu tego bitu na wartość zero, nie są podejmowane żadne działania. Jeżeli uchyb mieści się w granicach strefy nieczułości, przyjmowana jest zerowa wartość uchybu. W przeciwnym wypadku, wielkość uchybu nie jest modyfikowana przez ograniczenie strefą nieczułości. Po ustawieniu tego bitu na wartość 1, podejmowane są pewne działania związane ze strefą nieczułości. Jeżeli uchyb mieści się w granicach strefy nieczułości, przyjmowana jest zerowa wartość uchybu. Jeżeli jednak uchyb wykracza poza granice strefy nieczułości, wartość uchybu jest zmniejszana o wartości graniczne strefy nieczułości (uchyb = uchyb - wartość graniczna strefy nieczułości).</p> <p><b>Bit 4</b> = Mechanizm zapobiegający przekroczeniu wartości granicznych. Po ustawieniu wartości tego bitu na zero, mechanizm zapobiegający przekroczeniu wartości korzysta z algorytmu zerowania wstecz. Jeżeli sygnał wyjściowy mieści się w dopuszczalnych granicach, wartość ta zastępuje zakumulowaną wartość Y (zdefiniowaną na stronie 12-<b>Błąd! Nie zdefiniowano zakładki.</b>) bez względu na wartość wymaganą do wygenerowania sygnału wyjściowego mieszczącego się w dopuszczalnym zakresie wartości. Jeżeli wartość tego bitu jest równa jeden, wartość ta zastępuje zakumulowany wyraz Y przez wartość wyrazu Y zapamiętaną w momencie rozpoczęcia obliczeń. Dzięki takiemu rozwiązaniu, wstępna wartość ograniczająca Y jest pamiętana tak długo, jak długo sygnał wyjściowy mieści się w dopuszczalnych granicach.</p> <p><b>UWAGA:</b> Mechanizm zapobiegający przekroczeniu wartości dopuszczalnych jest dostępny wyłącznie w jednostkach centralnych 90-30 wersja 6.50 i późniejsze.</p> <p>Należy pamiętać, że bity są umieszczane jako kolejne potęgi liczby 2. Przykładowo, po ustawieniu wartości Parametru konfiguracyjnego na 0 w domyślnej konfiguracji regulatora PID, należy dodać 1 w celu zmiany sposobu obliczania uchybu z SP-SV na PV-SP lub dodać 2 w celu zmiany polaryzacji sygnału ustawiającego CV z CV=sygnal wyjściowy regulatora PID na CV=-sygnal wyjściowy regulatora PID lub dodać 4 w celu zmiany sposobu różniczkowania z różniczkowania uchybu na różniczkowanie sygnału PV.</p>
Manual Command- Sygnał sterujący w trybie ręcznym (13)	Wartość całkowita ze znakiem, równa sygnałowi ustawiającemu CV, jeżeli regulator PID pracuje w trybie automatycznym. Po przełączeniu regulatora w tryb ręcznego sterowania ( <b>Manual</b> ), wartość ta wykorzystywana jest do wyznaczenia wartości CV oraz wewnętrznej wartości bloku całkowitego w obrębie przedziału wyznaczonego przez Górną i Dolną granicy wartości sygnału sterującego oraz Minimalny czas narastania sygnału sterującego.

Tabela 12-16. Zestawienie parametrów bloku funkcyjnego PID (kontynuacja)

Parametr	Opis																								
Control Word- Słowo sterujące	<p>Jest to parametr wewnętrzny, którego wartość jest zwykle równa 0.</p> <p>Jeżeli wartość logiczna bitu zerowego jest równa 1, blok funkcyjny PID sterowany jest przez to Słowo konfiguracyjne oraz przez inne wewnętrzne parametry SP, PV i CV (proszę porównać z opisem poniżej). Dzięki temu urządzenia zewnętrzne, takie jak na przykład komputer, mogą przejąć sterownie od programu sterującego. Ostrzeżenie: Jeżeli nie chcesz żeby zaistniała taka sytuacja upewnij się, że wartość Słowa sterującego jest ustawiona na 0. Jeżeli młodszy bit posiada wartość 0, mogą zostać odczytane kolejne 4 bity w celu śledzenia stanu styków wejściowych PID tak długo, jak długo do styku PID Enable doprowadzany jest sygnał zasilający. Jest to słowo bitowe o strukturze dyskretnej (każdy z bitów posiada odrębne znaczenie), o następującym formacie:</p> <table border="0"> <tr> <td><b>Bit:</b></td> <td><b>Wartość słowa:</b></td> <td><b>Funkcja:</b></td> <td><b>Status lub działania zewnętrznego podejmowane po zmianie wartości bitu zerowego na 1:</b></td> </tr> <tr> <td>0</td> <td>1</td> <td>Wymuszenie</td> <td>Jeżeli jego wartość wynosi 0, następuje monitorowanie styków bloku, opisanych poniżej. Jeżeli 1, są one ustawiane przez urządzenia zewnętrzne.</td> </tr> <tr> <td>1</td> <td>2</td> <td>Manual/</td> <td>Jeżeli 1, blok jest w trybie <b>Manual</b>, w przypadku innych numerów, w trybie Auto.</td> </tr> <tr> <td>2</td> <td>4</td> <td>Enable</td> <td>Standardowo wartość 1, w przeciwnym wypadku blok nie jest nigdy wywoływany.</td> </tr> <tr> <td>3</td> <td>8</td> <td>UP/Raise</td> <td>Jeżeli 1 i blok pracuje w trybie ręcznego sterowania (bit 1=1), każde wykonanie bloku PID powoduje inkrementowanie sygnału CV.</td> </tr> <tr> <td>4</td> <td>16</td> <td>DN/Lower</td> <td>Jeżeli 1 i blok pracuje w trybie ręcznym, każde wykonanie bloku PID powoduje inkrementowanie sygnału CV.</td> </tr> </table>	<b>Bit:</b>	<b>Wartość słowa:</b>	<b>Funkcja:</b>	<b>Status lub działania zewnętrznego podejmowane po zmianie wartości bitu zerowego na 1:</b>	0	1	Wymuszenie	Jeżeli jego wartość wynosi 0, następuje monitorowanie styków bloku, opisanych poniżej. Jeżeli 1, są one ustawiane przez urządzenia zewnętrzne.	1	2	Manual/	Jeżeli 1, blok jest w trybie <b>Manual</b> , w przypadku innych numerów, w trybie Auto.	2	4	Enable	Standardowo wartość 1, w przeciwnym wypadku blok nie jest nigdy wywoływany.	3	8	UP/Raise	Jeżeli 1 i blok pracuje w trybie ręcznego sterowania (bit 1=1), każde wykonanie bloku PID powoduje inkrementowanie sygnału CV.	4	16	DN/Lower	Jeżeli 1 i blok pracuje w trybie ręcznym, każde wykonanie bloku PID powoduje inkrementowanie sygnału CV.
<b>Bit:</b>	<b>Wartość słowa:</b>	<b>Funkcja:</b>	<b>Status lub działania zewnętrznego podejmowane po zmianie wartości bitu zerowego na 1:</b>																						
0	1	Wymuszenie	Jeżeli jego wartość wynosi 0, następuje monitorowanie styków bloku, opisanych poniżej. Jeżeli 1, są one ustawiane przez urządzenia zewnętrzne.																						
1	2	Manual/	Jeżeli 1, blok jest w trybie <b>Manual</b> , w przypadku innych numerów, w trybie Auto.																						
2	4	Enable	Standardowo wartość 1, w przeciwnym wypadku blok nie jest nigdy wywoływany.																						
3	8	UP/Raise	Jeżeli 1 i blok pracuje w trybie ręcznego sterowania (bit 1=1), każde wykonanie bloku PID powoduje inkrementowanie sygnału CV.																						
4	16	DN/Lower	Jeżeli 1 i blok pracuje w trybie ręcznym, każde wykonanie bloku PID powoduje inkrementowanie sygnału CV.																						
SP (15)	Punkt pracy regulatora SP. (Nie konfigurowalny - wartość ustawiana i pamiętana przez sterownik). Jeżeli bit 0 ma wartość 1, wartość tego parametru musi być ustawiona z zewnątrz.																								
CV (16)	Sygnał ustawiający CV. (Nie konfigurowalny - ustawiany i pamiętany przez sterownik).																								
PV (17)	Wartość wielkości regulowanej PV. (Nie konfigurowalny - wartość ustawiana i pamiętana przez sterownik). Jeżeli bit 0 ma wartość 1, wartość tego parametru musi być ustawiona z zewnątrz.																								
Output - Polaryzacja sygnału ustawiającego (18)	(Nie konfigurowalny - ustawiany i pamiętany przez sterownik). Wartość całkowita ze znakiem reprezentująca bieżący stan parametru wyjściowego bloku funkcyjnego PID, przed zastosowanie opcjonalnej konwersji sygnału. Jeżeli nie jest zaprogramowana inwersja sygnału ustawiającego i bit polaryzacji sygnału (pierwszy bit parametru konfiguracyjnego) ma wartość 0, wartość ta jest równa wartości sygnału ustawiającego CV. Jeżeli inwersja sygnału ustawiającego jest zaprogramowana i bit polaryzacji sygnału (pierwszy bit parametru konfiguracyjnego) ma wartość 1, wartość ta jest równa wartości sygnału ustawiającego CV, z przeciwnym znakiem.																								
Diff Term Storage - Dane robocze dotyczące bloku różniczkującego (19)	Rejestr wykorzystywany przez blok funkcyjny PID do pamiętania wartości pośrednich. <i>W rejestrze tym nie wolno zapisywać żadnych danych.</i>																								
Int Term Storage - Dane robocze dotyczące bloku całkującego (20/21)	Rejestr wykorzystywany przez blok funkcyjny PID do pamiętania wartości pośrednich. <i>W rejestrze tym nie wolno zapisywać żadnych danych.</i>																								
Slew Term Storage - Dane odnośnie prędkości narastania sygnału ustawiającego (22)	Rejestr wykorzystywany przez blok funkcyjny PID do pamiętania wartości pośrednich. <i>W rejestrze tym nie wolno zapisywać żadnych danych.</i>																								
Clock - Zegar wewnętrzny (23-25)	Czas, który upłynął od momentu ostatniego wykonania algorytmu PID. <i>W rejestrach tych nie wolno zapisywać żadnych danych.</i>																								
Rejestr do przechowywania wartości Y (wewnętrzna zmienna sterownika) (26)	Parametr ten zawiera resztę z skalowania działki całkowania dla stabilnego, zerowego uchybu.																								
Lower and Upper Range- Górna i dolna granica zakresu wartości parametrów SP i PV (27/28)	Opcjonalne wartości całkowite w jednostkach bezwymiarowych sygnału ustawiającego PV, wykorzystywane do graficznej reprezentacji wartości SP i PV w oprogramowaniu Logicmaster.																								
Zarezerwowane (29-34 i 35-39)	Rejestry 29-34 są zarezerwowane do użytku wewnętrznego, a rejestry 35-39 są zarezerwowane do użytku zewnętrznego. Są one zarezerwowane dla GE Fanuc i nie mogą być wykorzystywane w innym celu.																								

## Parametry wewnętrzne w tablicy RefArray

Zgodnie z opisem przedstawionym w Tabeli 12-3 na poprzedniej stronie, blok PID odczytuje 13 parametrów użytkownika a pozostałe z 40 parametrów jest wykorzystywanych do wewnętrznych obliczeń. Standardowo nie ma potrzeby modyfikowania jakiegokolwiek z tych wartości. Jeżeli jednak blok PID wywołany jest w trybie automatycznym po długiej przerwie, można zapisać w parametrze %Ref+23 za pomocą funkcji SVCREQ #16 bieżący czas zegara w celu uaktualnienia czasu ostatniego wykonania bloku PID, co pozwoli na uniknięcie skoku członu całkującego. Jeżeli wartość bitu 0 Słowa sterującego (%Ref + 14) została ustawiona na 1, następane cztery bity Słowa sterującego będą wykorzystywane do sterowania stykami wejściowymi bloku PID (zgodnie z opisem zamieszczonym na poprzedniej stronie w Tabeli 12-3) oraz muszą być ustawione wartości Punktu pracy regulatora SP i wielkości regulowanej PV, ponieważ szczebel programu sterującego nie jest już w dalszym ciągu odpowiedzialny za sterowanie regulatorem PID.

## Wybór algorytmu sterowania PID (PIDISA lub PIDIND) oraz wzmocnień

Blok PID można zaprogramować w oparciu o algorytm o niezależnych wyrazach (PID\_INT) lub standardowy algorytm ISA (PID\_ISA). Jedyna różnica polega na definicji współczynnika wzmocnienia dla bloku całkującego i różniczkującego. Aby zrozumieć tę różnicę należy mieć świadomość podanych poniżej zależności:

W obydwu algorytmach PID uchyb jest obliczany jako wartość wyrażenia SP-PV lub wyrażenia PV-SP, jeżeli bit Sposób obliczania uchybu ma wartość 1. Warunek błędu jest zapisany w niższym bicie (bit 0) słowa konfiguracyjnego (%Ref+0012). Jeżeli uchyb jest liczony jako różnica (PV-SP) zmiana wartości regulowanej (PV) powoduje wzrost wartości sterowanej (CV). Jeżeli uchyb jest liczony jako różnica (PV-SP) zmiana wartości regulowanej (PV) powoduje wzrost wartości sterowanej (CV). Wprowadzenie bloku całkującego (I) zmienia sposób działania: Jeżeli uchyb jest obliczany jako (PV-SP), wartość sterująca (CV) jest zniższana, jeżeli wartość sterowana (PV) jest większa od wartości zadającej (SP). Jeżeli uchyb jest obliczany jako (PV-SP), wartość sterująca (CV) jest zwiększana jeżeli wartość sterowana (PV) jest większa od wartości zadającej (SP).

**Regulator bezpośredni (normalny):** Uchyb = pomiar – sygnał sterujący (PV-SP), Sposób obliczania błędu (bit 0) = 1

**Regulator odwrotny:** Uchyb = pomiar – sygnał sterujący (PV-SP), Sposób obliczania błędu (bit 0) = 1

Uwaga. **Direct Acting** is sometimes referred to as **Forward Acting**.

Blok różniczkujący zależy standardowo głównie od zmiany uchybu w czasie ostatniej realizacji algorytmu PID, blok może powodować dużą zmianę sygnału ustawiającego, jeżeli wartość zadana uległa zmianie. Jeżeli taka właściwość nie jest pożądana, można ustawić trzeci bit Słowa konfiguracyjnego, na 1, co spowoduje obliczanie wyrazu różniczkującego w oparciu o zmianę wartości zadanej PV. Wartość dt (przyrost czasu) jest wyznaczana poprzez odjęcie od bieżącego czasu sterownika czasu, który upłynął od momentu ostatniego wykonania algorytmu PID.

**dt** = Bieżący czas sterownika - czas, który upłynął od momentu ostatniego wykonania algorytmu PID

**Blok różniczkujący** = (Uchyb- Poprzedni uchyb)/dt lub (PV - poprzednia wartość PV)/dt jeżeli 3-i bit Słowa konfiguracyjnego ma wartość 1.

Algorytm PID o niezależnych wyrazach (PID\_INT) oblicza sygnał ustawiający jako:

**Sygnał ustawiający regulatora PID** =  $K_p * \text{Uchyb} + K_i * \text{Uchyb} * dt + K_d * \text{pochodna} + CV \text{ Bias}$

Standardowy algorytm ISA (PID\_ISA) korzysta z innego wzoru:

**Sygnał ustawiający regulatora PID** =  $K_c * (\text{Uchyb} + \text{Uchyb} * dt/T_i + T_d * \text{pochodna}) + CV \text{ Bias}$



gdzie  $K_c$  jest współczynnikiem wzmocnienia proporcjonalnego,  $T_i$  jest czasem całkowania a  $T_d$  jest czasem różniczkowania. Zaletą algorytmu ISA jest fakt, że zmiana parametru  $K_c$  powoduje zmianę wyrazu proporcjonalnego, jak również wyrazu różniczkującego i całkującego, co ułatwia dostrajanie zamkniętego układu regulacji. Jeżeli wzmocnienie regulatora PID wyrażone jest przy pomocy  $T_i$  i  $T_d$ , należy zastosować następujące wyrażenia

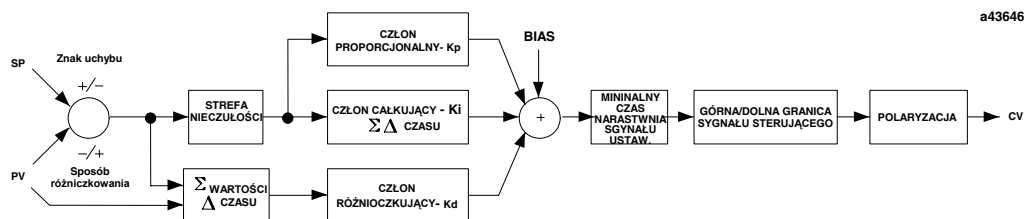
$$K_p = K_c \qquad K_i = K_c/T_i \qquad a \qquad K_d = K_c/T_d$$

do przekonwertowania ich do parametrów wejściowych użytkownika bloku PID.

Wymieniony powyżej składnik CV Bias jest dodatkowym składnikiem, niezależnym od pozostałych parametrów regulatora PID. Może zaistnieć potrzeba używania tylko proporcjonalnego wzmocnienia  $K_p$  i ustawienia CV na wartość nie zerową podczas gdy PV równa się SP a uchyb jest równy 0. W tym przypadku, ustaw CV Bias na żądane CV podczas gdy PV jest określone przez SP. Parametr CV Bias może być również wykorzystywany do sterowania ze sprzężeniem dodatnim, gdzie sygnał ustawiający CV danego regulatora PID jest zmieniany przez inny układ PID lub algorytm sterowania.

Jeżeli wykorzystywane jest wzmocnienie wyrazu całkującego  $K_i$ , parametr CV Bias jest zwykle równy 0, ponieważ blok całkujący automatycznie nie dopuszcza do otrzymania zerowej wartości sygnału ustawiającego. Wystarczy rozpocząć pracę w trybie ręcznym, a następnie za pomocą parametru Sygnał sterujący w trybie ręcznym (%Ref + 13) ustawić blok całkujący na żądaną wartość sygnału zadającego CV, po czym przejść do trybu sterowania automatycznego. Mechanizm ten pracuje również jeżeli  $K_i$  jest równe 0, za wyjątkiem faktu, że wartość wyrazu całkującego nie będzie zmieniana w zależności od wartości uchybu po przejściu do trybu automatycznego.

Zamieszczony poniżej schemat obrazuje pracę algorytmów PID:



Rysunek 12-4. Regulator PID o niezależnych wyrazach (PIDIND)

Algorytm ISA (PIDISA) jest bardzo podobny, różnica polega na wyeliminowaniu współczynnika  $K_p$  ze współczynników  $K_i$  i  $K_d$ , przez co wzmocnienie wyrazu całkującego wynosi  $K_p \cdot K_i$ , a wzmocnienie wyrazu różniczkującego jest równe  $K_p \cdot K_d$ . Znak uchybu, sposób różniczkowania oraz polaryzacja są wybierane poprzez ustawienie odpowiednich wartości parametru użytkownika Słowo konfiguracyjne.

## Wartości graniczne amplitudy i prędkości narastania sygnału zadającego CV

Blok PID nie wysyła obliczonego sygnału wyjściowego bezpośrednio do parametru CV. Obydwa algorytmy PID umożliwiają ograniczenie amplitudy i prędkości narastania sygnału ustawiającego PV. Maksymalna prędkość narastania jest określana w wyniku podzielenia maksymalnej wartości 100% CV (32000) przez minimalny czas narastania, jeżeli jest on określony to wartość jest większa od 0. Przykładowo, jeżeli minimalny czas narastania wynosi 100 sekund, ograniczenie prędkości narastania będzie wynosić 320 jednostek bezwymiarowych sygnału ustawiającego CV na sekundę. Jeżeli czas od momentu ostatniego wykonania wynosi 50 milisekund, nowa wartość sygnału ustawiającego CV nie może się zmienić o więcej, niż  $320 \cdot 50 / 1000$ , czyli o 16 jednostek bezwymiarowych sygnału ustawiającego CV, względem poprzedniej wartości.

Sygnał wyjściowy CV jest następnie porównywany z Górną i Dolną granicą wartości sygnału zadającego. Jeżeli przekroczona zostanie jedna z tych wartości, przyjmowana jest graniczna wartość sygnału

zadającego CV. Jeżeli nastąpiło przekroczenie zarówno wartości granicznych amplitudy jak i prędkości narastania sygnału, następuje dostosowanie wewnętrznej wartości wyrazu całkującego w celu uniknięcia przekraczania wartości dopuszczalnych.

Ostatecznie blok sprawdza polaryzację wyjściową (Drugi bit Słowa konfiguracyjnego o adresie %Ref+12) i zmienia znak sygnału, jeżeli wartość tego bitu jest równa 1.

CV= Sygnał wyjściowy regulatora PID z uwzględnieniem wartości granicznych lub –Sygnał wyjściowy regulatora PID z uwzględnieniem wartości granicznych, jeżeli ustawiono bit polaryzacji sygnału wyjściowego.

Jeżeli blok działa w trybie automatycznym, końcowa wartość CV jest umieszczana w parametrze Sygnał sterujący w trybie ręcznym o adresie %Ref+13. Jeżeli blok pracuje w trybie ręcznym, równanie PID jest pomijane jako, że CV jest ustawiany poprzez parametr Sygnał sterujący w trybie ręcznym, lecz wszystkie wartości graniczne amplitud wciąż są kontrolowane. Oznacza to, że przy pomocy parametru Sygnał sterujący w trybie ręcznym nie można określić wartości sygnału zadającego przekraczającej podane wartości graniczne, jak też nie jest możliwe narastanie sygnału z prędkością większą od określonej przy pomocy parametru minimalny czas narastania sygnału.

## Okres próbkowania bloku PID

Blok PID jest cyfrową implementacją analogowej funkcji sterowania, co powoduje, że czas próbkowania  $dt$  w równaniu zamkniętego układu regulacji nie jest nieskończenie mały, jak to ma miejsce w przypadku analogowych układów sterowania. Większość sterowanych procesów może być przybliżona jako wzmacnienie z inercją pierwszego lub drugiego rzędu, często z opóźnieniem czasowym. Blok PID wyznacza wartość zmiennej ustawiającej CV procesu oraz wykorzystuje bieżącą wartość wartości regulowanej PV do obliczania następnej wartości zmiennej CV. Kluczowym parametrem procesu jest stała czasowa, określająca prędkość narastania wielkości regulowanej PV w przypadku zmiany wielkości ustawiającej CV. Zgodnie z przedstawionym poniżej w punkcie "Metoda dostrajania współczynnika wzmacnienia regulatora PID" opisem, stała czasowa  $T_p+T_c$  dla systemu pierwszego rzędu jest opóźnieniem do momentu, kiedy wielkość regulowana PV dochodzi do 63% wartości końcowej, przy skokowej zmianie sygnału ustawiającego CV. Blok PID nie będzie w stanie sterować procesem, jeżeli Okres próbkowania jest większy od połowy globalnej stałej czasowej. Dłuższe okresy próbkowania nie zapewniają stabilności.

Okres próbkowania nie powinien być większy od globalnej stałej czasowej podzielone przez 10 (lub w najgorszym przypadku przez 5). Przykładowo, jeżeli wydaje się, że wartość regulowana PV dojdzie do około 2/3 swojej końcowej wartości w okresie 2 sekund, okres próbkowania nie powinien być mniejszy od 0.2 sekund, a w najgorszym przypadku 0.4 sekund. Z drugiej strony, okres próbkowania nie powinien być zbyt mały, tzn. mniejszy niż globalna stała czasowa podzielona przez 1000, bowiem składnik  $K_i * Uchyb * dt$  regulatora PID będzie zaokrąglany w dół do zera. Przykładowo, bardzo wolny proces, który zachodzi w ciągu 10 godzin, czyli 36000 sekund, w celu dojścia wartości regulowanej do poziomu 63% swojej wartości, powinien posiadać okres próbkowania wynoszący 40 sekund lub więcej.

Jeżeli proces nie zachodzi bardzo szybko, zwykle nie jest konieczne wprowadzanie wartości 0 dla okresu próbkowania, w celu wykonywania algorytmu PID w każdym cyklu. Jeżeli kilka bloków PID ma ustawiony okres próbkowania o wartości większej od czasu trwania cyklu, czas ten może ulegać znacznym wahaniom, ponieważ w jednym czasie może być rozwiązywanych wiele algorytmów PID. Najprostszym rozwiązaniem jest zastosowanie tablicy bitowej z wędrującą jedynką, sterującej dopływem sygnału do poszczególnych bloków PID.

## Wyznaczanie charakterystyk procesu

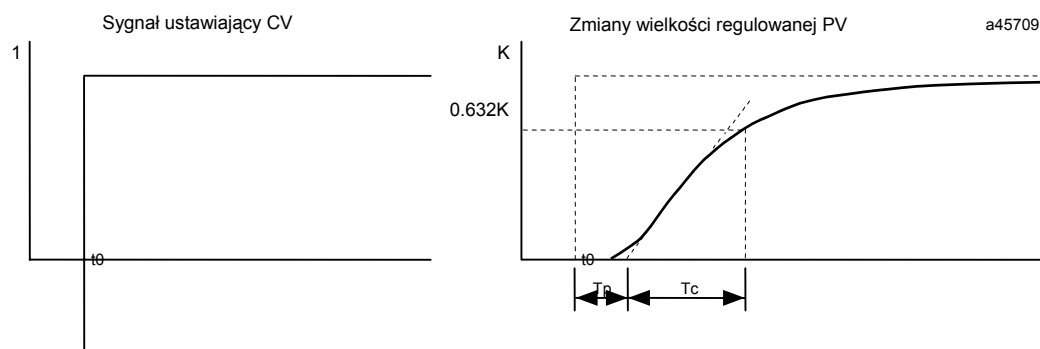
Współczynniki dla algorytmu PID,  $K_p$ ,  $K_i$  i  $K_d$  są wyznaczone na podstawie charakterystyki sterowanego procesu. W czasie wyznaczania parametrów zamkniętego układu regulacji PID należy znaleźć odpowiedź na dwa podstawowe pytania:

1. Jak zmienia się wartość wartości regulowanej PV przy zmianie sygnału ustawiającego CV o stałą wartość lub jak duże jest wzmocnienie otwartego układu regulacji?
2. Jaka jest szybkość reakcji systemu lub jak szybko zmienia się wartość wielkości regulowanej PV po skokowej zmianie wielkości ustawiającej CV?

Wiele procesów może być przybliżonych przez współczynnik wzmocnienia procesu, stałą czasową pierwszego lub drugiego rzędu oraz czas opóźnienia. W dziedzinie częstotliwości, transmitancja dla systemu ze zwłoką czasową pierwszego rzędu ma postać:

$$PV(s)/CV(s) = G(s) = K * e^{-Tp s} / (1 + Tc s)$$

Wykreślenie odpowiedzi skokowej w chwili  $t_0$  w dziedzinie czasu daje krzywą zmiany w układzie otwartym:



Na podstawie krzywej zmiany wielkości regulowanej PV można wyznaczyć następujące parametry procesu:

K	Wzmocnienie w otwartym układzie sterowania procesem = końcowa zmiana wielkości regulowanej PV/zmiana sygnału ustawiającego CV w chwili $t_0$ (zwrócić uwagę na brak indeksu przy współczynniku K).
$T_p$	Czas opóźnienia procesu lub czas po $t_0$ , po upływie, którego wartość PV zacznie się zmieniać.
$T_c$	Stała czasowa procesu pierwszego rzędu, czas mierzony od $T_p$ do momentu, kiedy wartość regulowana PV uzyska 63.3% wartości ostatecznej.

Zwykle najszybszą metodą pomiaru tych parametrów jest wywoływanie bloku PID w trybie ręcznym, z małymi zmianami sygnału ustawiającego CV, poprzez wprowadzanie odpowiednich wartości parametru Sygnał sterujący w trybie ręcznym i wykreślanie w funkcji czasu zmian wartości wielkości regulowanej PV. W przypadku wolnych procesów, czynność ta może być wykonywana ręcznie, natomiast w przypadku procesów szybszych, zachodzi potrzeba skorzystania z rejestratora danych lub komputera z odpowiednim oprogramowaniem. Wartość zmiany sygnału ustawiającego CV powinna być dostatecznie duża tak, aby powodowała dającą się zaobserwować zmianę wartości regulowanej PV, ale nie na tyle duża, aby doprowadzić do przerwania mierzonego procesu. Zmiana może przykładowo wynosić od 2 do 10% różnicy pomiędzy górną i dolną wartością graniczną sygnału ustawiającego.

## Dostrajanie parametrów regulatora PID

Wszystkie parametry bloku PID zależą całkowicie od rodzaju sterowanego procesu, nie można zalecić żadnych wartości, które na pewno okażą się słuszne, można jednakże zastosować prostą, iteracyjną metodę w celu znalezienia akceptowalnego współczynnika wzmocnienia.

1. Ustawić wartość wszystkich parametrów na 0, a następnie ustawić Górną i dolną wartość sygnału sterującego na odpowiednio maksymalną i minimalną oczekiwaną wartość CV. Ustawić Okres próbkowania na wartość szacowanej globalnej stałej czasowej procesu podzielonej przez liczbę z przedziału (0-100)
2. Uruchomić blok w trybie ręcznym, a następnie wprowadzać do parametru Sygnał sterujący w trybie ręcznym (%Ref +13) różne wartości w celu sprawdzenia, czy wartość ustawiająca CV dochodzi do górnej i dolnej wartości granicznej. Zanotować wartości regulowanej PV dla pewnych wartości ustawiającej CV, a następnie przypisać je do wartości zadanej SP.
3. Ustawić małe wzmocnienie Kp, przykładowo  $100 * \text{Maksymalna wartość CV} / \text{maksymalna wartość PV}$ , po czym przejść do trybu ręcznego. Zmienić skokowo wartość SP o 2 do 10% wartość maksymalnej wartości regulowanej PV i obserwować odpowiedzi wartości regulowanej PV. Zwiększyć wartość współczynnika wzmocnienia Kp jeżeli odpowiedź wartości regulowanej PV jest zbyt wolna lub też zmniejszyć ją, jeżeli zmiany wartości regulowanej PV są zbyt duże, albo oscylują, bez dochodzenia do stanu stabilności.
4. Po znalezieniu odpowiedniego współczynnika wzmocnienia Kp, zwiększać wartość Ki w celu znalezienia przeregulowania tłumionego do wartości ustalonej w ciągu 2 do 3 cykli. Może to wymagać zmniejszenia współczynnika Kp. Spróbować również zmienić skokową wielkość zmian oraz wprowadzić inne wartości sygnału ustawiającego CV.
5. Po znalezieniu odpowiednich współczynników Kp i Ki, spróbować zwiększać współczynnik Kd w celu szybszego uzyskiwania odpowiedzi na parametry wejściowe, jednak bez powstawania oscylacji. Współczynnik Kd jest bardzo często niepotrzebny oraz nie powinien być wykorzystywany w przypadku zakłóconego sygnału wielkości regulowanej PV.
6. Sprawdzić współczynnik wzmocnienia dla różnych wartości zadających SP oraz, jeżeli jest to potrzebne, określić strefę nieczułości i Minimalny czas narastania sygnału ustawiającego. W przypadku niektórych procesów konieczne może być ustawienie bitów Słowa konfiguracyjnego w celu zmiany znaku uchybu lub polaryzacji.

## Dobór współczynników wzmocnienia za pomocą metody Zieglera i Nicholasa

Po wyznaczeniu trzech parametrów modelu procesu,  $K$ ,  $T_p$  i  $T_c$  można wstępnie oszacować wartość współczynników wzmocnienia dla zamkniętego układu regulacji PID. Następujące podejście do tego problemu zaproponowane przez Ziegler'a i Nichols'a w 1940 zapewnia dobrą reakcję systemu przy wzmocnieniach dających stosunek amplitud 1/4. Stosunek amplitud jest to stosunek drugiej wartości szczytowej do pierwszej wartości szczytowej odpowiedzi zamkniętego układu regulacji.

1. Obliczyć szybkość reakcji regulatora:

$$R = K/T_c$$

2. Gdy regulator pracuje jedynie jako regulator proporcjonalny, policzyć współczynnik wzmocnienia wyrazu proporcjonalnego  $K_p$ :

$$K_p = 1/(R * T_p) = T_c/(K * T_p)$$

3. Gdy regulator pracuje jako proporcjonalny i całkujący, skorzystać z równania:

$$K_p = 0.9/(R * T_p) = 0.9 * T_c/(K * T_p)$$

$$K_i = 0.3 * K_p/T_p$$

4. Gdy regulator pracuje jako proporcjonalny, całkujący i różniczkujący, zastosować równanie:

$$K_p = G/(R * T_p) \quad \text{gdzie } G \text{ ma wartość z zakresu od } 1.2 \text{ do } 2.0$$

$$K_i = 0.5 * K_p/T_p$$

$$K_d = 0.5 * K_p * T_p$$

5. Sprawdzić, czy okres próbkowania ma wartość z zakresu  $(T_p + T_c)/10$  do  $(T_p + T_c)/1000$

Inna metoda "Idealnego dostrojenia" ma za zadanie zapewnić jak najlepszej odpowiedzi na zmiany wartości zadanej SP, opóźnionej wyłącznie przez opóźnienie procesu  $T_p$  lub czas martwy:

$$K_p = 2 * T_c/(3 * K * T_p)$$

$$K_i = T_c$$

$$K_d = K_i/4 \quad \text{jeżeli wykorzystywany jest składnik różniczkujący}$$

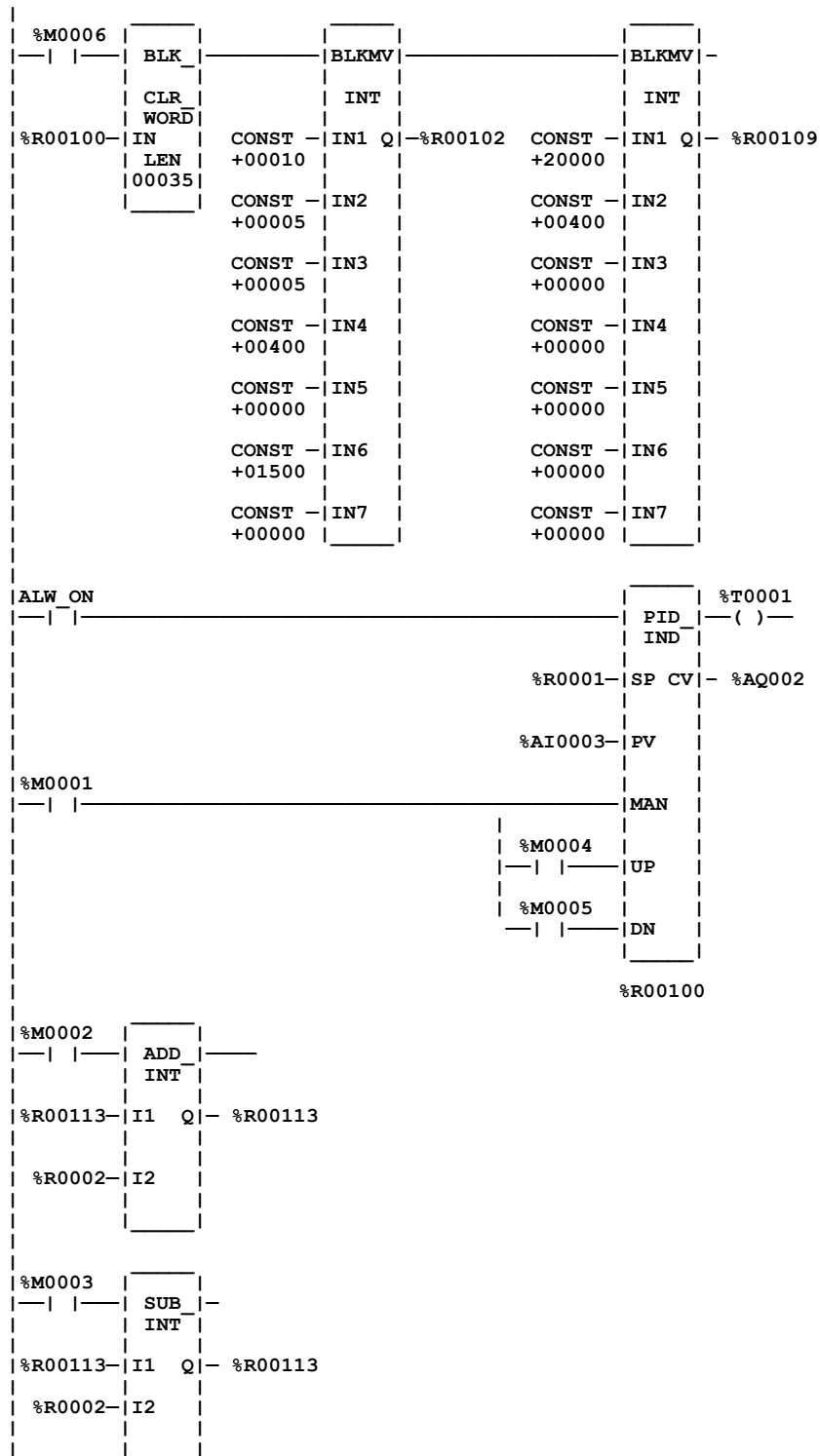
Po wyznaczeniu początkowych wartości współczynników wzmocnienia, można je przekonwertować na parametry użytkownika, będące liczbami całkowitymi. W celu uniknięcia problemów ze skalowaniem, wzmocnienie procesu  $K$  powinno być obliczane jako zmiana w jednostkach bezwymiarowych wartości regulowanej PV podzielonej przez skokową zmianę wartości wyjściowej CV w jednostkach bezwymiarowych CV, a nie w jednostkach inżynierskich parametrów PV czy CV. Również czas powinien być podawany w sekundach. Po wyznaczeniu współczynników  $K_p$ ,  $K_i$  i  $K_d$ , współczynniki  $K_p$  i  $K_d$  można pomnożyć przez 100 i wprowadzić je jako wartości całkowite, a współczynnik  $K_i$  należy pomnożyć przed wprowadzeniem do tabeli parametrów użytkownika przez 1000.

## Przykład prostego wywołania regulatora PID

W zamieszczonym poniżej przykładzie, Okres próbkowania wynosi 100 milisekund, współczynnik wzmocnienia Kp jest równy 4.00 a współczynnik wzmocnienia Ki jest równy 1.500. Wartość zadana SP zapamiętana w %R1, sygnał ustawiający Cv jest zapisywany w %AQ2, a wielkość regulowana zapisywana jest do %AI3. Muszą być również określone górna i dolna granica sygnału ustawiającego, w niniejszym przykładzie są to 20000 i 400, oraz opcjonalnie wprowadzono górną i dolną granice strefy nieczułości +5 i -5. Tablica RefArray zawierająca 40 słów rozpoczyna się od adresu %R100. Zwarcie styku %M0006 uaktywnia funkcje BLKMV, które ustawiają warunki początkowe przez kopiowanie wartości do bloku 14 słów rozpoczynających się od %R102 (%Ref+2). (Uwaga: aby zoptymalizować parametry w czasie dostrajania, należy w oprogramowaniu Logicmaster ustawić kursor na bloku PID i wcisnąć klawisz F10).

Blok funkcyjny PID można przełączyć w tryb ręcznego sterowania za pomocą zmiennej %M0001, co pozwala na zmianę parametru sygnał sterujący w trybie ręcznym (rejestr %Ref+13, czyli w niniejszym przykładzie %R0113). Bity %M0004 lub %M0005 umożliwiają zwiększenie lub zmniejszenie wartości zmiennej %R0113 o 1 przy każdym rozwiązywaniu algorytmu PID, w okresie co 100 milisekund. W celu przyspieszenia pracy w trybie ręcznym, za pomocą bitów %M0002 i %M0003 można w każdym cyklu pracy sterownika dodawać i odejmować wartość zapisaną w %R0002 do wartości %R0113. Sygnał wyjściowy %T0001 jest wysyłany pod warunkiem poprawnego wykonania algorytmu regulatora PID. Należy zwrócić uwagę, że niewszystkie rejestry zawierają się w bloku (40 rejestrów) ponieważ nie są wykorzystane w przykładzie lub nie są skonfigurowane (są wykorzystane przez sterownik). Dodatkowe informacje o parametrach zawarto w tabeli 12-8.

Adres	Wartość	Opis
%R0102	+00010	Sample Period (Okres próbkowania)
%R0103	+00005	Dead Band + (Górna granica strefy nieczułości)
%R0104	+00005	Dead Band – (dolna granica nieczułości)
%R0105	+00400	Proportional Gain – (Kp) (Współczynnik wzmocnienia proporcjonalnego Kp)
%R0106	+00000	Derivative Gain – (Kd) (Współczynnik wzmocnienia różniczkowego Kd)
%R0107	+01500	Integral Gain (Ki) (Współczynnik wzmocnienia całkowego Ki)
%R0108	+00000	CV Bias/Output Offset (Przesunięcie punktu pracy)
%R0109	+20000	Upper Clamp (Górna granica wartości sygnału ustawiającego)
%R0110	+00400	Lower Clamp (Dolna granica wartości sygnału sterującego)
%R0111	+00000	Minimum Slew Time (Minimalny czas narastania sygnału ustawiającego)
%R0112	+00000	Config. Word (Słowo konfiguracyjne)
%R0113	+00000	Manual Command (Sygnał sterujący w trybie ręcznym)
%R0114	+00000	Control Word (Słowo sterujące)
%R0115	+00000	Internal SP (Nie konfigurowalne)







Sterowniki serii 9030, 90-20 i Micro udostępniają szereg funkcji i bloków funkcyjnych. W Załączniku A podano rozmiar pamięci w bajtach oraz czas wykonywania w mikrosekundach każdej z instrukcji programowania. Rozmiar pamięci to liczba bajtów, zajmowanych przez każdą z instrukcji w drabinie logicznej programu sterującego.

Dla każdego elementu logicznego w tablicy podano dwa rodzaje czasów wykonywania:

Czas wykonywania	Opis
Aktywny	Czas wymagany na realizację elementu logicznego w sytuacji, gdy dopływa do niego sygnał wejściowy i element ten przesyła sygnał wyjściowy. Zwykle, najkrótsze czasy są uzyskiwane jeżeli blok umieszczony jest w pamięci RAM programu sterującego (zorientowanej rejestrowo), a nie w pamięci dyskretnej).
Nieaktywny	Czas wymagany na realizację elementu logicznego w sytuacji, gdy dopływa do niego sygnał wejściowy, lecz element nie jest aktywny, jak na przykład przełącznik czasowy przy dopływie sygnału zerującego.

#### Uwaga

Timers and counters are updated each time they are encountered in the logic, timers by the amount of time consumed by the last sweep and counters by one count. Liczniki i przełączniki czasowe uaktualnianie są za każdym razem ich wywołania w programie sterującym, wartość licznika jest zwiększana o czas trwania ostatniego cyklu, a wartość przełączników czasowych o jeden.

#### Uwaga

W przypadku jednostek centralnych 350, 351, 352 i 360, czasy wykonywania są identyczne, za wyjątkiem funkcji MOVE, której czas wykonywania przez jednostkę centralną CPU350 jest inny - proszę porównać z uwagą podaną pod tabelą, na stronie A-6.

Tabela A-1 Czasy wykonywania instrukcji

Funkcja Grupa	Funkcja	Aktywny				Nieaktywny				Przyrost czasu				Roz miar
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Przełączniki czasowe	Przełącznik czasowy bez pamięci	146	81	80	42	105	39	38	21	-	-	-	-	15
	Przełącznik czasowy z zanegowanym wejściem, bez pamięci	98	47	44	23	116	63	58	32	-	-	-	-	9
	Przełącznik	122	76	75	40	103	54	53	30	-	-	-	-	15
Liczniki	Licznik zliczający w górę	137	70	69	36	130	63	62	33	-	-	-	-	11
	Licznik zliczający w dół	136	70	69	37	127	61	61	31	-	-	-	-	11
Funkcje matematycz ne	Dodawanie (INT)	76	47	46	24	41	0	1	0	-	-	-	-	13
	Dodawanie (DINT)	90	60	60	34	41	1	0	0	-	-	-	-	13
	Odejmowanie (INT)	75	46	45	25	41	0	1	0	-	-	-	-	13
	Odejmowanie (DINT)	92	62	62	34	41	1	0	0	-	-	-	-	13
	Mnożenie (INT)	79	49	50	28	41	0	1	0	-	-	-	-	13
	Mnożenie (DINT)	108	80	101	43	41	1	0	0	-	-	-	-	13
	Dzielenie (INT)	79	51	50	27	41	0	1	0	-	-	-	-	13
	Dzielenie (DINT)	375	346	348	175	41	1	0	0	-	-	-	-	13
	Dzielenie Modulo (INT)	78	51	49	27	41	0	1	0	-	-	-	-	13
	Dzielenie Modulo (DINT)	134	103	107	54	41	1	0	0	-	-	-	-	13
	Pierwiastek kwadratowy (INT)	153	124	123	65	42	0	1	0	-	-	-	-	9
	Pierwiastek kwadratowy (DINT)	268	239	241	120	42	0	0	1	-	-	-	-	9
Relacje	Równe (INT)	66	35	36	19	41	1	1	0	-	-	-	-	9
	Równe (DINT)	86	56	54	29	41	1	0	0	-	-	-	-	9
	Różne (INT)	67	39	35	22	41	1	1	0	-	-	-	-	9
	Różne (DINT)	81	51	51	28	41	1	0	0	-	-	-	-	9
	Większe (INT)	64	33	35	20	41	1	1	0	-	-	-	-	9
	Większe (DINT)	89	59	58	32	41	1	0	0	-	-	-	-	9
	Większe lub równe (INT)	64	36	34	19	41	1	1	0	-	-	-	-	9
	Większe lub równe (DINT)	87	58	57	30	41	1	0	0	-	-	-	-	9
	Mniejsze (INT)	66	35		19	41	1	1	0	-	-	-	-	9
	Mniejsze (DINT)	87	57		30	41	1	1	0	-	-	-	-	9
	Mniejsze lub równe (INT)	66	36	34	21	41	1	1	0	-	-	-	-	9
	Mniejsze lub równe (DINT)	86	57	56	31	41	1	1	0	-	-	-	-	9
	Zakres (INT)	92	58	54	29	46	1	0	1	-	-	-	-	15
	Zakres (DINT)	106	75	57	37	45	0	0	0	-	-	-	-	15
	Zakres (WORD)	93	60	54	29	0	0	0	0	-	-	-	-	15

**Uwagi:** 1. Czasy (w mikrosekundach) podano dla jednostek centralnych 311, 313, 340 i 341 (Wersja 7 dla 331), na podstawie oprogramowania Logimaster 90-30/20 w wersji 5.01.

2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/ bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modułem HSC.
5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.

Tabela A-1 Czasy wykonywania instrukcji - kontynuacja

Funkcja Grupa	Funkcja	Aktywny				Nieaktywny				Przyrost czasu				Rozmiar	
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41		
Bit Działanie	Logiczne AND	67	37	37	22	42	0	0	1	-	-	-	-	13	
	Logiczne OR	68	38	38	21	42	0	0	1	-	-	-	-	13	
	Alternatywa wyłączająca OR	66	38	37	20	42	0	1	1	-	-	-	-	13	
	Negacja logiczna słowa bitowego NOT	62	32	31	17	42	0	1	1	-	-	-	-	9	
	Przesunięcie słowa bitowego w lewo	139	89	90	47	74	26	23	13	11.61	11.61	12.04	6.29	15	
	Przesunięcie słowa bitowego w prawo	135	87	85	45	75	26	24	13	11.63	11.62	12.02	6.33	15	
	Przesunięcie słowa bitowego w lewo w obiegu zamkniętym	156	127	126	65	42	1	1	0	11.70	11.78	12.17	6.33	15	
	Przesunięcie słowa bitowego w prawo w obiegu zamkniętym	146	116	116	62	42	1	1	0	11.74	11.74	12.13	6.27	15	
	Pozycja bitu	102	72	49	38	42	1	0	0	-	-	-	-	13	
	Ustawianie wartości pojedynczego bitu na 0	68	38	35	21	42	1	1	1	-	-	-	-	13	
	Sprawdzanie wartości pojedynczego bitu	79	49	51	28	41	0	0	1	-	-	-	-	13	
	Ustawianie wartości pojedynczego bitu na 1	67	37	37	20	42	0	0	0	-	-	-	-	13	
	Porównanie z maskowaniem (WORD)	217	154	141	74	107	44	39	21	-	-	-	-	25	
	Porównanie z maskowaniem (DWORD)	232	169	156	83	108	44	39	22	-	-	-	-	25	
	Przemieszczenie danych	Przemieszczanie (INT)	68	37	39	20	43	0	0	0	1.62	1.62	5.25	1.31	13
Przemieszczanie (BIT)		94	62	64	35	42	0	0	0	12.61	12.64	12.59	6.33	13	
Przemieszczanie (WORD)		67	37	40	20	41	0	0	0	1.62	1.63	5.25	1.31	13	
Przemieszczanie grupy wartości (INT)		76	48	50	28	59	30	30	16	-	-	-	-	27	
Przemieszczanie grupy wartości (WORD)		76	48	49	29	59	29	28	15	-	-	-	-	27	
Zerowanie fragmentu pamięci		56	28	27	14	43	0	0	0	1.35	1.29	1.40	0.78	9	
Rejestr przemieszczający bity (BIT)		201	153	153	79	85	36	34	18	0.69	0.68	0.71	0.37	15	
Rejestr przemieszczający słowa (WORD)		103	53	52	29	73	25	23	12	1.62	1.62	2.03	1.31	15	
Przemieszczanie jedyńki		165	101	99	53	96	31	29	16	0.07	0.07	0.08	0.05	15	
	COMM_REQ	1317	1272	1489	884	41	2	0	0	-	-	-	-	13	
Operacje	Kopiowanie danych	INT	230	201	177	104	72	41	40	20	1.29	1.15	10.56	2.06	21
		DINT	231	202	181	105	74	44	42	23	3.24	3.24	10.53	2.61	21
		BIT	290	261	229	135	74	43	42	23	-0.3	-0.3	-0.01	0.79	21
		BYTE	228	198	176	104	74	42	42	23	0.81	0.82	8.51	1.25	21
		WORD	230	201	177	104	72	41	40	20	1.29	1.15	10.56	2.06	21
	Szukanie wartości zadanej	INT	197	158	123	82	78	39	37	20	1.93	1.97	2.55	1.55	19
		DINT	206	166	135	87	79	38	36	21	4.33	4.34	4.55	2.44	19
		BYTE	179	141	117	74	78	38	36	21	1.53	1.49	1.83	1.03	19
		WORD	197	158	123	82	78	39	37	20	1.93	1.97	2.55	1.55	19

**Uwagi:** 1. Czasy (w mikrosekundach) podano dla jednostek centralnych 311, 313, 340 i 341 (Wersja 7 dla 331), na podstawie oprogramowania Logimaster 90-30/20 w wersji 5.01.

2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/ bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.

3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.

4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modułem HSC.

5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.

6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.

7. W przypadku instrukcji, dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.

Tabela A-1 Czasy wykonywania instrukcji - kontynuacja

Funkcja Grupa	Funkcja	Aktywny				Nieaktywny				Przyrost czasu				Roz miar	
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41		
	Szukanie wartości różnej														
	INT	198	159	124	83	79	39	36	21	1.93	1.93	2.48	1.52	19	
	DINT	201	163	132	84	79	37	35	21	6.49	6.47	6.88	3.82	19	
	BYTE	179	141	117	73	79	38	36	19	1.54	1.51	1.85	1.05	19	
	WORD	198	159	124	83	79	39	36	21	1.93	1.93	2.48	1.52	19	
	Szukanie wartości większej														
	INT	198	160	125	82	79	37	38	19	3.83	3.83	4.41	2.59	19	
	DINT	206	167	135	88	78	38	36	20	8.61	8.61	9.03	4.88	19	
	BYTE	181	143	118	73	79	37	36	19	3.44	3.44	3.75	2.03	19	
	WORD	198	160	125	82	79	37	38	19	3.83	3.83	4.41	2.59	19	
	Przeszukiwanie tablicy w celu znalezienia wartości większej lub równej wartości zadanej														
	INT	197	160	124	83	77	38	36	20	3.86	3.83	4.45	2.52	19	
	DINT	205	167	136	87	80	39	36	21	8.62	8.61	9.02	4.87	19	
	BYTE	180	142	118	75	79	37	37	20	3.47	3.44	3.73	2.00	19	
	WORD	197	160	124	83	77	38	36	20	3.86	3.83	4.45	2.52	19	
	Szukanie wartości mniejszej														
	INT	199	159	124	84	78	38	36	20	3.83	3.86	4.48	2.48	19	
	DINT	206	168	135	87	79	38	38	19	8.62	8.60	-1.36	4.88	19	
	BYTE	181	143	119	75	80	38	37	20	3.44	3.44	3.75	2.00	19	
	WORD	199	159	124	84	78	38	36	20	3.83	3.86	4.45	2.48	19	
	Szukanie wartości mniejszej lub równej														
	INT	200	158	124	82	79	38	37	21	3.79	3.90	4.45	2.55	19	
	DINT	207	167	137	88	78	39	37	19	8.60	8.61	9.01	4.86	19	
	BYTE	180	143	119	74	78	40	37	19	3.46	3.44	3.73	2.02	19	
	WORD	200	158	124	82	79	38	37	21	3.79	3.90	4.45	2.55	19	
	Funkcje konwersji	Konwersja na dane typu BCD-4	74	46	39	25	42	1	1	1	-	-	-	-	9
		Konwersja danych typu BCD na dane typu INT	77	50	34	25	42	1	1	1	-	-	-	-	9

- Uwagi:**
1. Czasy (w mikrosekundach) podano dla jednostek centralnych 311, 313, 340 i 341 (Wersja 7 dla 331), na podstawie oprogramowania Logicmaster 9030/20 w wersji 5.01.
  2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/liczbę bitów lub słów.
  3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modulem HSC.
  5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.
  7. W przypadku instrukcji dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.

Tabela A-1 Czasy wykonywania instrukcji - kontynuacja

Funkcja Grupa	Funkcja	Aktywny				Nieaktywny				Przyrost czasu				Roz miar
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Funkcje sterujące	Wywołanie podprogramu	155	93	192	85	41	0	0	0	-	-	-	-	7
	Natychmiastowe uaktualnienie stanu wejść/wyjść	309	278	323	177	38	1	0	0	-	-	-	-	12
	Algorytm PID-ISA	1870	1827	1812	929	91	56	82	30	-	-	-	-	15
	Algorytm PID-IND	2047	2007	2002	1017	91	56	82	30	-	-	-	-	15
	Instrukcje END	-	-	-	-	-	-	-	-	-	-	-	-	-
	Funkcje specjalne sterownika SVCREQ													
	# 6	93	54	63	45	41	2	0	0	-	-	-	-	9
	# 7 (Odczyt)	-	37	309	161	-	2	0	0	-	-	-	-	9
	# 7 (Ustawianie)	-	37	309	161	-	2	0	0	-	-	-	-	9
	#14	447	418	483	244	41	2	0	0	-	-	-	-	9
	#15	281	243	165	139	41	2	0	0	-	-	-	-	9
	#16	131	104	115	69	41	2	0	0	-	-	-	-	9
	#18	-	56	300	180	-	2	0	0	-	-	-	-	9
	#23	1689	1663	1591	939	43	1	0	0	-	-	-	-	9
	#26//30*	1268	1354	6680	3538	42	0	0	0	-	-	-	-	9
	#29	-	-	55	41	-	-	1	0	-	-	-	-	9
	Para zagnieżdżonych instrukcji MCR/ENDMCR	135	73	68	39	75	25	21	12	-	-	-	-	8

\* Czas dla funkcji specjalnej #26/30 został zmierzony przy pomocy licznika wysokiej częstotliwości, dla 16 punktowego modułu wyjściowego, zainstalowanego w 5 gnieździe.

- Uwagi:**
1. Czasy (w mikrosekundach) podano dla jednostek centralnych 311, 313, 340 i 341 (Wersja 7 dla 331), na podstawie oprogramowania Logimaster 90-30/20 w wersji 5.01.
  2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modułem HSC.
  5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.
  7. W przypadku instrukcji, dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.

Tabela A-2 Czasy wykonywania instrukcji CPU35x/36x

Funkcja Grupa	Funkcja	Aktywny	Nieaktywny	Przyrost czasu	Aktywny	Nieaktywny	Przyrost czasu	Rozmiar	
		350/351/36x	350/351/36x	350/351/36x	352	352	352		
Przełączniki czasowe	Przełącznik czasowy z pamięcią	4	6	–	4	5	–	15	
	Przełącznik	3	3	–	2	2	–	15	
	Przełącznik czasowy bez pamięci, z zanegowanym wejściem	3	3	–	3	2	–	15	
Liczniki	Licznik zliczający w górę	1	3	–	2	2	–	13	
	Licznik zliczający w dół	3	3	–	1	2	–	13	
Funkcje matematyczne	Dodawanie (INT)	2	0	–	1	0	–	13	
	Dodawanie (DINT)	2	0	–	2	0	–	19	
	Dodawanie (REAL)	52	0	–	33	0	–	17	
	Odejmowanie (INT)	2	0	–	1	0	–	13	
	Odejmowanie (DINT)	2	0	–	2	0	–	19	
	Odejmowanie (REAL)	53	0	–	34	0	–	17	
	Mnożenie (INT)	21	0	–	21	0	–	13	
	Mnożenie (DINT)	24	0	–	24	0	–	19	
	Mnożenie (REAL)	68	1	–	38	1	–	17	
	Dzielenie (INT)	22	0	–	22	0	–	13	
	Dzielenie (DINT)	25	0	–	25	0	–	19	
	Dzielenie (REAL)	82	2	–	36	2	–	17	
	Dzielenie Modulo (INT)	21	0	–	21	0	–	13	
	Dzielenie Modulo (DINT)	25	0	–	25	0	–	19	
	Pierwiastek kwadratowy (INT)	42	1	–	41	1	–	10	
	Pierwiastek kwadratowy (DINT)	70	0	–	70	0	–	13	
	Pierwiastek kwadratowy (REAL)	137	0	–	35	0	–	11	
	Funkcje trygonometryczne	SIN (REAL)	360	0	–	32	0	–	11
		COS (REAL)	319	0	–	29	0	–	11
		TAN (REAL)	510	1	–	32	1	–	11
ASIN (REAL)		440	0	–	45	0	–	11	
ACOS (REAL)		683	0	–	63	0	–	11	
ATAN (REAL)		264	1	–	33	1	–	11	
LOG (REAL)		469	0	–	32	0	–	11	
Funkcje logarytmiczne	LN (REAL)	437	0	–	32	0	–	11	
	EXP	639	0	–	42	0	–	11	
Funkcje wykładnicze	EXPT	89	1	–	54	1	–	17	
	Konwersja wartości kąta	65	1	–	32	1	–	11	
Konwersja wartości kąta	Konwersja radianów na stopnie	65	1	–	32	1	–	11	
	Konwersja stopni na radiany	59	0	–	32	0	–	11	

- Uwagi:**
1. Czasy (w mikrosekundach) dla jednostek centralnych 351 i 352 na podstawie oprogramowania Logicmaster 90-30/Micro w wersji 7.
  2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modulem HSC.
  5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.

Tabela A-2 Czasy wykonywania instrukcji CPU35x/36x - kontynuacja

Funkcja Grupa	Funkcja	Aktywny	Nieaktywny	Przyrost czasu	Aktywny	Nieaktywny	Przyrost czasu	Rozmiar
		350/351/36x	350/351/36x	350/351/36x	352	352	352	
Relacje	Równe (INT)	1	0	–	1	0	–	10
	Równe (DINT)	2	0	–	2	0	–	16
	Równe (REAL)	57	0	–	28	0	–	14
	Różne (INT)	1	0	–	1	0	–	10
	Różne (DINT)	1	0	–	1	0	–	16
	Różne (REAL)	62	0	–	31	0	–	14
	Większe (INT)	1	0	–	1	0	–	10
	Większe (DINT)	1	0	–	1	0	–	16
	Większe (REAL)	57	0	–	32	0	–	14
	Większe lub równe (INT)	1	0	–	1	0	–	10
	Większe lub równe (DINT)	1	0	–	1	0	–	10
	Większe lub równe (REAL)	57	1	–	31	1	–	14
	Mniejsze (INT)	1	0	–	1	0	–	10
	Mniejsze (DINT)	1	0	–	1	0	–	16
	Mniejsze (REAL)	58	1	–	36	1	–	14
	Mniejsze lub równe (INT)	1	0	–	1	0	–	10
	Mniejsze lub równe (DINT)	3	0	–	3	0	–	16
	Mniejsze lub równe (REAL)	37	0	–	37	0	–	14
	Zakres (INT)	2	1	–	2	1	–	13
	Zakres (DINT)	2	1	–	2	1	–	22
Zakres (WORD)	1	0	–	1	0	–	13	
Bit Działanie	Logiczne AND	2	0	–	2	0	–	13
	Logiczne OR	2	0	–	2	0	–	13
	Alternatywa wyłączająca OR	1	0	–	1	0	–	13
	Negacja logiczna słowa bitowego NOT	1	0	–	1	0	–	10
	Przesunięcie słowa bitowego w lewo	31	1	1.37	31	1	1.37	16
	Przesunięcie słowa bitowego w prawo	28	0	3.03	28	0	3.03	16
	Przesunięcie słowa bitowego w lewo w obiegu zamkniętym	25	0	3.12	25	0	3.12	16
	Przesunięcie słowa bitowego w prawo w obiegu zamkniętym	25	0	4.14	25	0	4.14	16
	Pozycja bitu	20	1	–	20	1	–	13
	Ustawianie wartości pojedynczego bitu na 0	20	0	–	20	0	–	13
	Sprawdzanie wartości pojedynczego bitu	20	0	–	20	0	–	13
	Ustawianie wartości pojedynczego bitu na 1	19	1	–	19	1	–	13
	Porównanie z maskowaniem (WORD)	52	0	–	52	0	–	25
	Porównanie z maskowaniem (DWORD)	50	0	–	49	0	–	25

- Uwagi:**
1. Czasy (w mikrosekundach) dla jednostek centralnych 351 i 352 na podstawie oprogramowania Logicmaster 90-30/Micro w wersji 7.
  2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modulem HSC.
  5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.
  7. W przypadku instrukcji dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.

Tabela A-2. Czasy wykonywania instrukcji CPU35x/36x - kontynuacja

Funkcja Grupa	Funkcja	Aktywne	Nieaktywne	Przyrost czasu	Aktywne	Nieaktywne	Przyrost czasu	Rozmiar
		350/351/ 36X	350/351/ 36X	350/351/36 X	352	352	352	
Przemieszczenie danych	Przemieszczanie (INT)	2	0	0.41	2	0	0.41	10
	Przemieszczanie (BIT)	28	0	4.98	28	0	4.98	13
	Przemieszczanie (WORD)	2	0	0.41	2	0	0.41	10
	Przemieszczanie (REAL)	24	1	0.82	24	1	0.82	13
	Przemieszczanie grupy wartości (INT)	2	0	–	2	0	–	28
	Przemieszczanie grupy wartości (WORD)	4	4	–	3	0	–	28
	Przemieszczanie grupy wartości (REAL)	41	0	–	41	0	–	13
	Zerowanie fragmentu pamięci	1	0	0.24	1	0	0.24	11
	Rejestr przemieszczający bity (BIT)	49	0	0.23	46	0	0.23	16
	Rejestr przemieszczający słowa (WORD)	27	0	0.41	27	0	0.41	16
	Przemieszczanie jedynek	38	22	0.02	38	22	0.02	16
	COMM_REQ	765	0	–	765	0	–	13
Operacje	Kopiowanie danych							
	INT	54	0	0.97	54	0	0.97	22
	DINT	54	0	0.81	54	0	0.81	22
	BIT	69	0	0.36	69	0	0.36	22
	BYTE	54	1	0.64	54	1	0.64	22
	WORD	54	0	0.97	54	0	0.97	22
	Szukanie wartości zadanej							
	INT	37	0	0.62	37	0	0.62	19
	DINT	41	1	1.38	41	1	1.38	22
	BYTE	35	0	0.46	35	0	0.46	19
	WORD	37	0	0.62	37	0	0.62	19
	Szukanie wartości różnej							
	INT	37	0	0.62	37	0	0.62	19
	DINT	38	0	2.14	38	0	2.14	22
	BYTE	37	0	0.47	37	0	0.47	19
	WORD	37	0	0.62	37	0	0.62	19
	Szukanie wartości większej							
	INT	37	0	1.52	37	0	1.52	19
	DINT	39	0	2.26	39	0	2.26	22
	BYTE	36	1	1.24	36	1	1.24	19
	WORD	37	0	1.52	37	0	1.52	19
	Szukanie wartości większej lub równej							
	INT	37	0	1.48	37	0	1.48	19
	DINT	39	0	2.33	39	0	2.33	22
	BYTE	37	1	1.34	37	1	1.34	19
	WORD	37	0	1.48	37	0	1.48	19

- Uwagi:**
1. Czasy (w mikrosekundach) dla jednostek centralnych 35x i 36x na podstawie oprogramowania Logimaster 90-30/Micro w wersji 7.
  2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modulem HSC.
  5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.
  7. W przypadku instrukcji dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.



Tabela A-2 Czasy wykonywania instrukcji CPU35x/36x - kontynuacja

Funkcja Grupa	Funkcja	Aktywny	Nieaktywny	Przyrost czasu	Aktywny	Nieaktywny	Przyrost czasu	Rozmiar
		350/351/36x	350/351/36x	350/351/36x	352	352	352	
	Szukanie wartości mniejszej							
	INT	37	0	1.52	37	0	1.52	19
	DINT	41	1	2.27	41	1	2.27	22
	BYTE	37	0	1.41	37	0	1.41	19
	WORD	37	0	1.52	37	0	1.52	19
	Szukanie wartości mniejszej lub równej							
	INT	38	0	1.48	38	0	1.48	19
	DINT	40	1	2.30	40	1	2.30	22
	BYTE	37	0	1.24	37	0	1.24	19
	WORD	38	0	1.48	38	0	1.48	19
Funkcje konwersji	Konwersja na dane typu BCD-4	19	1	-	19	1	-	10
	Konwersja na dane typu BCD	21	1	-	21	1	-	10
	Konwersja na dane typu REAL	27	0	-	21	0	-	8
	Konwersja na dane typu WORD	28	1	-	30	1	-	11
	Przybliżenie do liczby typu INT	32	0	-	32	0	-	11
	Przybliżenie do liczby typu DINT	63	0	-	31	0	-	11
Funkcje sterujące	Wywołanie podprogramu	72	1	-	73	1	-	7
	Natychmiastowe uaktualnienie stanu wejść/wyjść	114	1	-	115	1	-	13
	Algorytm PID-ISA*	162	34	-	162	34	-	16
	Algorytm PID-IND*	146	34	-	146	34	-	16
	Instrukcje END	-	-	-	-	-	-	-
	Funkcje specjalne sterownika SVCREQ							
	#6	22	1	-	22	1	-	10
	# 7 (Odczyt)	75	1	-	75	1	-	10
	# 7 (Ustawianie)	75	1	-	75	1	-	10
	#14	121	1	-	121	1	-	10
	#15	46	1	-	46	1	-	10
	#16	36	1	-	36	1	-	10
	#18	261	1	-	261	1	-	10
	#23	426	0	-	426	0	-	10
	#26//30**	2260	1	-	2260	1	-	10
	#29	20	0	-	20	0	-	10
	#43							
	Para zagnieżdżonych instrukcji MCR/ENDMCR Łączone	1	1	-	1	1	-	4
	SER	(Patrz tabelą A-3)	26.50		(Porównać z tabelą A-3)			

\* Czasy dla regulatora PID w powyższej tabeli podano dla jednostek centralnych 351, wersja 6.5.

\*\* Czas dla funkcji specjalnej #26/30 został zmierzony przy pomocy licznika o dużej prędkości, dla 16 punktowego modułu wyjściowego, zainstalowanego w 5 gnieździe.

- Uwagi:**
1. Czasy (w mikrosekundach) dla jednostek centralnych 35x i 36x na podstawie oprogramowania Logicmaster 90-30/Micro w wersji 7.
  2. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  3. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  4. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modułem HSC.
  5. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  6. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.
  7. W przypadku instrukcji, dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.

Tabela A-3 Czas wykonywania bloku funkcyjnego SER

Konfigurowanie	Przykład	Czas ( $\mu$ s)
Sygnal wyjściowy nie wysyłany (disabled)	—	26.50
<b>Sąsiadujące</b>		
8 kanałów	%I1—8	79.94
16 kanałów	%I1—16	80.58
24 kanałów	%I1—24	81.56
32 kanałów	%I1—32	81.73
8+8 sąsiadujących kanałów	%I1—8 i %Q1—8	111.03
8 + 8 + 8 sąsiadujących kanałów	%I1—8, %Q1—8 i %M1—8	143.38
8 + 8 + 8 + 8 sąsiadujących kanałów	%I1—8, %Q1—8 i %M1—8 i %T1—8	175.79
<b>Dla nie sąsiadujących</b>		
	%I1, %M10, %Q3, etc.	
8 kanałów		299.64
16 kanałów		552.83
24 kanałów		806.35
32 kanałów		1059.85
<b>Reset</b>		
z 8 kanałami	—	162.63
z 16 kanałami	—	267.51
z 24 kanałami	—	372.73
z 32 kanałami	—	477.95

**Uwagi:** Jeżeli określono gniazdo w module wejść, każdy z czasów (**Dla sąsiadujących i nie sąsiadujących**) należy dodatkowo zwiększyć o 46  $\mu$ s.

W momencie wystąpienia wyzwolenia, należy dodatkowo dodać 29  $\mu$ s w przypadku korzystania z formatu BCD lub 148  $\mu$ s w przypadku korzystania z formatu POSIX.

Czasy podano dla kasowania oraz dla maksymalnej wielkości bufora – 1024 próbki. (Kasowanie powoduje usunięcie wszystkich próbek z bufora próbek).

Tabela A-4 Czasy wykonywania instrukcji – CPU 37x

Funkcja Grupa	Funkcja	Aktywny	Nieaktywny	Przyrost czasu	Rozmiar	
		37x	37x	37x		
Przełączniki czasowe	Przełącznik czasowy z pamięcią	4	5	–	15	
	Przełącznik	2	2	–	15	
	Przełącznik czasowy bez pamięci, z zanegowanym wejściem	3	2	–	15	
Liczniki	Licznik zliczający w górę	2	2	–	13	
	Licznik zliczający w dół	1	2	–	13	
Funkcje matematyczne	Dodawanie (INT)	1	0	–	13	
	Dodawanie (DINT)	2	0	–	19	
	Dodawanie (REAL)	5	0	–	17	
	Odejmowanie (INT)	1	0	–	13	
	Odejmowanie (DINT)	2	0	–	19	
	Odejmowanie (REAL)	5	0	–	17	
	Mnożenie (INT)	5	0	–	13	
	Mnożenie (DINT)	5	0	–	19	
	Mnożenie (REAL)	5	0	–	17	
	Dzielenie (INT)	5	0	–	13	
	Dzielenie (DINT)	5	0	–	19	
	Dzielenie (REAL)	5	0	–	17	
	Dzielenie Modulo (INT)	5	0	–	13	
	Dzielenie Modulo (DINT)	5	0	–	19	
	Pierwiastek kwadratowy (INT)	5	0	–	10	
	Pierwiastek kwadratowy (DINT)	10	0	–	13	
	Pierwiastek kwadratowy (REAL)	5	0	–	11	
	Funkcje trygonometryczne	SIN (REAL)	10	0	–	11
		COS (REAL)	10	0	–	11
TAN (REAL)		10	0	–	11	
ASIN (REAL)		10	0	–	11	
ACOS (REAL)		10	0	–	11	
ATAN (REAL)		5	0	–	11	
Funkcje logarytmiczne	LOG (REAL)	5	0	–	11	
	LN (REAL)	5	0	–	11	
Funkcje wykładnicze	EXP	10	0	–	11	
	EXPT	10	0	–	17	
Konwersja wartości kąta	Konwersja radianów na stopnie	5	0	–	11	
	Konwersja stopni na radiany	5	0	–	11	

- Uwagi:**
1. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/ bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  2. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  3. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modulem HSC.
  4. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  5. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.

Tabela A-4 Czasy wykonywania instrukcji – CPU37x - kontynuacja

Funkcja Grupa	Funkcja	Aktywny	Nieaktywny	Przyrost czasu	Rozmiar
		37x	37x	37x	
Relacje	Równe (INT)	1	0	–	10
	Równe (DINT)	2	0	–	16
	Równe (REAL)	5	0	–	14
	Różne (INT)	1	0	–	10
	Różne (DINT)	1	0	–	16
	Różne (REAL)	5	0	–	14
	Większe (INT)	1	0	–	10
	Większe (DINT)	1	0	–	16
	Większe (REAL)	5	0	–	14
	Większe lub równe (INT)	1	0	–	10
	Większe lub równe (DINT)	1	0	–	10
	Większe lub równe (REAL)	5	0	–	14
	Mniejsze (INT)	1	0	–	10
	Mniejsze (DINT)	1	0	–	16
	Mniejsze (REAL)	5	0	–	14
	Mniejsze lub równe (INT)	1	0	–	10
	Mniejsze lub równe (DINT)	3	0	–	16
	Mniejsze lub równe (REAL)	5	0	–	14
	Zakres (INT)	2	0	–	13
	Zakres (DINT)	2	0	–	22
Zakres (WORD)	1	0	–	13	
Bit Działanie	Logiczne AND	2	0	–	13
	Logiczne OR	2	0	–	13
	Alternatywa wyłączająca OR	1	0	–	13
	Negacja logiczna słowa bitowego NOT	1	0	–	10
	Przesunięcie słowa bitowego w lewo	5	0	1	16
	Przesunięcie słowa bitowego w prawo	5	0	1	16
	Przesunięcie słowa bitowego w lewo w obiegu zamkniętym	5	0	1	16
	Przesunięcie słowa bitowego w prawo w obiegu zamkniętym	5	0	1	16
	Pozycja bitu	5	0	–	13
	Ustawianie wartości pojedynczego bitu na 0	5	0	–	13
	Sprawdzanie wartości pojedynczego bitu	5	0	–	13
	Ustawianie wartości pojedynczego bitu na 1	5	0	–	13
	Porównanie z maskowaniem (WORD)	9	0	–	25
	Porównanie z maskowaniem (DWORD)	10	0	–	25

- Uwagi:**
1. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/ bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  2. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  3. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modułem HSC.
  4. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  5. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.

Tabela A-4 Czasy wykonywania instrukcji – CPU37x - kontynuacja

Funkcja Grupa	Funkcja	Aktywny	Nieaktywny	Przyrost czasu	Rozmiar
		37x	37x	37x	
Przemieszczenie danych	Przemieszczanie (INT)	2	0	1	10
	Przemieszczanie (BIT)	5	0	1	13
	Przemieszczanie (WORD)	2	0	1	10
	Przemieszczanie (REAL)	5	0	1	13
	Przemieszczanie grupy wartości (INT)	2	0	–	28
	Przemieszczanie grupy wartości (WORD)	3	0	–	28
	Przemieszczanie grupy wartości (REAL)	11	1	–	13
	Zerowanie fragmentu pamięci	1	0	1	11
	Rejestr przemieszczający bity (BIT)	10	0	1	16
	Rejestr przemieszczający słowa (WORD)	15	0	1	16
	Przemieszczanie jedyńki COMM_REQ	14	10	1	16
	COMM_REQ	200	200	–	13
Operacje	Kopiowanie danych				
	INT	10	0	1	22
	DINT	15	0	1	22
	BIT	10	0	1	22
	BYTE	10	0	1	22
	WORD	10	0	1	22
	Szukanie wartości zadanej				
	INT	5	0	1	19
	DINT	5	0	2	22
	BYTE	5	0	1	19
	WORD	5	0	1	19
	Szukanie wartości różnej				
	INT	5	0	1	19
	DINT	10	0	2	22
	BYTE	5	0	2	19
	WORD	5	0	2	19
	Szukanie wartości większej				
	INT	5	0	1	19
	DINT	5	0	2	22
	BYTE	10	0	1	19
	WORD	5	0	1	19
	Szukanie wartości większej lub równej				
	INT	5	0	1	19
	DINT	5	0	2	22
BYTE	5	0	1	19	
WORD	5	0	1	19	

- Uwagi:**
1. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  2. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  3. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modulem HSC.
  4. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  5. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.
  6. W przypadku instrukcji, dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.

Tabela A-4 Czasy wykonywania instrukcji – CPU37x - kontynuacja

Funkcja Grupa	Funkcja	Aktywny	Nieaktywny	Przyrost czasu	Rozmiar
		37x	37x	37x	
	Szukanie wartości mniejszej				
	INT	5	0	1	19
	DINT	10	0	2	22
	BYTE	5	0	1	19
	WORD	5	0	1	19
	Szukanie wartości mniejszej lub równej				
	INT	5	0	1	19
	DINT	5	0	2	22
Funkcje konwersji	Konwersja na dane typu BCD-4	5	0	–	10
	Konwersja na dane typu BCD	5	0	–	10
	Konwersja na dane typu REAL	5	0	–	8
	Konwersja na dane typu WORD	5	0	–	11
	Przybliżenie do liczby typu INT	5	0	–	11
	Przybliżenie do liczby typu DINT	5	0	–	11
Funkcje sterujące	Wywołanie podprogramu	15	0	–	7
	Natychmiastowe uaktualnienie stanu wejść/wyjść	5	0	–	13
	Algorytm PID-ISA	14	10	–	16
	Algorytm PID-IND	14	10	–	16
	Instrukcje END	–	–	–	–
	Funkcje specjalne sterownika SVCREQ				
	#6	5	0	–	10
	#7 (Odczyt)	10	0	–	10
	#7 (Ustawianie)	5	0	–	10
	#14	15	0	–	10
	#15	5	0	–	10
	#16	10	0	–	10
	#18	255	0	–	10
	#23	25	0	–	10
	#26//30**	155	0	–	10
	#29	5	0	–	10
	Para zagnieżdżonych instrukcji MCR/ENDMCR	1	0	–	4
Zdarzenie sekwencyjne Sekwencyjny rejestrator zdarzeń (SER) 8 kanałowy	60	0	=		
Zdarzenie sekwencyjne Sekwencyjny rejestrator zdarzeń (SER) 16 kanałowy	199	0	=		

\*\* Czas dla funkcji specjalnej #26/30 został zmierzony przy pomocy licznika o dużej prędkości, dla 16 punktowego modułu wyjściowego, zainstalowanego w 5 gnieździe.

- Uwagi:**
1. W przypadku funkcji wykonujących operacje na tablicach przyrost czasu podany jest w jednostkach o zadanej długości: dla funkcji do operacji bitowych w mikrosekundach/bit, a dla funkcji do przeprowadzania operacji na danych w mikrosekundach/ liczbę bitów lub słów.
  2. Czas w trybie Enabled podano dla modułów o pojedynczej długości typu %R, %AI i %AQ.
  3. Czas dla COMMREQ został zmierzony pomiędzy jednostką centralną a modulem HSC.
  4. W przypadku funkcji DOIO jest to czas wysyłania parametrów wyjściowych do modułu z wyjściami dyskretnymi.
  5. Jeżeli istnieje więcej niż jedna możliwość, podany czas jest czasem najmniej korzystnym.
  6. W przypadku instrukcji dla których w kolumnie Przyrost czasu podano wartość, wartość tę należy pomnożyć przez (Długość - 1), a następnie dodać ten wynik do czasu podstawowego.

## Czasy wykonywania dla przekaźników i styków

Zamieszczona poniżej tabela podaje czasy wykonywania dla przekaźników i styków w jednostkach centralnych serii 90-30.

**Tabela A-5 Czasy wykonywania dla przekaźników i styków**

Model jednostki centralnej	Czas wykonywania dla 1000 styków/ przekaźników logicznych
37x	0,15 ms
35x i 36x	0,22 ms
340/341	0,3 ms
331	0,4 ms
313/323	0,6 ms
311	18,0 ms

## Rozmiary instrukcji – CPU350 - 374

Rozmiary pamięci w poniższej tabeli odnoszą się do ilości bajtów wymaganej przez instrukcję w programie sterującym.

**Tabela A-6 Rozmiary instrukcji – CPU350 -374**

Funkcja	Rozmiar pamięci
Funkcja logiczna AND języka drabinkowego	1
Funkcja logiczna OR języka drabinkowego	1
Utworzenie duplikatu elementu na samej gorze stosu	1
Wypchnięcie jednej pozycji	1
Zainicjowanie stosu	1
Etykieta	5
Skok	5
Wszystkie pozostałe instrukcje	3
Bloki funkcyjne - proszę porównać z Tabelą A-2	różne





Sterowniki serii 90-30, serii 90-20 i serii 90-Micro posiadają dwie tabele błędów działania: jedna dla błędów działania układów wejść/wyjść, a druga dla wewnętrznych błędów działania sterownika. Informacje zawarte w niniejszym załączniku są pomocne przy analizowaniu formatu struktury komunikatów zapisanych w jednej z tych dwóch tabel. Obydwie tabele zawierają podobne dane. Zawartość tych tabel zapisana jest tylko w sterowniku. Dlatego wykorzystując oprogramowanie należy nawiązać komunikację ze sterownikiem (tryb ONLINE lub MONITOR).

- W tabeli błędów działania sterownika zapisane są:
  - Miejsce wystąpienia błędu
  - Opis błędu
  - Data i czas wystąpienia błędu
- W tabeli błędów działania układów wejść/wyjść zapisane są:
  - Miejsce wystąpienia błędu
  - Adres elementu
  - Kategoria błędu
  - Typ błędu
  - Data i czas wystąpienia błędu

## **Tabela błędów działania sterownika**

Tabela błędów działania sterownika może być odczytywana za pomocą oprogramowania. Sposób odczytywania tablicy błędów z poziomu oprogramowania opisano w pomocy komputerowej oraz w podręczniku *Logicmaster 90 Series 90-30/20/Micro Programming Software User's Manual*, GFK-0466.



Poniżej opisano każde z pól tego komunikatu. Podano również dopuszczalne wartości każdego z tych pól.

### **Wskaźnik długości bloku szczegółowych informacji o błędzie**

Bajt ten oznacza czy blok dodatkowych informacji o błędzie ma 8 czy 24 bity.

<b>Typ</b>	<b>Kod</b>	<b>Szczegółowe informacje o błędzie</b>
Short Integer	00	8 bajtów
Długie	01	24 bajtów

### **Pole wolne**

Tych sześć bajtów ma na celu zapewnienie takiej samej długości tablicy błędów działania sterownika, jak długość tabeli błędów działania układów wejść/wyjść.

### **Kaseta**

Wartość od 0 do 7. 0 oznacza kasetę główną sterownika. Kasety 1 do 7 są kasetami rozszerzającymi, podłączanymi do sterownika za pomocą kabla rozszerzającego.

### **Gniazdo**

Gniazda numerowane są od 0 do 9. Jednostka centralna jest zawsze zamontowana w gnieździe 1 (w kasecie 0).

### **Zadanie**

Zadania numerowane są od 0 do +65 535. Czasami numer zadania wykorzystywany jest przez użytkownika w celu uzyskania dodatkowych informacji o błędzie.

## Grupy błędów działania sterownika

Grupa błędów jest pierwszym kryterium do klasyfikacji błędów. Podaje ona ogólną kategorię do której błąd należy. Tekst objaśniający błąd, wyświetlany przez oprogramowanie LogiMaster 90-30/20/Micro, zależy właśnie od grupy błędów i kodów błędów.

W tabeli B-1 podano wszystkie grupy błędów, które można wyróżnić w działaniu sterownika.

Ostatnia grupa, bez przyporządkowanego numeru, Błąd niesklasyfikowany została zadeklarowana w celu obsługi nowych błędów w systemie, dla których sterownik nie zna kodu. Wszystkie nierozpoznane błędy są klasyfikowane do tej grupy.

**Tabela B-1 Grupy błędów działania sterownika:**

Numer grupy		Nazwa grupy	Waga błędów
Dziesiętne	Heksadecymalne		
1	1	Loss of, or missing, rack	Krytyczny
4	4	Loss of, or missing, option module	Diagnostyczny
5	5	Addition of, or extra, rack	Diagnostyczny
8	8	Addition of, or extra, option module	Diagnostyczny
11	B	System configuration mismatch	Krytyczny
12	C	System bus error	Diagnostyczny
13	D	PLC CPU hardware failure	Krytyczny
14	E	Non-fatal module hardware failure	Diagnostyczny
16	10	Option module software failure	Diagnostyczny
17	11	Program block checksum failure	Krytyczny
18	12	Low battery signal	Diagnostyczny
19	13	Constant sweep time exceeded	Diagnostyczny
20	14	PLC system fault table full	Diagnostyczny
21	15	I/O fault table full	Diagnostyczny
22	16	User Application fault	Diagnostyczny
-	-	Additional PLC fault codes	Zgodnie z informacjami
128	80	System bus failure	Krytyczny
129	81	No user's program on power-up	Informacyjny
130	82	Corrupted user RAM detected	Krytyczny
132	84	Password access failure	Informacyjny
135	87	PLC CPU software failure	Krytyczny
137	89	PLC sequence-store failure	Krytyczny

## Waga błędu

Można wyróżnić trzy rodzaje wag błędów. Wagi przyporządkowane do błędów w sterownikach serii 90-30 nie mogą być zmieniane przez użytkownika.

Tabela B-2 Tabela wag błędów działania sterownika:

Waga błędu	Działania realizowane przez jednostk centraln	Kod
Informacyjny	Zarejestrowanie błędu w tabeli błędów	1
Diagnostyczny	Zarejestrowanie błędu w tabeli błędów Ustawienie zmiennych systemowych informujących o wystąpieniu błędu	2
Krytyczny	Zarejestrowanie błędu w tabeli błędów Ustawienie zmiennych systemowych informujących o wystąpieniu błędu Przejsście do trybu STOP	3

## Kod błędu

Kod błędu zawiera dalsze informacje o błędzie. Każda z grup błędów posiada swój własny zestaw kodów błędów. W Tabeli B-3 zestawiono kody błędów dla grupy Uszkodzone oprogramowanie jednostki centralnej (Grupa 87H)

Tabela B-3 Kody błędów alarmowych dla grupy PLC CPU Software

Dziesiąte	Heksadecym.	Nazwa
20	14	Uszkodzenie pamięci programu sterującego
39	27	Uszkodzenie pamięci programu sterującego
82	52	Błąd transmisji przez magistralę komunikacyjną
90	5A	Zatrzymanie na żądanie użytkownika
Wszystkie pozostałe		Błąd jednostki centralnej

W Tabeli B-4 zamieszczono kody błędów dla pozostałych grup.

Tabela B-4 Kody błędów alarmowych dla sterownika

Dziesiętnie	Heksadecym.	Nazwa
<i>Kody błędów sterownika dla grupy Loss of Option Module (Brak modułu dodatkowego)(4)</i>		
44	2C	Nieudane ponowne uruchomienie modułu dodatkowego
45	2D	Nieudane ponowne uruchomienie modułu dodatkowego
255	FF	Błąd w komunikacji z modułem dodatkowym
79	4F	Brak karty rozszerzeń
<i>Kody błędów dla grupy Reset of, Addition of, or Extra Option Module (Ponowne uruchomienie, dodanie modułu lub dodatkowy, wyspecjalizowany moduł (8)</i>		
2	2	Zakończenie ponownego uruchamiania modułu
04	4	Niezadeklarowana karta rozszerzeń
05	5	Ponowne uruchomienie kraty rozszerzeń
	Wszystkie pozostałe	Ponowne uruchomienie, dodanie lub brak konfiguracji dla modułu dodatkowego
<i>Kody błędów dla grupy Option Module Software Failure (Uszkodzenie oprogramowania modułu dodatkowego (10)</i>		
1	1	Nieobsługiwany typ płyty
2	2	Przepełnienie skrzynki pocztowej COMREQ dla komunikatu wyjściowego powodującego wywołanie funkcji COMREQ
3	3	Skrzynka pocztowa COMREQ przepełniona w czasie wysyłania odpowiedzi
5	5	Magistrala komunikacyjna sterownika; nie zrealizowane żądanie
11	B	Błąd zasobów (alokacja pamięci, przepełnienie tabeli itp.)
13	D	Błąd w programie sterującym użytkownika
401	191	Uszkodzenie oprogramowania modułu, żądanie ponownego wczytania
<i>Kody błędów dla grupy System Configuration Mismatch (Błędna konfiguracja systemu (B)</i>		
8	8	Błąd w konfiguracji dodatkowych modułów analogowych
10	A	Nie obsługiwana funkcja
23	17	Brak pamięci dla programu sterującego
58	3A	Błąd konfiguracji karty rozszerzeń
<i>Kody błędów dla grupy System Bus Error (Błąd magistrali systemowej) (C)</i>		
	Wszystkie pozostałe	Błąd magistrali systemowej
<i>Kody błędów dla grupy Program Block Checksum (Błędna suma kontrolna programu sterującego) (11)</i>		
3	3	Błędna suma kontrolna dla programu lub bloku programu
<i>Kody błędów dla grupy Low Battery Signal (Rozładowane baterie)</i>		
0	0	Uszkodzone baterie jednostki centralnej sterownika lub innego modułu
1	1	Rozładowane baterie jednostki centralnej sterownika lub innego modułu
<i>Kody błędów dla grupy Application Fault (Uszkodzenie programu sterującego użytkownika)(16)</i>		
2	2	Przekroczenie czasu przez zegar alarmowy sterownika
5	5	Dana instrukcja nie może pracować w trybie COMREQ – WAIT
6	6	COMREQ –Niewłaściwy identyfikator ID zadania
7	7	Przepełnienie stosu aplikacji
<i>Kody błędów dla grupy System Bus Failure Group (Błąd magistrali systemowej) (80)</i>		
1	1	System operacyjny
<i>Kody błędów dla grupy Corrupted User RAM on Powerup (Wykrycie w czasie rozruchu uszkodzenia pamięci RAM użytkownika) (82)</i>		
1	1	Uszkodzenie pamięci RAM użytkownika w czasie rozruchu
2	2	Wykryty niewłaściwy kod logiczny Opcode
3	3	PLC_ISCP_PC_OVERFLOW
4	4	PRG_SYNTAX_ERR
<i>Kody błędów dla grupy PLC CPU Hardware Faults (Uszkodzenie sprzętowe jednostki centralnej sterownika) (D)</i>		
	Wszystkie pozostałe	Uszkodzenie sprzętowe jednostki centralnej

## Szczegółowe informacje o błędzie

W polu tym zapisane są szczegółowe informacje o błędzie. W poniższym przykładzie pokazano sposób prezentacji błędów:

### Przykład – Błąd pamięci RAM

Dla czterech kodów błędów, z grupy Błędna konfiguracja systemu podawane są dodatkowe informacje:

Tabela B-5 Informacje o błędzie działania sterownika - Wykryty niewłaściwy kod logiczny OPCODE

Szczegółowe informacje o błędzie	Niewłaściwy numer modelu
[0]	Zawartość rejestru błędów ISCP
[1]	Niewłaściwy kod OPCODE
[2,3]	Licznik programu ISCP
[4,5]	Numer funkcji

W przypadku wykrycia uszkodzenia pamięci RAM jednostki centralnej sterownika (jedno z pierwszych uszkodzeń, podawanych w grupie Uszkodzenia sprzętowe jednostki centralnej sterownika), w pierwszych czterech bajtach tego pola zapamiętywany jest adres wystąpienia uszkodzenia.

## Czas i data wystąpienia błędu działania sterownika

### Uszkodzenie jednostki centralnej (błąd pamięci RAM)

Czas i data wystąpienia błędu działania sterownika zapisywane są w sześciu bajtach. (Wartości są przedstawiane w kodzie BCD).

Tabela B-6 Czas i data wystąpienia błędu działania sterownika

Numer bajtu	Opis
1	Sekundy
2	Minuty
3	Godziny
4	Dzień miesiąca
5	Miesiąc
6	Rok





Poniżej opisano każde z pól tego komunikatu. Podano również dopuszczalne wartości każdego z tych pól.

### Wskaźnik długości bloku szczegółowych informacji o błędzie

Bajt ten informuje o długości bloku szczegółowych informacji o błędzie, 5 bajtów lub 21 bajtów.

Tabela B-7 Długość bloku szczegółowych informacji o błędzie

Typ	Kod	Długość bloku szczegółowych informacji o błędzie
Krótki	02	5 bajtów
Długi	03	21 bajtów

### Adres zmiennej

Adres elementu o długości trzech bajtów jest adresem pamięci wejść/wyjść oraz pozycji (lub przemieszczenia) odpowiadającej punktowi, dla którego stwierdzony został błąd. W przypadku jeżeli uszkodzenie wystąpiło w bloku Genius PAGE Genius block fault lub w zintegrowanym module analogowym, adres elementu odnosi się do pierwszego punktu bloku, w którym stwierdzono uszkodzenie.

Tabela B-8 Adres elementu wejść/wyjść

Bajt	Opis	Przedział
0	Typ pamięci	0 –FF
1–2	Pozycja	0 –7FF

Bajt określający typ pamięci może przyjmować wartości podane w zamieszczonej poniżej tabeli.

Tabela B-9 Typ pamięci adresu elementu wejść/wyjść

Nazwa	Wartość (Heksadecymalnie)
Wejście analogowe	0A
Wyjście analogowe	0C
Zgrupowane wejścia analogowe	0D
Wejście dyskretne	10 lub 46
Wyjście dyskretne	12 lub 48
Zgrupowane wyjścia dyskretne	1F

### Adres wystąpienia błędu działania układów wejść/wyjść

Adres wystąpienia błędu działania układów wejść/wyjść jest adresem o długości sześciu bajtów określającym kasetę, gniazdo, magistralę, blok i adres punktu wejścia/ wyjścia dla którego stwierdzone zostało uszkodzenie. Adres punktu jest słowem, wszystkie pozostałe adresy mają długość jednego bajtu. Dla danego uszkodzenia mogą nie być podawane wszystkie pięć wartości.

Jeżeli adres wystąpienia błędu w działaniu układów wejść/wyjść nie zawiera wszystkich pięciu adresów, miejsce do którego podawane są prawidłowe wartości oznaczone jest przez wprowadzenie liczby heksadecymalnej 7F. Przykładowo, jeżeli bajt magistrali ma wartość 7F, zlokalizowano uszkodzenie w module. Podawane są więc wyłącznie poprawne wartości dla kasyety i gniazda.

## Kaseta

Wartość od 0 do 7. 0 oznacza kasetę główną sterownika tj. kasetę z jednostką centralną. Kasety o numerach 1 do 7 są kasetami rozszerzającymi.

## Gniazdo

Gniazda numerowane są od 0 do 9. Jednostka centralna jest zawsze zamontowana w gnieździe 1 (w kasecie 0).

## Punkt

Punkty mogą mieć przyporządkowane numery od 1 do 1024 (w systemie dziesiętnym). Podana wartość informuje który z punktów w bloku został uszkodzony, o ile jest to jest to uszkodzenie z typu uszkodzenie elementu.

## Grupy błędów działania układów wejść/wyjść

Grupa błędu jest pierwszym kryterium do klasyfikacji błędu. Podaje ona ogólną kategorię do której błąd należy. Tekst objaśniający błąd, wyświetlany przez oprogramowanie Logicmaster 90-30/20/Micro, zależy właśnie od grupy błędu i kodów błędu.

W tabeli B-10 podano wszystkie grupy błędów działania układów wejść/wyjść. Grupy o numerach mniejszych od 80 (heksadecymalnie) zawierają znane błędy.

Ostatnia grupa, bez przyporządkowanego numeru, *Nie sklasyfikowany błąd działania sterownika* została zadeklarowana w celu obsługi nowych błędów w systemie, dla których sterownik nie zna kodu. Wszystkie nierozpoznane błędy są klasyfikowane do tej grupy.

Tabela B-10 Grupy błędów działania układów wejść/wyjść

Numer grupy	Nazwa grupy	Waga błędu
3	Uszkodzenie lub brak modułu wejść/wyjść	Diagnostyczny
7	Nie skonfigurowany lub nowy moduł wejść/wyjść.	Diagnostyczny
9	Uszkodzenie kontrolera magistrali lub magistrali wejść/wyjść	Diagnostyczny
A	Uszkodzenie modułu wejść/wyjść	Diagnostyczny
–	Nie sklasyfikowany błąd działania układów wejść/wyjść	Zgodnie z informacjami

## Wagi błędów działania układów wejść/wyjść

Zakres działań podejmowanych w przypadku wystąpienia błędu zależy od wagi tego błędu. Wagi błędów wymieniono w Tabelicy B-11.

Tabela B-11 Wagi błędów działania układów wejść/wyjść

Waga błędu	Działania realizowane przez jednostkę centralną	Kod
Informacyjny	Zarejestrowanie błędu w tabeli błędów	1
Diagnostyczny	Zarejestrowanie błędu w tabeli błędów Ustawienie zmiennych systemowych informujących o wystąpieniu błędu	2
Krytyczny	Zarejestrowanie błędu w tabeli błędów Ustawienie zmiennych systemowych informujących o wystąpieniu błędu Przejsie do trybu STOP	3

## Blok szczegółowych informacji o błędach działania układów wejść/wyjść

Każda z tabel błędów działania układów wejść/wyjść może zawierać do 5 bajtów dodatkowych informacji o zaistniałym błędzie.

## Symboliczne informacje o błędach

W tabeli B-12 wymieniono dane, wymagane do konfigurowania obwodu bloku.

Tabela B-12 Blok szczegółowych informacji o błędach działania układów wejść/wyjść

Wartość dziesiętna	Heksadecy malnie	Opis
<i>Konfiguracja obwodu</i>		
	1	Obwód jest wejściem trójstanowym
	2	Obwód jest wejściem
	3	Obwód jest wyjściem

## Wagi błędów symbolicznych

Wymuszenie obwodów jest zapisywane do tablicy błędów jako błąd informacyjny. Wszystkie pozostałe błędy są błędami diagnostycznymi lub krytycznymi.

Błędy: niezgodność numeru modułu, niezgodność typu wejść/wyjść oraz brak modułów wejść/wyjść są zapisywane w tabeli błędów działania sterownika, w grupie Błędna konfiguracja systemu. Nie są one zapisywane w tabeli błędów działania układów wejść/wyjść.

## Czas zarejestrowania błędu

Czas i data wystąpienia błędu działania sterownika zapisywane są w sześciu bajtach. Wartości są podane w kodzie BCD.

Tabela B-13 Czas i data wystąpienia błędu działania układów wejść/wyjść

Numer bajtu	Opis
1	Sekundy
2	Minuty
3	Godziny
4	Dzień miesiąca
5	Miesiąc
6	Rok

Wprowadzenie znaku & i następnie nazwy mnemonicznej instrukcji umożliwia szybkie wpisanie czy odszukanie instrukcji w czasie edycji programu sterującego. W przypadku niektórych instrukcji, można również podać adres zmiennej lub nazwę pomocniczą, etykietę lub adres położenia zmiennej.

W niniejszym załączniku wyszczególniono nazwy mnemoniczne dla instrukcji oprogramowania Logicmaster 90-30/20/Micro. Pełne nazwy mnemoniczne podane są w 3 kolumnie tej tabeli, natomiast najkrótsze nazwy, które mogą być wprowadzane dla każdej z instrukcji podano w kolumnie 4.

W dowolnym momencie pracy z oprogramowaniem można wyświetlić ekran z pomocą, zawierający te nazwy mnemoniczne. należy w tym celu wcisnąć klawisze ALT i I.

Grupa funkcji	Instrukcja	Nazwa mnemoniczna						
		Wszystkie	INT	DINT	BIT	BYTE	WORD	REAL
Styki	Dowolny styk	&CON	&CON					
	Styk normalnie otwarty	&NOCON	&NOCON					
	Styk normalnie zwarty	&NCCON	&NCCON					
	Styk kontynuacji	&CONC	&CONC					
Przełączniki	Dowolny przełącznik	&COI	&COI					
	Przełącznik o stykach otwartych	&NOCOI	&NOCOI					
	Przełącznik o stykach zamkniętych	&NCCOI	&NCCOI					
	Przełącznik uaktywniany z boczem narastającym sygnału	&PCOI	&PCOI					
	Przełącznik uaktywniany z boczem opadającym sygnału	&NCOI	&NCOI					
	Przełącznik ustawialny SET	&SL	&SL					
	Przełącznik ustawialny RESET	&RL	&RL					
	Przełącznik SET z pamięcią	&SM	&SM					
	Przełącznik RESET z pamięcią	&RM	&RM					
	Przełącznik o stykach otwartych z pamięcią	&NOM	&NOM					
	Przełącznik o stykach zamkniętych, z pamięcią	&NCM	&NCM					
	Przełącznik kontynuacji	&COILC	&COILC					
	Połączenia	Połączenie poziome	&HO	&HO				
Połączenie pionowe		&VE	&VE					
Przełączniki czasowe	Przełącznik czasowy z pamięcią	&ON	&ON					
	Przełącznik czasowy z zanegowanym wejściem, bez pamięci	&TM &OF	&TM &OF					
Liczniki	Licznik zliczający w górę	&UP	&UP					
	Licznik zliczający w dół	&DN	&DN					



Grupa funkcji	Instrukcja	Nazwa mnemoniczna							
		Wszystkie	BCD-4	INT	DINT	BIT	BYTE	WORD	REAL
Funkcje matematyczne	Dodawanie Odejmowanie Mnożenie Dzielenie bez reszty Dzielenie modulo Pierwiastek kwadratowy Sinus (SIN) Cosinus Tangens (TAN) Arcus sinus Arcus cosinus Arcus tangens Logarytm dziesiętny Logarytm naturalny (LN) Potęga liczby e Potęga liczby x	&AD &SUB &MUL &DIV &MOD &SQ &SIN &COS &TAN &ASIN &ACOS &ATAN &LOG &LN &EXP &EXPT		&AD_I &SUB_I &MUL_I &DIV_I &MOD_I &SQ_I	&AD_DI &SUB_DI &MUL_DI &DIV_DI &MOD_DI &SQ_DI				&AD_R &SUB_R &MUL_R &DIV_R &MOD_R&SQ_R
Relacje	Równy Różny Większy Większy lub równy Mniejszy Mniejszy lub równy	&EQ &NE &GT &GE &LT &LE		&EQ_I &NE_I &GT_I &GE_I &LT_I &LE_I	&EQ_DI &NE_DI &GT_DI &GE_DI &LT_DI &LE_DI			&EQ_R &NE_R &GT_R &GE_R &LT_R &LE_R	
Operacje bitowe	AND OR Alternatywa wyłączająca OR NOT Przesunięcie słowa bitowego w lewo Przesunięcie słowa bitowego w prawo Przesunięcie słowa bitowego w lewo w obiegu zamkniętym Przesunięcie słowa bitowego w prawo w obiegu zamkniętym Sprawdzanie wartości pojedynczego bitu Ustawianie wartości pojedynczego bitu na 1 Ustawianie wartości pojedynczego bitu na 0 Pozycja bitu Porównanie z maskowaniem	&AN &OR &XO &NOT &SHL &SHR &ROL &ROR &BT &BS &BCL &BP &MCOMP					&AN_W &OR_W &XO_W &NOT_W &SHL_W &SHR_W &ROL_W &ROR_W &BT_W &BS_W &BCL_W &BP_W &MCM_W		
Funkcje konwersji	Konwersja na dane typu INT Konwersja na dane typu DINT Konwersja danych typu BCD na dane typu INT Konwersja na dane typu REAL Konwersja na dane typu WORD Przybliżenie do liczby typu INT Przybliżenie do liczby typu DINT	&TO_INT &TO_DINT &BCD4 &TO_REAL &TO_W &TRINT &TRDINT	&TO_INT_BCD4		&TO_REAL_DI		&TO_REAL_W	&BCD4_R	

Grupa funkcji	Instrukcja	Nazwa mnemoniczna						
		Wszystkie	INT	DINT	BIT	BYTE	WORD	REAL
Przemieszczenie danych	Przemieszczenie Przemieszczanie bloku Zerowanie fragmentu pamięci Rejestr przemieszczający Przemieszczenie jedynek Inicjalizacja komunikacji z jednym z modułów sterownika	&MOV &BLKM &BLKC &SHF &BI &COMMR	&MOV_I &BLKM_I		&MOV_BI  &SHF_BI		&MOV_W &BLKM_W  &AR_W	&MOV_R &BLKM_R
Operacje tablicowe	Kopiowanie danych Szukanie wartości zadanej Szukanie wartości różnej Szukanie wartości większej Szukanie wartości większej lub równej Szukanie wartości mniejszej Szukanie wartości mniejszej lub równej	&AR &SRCHE &SRCHN &SRCHGT &SRCHGE &SRCHLT &SRCHLE	&AR_I &SRCHE_I &SRCHN_I &SRCHGT_I &SRCHGE_I &SRCHLT_I &SRCHLE_I	&AR_DI &SRCHE_DI &SRCHN_DI &SRCHGT_DI &SRCHGE_DI &SRCHLT_DI &SRCHLE_DI	&AR_BI	&AR_BY &SRCHE_BY &SRCHN_BY &SRCHGT_BY &SRCHGE_BY &SRCHLT_BY &SRCHLE_BY	&AR_W &SRCHE_W &SRCHN_W &SRCHGT_W &SRCHGE_W &SRCHLT_W &SRCHLE_W	
Funkcje sterujące	Wywołanie podprogramu Natychmiastowe uaktualnienie stanu wejść/wyjść SER Algorytm PID-ISA Algorytm PID-IND Zerowanie SFC Koniec Komentarz Funkcje SVCREQ Początek fragmentu programu sterującego wykonywanego bez dopływu sygnału Koniec fragmentu programu sterującego, wykonywanego bez doprowadzania sygnału MCRN ENDMCRN Skok Zagnieżdżona instrukcja JUMP: Etykieta Zagnieżdżona instrukcja Label	&CA &DO &SER &PIDIS &PIDIN &SFCR &END &COMME &SV &MCR &ENDMCR &MCRN &ENDMCRN &JUMP &JUMPN &LABEL &LABELN						





W niniejszym załączniku opisano skróty klawiszowe wykorzystywane w oprogramowaniu. Podane tu informacje mogą zostać również wyświetlone na ekranie. Należy w tym celu wcisnąć klawisze ALT-K.

Skróty klawiaturowe	Opis	Skróty klawiaturowe	Opis
<i>Klawisze dostępne z poziomu całego pakietu oprogramowania</i>			
ALT-A	Przerwij	CTRL-Break	Zamknięcie programu
ALT-C	Wymazanie zawartości pola	Esc	Powiększenie
ALT-M	Zmiana trybu programatora	CTRL-Home	Przywołanie poprzedniej linii poleceń
ALT-R	Zmiana trybu pracy sterownika (Run/ Stop)	CTRL-End	Przywołanie następnej linii poleceń
ALT-E	Wyświetlenie/ ukrycie	CTRL- ←	Przesunięcie kursora w lewo w obrębie pola
ALT-J	Wyświetlenie/ ukrycie linii poleceń	CTRL- →	Przesunięcie kursora w prawo w obrębie pola
ALT-L	Wyświetlenie zawartości kartoteki	CTRL-D	Zmniejszenie adresu zmiennej
ALT-P	Wydruk ekranu	CTRL-U	Zwiększenie adresu zmiennej
ALT-H	Pomoc	Tabulator	Zmiana/zwiększenie wartości pola
ALT-K	Pomoc na temat skrótów klawiszowych	Shift-Tab	Zmiana/zmniejszenie wartości pola
ALT-I	Pomoc na temat nazw mnemonicicznych	Enter	Zaakceptowanie wartości wprowadzonej w polu
ALT-N	Przełączenie opcji wyświetlania	CTRL-E	Wyświetlenie ostatniego błędu systemowego
ALT-T	Włączenie trybu uczenia (Teach)	F12 lub szary minus (-)	Zmiana wartości zmiennej dyskretnej
ALT-Q	Wyłączenie trybu uczenia (Teach)	F11 lub szary klawisz mnożenia (*)	Wymuszona zmiana wartości zmiennej dyskretnej
ALT-N	Odtworzenie pliku n (n=0 do 9)		
<i>Skróty klawiszowe dostępne wyłącznie w edytorze programu</i>			
ALT-B	Włączenie/ wyłączenie sygnału dźwiękowego edytora	Szary klawisz dodawania (+)	Akceptacja szczebla
ALT-D	Usunięcie elementu szczebla/ Usunięcia szczebla	Enter	Akceptacja szczebla
ALT-S	Zapisanie bloku w sterowniku i na dysku	CTRL-PgUp	Poprzedni szczebel
ALT-X	Wyświetlenie współczynnika powiększenia	CTRL-PgDn	Następny szczebel
ALT-U	Uaktualnienie pliku na dysku	~	Połączenie poziome
ALT-V	Wyświetlenie okna z tabelą zmiennych		Połączenie pionowe
ALT-F2	Przejdź do tabeli zmiennych	Tabulator	Przejdź do następnego parametru
<i>Specjalne skróty klawiszowe</i>			
ALT-O	Zmiana hasła. Skróót dostępny wyłącznie w oknie dialogowym do zmiany hasła w oprogramowaniu do konfiguracji.		

Karta z pomocą zamieszczona na następnej stronie skrótów klawiszowe i nazwy mnemoniczne dla oprogramowania Logicmaster 90-30/20/Micro. Karta ta sporządzona jest w trzech kopiach oraz posiada perforowane krawędzie, co ułatwia odłączenie jej od podręcznika.

Przy korzystaniu z liczb zmiennoprzecinkowych należy pamiętać o kilku zasadach. W pierwszym punkcie omówiono zagadnienia ogólne. Instrukcje do wprowadzania i wyświetlania liczb zmiennoprzecinkowych podano na stronie E-4 i stronach następnych.

### Uwaga

Operacje zmiennoprzecinkowe dostępne są **wyłącznie** w jednostkach centralnych 35x i 36x, wersja 9,00 lub nowsza oraz we wszystkich wersjach jednostek centralnych CPU352 i CPU37x.

## Liczby zmiennoprzecinkowe

Oprogramowanie pozwala na edycję, wyświetlanie, zapisywanie oraz przeglądanie liczb zmiennoprzecinkowych. Liczby rzeczywiste są również parametrami wywołania niektórych funkcji. Należy jednak pamiętać, że warunkiem korzystania w oprogramowaniu z liczb zmiennoprzecinkowych jest posiadanie odpowiedniej jednostki centralnej (proszę porównać z zamieszczoną powyżej uwagą). Liczby zmiennoprzecinkowe reprezentowane są w notacji dziesiętnej, z wyświetlaniem sześciu znaczących cyfr.

### Uwaga

Terminy "liczby zmiennoprzecinkowe" i "liczby rzeczywiste" są w niniejszym podręczniku używane wymiennie, oznaczają ten sam typ danych.

Stosowany jest następujący format. Dla liczb z zakresu 9 9999999 do 0.0001 na wyświetlaczu nie jest pokazywany wykładnik, a liczba cyfr znaczących wynosi sześć lub siedem. Przykładowo:

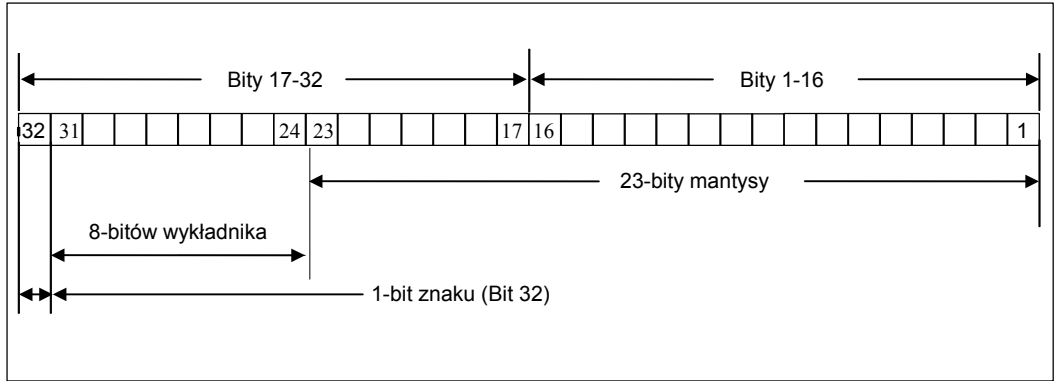
Wartość	Format wyświetlania	Opis
.000123456789	+0001234567	Dziesięć cyfr, sześć lub siedem cyfr znaczących
-12.345e-2	-.1234500	Siedem cyfr, sześć lub siedem cyfr znaczących
1234	+1234.000	Siedem cyfr, sześć lub siedem cyfr znaczących

Jeżeli wartość nie mieści się w podanym powyżej przedziale, wyświetlanych jest jedynie sześć cyfr znaczących, w następującym formacie: +1.23456E+12

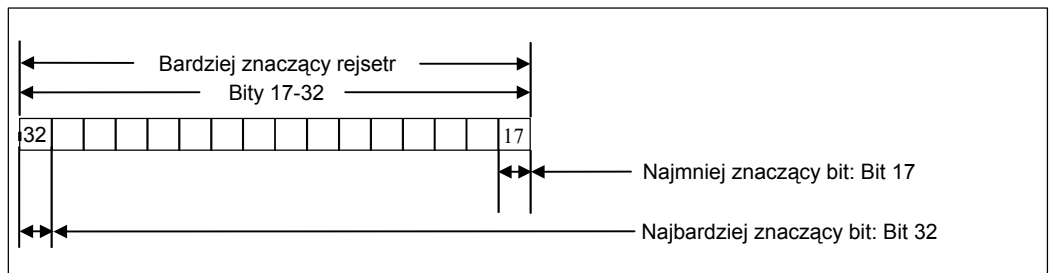
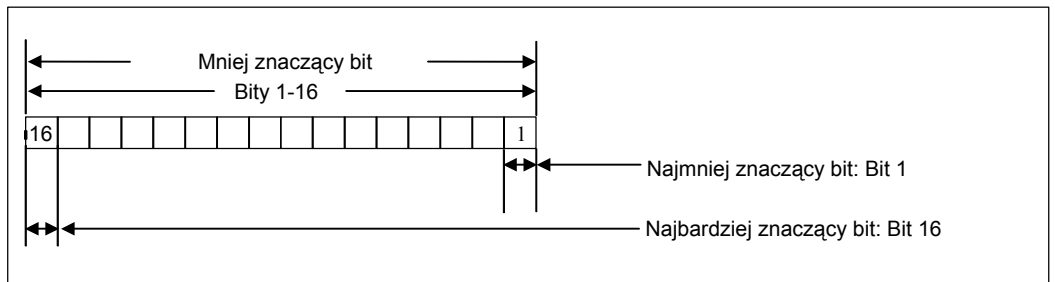


### Format wewnętrznej reprezentacji liczb zmiennoprzecinkowych

Liczby zmiennoprzecinkowe pamiętane są w standardowym formacie IEEE pojedynczej precyzji. Do reprezentacji tych liczb wymagane są 32 bity, czyli dwa (sąsiednie) 16 bitowe rejestry sterownika. Sposób kodowania tych bitów podano na zamieszczonym poniżej schemacie.



Kolejny schemat pokazuje wykorzystanie rejestrów przez liczbę zmiennoprzecinkowa pojedynczej precyzji. Przykładowo, jeżeli liczba zmiennoprzecinkowa zajmuje rejestry %R0005 i %R0006, rejestr %R0005 jest rejestrem mniej znaczącym, a rejestr %R0006 rejestrem bardziej znaczącym.



## Wartości liczb zmiennoprzecinkowych

Zamieszczona poniżej tabela umożliwia obliczenie wartości liczby zmiennoprzecinkowej na podstawie zawartości dwóch sąsiadujących ze sobą rejestrów.

Wykładnik (e)	Mantysa (f)	Wartość liczby zmiennoprzecinkowej
255	Różna od zera	Niepoprawna wartość
255	0	$-1^s * \infty$
$0 < e < 255$	Dowolna wartość	$-1^s * 2^{e-127} * 1.f$
0	Różna od zera	$-1^s * 2^{-126} * 0.f$
0	0	0

f = Mantysa: Mantysa jest funkcją binarną.

e = Wykładnik. Wykładnik jest liczbą całkowitą E taką że: E+127 jest potęgą liczby 2 przez którą musi być pomnożona mantysa w celu otrzymania wartości zmiennoprzecinkowej.

s = bit znaku.

\* = operator mnożenia.

Przykładowo, zanalizowana zostanie reprezentacja liczby zmiennoprzecinkowej 12.5. Reprezentacja zmiennoprzecinkowa tej wartości, zgodnie ze standardem IEEE, wygląda następująco:

01000001 01001000 00000000 00000000

lub 41480000 (heksadecymalnie). Najbardziej znaczący bit (bit znaku) jest równy zero (s=0). Następnym osiem najbardziej znaczących bitów jest równie 10000010, czyli 130 w systemie dziesiętnym (e=130).

Mantysa pamiętana jest jako dziesiętna liczba binarna, z kropką dziesiętną poprzedzającą najbardziej znaczące 23 bity. Tak więc, najbardziej znaczący bit w mantysie jest wielokrotnością  $2^{-1}$ , następny najbardziej znaczący bit jest wielokrotnością  $2^{-2}$ , itd., do najmniej znaczącego bitu, który jest wielokrotnością  $2^{-23}$ . 23 bity mantysy wyglądają następująco:

1001000 00000000 00000000

Tak więc wartość mantysy wynosi w takim przypadku .5625 (tzn.  $2^{-1} + 2^{-4}$ )

Ponieważ  $e > 0$  i  $e < 255$ , zastosowana zostanie trzecia formuła z podanej powyżej tabeli:

$$\begin{aligned}
 \text{wartość} &= -1^s * 2^{e-127} * 1.f \\
 &= -1^0 * 2^{130-127} * 1.5625 \\
 &= 1 * 2^3 * 1.5625 \\
 &= 8 * 1.5625 \\
 &= 12.5
 \end{aligned}$$

Przedstawiona powyżej analiza udowadnia poprawność tej reprezentacji binarnej.

Zakres wartości, które mogą być reprezentowane w ten sposób wynosi:  $\pm 1.401298E-45$  do  $\pm 3.402823E+38$ .

## Wprowadzanie i wyświetlanie liczb zmiennoprzecinkowych

Dla mantysy można wprowadzić i zapamiętać sześć lub siedem znaczących cyfr dokładności, nie mniej jednak oprogramowanie wyświetla wyłącznie pierwszych sześć cyfr mantysy. Mantysa może być poprzedzona znakiem dodatnim lub ujemnym. Jeżeli znak nie zostanie wprowadzony, przyjmowane jest domyślnie, że jest to znak dodatni.

Po wykładniku należy wprowadzić literę **E** lub **e**, a mantysa musi zawierać kropkę dziesiętną co pozwoli na uniknięcie pomylenia jej z liczbą heksadecymalną. Należy również podać znak wykładnika, jeżeli jednak nie zostanie on podany, domyślnie przyjmowane jest, że jest on dodatni. Jeżeli wykładnik nie zostanie wprowadzony, przyjmuje się domyślnie, że jest on równy zero. Nie dopuszcza się stosowania znaku spacji w liczbach zmiennoprzecinkowych.

W celu zwiększenia łatwości korzystania, zarówno w linii poleceń jak i w polach danych dopuszczalne jest stosowanie kilku różnych formatów. Dopuszczalne jest między innymi wprowadzanie liczby całkowitej, liczby dziesiętnej oraz liczby dziesiętnej z wykładnikiem. Tak wprowadzone liczby są konwertowane na standardowy format wyświetlania, natychmiast po wprowadzeniu danych i wciśnięciu klawisza **Enter**.

Poniżej podano przykłady poprawnie wprowadzonych wartości zmiennoprzecinkowych oraz format ich standardowego wyświetlania.

Wartość	Sposób wyświetlania w Logicmaster
250	+250.0000
+4	+4.000000
-2383019	-2383019.
34.	+34.00000
-.0036209	-.003620900
12.E+9	+1.20000E+10
-.0004E-11	-4.00000E-15
731.0388	+731.0388
99.20003e-29	+9.92000E-28

Następna tabela zawiera przykłady złych lub błędnie wprowadzonych wartości zmiennoprzecinkowych.

Błędnie wprowadzona wartość	Objaśnienia
-433E23	Brak kropki dziesiętnej. Logicmaster wyświetla komunikat "Bad numeric value"
10e-19	Brak kropki dziesiętnej. Logicmaster wyświetla komunikat "Bad numeric value"
1 0.e19	Pomiędzy cyframi 1 i 0 znajduje się spacja. Liczby rzeczywiste należy wprowadzać bez spacji pomiędzy cyframi bądź znakami. Logicmaster traktuje taki wpis jako niewłaściwa wartość +1.000000.
4.1e 19	Pomiędzy literą e a cyfrą 19 znajduje się spacja. Liczby rzeczywiste należy wprowadzać bez spacji pomiędzy cyframi bądź znakami. Logicmaster traktuje taki wpis jako niewłaściwa wartość +4,100000.

## Błędy w operacjach i liczbach zmiennoprzecinkowych

### Wartość nieskończenie duża lub nieskończenie mała

Jeżeli w jednostkach centralnych 352 lub 374, w wyniku działania funkcji operującej na danych typu REAL otrzymana zostanie wartość większa od  $3.402823E+38$  lub mniejsza od  $-3.402823E+38$  generowany jest błąd. W przypadku pozostałych sterowników serii 90-30, zakres dopuszczalnych wartości liczb zmiennoprzecinkowych jest większy od  $2^{16}$  lub mniejszy od  $-2^{16}$ . Jeżeli wartość nie mieści się w tym przedziale, nie jest wysyłany sygnał wyjściowy OK, a wynik w takiej sytuacji ma dodatnią, wartość nieskończenie dużą, tzn. większą od  $3.402823E+38$  w przypadku jednostki centralnej CPU352 lub większą od  $2^{16}$  w przypadku pozostałych modeli) lub też ma wartość ujemną, nieskończenie dużą (dla wartości mniejszych od  $-3.402823E+38$  lub od  $-2^{16}$  dla pozostałych modeli). Wystąpienie takiej sytuacji może zostać stwierdzone poprzez zbadanie wartości sygnału wyjściowego OK.

Nazwa mnemoniczna	Wartość wyświetlana w szczeblu drabiny logicznej	Wartość w tabeli zmiennych (heks)	Opis
POS_INF	+OVERFLOW	7F80 0000	Wartość dodatnia, nieskończenie duża IEEE, w systemie heksadecymalnym.
NEG_INF	-OVERFLOW	FF80 0000	Wartość ujemna, nieskończenie duża IEEE, w systemie heksadecymalnym.

### Uwaga

Przy korzystaniu z oprogramowania wykonującego operacje na liczbach zmiennoprzecinkowych (wszystkie modele w których można korzystać z liczb zmiennoprzecinkowych za wyjątkiem jednostki centralnej 352), wartości są zaokrąglane do zera (0) dla  $\pm 1.175494E-38$ .

Jeżeli otrzymane wartości nieskończenie duże zostaną wykorzystane jako parametr innej funkcji operującej na danych typu REAL, może to być powodem nieokreślonego wyniku. Ten nieokreślony wynik określany jest terminem NaN (Not a Number – nie liczba). Przykładowo, wynik dodawania dodatniej wartości nieskończenie dużej do ujemnej wartości nieskończenie małej jest nieokreślony. Jeżeli więc parametrami wywołania funkcji ADD\_REAL są dodatnia wartość nieskończenie duża i ujemna wartość nieskończenie duża, otrzymany wynik będzie równy NaN.

### NaN

NaN jest niezdefiniowaną wartością taką jak wynik z dzielenia zera przez zera. Nie rozróżnia się dodatnich ani ujemnych wartości NaN. Poniższa część ma pomóc w identyfikacji sytuacji, w których otrzymano wynik NaN.



## Kody NaN – CPU352 lub CPU374

W jednostkach centralnych 352 i 374, każda funkcja operująca na danych typu REAL, generuje wartość NaN jednoznacznie identyfikującą funkcję. W oprogramowaniu Logimaster sygnalizowane będzie to komunikatem “OVERFLOW”. (Jeśli termin “OVERFLOW” będzie poprzedzony znakiem plus lub minus, będzie to oznaczać dodatnią lub ujemną nieskończoność).

Kody NaN w jednostkach centralnych 352 i 374		
Nazwa mnemoniczna	Wartość w tabeli zmiennych (heks)	Opis
NaN_ADD.	7F81 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie dodawania liczb rzeczywistych.
NaN_SUB	7F81 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie odejmowania liczb rzeczywistych.
NaN_MUL	7F82 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie mnożenia liczb rzeczywistych
NaN_DIV	7F83 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie dzielenia liczb rzeczywistych
NaN_SQRT	7F84 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania pierwiastka kwadratowego dla liczby rzeczywistej.
NaN_LOG	7F85 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania logarytmu z liczby rzeczywistej.
NaN_POW0	7F86 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania potęgi liczby rzeczywistej.
NaN_SIN	7F87 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania sinusa liczby rzeczywistej.
NaN_COS	7F88 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania cosinusa liczby rzeczywistej.
NaN_TAN	7F89 FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania tangensa liczby rzeczywistej.
NaN_ASIN	7F8A FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania arcus sinusa liczby rzeczywistej.
NaN_ACOS	7F8B FFFF	Wartość heksadecymalna, otrzymywana w przypadku wystąpienia błędu w czasie obliczania arcus cosinusa liczby rzeczywistej.
NaN_BCD	7F8C FFFF	Błąd w czasie konwersji liczby w formacie BCD-4 na liczbę rzeczywistą
REAL_INDEF	FFC0 0000	Nieokreślona liczba rzeczywista, błąd spowodowany próbą dzielenia przez 0.

## Kody NaN – CPU35x, 36x, 37x (wyłączając CPU352)

Wszystkie pozostałe jednostki centralne pozwalające operować na liczbach zmiennoprzecinkowych (wyłączając CPU352) zwracają jedną wartość NaN. FFFF FFFF. W oprogramowaniu Logimaster sygnalizowane będzie to komunikatem “OVERFLOW”.

Typy NaN w jednostkach centralnych 35x, 36x, i 37x (wyłączając CPU352)		
Nazwa mnemoniczna	Wartość w tabeli zmiennych (heks)	Opis
NaN_SW	FFFF FFFF	Kod zmiennoprzecinkowy (programowy) dla wszystkich NaN

## Przepływ sygnału sterującego dla liczb nieskończonych i NaN

Jeżeli wynik równy NaN zostanie przesłany jako parametr do innej funkcji, wynik tej funkcji również będzie równy NaN (niepoprawna wartość). Przykładowo, jeśli wartość NaN\_ADD jest parametrem funkcji SUB\_REAL, to wynikiem jej działania będzie NaN\_ADD. Jeśli oba parametry wejściowe będą równe NaN, wtedy wynik będzie równy pierwszemu z nich. Taki mechanizm przekazywania wartości NaN pozwala na zidentyfikowanie funkcji, w której wystąpił błąd.

### Uwaga

Jeżeli wynik jest równy NaN, nie jest wysyłany sygnał wyjściowy ok.

W zamieszczonej poniżej tabeli pokazano czy wysyłany będzie sygnał wyjściowy przy operacjach takich jak dodawanie, mnożenie itp., w zależności czy na wejściu znajduje się liczba, czy nieskończoność. Jak pokazano wcześniej wyjścia, w których przekroczony został zakres wyświetlane będą jako POS\_INF lub NEG\_INF.

Tabela E-1 Przepływ sygnału przy operacjach zmiennoprzecinkowych

Działanie	Wartość parametru 1	Wartość parametru 2	Wyjście	Przepływ sygnału
Wszystkie	Liczba	Liczba	Wartość nieskończenie duża dodatnia lub ujemna	Jednokierunkowe
Wszystkie działania za wyjątkiem dzielenia	Nieskończoność	Liczba	Nieskończoność	Dwukierunkowe
Wszystkie	Liczba	Nieskończoność	Nieskończoność	Dwukierunkowe
Dzielenie bez reszty	Nieskończoność	Liczba	Nieskończoność	Jednokierunkowe
Wszystkie	Liczba	Liczba	NaN	Jednokierunkowe

Podręcznik ten został napisany dla użytkowników pracujących z oprogramowaniem Logicmaster (pracującym w systemie operacyjnym DOS). Informacje o oprogramowaniu pracującym w środowisku Windows tj. CIMPLICITY® Machine Edition Logic Developer i VersaPro®, można znaleźć w ich systemach pomocy on-line. Użytkownicy pracujący w środowisku Windows powinni zwrócić uwagę, że poszczególne instrukcje są wyświetlane w inny sposób niż ma to miejsce w programie Logicmaster (pomimo że dotyczą one tego samego sterownika). W systemie pomocy oprogramowania pracującego w środowisku Windows znajdują się dokładne informacje o poszczególnych instrukcjach.

Informacje pomoce w pracy z innymi programami można znaleźć w podręcznikach:

*VersaPro™ Programming Software User's Guide, GFK-1670*

*CIMPLICITY® Machine Edition Getting Started Guide, GFK-1868*

## Uwagi:

### Obsługa funkcji DRUM Sequencer

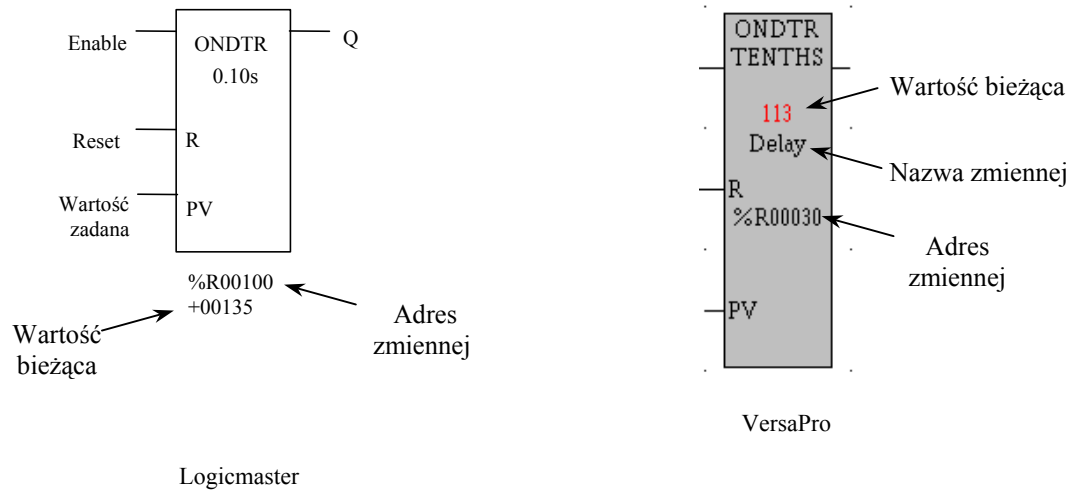
Funkcja ta obsługiwana przez jednostki centralne 350-364 w wersji 10,0 i późniejszych oraz przez wszystkie jednostki centralne 37x, lecz nie jest obsługiwana przez oprogramowanie Logicmaster dlatego też w podręczniku tym nie opisano tej instrukcji. Jest ona obsługiwana przez oprogramowanie VersaPro od wersji 1.1 oraz przez wszystkie wersje Logic Developer. Informacje o tej funkcji można znaleźć w pomocy on-lin oprogramowania.

### Znaczniki początku i końca programu

Znaczniki takie widoczne są w oprogramowaniu Logicmaster, nie są natomiast widoczne w oprogramowaniu pracującym pod kontrolą systemu Windows.

### Lokalizacja adresu słowa kontrolnego

Niektóre instrukcje takie jak przekaźniki czasowe, liczniki czy bit sequencer wymagają grupy kolejnych słów do wewnętrznych obliczeń. Grupa tak jest zwykle nazywany blokiem sterującym. W oprogramowaniu Logicmaster adres pierwszego słowa bloku sterującego jest wyświetlany pod instrukcją w programie sterującym (rejestr %R00100 na poniższym rysunku). W oprogramowaniu pracującym w środowisku Windows, adres pierwszej zmiennej jest wyświetlany wewnątrz bloku instrukcji (rejestr %R00030 na poniższym rysunku). W oprogramowaniu VersaPro oprócz adresu zmiennej wyświetlana jest również jej nazwa („Delay” na poniższym rysunku). Jeśli zmienna nie ma przypisanej nazwy, w obu miejscach zostanie wyświetlony adres. Na prawym rysunku (oprogramowanie VersaPro) powyżej nazwy wyświetlana jest wartość 113, jest to bieżąca wartość zapisana w tej zmiennej.



### Różnice w wyświetlaniu liczb rzeczywistych

Występują różnice pomiędzy programami w sposobie wyświetlania wartości nieokreślonych takich jak próba dzielenia przez zero. W załączniku E niniejszego podręcznika przedstawiono sposoby wyświetlania takich wartości, zarówno w programie sterującym jak i tablicy zmiennych.

## A

ACOS, 6-13  
 ADD, 6-2  
 ADD\_IOM, 2-29  
 ADD\_SIO, 2-29  
 Alarm  
     Kody błędów, B-5  
 Alarmy, 3-2  
 ALT klawisz, D-1  
 AND, 8-3  
 ANY\_FLT, 2-30  
 APL\_FLT, 2-29  
 Arcus cosinus, 6-13  
 Arcus sinus, 6-13  
 Arcus tangens, 6-13  
 ARRAY\_MOVE, 10-2  
 ASIN, 6-13  
 ATAN, 6-13  
 Automatyczne zerowanie statystyk, 12-97  
 Awaria komunikacji podczas przesyłania danych do sterownika, 3-16

## B

BAD\_PWD, 2-29  
 BCD-4, 2-27, 11-2  
 BCLR, 8-14  
 BIT, 2-27  
 BITSEQ, 9-13  
     zapotrzebowanie na pamięć, 9-14  
 BLKCLR, 9-9  
 BLKMOV, 9-6  
 Bloki podprogramów, 2-22  
 Bloki programu  
     wywoływanie bloków, 2-23  
     wywoływanie bloków C, 2-23  
     wywoływanie podprogramów, 2-23  
 Blokowanie  
     permanently locking a subroutine, 2-44  
 Blokowanie/ odblokowywanie podprogramów, 2-44  
 Błąd oprogramowania, modułu dodatkowego, 3-11  
 Błąd programu sterującego, 3-12  
 Błąd w działaniu programu sterującego, 3-12  
 Błąd w oprogramowaniu jednostki centralnej sterownika, 3-14  
 Błąd w oprogramowaniu wyspecjalizowanego modułu dodatkowego, 3-11  
 Błędy  
     klasy błędów, 3-2  
 Błędy diagnostyczne  
     rozładowane baterie, 3-11  
 Błędy działania, dodatkowe skutki, 3-5  
 Błędna suma kontrolna programu, 3-11  
 Błędna suma kontrolna programu, 3-11  
 Błędna suma kontrolna, programu, 3-11  
 Błędy i ich usuwanie  
     grupy błędów wejść/wyjść, B-11  
 Błędy  
     grupy błędów działania sterownika, B-4  
 Błędy, 3-2  
     Awaria komunikacji podczas przesyłania danych do sterownika, 3-16  
     błąd w działaniu programu sterującego, 3-12  
     błąd w oprogramowaniu jednostki centralnej sterownika, 3-14  
     błędna suma kontrolna programu, 3-11  
     błędna suma kontrolna programu, 3-11  
     brak programu sterującego, 3-13  
     brak zadeklarowanego modułu wejść/wyjść, 3-17  
     dodatkowe informacje o błędzie, 3-7

dodatkowe skutki wystąpienia, 3-5  
 dołączono niezadeklarowany moduł wejść/wyjść, 3-18  
 grupy błędów wejść/wyjść, B-11  
 interpretacja tabeli błędów działania sterownika, 3-8  
 Interpretacja tabeli działania układów wejść/wyjść, 3-17  
 interpretowanie, B-1  
 klasy błędów, 3-2  
 kody błędów, B-5  
 Niepoprawne hasło dostępu do sterownika, 3-13  
 reakcja systemu na błędy, 3-3  
 reakcja systemu na błędy działania, 3-3  
 rozładowane baterie, 3-11  
 tabela błędów działania sterownika, 3-3, 3-5  
 tabela błędów działania wejść/wyjść, 3-3  
 usterka układu wejść/wyjść, 3-2  
 usterka wewnętrzna, 3-2  
 usterki eksploatacyjne, 3-2  
 wagi błędów, 3-3  
 wagi błędów działania sterownika, B-5  
 wagi błędów wejść/wyjść, B-12  
 wyświetlanie dodatkowych informacji o błędzie, 3-7  
 zmienne, 3-4

Błędy diagnostyczne, 3-4  
     błąd w działaniu programu sterującego, 3-12  
     brak zadeklarowanego modułu wejść/wyjść, 3-17  
     dołączono niezadeklarowany moduł wejść/wyjść, 3-18  
 Błędy działania sterownika - opis i ich usuwanie:, 3-1  
 Błędy i ich usuwanie  
     awaria komunikacji podczas przesyłania danych do sterownika, 3-16  
     błąd w oprogramowaniu jednostki centralnej, 3-14  
     brak programu sterującego, 3-13  
     brak zadeklarowanego modułu wejść/wyjść, 3-17  
     dołączono niezadeklarowany moduł wejść/wyjść, 3-18  
     niepoprawne hasło dostępu do sterownika, 3-13  
     rozładowane baterie, 3-11  
     wykrycie uszkodzenia pamięci programu sterującego podczas rozruchu, 3-13  
     wyświetlanie dodatkowych informacji o błędzie, 3-7  
 Błędy i sposób ich usuwania  
     tabela błędów działania sterownika, 3-5  
 Błędy informacyjne, 3-4  
     brak programu sterującego, 3-13  
     niepoprawne hasło dostępu do sterownika, 3-13  
 Błędy krytyczne, 3-4  
     awaria komunikacji podczas przesyłania danych do sterownika, 3-16  
     błąd w oprogramowaniu jednostki centralnej sterownika, 3-14  
     błędna suma kontrolna programu, 3-11  
     wykrycie uszkodzenia pamięci programu sterującego podczas rozruchu, 3-13  
 Błędy, interpretowanie, B-1  
 BPOS, 8-16  
 Brak programu sterującego, 3-13  
 Brak zadeklarowanego modułu wejść/wyjść, 3-17  
 BSET, 8-14  
 BTST, 8-12  
 BYTE, 2-27

## C

CALL, 12-3  
 CFG\_MM, 2-29  
 COMMENT, 12-45  
 COMMREQ, 9-17  
     kod błędu, opis usuwanie, 3-11

COS, 6-13  
 Cosinus, 6-13  
 CTRL klawisz, D-1  
 Cykl o stałym czasie trwania, 2-41  
 cykl pracy jednostki centralnej, 2-2  
 Cykl pracy sterownika  
   komunikacja modułu PCM ze sterownikiem, 2-14  
 Cykl pracy sterownika, 2-2  
   czasy odczytywania dla jednostek centralnych  
   35x/36x/37x, 2-6  
   czasy odczytywania dla jednostek centralnych  
   35x/36x/37x, 2-7  
   inicjalizacja, 2-10  
   komunikacja modułu DSM ze sterownikiem, 2-16  
   komunikacja z programatorem, 2-11  
   konfiguracja systemowa, 2-13  
   obsługa wyjść, 2-11  
   odczytywanie danych wejściowych, 2-10  
   rozwiązania programowe, 2-11  
   stały czas trwania cyklu, 2-16  
   tryb STOP, 2-17  
   tryb ze stałym czasem trwania, 2-41  
   tryb ze stałym czasem trwania cyklu, 2-16  
   warianty standardowego cyklu pracy sterownika, 2-16  
   wykonywanie programu sterującego, 2-11  
 Cykl pracy sterownika, standardowy, 2-2  
 Cykl, obliczania czasu trwania, 2-9  
 Cykl, sterownika  
   czasy odczytywania dla jednostek centralnych  
   35x/36x/37x, 2-7  
   rozwiązania programowe, 2-11  
 Cykl, sterownika, 2-2  
   czasy odczytywania dla jednostek centralnych  
   35x/36x/37x, 2-6  
   komunikacja modułu DSM ze sterownikiem, 2-16  
   komunikacja modułu PCM ze sterownikiem, 2-14  
   konfiguracja systemowa, 2-13  
   obsługa wyjść, 2-11  
   tryb ze stałym czasem trwania cyklu, 2-16  
   wykonywanie programu sterującego, 2-11  
 Cykl, Sterownika  
   odczytywanie danych wejściowych, 2-10  
 Czasu wykonywania instrukcji  
   modele 35x-36x, A-8  
 Czasy odczytywania dla jednostek centralnych  
   35x/36x/37x, 2-6, 2-7  
 Czasy wykonywania instrukcji, A-1  
   modele standardowe, A-3  
   SER, A-15  
 Czasy wykonywania przekaników i styków, A-24

## D

Dane diagnostyczne, 2-49  
 Dane globalne, 2-49  
 Dane globalne Genius, 2-49  
 DEG, 6-17  
 DINT, 2-27, 11-6  
 DIV, 6-2  
 DNCTR, 5-14  
 Dodawanie, 6-2  
 DOIO, 12-4  
   rozszerzona funkcja DOIO dla CPU331 i wyższych,  
   12-10  
 Dołączony niezadeklarowany moduł wejść/wyjść, 3-18  
 Domyślne wartości modułów wyjść, 2-48  
 Działanie sterownika, 2-1  
 Działanie systemu, 2-1  
 Działanie systemu sterownika, 2-1  
 Dzielenie, 6-2

## E

EDITLOCK, 2-44  
 END, 12-30  
 ENDMCR, 12-40  
 ENDMCR, postać nie zagnieżdżona, 12-40  
 EQ, 7-1  
 EXP, 6-15  
 EXPT, 6-15

## F

Funkcja Bit sequencer, 9-13  
 funkcja Block clear, 9-9  
 Funkcja Block move, 9-6  
 Funkcja Call, 12-3  
 Funkcja Comment, 12-45  
 Funkcja Communication request, 9-17  
 Funkcja Do I/O, 12-4  
   rozszerzona funkcja DOIO dla CPU331 i wyższych,  
   12-10  
 Funkcja End, 12-30  
 Funkcja MCR, 12-32  
 Funkcja modułu, 6-9  
 Funkcja Move, 9-2  
 Funkcja Range, 7-4  
 Funkcja SER, 12-11  
 Funkcja Service request  
   Automatyczne zerowanie statystyk (#49), 12-97  
   Odczyt czasu trwania cyklu (#9), 12-67  
   Odczyt identyfikatora sterownika (#11), 12-69  
   Odczyt nazwy folderu (#10), 12-68  
   Odczyt ostatnio zarejestrowanego komunikatu o  
   błędzie działania, 12-73  
   Odczyt sumy kontrolnej programu sterującego i  
   konfiguracji, 12-83  
   Odczyt wartości z programatora (#2), 12-53  
   Odczyt/zmiana liczby słów sumy kontrolnej  
   programu sterującego, 12-59  
   Odczyt/zmiana wskazań zegara czasu  
   rzeczywistego, 12-61  
   Pominięcie obsługi wejść/wyjść w następnym  
   cyklu, 12-87  
   Ponowne uruchomienie po błędzie krytycznym  
   automatycznego zerowania (#48), 12-95  
   Porównanie rzeczywistej konfiguracji modułów  
   wejść/ wyjść sterownika ze zdefiniowaną, 12-85  
   Szybki dostęp do statusu komunikacji przez szynę  
   komunikacyjną, 12-88  
   Wymazanie komunikatów z tablicy błędów  
   sterownika i układów wejść/wyjść, 12-72  
   Zatrzymanie sterownika, 12-71  
   zegar czasu trwania przerwy w zasilaniu  
   sterownika, 12-86  
   zegar odmierający czas pracy sterownika, 12-80  
   Zerowanie zegara wyłączającego (#8), 12-66  
   Zmiana trybu i czasu trwania komunikacji z  
   programatorem (#3), 12-55  
   Zmiana trybu komunikacji systemowej oraz czasu  
   trwania komunikacji systemowej (#4), 12-57  
   Zmiana/odczyt czasu trwania cyklu pracy w trybie  
   ze stałym czasem trwania cyklu (#1), 12-50  
 Funkcja Shift register, 9-10  
 Funkcja SVCREQ,  
   lista, 12-47  
 Funkcja Truncate, 11-12  
 Funkcje do operacji na bitach  
   AND, 8-3  
   BCLR, 8-14  
   BPOS, 8-16  
   BSET, 8-14

- BTST, 8-12
  - MCMP, 8-18
  - OR, 8-3
  - ROR, 8-10
  - SHL, 8-8
  - SHR, 8-8
  - XOR, 8-5
  - Funkcje do operacji na danych, 9-1
  - Funkcje do operacji na bitach, 8-1
    - ROL, 8-10
  - Funkcje do przemieszczania danych
    - BITSEQ, 9-13
    - BLKCLR, 9-9
    - COMMREQ, 9-17
  - Funkcje do przemieszczenia danych
    - BLKMOV, 9-6
    - MOVE, 9-2
    - SHFR, 9-10
  - Funkcje konwersji
    - REAL, 11-8
  - Funkcje konwersji, 11-1
    - BCD-4, 11-2
    - DINT, 11-6
    - INT, 11-4
    - TRUN, 11-12
    - WORD, 11-10
  - Funkcje logarytmiczne, 6-15
    - logarytm dziesiętny, 6-15
    - logarytm naturalny, 6-15
  - Funkcje matematyczne, 6-1
    - ACOS, 6-13
    - ADD, 6-2
    - ASIN, 6-13
    - ATAN, 6-13
    - COS, 6-13
    - DEG, 6-17
    - DIV, 6-2
    - EXP, 6-15
    - EXPT, 6-15
    - LN, 6-15
    - LOG, 6-15
    - MOD, 6-9
    - MUL, 6-2
    - RAD, 6-17
    - SIN, 6-13
    - SQRT, 6-11
    - SUB, 6-2
    - TAN, 6-13
  - Funkcje przekaźników, 4-1
    - przełącznik kontynuacji, 4-9
    - przełącznik RESET, 4-6
    - przełącznik RESET z pamięcią, 4-8
    - przełącznik SET, 4-6
    - przełącznik uaktywniany z boczem narastającym sygnału, 4-5
    - przełącznik z pamięcią SET, 4-7
    - przełączniki, 4-2, 4-3
    - przełączniki o stykach zwartych z pamięcią, 4-5
    - styk kontynuacji, 4-9
    - styki, 4-1
  - Funkcje relacji, 7-1
    - EQ, 7-1
    - GE, 7-1
    - GT, 7-1
    - LE, 7-1
    - LT, 7-1
    - NE, 7-1
    - RANGE, 7-4
  - Funkcje Service request
    - kontrola występowania wymuszeń zmiany wartości zmiennych wejściowych i wyjściowych, 12-82
  - Funkcje sterujące, 12-1
    - CALL, 12-3
    - COMMENT, 12-45
    - DOIO, 12-4
      - rozszerzona funkcja DOIO dla CPU331 i wyższych, 12-10
    - END, 12-30
    - ENDMCR, 12-40
    - JUMP, 12-41
    - LABEL, 12-43
    - MCR, 12-32
    - PID, 12-98
    - Sequential Event Recorder, 12-12
    - SER, 12-11, 12-12
    - SVCREQ, 12-47
  - Funkcje wykładnicze, 6-15
    - potęga e, 6-15
    - potęga liczby x, 6-15
- ## G
- GE, 7-1
  - Generator sygnału prostokątnego, 2-42
  - Grupy błędów, B-4
  - Grupy błędów, B-11
  - GT, 7-1
- ## H
- Hasła, 2-43
  - HRD\_CPU, 2-29
  - HRD\_FLT, 2-30
  - HRD\_SIO, 2-29
- ## I
- Inicjalizacja cyklu sterownika, 2-10
  - Instrukcja Jump, 12-41
  - Instrukcja Label, 12-43
  - Instrukcje
    - funkcje do operacji na bitach, 8-1
    - funkcje do operacji na danych, 9-1
    - funkcje konwersji, 11-1
    - funkcje relacji, 7-1
    - funkcje sterujące, 12-1
    - matematyczne, 6-1
    - operacje tablicowe, 10-1
  - Instrukcje do programowania
    - funkcje matematyczne, 6-1
    - funkcje relacji, 7-1
    - nazwy mnemoniczne, C-1
    - styki i przełączniki, 4-1
    - funkcje do operacji na bitach, 8-1
    - funkcje do operacji na danych, 9-1
    - funkcje relacji, 7-1
    - funkcje sterujące, 12-1
    - nazwy mnemoniczne, C-1
    - funkcje matematyczne, 6-1
    - operacje tablicowe, 10-1
    - funkcje konwersji, 11-1
  - INT, 2-27, 11-4
  - Interpretacja i usuwanie błędów
    - grupy błędów działania sterownika, B-4
  - Interpretowanie i usuwanie błędów
    - interpretowanie błędów, B-1
  - IO\_FLT, 2-30
  - IO\_PRES, 2-30
- ## J
- JUMP, 12-41

## K

Kategoria błędów, 3-17  
Kategoria błędu, 3-17  
Klonwersja na dane typu INT, 11-4  
Kody błędów, B-5  
Komunikacja modułu DSM ze sterownikiem, 2-16  
Komunikacja modułu PCM ze sterownikiem, 2-14  
Komunikacja z programatorem, 2-11  
Komunikacja ze sterownikiem, 2-14, 2-16  
Konfiguracja systemowa, 2-13  
Konsrwacja, 3-1  
Kontrola występowania wymuszeń zmiany wartości zmiennych wejściowych i wyjściowych, 12-82  
Konwersja miar kąta, 6-17  
Konwersja na dane typu BCD-4, 11-2  
Konwersja na dane typu DINT, 11-6  
Konwersja na dane typu Word, 11-10  
Korzystanie z liczb zmiennoprzecinkowych, E-1

## L

LABEL, 12-43  
LE, 7-1  
Liczba całkowita podwójnej precyzji, 2-27  
Liczby zmiennoprzecinkowe, E-1  
    błędy w operacjach i liczbach zmiennoprzecinkowych, E-6  
    format wewnętrznej reprezentacji liczb zmiennoprzecinkowych, E-3  
    nazewnictwo, E-2  
    wartości liczb zmiennoprzecinkowych, E-4  
    wprowadzanie i wyświetlanie liczb zmiennoprzecinkowych, E-5  
Licznik zliczający w dół, 5-14  
Licznik zliczający w górę, 5-12  
Liczniki  
    blok danych sterujących, 5-1  
    DNCTR, 5-14  
    UPCTR, 5-12  
LN, 6-15  
LOG, 6-15  
Logarytm dziesiętny, 6-15  
Logarytm naturalny, 6-15  
Logiczne AND, 8-3  
Logiczne NOT, 8-7  
Logiczne OR, 8-3  
Logiczne XOR, 8-5  
Lokalizowanie bitu, 8-16  
LOS\_IOM, 2-29  
LOS\_SIO, 2-29  
LOW\_BAT, 2-29  
LT, 7-1

## M

MCR, 12-32  
Mechanizm blokowania, 2-44  
Mechanizm blokowania podprogramów  
    blokada edycji, 2-44  
    blokada podglądu, 2-44  
Mechanizmy zabezpieczenia przed dostępem blokowanie/ odblokowywanie podprogramów, 2-44  
    poziomy uprawnień, 2-43  
    zmiana poziomu uprawnień, 2-43  
Mechanizmy zabezpieczenia  
    hasła, 2-43  
Mniejszy lub równy, 7-1  
Mnożenie, 6-2  
MOD, 6-9

Model 20 moduły wejść/wyjść, 1-2  
Model 30 I/O modules, 2-46  
Moduły wejść/ wyjść  
    sterowniki serii 90/20, 2-45  
    sterowniki serii 90/30, 2-45  
Moduły wejść/ wyjść sterowników serii 90/30, 2-45  
Moduły wejść/ wyjść sterowników serii 90/30L  
    dane diagnostyczne, 2-49  
Moduły wejść/wyjść serii 90-20, 2-46  
Moduły wejść/wyjść sterowników serii 90-20, 2-51  
MOVE, 9-2  
MSKCMP, 8-18  
MUL, 6-2

## N

NaN, E-6  
Nazwy mnemoniczne instrukcji, C-1  
NE, 7-1  
Niepoprawne hasło dostępu do sterownika, 3-13  
NOT, 8-7  
Not a Number, E-6

## O

Obliczanie czasu trwania cyklu pracy sterownika, 2-9  
Obsługa błędów, 3-2  
    alarmy, 3-2  
    błędy działania, 3-3  
    procesor alarmowy, 3-2  
Obsługa wyjść, 2-11  
Obsługa, wyjść, 2-11  
Odczyt czasu trwania cyklu, 12-67  
Odczyt identyfikatora sterownika, 12-69  
Odczyt nazwy folderu, 12-68  
Odczyt ostatnio zarejestrowanego komunikatu o błędzie działania, 12-73  
Odczyt sumy kontrolnej programu sterującego i konfiguracji, 12-83  
Odczyt trybu pracy sterownika, 12-70  
Odczyt wartości z programatora, 12-53  
Odczyt/zmiana liczby słów sumy kontrolnej programu sterującego, 12-59  
Odczyt/zmiana wskazań zegara czasu rzeczywistego, 12-61  
Odczytywanie danych wejściowych, 2-10  
Odczytywanie danych, wejściowych, 2-10  
Odejmnowanie, 6-2  
Odnajdywanie i usuwanie usterek  
    interpretowanie błędów, B-1  
OFDT, 5-9  
ONDTR, 5-4  
Operacje tablicowe, 10-1  
    ARRAY\_MOVE, 10-2  
    SRCH\_GE, 10-9  
    szukanie wartości mniejszej lub równej, 10-9  
Opis błędów i ich usuwanie  
    błędna suma kontrolna programu, 3-11  
Opis błędów i ich usuwanie  
    reset of, addition of, or extra, option module, 3-9  
Opis błędów i ich usuwanie  
    application fault, 3-12  
Opis błędów i ich usuwanie, 3-1  
    błędy konfiguracji, 3-9  
    constant sweep time exceeded, 3-12  
    dodatkowe informacje o błędzie, 3-7  
    interpretacja tabeli błędów działania sterownika, 3-8  
    interpretacja tabeli błędów działania układów wejść/wyjść, 3-17  
    kategoria błędu, 3-17  
    obsługa błędów, 3-2



opis błędu, 3-17  
 tabela błędów działania sterownika, 3-5  
 tabela błędów działania wejść/wyjść, 3-5  
 typ błędu, 3-17  
 Opis błędu, 3-17  
 Opis działania  
 Cykl pracy sterownika, 2-2  
 moduły wejść/wyjść serii 90/20, 2-45  
 moduły wejść/wyjść serii 90-30, 2-45  
 organizacja programu, typy danych i typy zmiennych, 2-21  
 Opis działania systemu  
 zegary, 2-40  
 OR, 8-3  
 Organizacja programu  
 liczby zmiennoprzecinkowe, E-1  
 pamięć stanu, 2-26  
 struktura bloku funkcyjnego, 2-31  
 typy danych, 2-27  
 typy zmiennych, 2-24  
 zmienne systemowe, 2-28  
 Organizacja programu, typy danych i typy zmiennych, 2-21  
 OV\_SWP, 2-29

## P

Pamięć stanu, 2-26  
 Pamięć, uszkodzona, 3-8  
 Parametry bloków funkcyjnych, 2-33  
 PB\_SUM, 2-29  
 PID, 12-98  
 Pierwiastek kwadratowy, 6-11  
 Podręczniki  
 moduły wejść/wyjść, 2-46  
 Podstawowe informacje o instrukcjach języka drabinkowego  
 parametry bloków funkcyjnych, 2-33  
 Podstawowe informacje o instrukcjach języka drabinkowego  
 format bloków funkcyjnych, 2-31  
 Pominięcie obsługi wejść/wyjść w następnym cyklu, 12-87  
 Ponowne uruchomienie po błędzie krytycznym automatycznego zerowania, 12-95  
 Porównanie rzeczywistej konfiguracji modułów wejść/wyjść sterownika ze zdefiniowaną, 12-85  
 Porównanie z maskowaniem, 8-18  
 Potęga liczby e, 6-15  
 Potęga liczby x, 6-15  
 Poziomy uprawnień, 2-43  
 zmiana, 2-43  
 Procedury wykonywane okresowo, 2-24  
 Procesor alarmowy, 3-2  
 Program  
 blok podprogramu, 2-22  
 Przekaznik  
 sprawdzanie wielokrotnego wykorzystania zmiennych w przekaznikach, 4-7  
 Przekaznik czasowy wyłączający, 5-9  
 Przekaznik czasowy załączający, 5-4, 5-6  
 Przekaznik kontynuacji, 4-9  
 Przekaznik o stykach otwartych z pamięcią, 4-5  
 Przekaznik o stykach zwartych, 4-5  
 Przekaznik o stykach zwartych z pamięcią, 4-5  
 Przekaznik RESET, 4-6  
 Przekaznik RESET z pamięcią, 4-8  
 Przekaznik SET z pamięcią, 4-7  
 Przekaznik uaktywniany zboczem narastającym sygnału, 4-5  
 Przekaznik uaktywniany zboczem opadającym sygnału, 4-6

Przekaznik ustawialny SET, 4-6  
 Przekazniki, 4-2, 4-3  
 przekaznik o stykach otwartych z pamięcią, 4-5  
 przekaznik o stykach zwartych, 4-5  
 przekaznik RESET, 4-6  
 przekaznik RESET z pamięcią, 4-8  
 przekaznik SET, 4-6  
 przekaznik uaktywniany zboczem opadającym sygnału, 4-6  
 przekaznik ustawialny SET, 4-7  
 styk kontynuacji, 4-9  
 Przekazniki czasowe  
 blok danych sterujących, 5-1  
 generator sygnału postokątnego, 2-42  
 OFDT, 5-9  
 ONDTR, 5-4  
 stały czas trwania cyklu, 2-41  
 TMR, 5-6  
 zegar wyłączający, 2-41  
 Przekroczony czas trwania cyklu, 3-12  
 Przepływy sygnału, 2-34  
 Przesunięcie bitowe w prawo, 8-8  
 Przesunięcie bitowe w lewo, 8-8  
 Przesunięcie bitowe w prawo w obiegu zamkniętym, 8-10  
 Przkazniki  
 przekaznik o stykach zwartych z pamięcią, 4-5  
 Przesunięcie bitowe w lewo w obiegu zamkniętym, 8-10  
 Przykłady  
 SER, 12-25

## R

RAD, 6-17  
 RANGE, 7-4  
 REAL  
 Konwersja na dane typu REAL, 11-8  
 Korzystanie z liczb zmiennoprzecinkowych, E-1  
 struktura danych, 2-27  
 Regulator PID, 12-98  
 Rejestry  
 rejestry systemowe, 2-24  
 Reset Smart Module, 12-84  
 ROL, 8-10  
 ROR, 8-10  
 Rozładowane baterie, 3-11  
 Rozruch i wyłączenie sterownika, 2-36  
 Rozruch i wyłączenie sterownika, 2-36, 2-39  
 Rozruch sterownika, 2-36  
 Rozszerzona funkcja DOIO dla CPU331 i wyższych, 12-10  
 Rozwiązania programowe, 2-11  
 Równy, 7-1  
 Różny, 7-1

## S

Schemat pracy sterownika, 2-1  
 Sekwencyjny rejestrator zdarzeń, 12-11, 12-12  
 SER, 12-11  
 Service Request  
 Odczyt/zmiana liczby słów sumy kontrolnej programu sterującego, 12-59  
 SFT\_CPU, 2-29  
 SFT\_FLT, 2-30  
 SHFR, 9-10  
 SHL, 8-8  
 SHR, 8-8  
 SIN, 6-13  
 Sinus, 6-13  
 SNPX\_RD, 2-28

- SNPX\_WT, 2-28  
 SNPXACT, 2-28  
 Sprawdzenie wartości bitu, 8-12  
 SQRT, 6-11  
 SRCH\_GE, 10-9  
 SRCH\_LE, 10-9  
 Standard program sweep mode, 2-2  
 Standard program sweep variations, 2-16  
 Standardowy cykl pracy sterownika, 2-2  
 Stanu pamięć, 2-26  
 Sterowniki 90-20 system wejść/wyjść  
 moduły wejść/wyjść model 20, 1-2  
 Sterowniki serii 90-20  
 moduły wejść/ wyjść, 2-51  
 Sterowniki serii 90-30  
 moduły wejść/wyjść, 2-46  
 STOP mode, 2-17  
 STOR\_ER, 2-29  
 Struktura bloku funkcyjnego, 2-31  
 przekaźniki, 2-31  
 Struktura programu  
 blok podprogramu, 2-22  
 wywoływanie bloków, 2-23  
 wywoływanie bloków C, 2-23  
 wywoływanie podprogramów, 2-23  
 Struktura wejść/wyjść, sterowniki 90-30, 2-45  
 Styk kontynuacji, 4-9  
 Styk otwarty, 4-3  
 Styk zwarty, 4-3  
 Styki, 4-1  
 styk kontynuacji, 4-9  
 styk otwarty, 4-3  
 styk zwarty, 4-3  
 Styki i przekaźniki, 4-1  
 przekaźnik uaktywniany zboczem narastającym  
 sygnału, 4-5  
 przekaźniki, 4-2  
 przekaźniki o stykach otwartych z pamięcią, 4-5  
 przekaźniki o stykach zwartych, 4-5  
 przekaźnik uaktywniany zboczem opadającym  
 sygnału, 4-6  
 styk otwarty, 4-3  
 styk zwarty, 4-3  
 szyny poziome i pionowe, 4-8  
 Styki, przekaźniki i połączenia, 4-1  
 SUB, 6-2  
 SY\_FLT, 2-30  
 SY\_PRES, 2-30  
 System wejść/wyjść sterowników 90-30  
 domyślne wartości modułów wyjść, 2-48  
 System wejść/wyjść sterowników 90-20  
 moduły wejść/wyjść model 20, 1-2  
 System wejść/wyjść sterowników 90-30  
 dane globalne, 2-49  
 formaty danych wejść/wyjść, 2-48  
 struktura wejść/wyjść, 2-45  
 System wejść/wyjść, sterowniki 90-30  
 domyślne wartości modułów wyjść, 2-48  
 System wejść/wyjść, sterowniki 90-30  
 formaty danych wejść/wyjść, 2-48  
 Szukanie wartości mniejszej lub równej, 10-9  
 Szukanie wartości większej lub równej, 10-9  
 Szybki dostęp do statusu komunikacji przez szynę  
 komunikacyjną, 12-88  
 Szyna pionowa, 4-8  
 Szyna pozioma, 4-8  
 Szyny poziome i pionowe, 4-8
- T**
- Tabela błędów działania sterownika  
 kasety, B-3
- Tabela błędów działania wejść/wyjść  
 interpretowanie błędu, B-1  
 Tabela błędów działania sterownika  
 grupy błędów, B-4  
 Tabela błędów działania sterownika, 3-3, 3-5, B-1  
 czas zarejestrowania błędu, B-8  
 długi/krótki wskaźnik, B-3  
 gniazdo, B-3  
 interpretowanie błędu, B-1  
 kody błędów, B-5  
 objaśnienia, 3-8  
 opis, 3-17  
 pole wolne, B-3  
 szczegółowe informacje o błędzie, B-8  
 wagi błędów, B-5  
 zadanie, B-3  
 Tabela błędów działania sterownika:, 3-5  
 Tabela błędów działania układów wejść/wyjść, 3-5  
 gniazdo, B-11  
 grupy błędów, B-11  
 interpretacja, 3-17  
 punkt, B-11  
 Tabela błędów działania wejść/wyjść, 3-3, 3-5  
 adres błędu, B-10  
 adres elementu, B-10  
 czas zarejestrowania błędów, B-13  
 długi/krótki wskaźnik, B-10  
 kasety, B-11  
 szczegółowe informacje o błędzie, B-12  
 wagi błędów, B-12  
 Tabela błędów sterownika  
 interpretacja, 3-8  
 Tabela działania sterownika, 3-3  
 Tabela działania wejść/wyjść, B-9  
 Tabela błędów działania wejść/wyjść  
 szczegółowe informacje o błędzie, B-12  
 TAN, 6-13  
 Tangens, 6-13  
 Time-of-day clock, 2-40  
 TMR, 5-6  
 TRUN, 11-12  
 tryb STOP, 2-17  
 Tryb ze stałym czasem trwania cyklu, 2-16  
 Tryb ze stałym czasem trwania cyklu pracy sterownika,  
 2-41  
 Tryby realizacji faz komunikacji, 2-17  
 Typ błędu, 3-17  
 Typy danych, 2-27  
 BAJT, 2-27  
 BCD-4, 2-27  
 BIT, 2-27  
 DINT, 2-27  
 REAL, 2-27  
 WORD, 2-27  
 Typy zmiennych, 2-24  
 wejścia dyskretne, 2-24  
 wewnętrzne zmienne dyskretne, 2-25  
 wyjścia analogowe, 2-24  
 wyjścia dyskretne, 2-25  
 zmienne dyskretne, 2-24  
 zmienne systemowe, 2-24, 2-25, 2-28
- U**
- UPCTR, 5-12  
 Ustawianie bitu, 8-14  
 Usterka układu wejść/wyjść, 3-2  
 Usterki eksploatacyjne, 3-2  
 Usterki wewnętrzne, 3-2  
 Usuwanie zakłóceń  
 wyświetlanie dodatkowych informacji o błędzie, 3-7

Uszkodzona pamięć, 3-8

## V

VersaPro  
uwagi dla użytkowników, 1-2  
VIEWLOCK, 2-44

## W

Wagi błędów, 3-3, 3-9  
błędy diagnostyczne, 3-4  
błędy informacyjne, 3-4  
błędy krytyczne, 3-4  
wagi błędów działania sterownika, B-5  
wagi błędów wejść/wyjść, B-12  
Wagi błędów i działania podejmowane w momencie ich wystąpienia, 3-9  
Wagi błędów, dodatkowe, 3-5  
Warianty standardowego cyklu pracy sterownika, 2-16  
Wewnętrzne zmienne dyskretne, 2-25  
Większy, 7-1  
Większy lub równy, 7-1  
WORD, 2-27, 11-10  
Wykonywanie programu sterującego, 2-11  
Wykrycie uszkodzenia pamięci programu sterującego podczas rozruchu, 3-13  
Wykrywanie i usuwanie usterek, 3-1  
błędy konfiguracji, 3-9  
dodatkowe informacje o błędach, 3-7  
interpretacja tabeli błędów działania sterownika, 3-8  
interpretacja tabeli błędów działania układów wejść/wyjść, 3-17  
tabela błędów działania sterownika, 3-5  
tabela błędów działania wejść/wyjść, 3-5  
Wyłączanie sterownika, 2-39  
Wymazanie komunikatów z tablicy błędów sterownika i układów wejść/wyjść, 12-72  
Wymuszenie zmiany wartości, 2-26

## X

XOR, 8-5

## Z

Zatrzymanie sterownika, 12-71  
Zegar cyklu o stałym czasie trwania, 2-41  
Zegar czasu rzeczywistego, 2-40  
zegar czasu trwania przerwy w zasilaniu sterownika, 12-86  
Zegar czasu trwania przerwy w zasilaniu sterownika, 2-41  
Zegar odmierzający czas pracy sterownika, 2-40  
Zegar odmierzający czas pracy sterownika, 12-80  
Zegar wyłączający, 2-41  
Zegary, 2-40  
zegar czasu pracy, 2-40  
zegar czasu rzeczywistego, 2-40  
zegar czasu trwania przerwy w zasilaniu sterownika, 2-41  
Zerowanie bitu, 8-14  
Zerowanie zegara wyłączającego, 12-66  
Zestaw instrukcji styki i przekaźniki, 4-1  
Zmiana poziomu uprawnień, 2-43  
Zmiana stanu, 2-26  
Zmiana trybu i czasu trwania komunikacji z programatorem, 12-55

Zmiana trybu komunikacji systemowej oraz czasu trwania komunikacji systemowej, 12-57

Zmienne  
dyskretne tymczasowe, 2-25  
Zmienne do obsługi błędów, 3-4  
Zmienne dyskretne  
zmienne systemowe statusu, 2-25, 2-28  
Zmienne dyskretne, 2-24  
dane globalne, 2-25  
tymczasowe, 2-25  
wewnętrzne zmienne dyskretne, 2-25  
wyjścia dyskretne, 2-25  
Zmienne dyskretne  
wejścia dyskretne, 2-24  
Zmienne globalne, 2-25  
Zmienne przypisane wejściom analogowym, 2-24  
zmienne przypisane wejściom dyskretnym, 2-24Z  
Zmienne przypisane wyjściom analogowym, 2-24  
Zmienne przypisane wyjściom dyskretnym, 2-25  
Zmienne rejestrowe, 2-24  
adresy rejestrów, 2-24  
wejścia analogowe, 2-24  
wyjścia analogowe, 2-24  
Zmienne statusu, systemowe, 2-25, 2-28  
Zmienne systemowe, 2-24  
Zmienne systemowe statusu, 2-25, 2-28  
ADD\_IOM, 2-29  
ADD\_SIO, 2-29  
ANY\_FLT, 2-30  
BAD\_PWD, 2-29  
CFG\_MM, 2-29  
HRD\_CPU, 2-29  
HRD\_FLT, 2-30  
HRD\_SIO, 2-29  
IO\_FLT, 2-30  
LOS\_IOM, 2-29  
LOS\_SIO, 2-29  
LOW\_BAT, 2-29  
OV\_SWP, 2-29  
PB\_SUM, 2-29  
SFT\_CPU, 2-29  
SFT\_FLT, 2-30  
SNPX\_RD, 2-28  
SNPX\_WT, 2-28  
STOR\_ER, 2-29  
SY\_FLT, 2-30  
SY\_PRES, 2-30  
Zmienne systemowe statusu  
APL\_FLT, 2-29  
Zmienne systemowe statusu  
IO\_PRES, 2-30  
Zmienne tymczasowe, dyskretne, 2-25  
Zmienne systemowe statusu  
SNPXACT, 2-28

## Ż

Żądanie komunikacji  
kod błędu, opis i usuwanie, 3-11