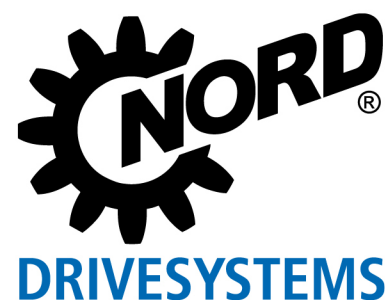


BU 0550 – pl

Funkcjonalność PLC

Dodatkowa instrukcja dla urządzeń NORDAC



Spis treści

1	Wprowadzenie	7
1.1	Informacje ogólne.....	7
1.1.1	Dokumentacja.....	7
1.1.2	Historia dokumentu.....	7
1.1.3	Ochrona praw autorskich.....	8
1.1.4	Wydawca.....	8
1.1.5	Uwagi dotyczące niniejszej instrukcji.....	8
1.2	Dodatkowo obowiązujące dokumenty.....	9
1.3	Zasady typograficzne.....	9
1.3.1	Wskazówki ostrzegawcze.....	9
1.3.2	Inne wskazówki.....	9
1.4	Zastosowanie zgodne z przeznaczeniem.....	9
2	Bezpieczeństwo	10
2.1	Dobór i kwalifikacje personelu.....	10
2.1.1	Kwalifikacje personelu.....	10
2.1.2	Wykwalifikowany elektryk.....	10
2.2	Zasady bezpieczeństwa.....	10
3	PLC	11
3.1	Uwagi ogólne.....	11
3.1.1	Specyfikacja PLC.....	12
3.1.2	Struktura PLC.....	13
3.1.2.1	Pamięć.....	13
3.1.2.2	Obraz procesu.....	13
3.1.2.3	Zadanie programowe.....	14
3.1.2.4	Przetwarzanie wartości zadanej.....	14
3.1.2.5	Przetwarzanie danych przez akumulator.....	14
3.1.3	Zakres funkcji.....	15
3.1.3.1	Motion Control Lib.....	15
3.1.3.2	Reduktor elektroniczny z funkcją latającej piły.....	15
3.1.3.3	Wizualizacja.....	15
3.1.3.4	Regulator procesu.....	16
3.1.3.5	Komunikacja CANopen.....	16
3.2	Tworzenie programów PLC.....	17
3.2.1	Ładowanie, zapisywanie i drukowanie.....	17
3.2.2	Edytor.....	18
3.2.2.1	Zmienne i deklaracja FB.....	19
3.2.2.2	Okno wprowadzania.....	20
3.2.2.3	Okno wyświetlania punktu kontrolnego i punktu przerwania.....	21
3.2.2.4	Okno komunikatów PLC.....	21
3.2.3	Przesłanie programu do urządzenia.....	22
3.2.4	Debugowanie.....	23
3.2.4.1	Punkty obserwacyjne (Watchpoints).....	23
3.2.4.2	Punkty przerwania (Breakpoints).....	23
3.2.4.3	Pojedyncze kroki (Single Step).....	24
3.2.5	Konfiguracja PLC.....	25
3.3	Bloki funkcyjne.....	26
3.3.1	CANopen.....	26
3.3.1.1	Przegląd.....	26
3.3.1.2	FB_NMT.....	27
3.3.1.3	FB_PDOConfig.....	28
3.3.1.4	FB_PDOResceive.....	31
3.3.1.5	FB_PDOSend.....	33
3.3.2	Reduktor elektroniczny z funkcją latającej piły.....	35
3.3.2.1	Przegląd.....	36
3.3.2.2	FB_FlyingSaw.....	36
3.3.2.3	FB_Gearing.....	38
3.3.3	Motion Control.....	39
3.3.3.1	MC_Control.....	41
3.3.3.2	MC_Control_MS.....	43
3.3.3.3	MC_Home.....	44

3.3.3.4	MC_Home (SK 5xxP)	45
3.3.3.5	MC_MoveAbsolute	47
3.3.3.6	MC_MoveAdditive	49
3.3.3.7	MC_MoveRelative	50
3.3.3.8	MC_MoveVelocity	51
3.3.3.9	MC_Power	53
3.3.3.10	MC_ReadActualPos	55
3.3.3.11	MC_ReadParameter	56
3.3.3.12	MC_ReadStatus	57
3.3.3.13	MC_Reset	58
3.3.3.14	MC_Stop	59
3.3.3.15	MC_WriteParameter_16 / MC_WriteParameter_32	60
3.3.4	Standard	61
3.3.4.1	Licznik odliczający w dół CTD	61
3.3.4.2	Licznik odliczający do góry CTU	62
3.3.4.3	Licznik odliczający do góry i w dół CTUD	63
3.3.4.4	R_TRIG i F_TRIG	65
3.3.4.5	R \bar{S} Flip Flop	66
3.3.4.6	SR Flip Flop	67
3.3.4.7	Opóźnienie wyłączenia TOF	68
3.3.4.8	Opóźnienie włączenia TON	69
3.3.4.9	Impuls czasowy TP	70
3.3.5	Dostęp do obszarów pamięci przetwornicy częstotliwości	71
3.3.5.1	FB_ReadTrace	71
3.3.5.2	FB_WriteTrace	73
3.3.6	Wizualizacja panelu ParameterBox	75
3.3.6.1	Przegląd wizualizacji	75
3.3.6.2	FB_DINTToPBOX	76
3.3.6.3	FB_STRINGToPBOX	79
3.3.7	FB_Capture (rejestracja szybkich zdarzeń)	81
3.3.8	FB_DinCounter	83
3.3.9	FB_FunctionCurve	85
3.3.10	FB_PIDT1	86
3.3.11	FB_ResetPosition	88
3.3.12	FB_Weigh	89
3.4	Operatory	91
3.4.1	Operatory arytmetyczne	91
3.4.1.1	ABS	91
3.4.1.2	ADD i ADD(92
3.4.1.3	DIV i DIV(93
3.4.1.4	LIMIT	93
3.4.1.5	MAX	94
3.4.1.6	MIN	94
3.4.1.7	MOD i MOD(95
3.4.1.8	MUL i MUL(95
3.4.1.9	MUX	96
3.4.1.10	SUB i SUB(96
3.4.2	Zaawansowane operatory matematyczne	97
3.4.2.1	COS, ACOS, SIN, ASIN, TAN, ATAN	97
3.4.2.2	EXP	98
3.4.2.3	LN	98
3.4.2.4	LOG	99
3.4.2.5	SQRT	99
3.4.3	Operatory bitowe	100
3.4.3.1	AND i AND(100
3.4.3.2	ANDN i ANDN(101
3.4.3.3	NOT	102
3.4.3.4	OR i OR(103
3.4.3.5	ORN i ORN(104
3.4.3.6	ROL	105
3.4.3.7	ROR	105
3.4.3.8	S i R	106
3.4.3.9	SHL	106
3.4.3.10	SHR	107
3.4.3.11	XOR i XOR(108
3.4.3.12	XORN i XORN(109
3.4.4	Operatory ładowania i zapisu	110
3.4.4.1	LD	110

3.4.4.2	LDN	110
3.4.4.3	ST	111
3.4.4.4	STN	111
3.4.5	Operatory porównywania.....	112
3.4.5.1	EQ	112
3.4.5.2	GE	112
3.4.5.3	GT	113
3.4.5.4	LE	113
3.4.5.5	LT	114
3.4.5.6	NE	114
3.5	Wartości procesowe.....	115
3.5.1	Wejścia i wyjścia.....	115
3.5.2	Wartości zadane i rzeczywiste PLC.....	123
3.5.3	Wartości zadane i rzeczywiste magistrali.....	126
3.5.4	ControlBox i ParameterBox.....	131
3.5.5	Parametry informacyjne.....	132
3.5.6	Błędy PLC.....	136
3.5.7	Parametry PLC.....	137
3.6	Języki.....	139
3.6.1	Lista instrukcji (AWL / IL).....	139
3.6.1.1	Informacje ogólne	139
3.6.2	Tekst strukturalny (ST).....	143
3.6.2.1	Informacje ogólne	143
3.6.2.2	Instrukcje	145
3.7	Skoki.....	149
3.7.1	JMP.....	149
3.7.2	JMPC.....	149
3.7.3	JMPCN.....	149
3.8	Konwersja typów.....	150
3.8.1	BOOL_TO_BYTE.....	150
3.8.2	BYTE_TO_BOOL.....	150
3.8.3	BYTE_TO_INT.....	151
3.8.4	DINT_TO_INT.....	151
3.8.5	INT_TO_BYTE.....	152
3.8.6	INT_TO_DINT.....	152
3.9	Komunikaty o usterkach PLC.....	153
4	Parametry.....	154
5	Załącznik.....	155
5.1	Wskazówki serwisowe i dotyczące uruchamiania.....	155
5.2	Dokumenty i oprogramowanie.....	155
5.3	Skróty.....	156

1 Wprowadzenie

1.1 Informacje ogólne

1.1.1 Dokumentacja

Oznaczenie:	BU 0550
Numer art.:	6075513
Seria:	Funkcjonalność PLC przetwornic częstotliwości i starterów serii
	NORDAC PRO (SK 500P ... SK 550P) (SK 520E ... SK 545E)
	NORDAC Flex (SK 200E ... SK 235E)
	NORDAC Base (SK 180E / SK 190E)
	NORDAC Link (SK 250E-FDS ... SK 280E-FDS)
	NORDAC Link (SK 155E-FDS / SK 175E-FDS)

1.1.2 Historia dokumentu

Wydanie	Seria	Wersja	Uwagi
Numer zamówienia		Oprogramowanie	
BU 0550, wrzesień 2011 6075513/ 3911	SK 540E ... SK 545E	V 2.0 R0	Pierwsze wydanie
Kolejne modyfikacje: październik 2011, luty 2013, luty 2017 Przegląd zmian ww. wydań znajduje się w odpowiednim dokumencie.			
BU 0550, maj 2019 6075513/ 1919	SK 500P ... SK 550P SK 540E ... SK 545E SK 520E ... SK 535E SK 200E ... SK 235E SK 180E / SK 190E SK 250E-FDS ... SK 280E-FDS SK 155E-FDS / SK 175E-FDS	V 1.0 R1 V 2.4 R0 V 3.2 R0 V 2.2 R0 V 1.3 R0 V 1.3 R0 V 1.2 R0	<ul style="list-style-type: none"> Implementacja typów urządzeń NORDAC PRO SK 500P ... SK 550P Dopasowania i korekty

1.1.3 Ochrona praw autorskich

Dokument, który jest częścią składową opisanego urządzenia lub opisanej funkcjonalności, należy udostępnić każdemu użytkownikowi w odpowiedniej formie.

Każda edycja lub modyfikacja dokumentu jest zabroniona.

1.1.4 Wydawca

Getriebebau NORD GmbH & Co. KG

Getriebebau-Nord-Straße 1

22941 Bargteheide, Germany

<http://www.nord.com/>

Tel. +49 (0) 45 32 / 289-0


Faks +49 (0) 45 32 / 289-2253

1.1.5 Uwagi dotyczące niniejszej instrukcji

Niniejsza instrukcja powinna pomóc w uruchomieniu funkcjonalności PLC przetwornicy częstotliwości lub startera silnika firmy Getriebebau NORD GmbH & Co. KG (zwanej krótko NORD). Jest skierowana do wykwalifikowanych elektryków, którzy planują, projektują, instalują i przygotowują programy PLC dla urządzenia (📖 punkt 2.1 "Dobór i kwalifikacje personelu"). Informacje zamieszczone w niniejszej instrukcji są podane przy założeniu, że wykwalifikowani elektrycy, którym powierzono wykonywanie prac, opanowali obsługę elektronicznej techniki napędowej, a w szczególności urządzeń firmy NORD.

Niniejsza instrukcja zawiera wyłącznie informacje i opisy funkcjonalności PLC oraz informacje dodatkowe istotne dla funkcjonalności PLC dotyczące urządzenia firmy NORD Getriebebau GmbH & Co. KG.

1.2 Dodatkowo obowiązujące dokumenty

Niniejsza instrukcja obowiązuje wyłącznie wraz z instrukcją obsługi stosowanego urządzenia. Tylko w połączeniu z niniejszym dokumentem są dostępne wszystkie informacje wymagane do bezpiecznego uruchomienia zadania napędowego. Lista dokumentów znajduje się w  punkcie 5.2 "Dokumenty i oprogramowanie".

Wymagane dokumenty znajdują się po adresem www.nord.com.

1.3 Zasady typograficzne

1.3.1 Wskazówki ostrzegawcze

Wskazówki ostrzegawcze dotyczące bezpieczeństwa użytkowników i interfejsów magistralowych są oznaczone w następujący sposób:

NIEBEZPIECZEŃSTWO

Ta wskazówka ostrzegawcza ostrzega przed zagrożeniami dla ludzi, które prowadzą do poważnych obrażeń lub do śmierci.

OSTRZEŻENIE

Ta wskazówka ostrzegawcza ostrzega przed zagrożeniami dla ludzi, które mogą prowadzić do poważnych obrażeń lub do śmierci.

OSTROŻNIE

Ta wskazówka ostrzegawcza ostrzega przed zagrożeniami dla ludzi, które mogą prowadzić do lekkich i średniociężkich obrażeń.

UWAGA


Ta wskazówka ostrzegawcza ostrzega przed szkodami materialnymi.

1.3.2 Inne wskazówki

Informacja

Ta wskazówka przedstawia porady i ważne informacje.

1.4 Zastosowanie zgodne z przeznaczeniem

Funkcjonalność PLC firmy Getriebbau NORD GmbH & Co. KG jest wspomaganym programowo funkcjonalnym rozszerzeniem dla przetwornic częstotliwości i starterów silników firmy NORD. Jest nierozłącznie połączona z urządzeniem i nie można jej stosować niezależnie od niego. Obowiązują w sposób nieograniczony zasady bezpieczeństwa urządzenia, które są opisane w odpowiedniej instrukcji ( punkt 5.2 "Dokumenty i oprogramowanie").

Funkcjonalność PLC jest stosowana przede wszystkim do rozwiązywania kompleksowych zadań napędowych z jednym lub kilkoma urządzeniami elektronicznej techniki napędowej oraz do uproszczenia związanych z napędem funkcji sterowania i monitorowania przez odpowiednio wyposażone urządzenie.

2 Bezpieczeństwo

2.1 Dobór i kwalifikacje personelu

Funkcjonalność PLC powinni uruchamiać wyłącznie wykwalifikowani elektrycy. Muszą oni posiadać wymaganą wiedzę o funkcjonalności PLC, stosowanej elektronicznej technice napędowej, narzędziach konfiguracyjnych (np. oprogramowaniu NORD CON) i urządzeniach peryferyjnych związanych z zadaniem napędowym (m.in. sterowniki).

Ponadto wykwalifikowani elektrycy muszą być zaznajomieni z instalacją, uruchomieniem i eksploatacją czujników i elektronicznej techniki napędowej oraz powinni znać i przestrzegać wszystkich przepisów zapobiegania wypadkom, dyrektyw i ustaw obowiązujących w miejscu użytkowania.

2.1.1 Kwalifikacje personelu

Wykwalifikowany personel obejmuje osoby, które ze względu na swoje fachowe wykształcenie i doświadczenie posiadają wystarczającą wiedzę z zakresu specjalnej dziedziny oraz znają odpowiednie przepisy ochrony pracy i zapobiegania wypadkom, a także ogólnie uznane przepisy techniczne.


Personel musi być uprawniony przez użytkownika urządzenia do wykonywania wymaganych czynności.

2.1.2 Wykwalifikowany elektryk

Wykwalifikowany elektryk to osoba, która ze względu na swoje fachowe wykształcenie i doświadczenie posiada wystarczającą wiedzę dotyczącą


- włączania, wyłączania, odłączania, uziemiania i oznaczania obwodów prądowych i urządzeń,
- prawidłowej konserwacji i stosowania urządzeń ochronnych zgodnie z ustalonymi normami bezpieczeństwa,
- pomocy medycznej poszkodowanym w nagłych wypadkach.

2.2 Zasady bezpieczeństwa

Stosować funkcję technologiczną Funkcjonalność PLC i urządzenie firmy Getriebebau NORD GmbH & Co. KG wyłącznie zgodnie z przeznaczeniem,  punkt 1.4 "Zastosowanie zgodne z przeznaczeniem".

Przestrzegać zaleceń zawartych w niniejszej instrukcji, aby zapewnić bezpieczne stosowanie funkcji technologicznej.

Uruchamiać urządzenie wyłącznie w stanie technicznie niezmienionym i z wymaganymi osłonami. Wszystkie przyłącza i kable powinny znajdować się w nienagannym stanie.

Prace przy urządzeniu powinien wykonywać wyłącznie wykwalifikowany personel,  punkt 2.1 "Dobór i kwalifikacje personelu".

3 PLC

3.1 Uwagi ogólne

Przetwornice częstotliwości NORD serii SK 180E/SK 190E, SK 2xxE, SK 2xxE-FDS, SK 520E – SK 545E i SK 5xxP oraz startery silników serii SK 155E-FDS/SK 175E-FDS posiadają system przetwarzania logicznego, który jest oparty na normie IEC61131-3 obowiązującej dla sterowników programowalnych (PLC). Szybkość reakcji i moc obliczeniowa PLC jest odpowiednia do wykonywania prostych zadań w środowisku przetwornicy. Wejścia przetwornicy lub informacje napływające przez magistralę polową można monitorować, analizować i przetwarzać na odpowiednie wartości zadane dla przetwornicy częstotliwości. We współpracy z innymi urządzeniami NORD możliwa jest również wizualizacja stanów urządzenia i wprowadzanie specjalnych parametrów użytkownika. W rezultacie uzyskuje się w ograniczonym zakresie potencjalne oszczędności dzięki wyeliminowaniu dotychczasowego zewnętrznego rozwiązania PLC. Obsługiwanym językiem programowania jest AWL. AWL jest komputerowym zorientowanym maszynowo tekstowym językiem programowania, którego zakres i zastosowanie jest określone w normie IEC61131-3.

 Informacja

Programowanie i pobieranie danych do urządzenia odbywa się wyłącznie za pomocą oprogramowania NORD NORD CON.

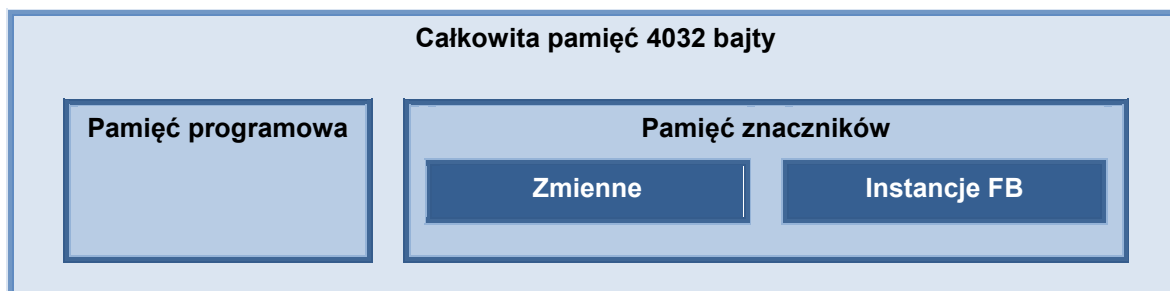
3.1.1 Specyfikacja PLC

Funkcja	Specyfikacja		
Standard	Na bazie normy IEC61131-3		
Język	Lista instrukcji (IL), tekst strukturalny (ST)		
Zadanie	Zadanie cykliczne, wywołanie programu co 5 ms		
Moc obliczeniowa	Ok. 200 poleceń AWL na 1 ms		
Pamięć programowa	SK 5xxP, SK 520E ... SK 545E, SK 2xxE, SK 2x0E-FDS	SK 190E / SK 180E	SK 155E-FDS / SK 175E-FDS
	8128 bajtów dla znaczników, funkcji i programu PLC	2032 bajty dla znaczników, funkcji i programu PLC	2028 bajtów dla znaczników, funkcji i programu PLC
Maks. możliwa liczba poleceń	Ok. 2580 poleceń	Ok. 660 poleceń	Ok. 660 poleceń
	Uwaga: Jest to wartość średnia, intensywne stosowanie znaczników, danych procesowych i funkcji znacznie minimalizuje możliwą liczbę wierszy, patrz punkt Zasoby.		
Swobodnie dostępne skrzynki CAN Mailbox	20		
Obsługiwane urządzenia	SK 5xxP SK 54xE SK 53xE / SK 52xE od V3.0 SK 2xxE od V2.0 SK 2x0E-FDS SK 180E / SK 190E SK 155E-FDS / SK 175E- FDS		

3.1.2 Struktura PLC

3.1.2.1 Pamięć

Pamięć w PLC jest podzielona na pamięć programową i pamięć znaczników. W obszarze pamięci znaczników, oprócz zmiennych, są przechowywane instancje bloków funkcyjnych. Instancja to obszar pamięci, w którym są przechowywane wszystkie wewnętrzne zmienne wejściowe i wyjściowe FB. Każda deklaracja FB wymaga własnej instancji. Granica między pamięcią programową i pamięcią znaczników jest określana dynamicznie, w zależności od wielkości obszaru znaczników.



W pamięci znaczników w obszarze zmiennych są przechowywane dwie różne klasy:

[VAR]

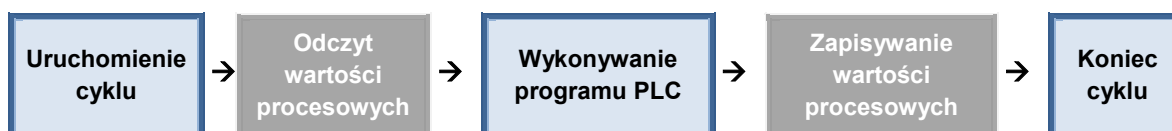
Zmienna pamięci do przechowywania informacji pomocniczych i stanów. Zmienne tego typu są na nowo inicjowane podczas każdego uruchomienia PLC. Podczas cyklu wykonywanego przez PLC zawartość pamięci pozostaje zachowana.

[VAR_ACCESS]

Służy do odczytu i opisu danych procesowych (wejścia, wyjścia, wartości zadane itd.) przetwornicy częstotliwości. Wartości te są tworzone na nowo w każdym cyklu PLC.

3.1.2.2 Obraz procesu

Urządzenie charakteryzuje kilka parametrów fizycznych, takich jak moment obrotowy, prędkość obrotowa, pozycja, wejścia, wyjścia itd. Parametry te są podzielone na wartości rzeczywiste i zadane. Można je załadować do obrazu procesu PLC i wpływać na nie. Wymagane wartości procesowe muszą być zdefiniowane na liście zmiennych w klasie VAR_ACCESS. W każdym cyklu PLC wszystkie dane procesowe przetwornicy zdefiniowane na liście zmiennych są wczytywane na nowo. Na końcu każdego cyklu PLC zapisywalne dane procesowe są przekazywane ponownie do przetwornicy, patrz poniższy rysunek.



Taki przebieg oznacza konieczność zaprogramowania cyklicznej sekwencji programowej. Programowanie pętli w celu oczekiwania na określone zdarzenia (np. zmiana poziomu na wejściu) nie daje oczekiwanego rezultatu. W przypadku bloków funkcyjnych, które mają dostęp do wartości procesowych, zachowanie to jest inne. Tutaj wartości procesowe są odczytywane po wywołaniu bloku funkcyjnego i zapisywane natychmiast po zakończeniu bloku.

i Informacja

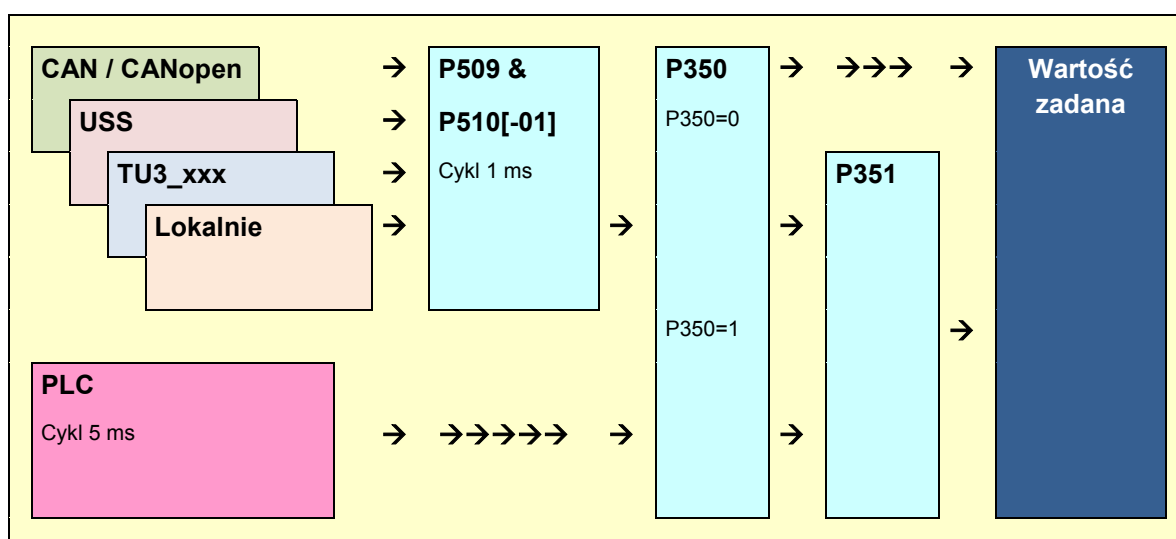
Gdy są stosowane bloki Motion MC_Power, MC_Reset, MC_MoveVelocity, MC_Move, MC_Home i MC_Stop, nie wolno stosować wartości procesowych „PLC_Control_Word” i „PLC_Set_Val1” do „PLC_Set_Val5”. W przeciwnym wypadku zmiany bloków funkcyjnych będą nadpisywane przez wartości z listy zmiennych.

3.1.2.3 Zadanie programowe

Wykonywanie programu w PLC odbywa się w ramach jednego zadania. Zadanie jest wywoływane cyklicznie co 5 ms, a jego maks. czas przetwarzania wynosi 3 ms. Jeżeli w tym czasie nie można wykonać dłuższego programu, wykonywanie programu zostanie przerwane i będzie kontynuowane w następnym zadaniu 5 ms.

3.1.2.4 Przetwarzanie wartości zadanej

Przetwornica dysponuje wieloma źródłami wartości zadanych, które ostatecznie łączą się ze sobą za pomocą kilku parametrów w jedną wynikową wartość zadaną przetwornicy częstotliwości.



Przy aktywowanym PLC (P350=1) wstępny wybór wartości zadanych ze źródeł zewnętrznych (głównych wartości zadanych) odbywa się za pomocą parametrów P509 i P510 [-01]. Parametr P351 ostatecznie określa, które wartości zadane są wykorzystywane: pochodzące z PLC lub od wartości otrzymywanych przez parametr P509/P510 [-01]. Dopuszczalne jest również wykorzystanie obu źródeł. W przypadku pomocniczych wartości (P510[-02]) funkcja PLC nie powoduje żadnych zmian. Wszystkie źródła pomocniczych wartości zadanych i PLC przekazują pomocnicze wartości zadane do przetwornicy częstotliwości na równych prawach.

3.1.2.5 Przetwarzanie danych przez akumulator

Akumulator stanowi centralną jednostkę obliczeniową PLC. Niemal wszystkie polecenia AWL działają wyłącznie w połączeniu z akumulatorem. NORD PLC posiada trzy akumulatory. Są to 32-bitowe akumulatory Akku1 i Akku2 oraz AE w formacie BOOL. AE jest stosowany do wszystkich boolowskich operacji ładowania, pamięciowych i porównywania. Załadowana wartość boolowska jest przedstawiana w AE. Operandy porównywania przesyłają wynik do AE, a skoki warunkowe są wywoływane przez AE. Akku1 i Akku2 są stosowane dla wszystkich operandów w formacie danych BYTE, INT i DINT. Akku1 jest akumulatorem głównym, a Akku2 obsługuje tylko funkcje pomocnicze. Wszystkie operandy ładowania i pamięciowe są wykonywane przez Akku1. Wszystkie operatory arytmetyczne zapisują wynik w Akku1. Podczas każdego polecenia ładowania zawartość Akku1

zostaje przesunięta do Akku2. Kolejny operator może połączyć oba akumulatory lub dokonać analizy i ponownie zapisać wynik w Akku1, zwanym jest dalej „Akku”.

3.1.3 Zakres funkcji

Sterownik PLC obsługuje szereg operatorów, funkcji i standardowych modułów funkcyjnych zdefiniowanych w normie IEC1131-3. Szczegółowy opis znajduje się w kolejnych rozdziałach. Ponadto objaśniono dodatkowo obsługiwane bloki funkcyjne.

3.1.3.1 Motion Control Lib

Biblioteka Motion Control Lib jest oparta na specyfikacji PLCopen „Function blocks for motion control”. Zawiera przede wszystkim bloki funkcyjne do przemieszczania napędu. Ponadto są dostępne bloki funkcyjne do odczytu i zapisu parametrów urządzenia.

3.1.3.2 Reduktor elektroniczny z funkcją latającej piły

Przetwornica częstotliwości dysponuje funkcjami reduktora elektronicznego (praca synchroniczna w trybie pozycjonowania) i latającej piły. Dzięki tym funkcjom przetwornica może zostać zsynchronizowana kątowo z innym napędem. Ponadto dzięki dodatkowej funkcji latającej piły jest możliwa dokładna synchronizacja pozycji z pracującym napędem. Tryb reduktora elektronicznego można uruchomić i zatrzymać w dowolnym momencie. Pozwala to na połączenie klasycznej regulacji położenia z poleceniami przemieszczeń i funkcją reduktora. Funkcja reduktora wymaga obecności przetwornicy częstotliwości NORD z wewnętrzną magistralą CAN na osi Master.

3.1.3.3 Wizualizacja

Za pomocą paneli ControlBox i ParameterBox możliwa jest wizualizacja stanu pracy i parametryzacja przetwornicy częstotliwości. Alternatywnie do wyświetlania informacji można również wykorzystać funkcjonalność CANopen Master panelu magistrali CAN PLC.

ControlBox

Najprostszą metodą wizualizacji jest użycie panelu ControlBox. Za pomocą dwóch wartości procesowych można uzyskać dostęp do 4-pozycyjnego wyświetlacza i stanu klawiatury. Pozwala to na bardzo szybkie tworzenie prostych aplikacji HMI. Aby PLC mógł uzyskać dostęp do wyświetlacza, parametr P001 musi być ustawiony na wartość „PLC-Controlbox Value”. Kolejną cechą szczególną jest brak dostępu do menu parametrów za pomocą przycisków strzałek. Zamiast tego należy równocześnie nacisnąć przyciski „On” i „Enter”.

ParameterBox

W trybie wizualizacji można ustawić za pomocą PLC każdy z 80 znaków na wyświetlaczu P-Box (4 wiersze po 20 znaków). Można przedstawiać zarówno liczby, jak i teksty. Ponadto PLC może przekazywać naciśnięcia klawiatury na P-Box. Umożliwia to realizację bardziej złożonych funkcji HMI (wyświetlanie wartości rzeczywistych, zmiana ekranu, przekazywanie wartości zadanych itd.). Dostęp do wyświetlacza P-Box odbywa się za pomocą bloków funkcyjnych w PLC. Wizualizacja odbywa się przez wyświetlanie wartości roboczych na panelu ParameterBox. Wyświetlane wartości robocze można ustawić za pomocą parametru P1003 panelu P-Box. Parametr ten znajduje się w pozycji menu głównego „Wyświetlacz”. Parametr P1003 musi być ustawiony na wartość „Wyświetlacz PLC”. Za pomocą przycisków strzałek w prawo lub w lewo można ponownie wybrać wyświetlane wartości robocze. Zostanie wtedy pokazany wyświetlacz sterowany przez PLC. Ustawienie pozostaje zachowane nawet po ponownym włączeniu.

3.1.3.4 Regulator procesu

Regulator procesu jest regulatorem PID-T1 o ograniczonej wartości wyjściowej. Za pomocą tego modułu funkcyjnego można w łatwy sposób skonfigurować w PLC złożone reguły sterowania, dzięki którym można znacznie efektywniej wykonywać szereg procesów, jak np. regulację ciśnienia, niż w przypadku powszechnie stosowanych regulatorów dwupunktowych.

3.1.3.5 Komunikacja CANopen

Oprócz standardowych kanałów komunikacyjnych PLC oferuje dalsze możliwości komunikacji. Za pośrednictwem interfejsu magistrali CAN przetwornicy częstotliwości lub magistrali systemowej można ustanowić dodatkowe relacje komunikacyjne z innymi urządzeniami. Stosowanym protokołem jest CANopen. Komunikacja jest ograniczona do przesyłania danych PDO i poleceń NMT. Funkcja PLC nie wpływa na standardową komunikację CANopen w przetwornicy częstotliwości za pomocą SDO, PDO1, PDO2 i Broadcast.


PDO (Prozess Data Objects)

Za pomocą PDO można sterować i monitorować inne przetwornice częstotliwości. Do PLC można również podłączyć urządzenia od innych dostawców. Mogą to być moduły WE/WY, enkodery CANopen, panele itd. Pozwala to na dowolne zwiększenie liczby wejść/wyjść przetwornicy częstotliwości, a także na wykorzystywanie wyjść analogowych.

NMT (Network Management Objects)

Wszystkie urządzenia CANopen muszą zostać ustawione przez moduł nadrzędny magistrali w stan „Operational” magistrali CANopen. Tylko w tym stanie magistrali jest możliwa komunikacja PDO. Jeżeli magistrala CANopen nie posiada modułu nadrzędnego, zadanie to może zrealizować PLC. Do tego celu służy moduł funkcyjny FB_NMT.

3.2 Tworzenie programów PLC

Programy PLC są tworzone wyłącznie za pomocą programu PC NORD CON. Edytor PLC można otworzyć za pomocą punktu menu „Plik/Nowy/Program PLC” lub za pomocą ikony . Przycisk ten jest aktywny tylko wtedy, gdy w przeglądarce urządzeń jest wybrane urządzenie z funkcjonalnością PLC.

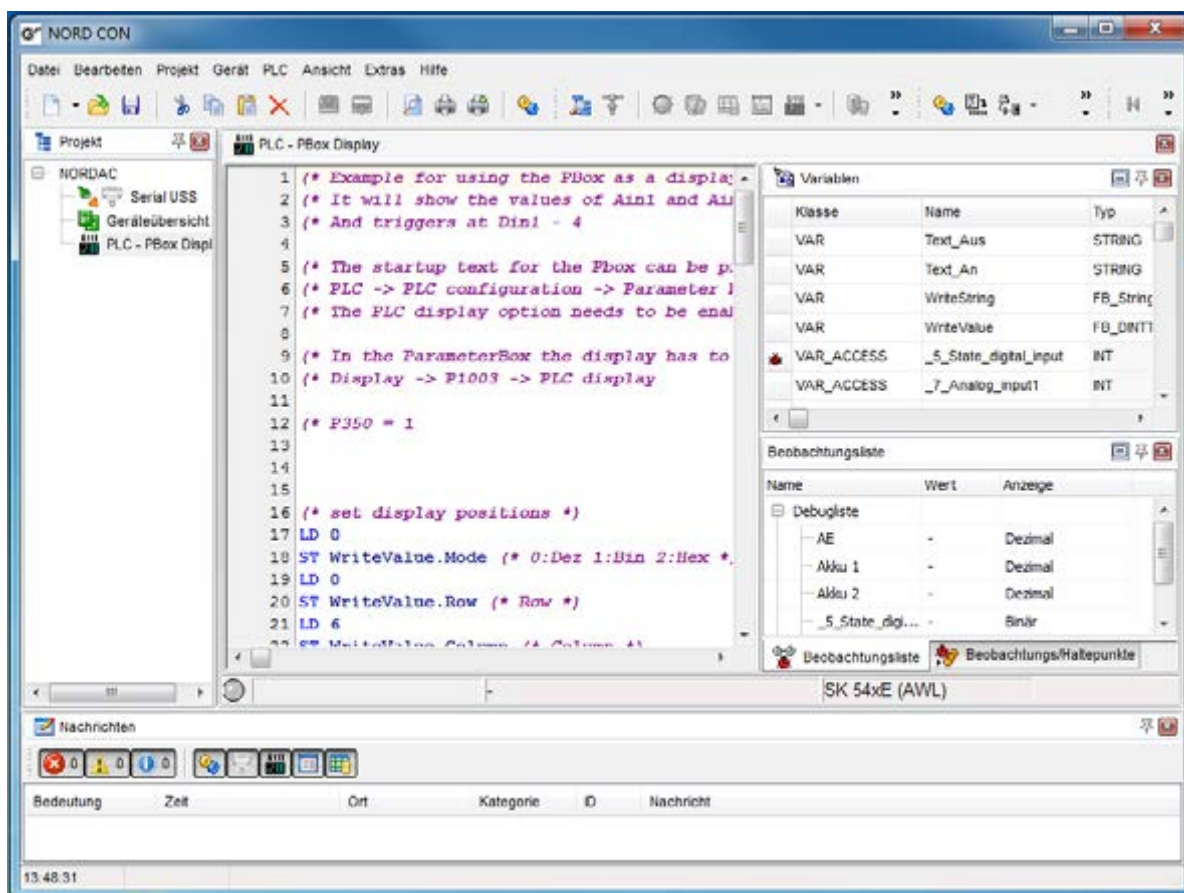
3.2.1 Ładowanie, zapisywanie i drukowanie

Funkcje ładowania, zapisywania i drukowania są wykonywane za pomocą odpowiednich punktów w menu głównym lub pasków narzędzi. Podczas otwierania zaleca się ustawienie typu pliku w oknie dialogowym „Otwórz” na „Program PLC” (*.awlx, *.nstx). Dzięki temu będą wyświetlane tylko pliki, które można odczytać za pomocą edytora PLC. Aby zapisać utworzony program PLC, musi być aktywne okno edytora PLC. Program PLC można zapisać za pomocą polecenia „Zapisz” lub „Zapisz jako”. Polecenie „Zapisz jako” pozwala określić typ pliku (program PLC (*.awlx, *.nstx)) podczas zapisywania. Aby wydrukować program PLC, musi być aktywne odpowiednie okno PLC. Aby rozpocząć wydruk, należy wybrać polecenie „Plik/Drukuj” lub nacisnąć odpowiednią ikonę.

Programy PLC można również zapisać w postaci zabezpieczonego programu PLC. W tym celu użytkownik musi ustawić typ pliku na „Zabezpieczone pliki STL” lub „Zabezpieczone pliki ST” w oknie wyboru pliku. W rezultacie program PLC zostanie zapisany w wersji zaszyfrowanej (*.awls lub *.nsts) i normalnej (*.awlx, *.nstx). Zaszyfrowany program PLC może zostać przesłany tylko do urządzenia (patrz).

3.2.2 Edytor

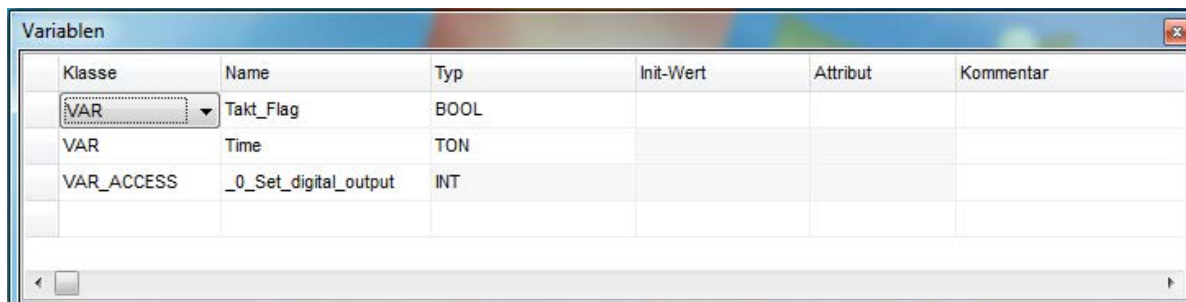
Edytor PLC jest podzielony na cztery różne okna.



Poszczególne okna zostały bardziej szczegółowo objaśnione w kolejnych punktach.

3.2.2.1 Zmienne i deklaracja FB

W tym oknie są deklarowane wszystkie zmienne, wartości procesowe i bloki funkcyjne wymagane w programie.



Klasse	Name	Typ	Init-Wert	Attribut	Kommentar
VAR	Takt_Flag	BOOL			
VAR	Time	TON			
VAR_ACCESS	_0_Set_digital_output	INT			

Zmienne

Zmienne są tworzone przez ustawienie wartości „VAR” w polu Klasa. Nazwa zmiennej może być wybrana dowolnie. W polu Typ można wybrać BOOL, BYTE, INT i DINT. W polu Wartość Init można wprowadzić wartość początkową zmiennej.

Wartości procesowe

Są tworzone przez wybór wartości „VAR_ACCESS” w polu Klasa. Nazwa zmiennej nie może być wybrana dowolnie, a pole Wartość Init jest dla tego typu zablokowane.

Moduły funkcyjne

W polu Klasa wybrać wartość „VAR”. Nazwa instancji modułu funkcyjnego (FB) może być wybrana dowolnie. Żądany FB można wybrać w polu Typ. Dla FB nie można ustawić wartości w polu Wartość Init.

Wszystkie punkty menu, które dotyczą okna zmiennych, są wywoływane za pomocą menu kontekstowego. Można w ten sposób dodawać i usuwać wpisy. Można również aktywować zmienne i zmienne procesowe do obserwacji (funkcja Watchpoint) lub debugowania (Breakpoint).

3.2.2.2 Okno wprowadzania

Okno wprowadzania służy do wprowadzania i przedstawiania programu AWL. Posiada następujące funkcje:

- Wyróżnianie elementów składni
- Zakładka
- Deklaracja zmiennych
- Debugowanie

Wyróżnianie elementów składni

Jeżeli edytor rozpoznał polecenie i przypisaną do niego zmienną, polecenie jest wyświetlane na niebiesko, a zmienna na czarno. Jeżeli tak nie jest, tekst jest przedstawiany cienką, skośną, czarną czcionką.

Zakładka

Ponieważ programy w edytorze mogą osiągać znaczną długość, funkcja zakładek pozwala zaznaczać ważne miejsca w programie w celu szybkiego do nich przejścia. Aby zaznaczyć wiersz, kursor musi znajdować się w wybranym wierszu. Wiersz z wybraną zakładką można zaznaczyć za pomocą punktu menu „Przełącz zakładkę” (menu prawego przycisku myszy). W celu przejścia do zakładki należy użyć punktu menu „Przejdź do zakładki”.

Deklaracja zmiennych

Za pomocą menu edytora „Dodaj zmienną” (prawy przycisk myszy) można zadeklarować w edytorze nowe zmienne.

Debugowanie

W edytorze można określić pozycję punktów przerwania i punktów kontrolnych dla funkcji debugowania. Można tego dokonać za pomocą punktów menu „Przełącz punkt przerwania” (Breakpoints) i „Przełącz punkt obserwacyjny” (Watchpoints). Pozycję punktów przerwania można również określić, klikając lewą krawędź okna edytora. Zmienne i wartości procesowe, które mają być odczytane z przetwornicy częstotliwości podczas debugowania, muszą być zaznaczone. Można tego dokonać w edytorze za pomocą punktów menu „Debuguj zmienną” i „Obserwuj zmienną”. W tym celu odpowiednia zmienna musi być zaznaczona, zanim zostanie wybrany żądany punkt menu.

3.2.2.3 Okno wyświetlania punktu kontrolnego i punktu przerwania

Okno posiada dwie zakładki opisane poniżej.

Punkty przerwania

W tym oknie są przedstawione wszystkie ustawione punkty przerwania i punkty kontrolne. Można je włączać i wyłączać za pomocą pól wyboru i usuwać za pomocą przycisku „Usuń”. Za pomocą prawego przycisku myszy można wywołać odpowiednie menu.

Lista obserwowanych

Tutaj są wyświetlane wszystkie zmienne wybrane do obserwacji. W kolumnie Wartość jest wyświetlana aktualna zawartość. Format wyświetlania można wybrać w kolumnie Wyświetlanie.

3.2.2.4 Okno komunikatów PLC

W tym oknie są wyprowadzane wszystkie komunikaty o stanie i błędach PLC. W przypadku prawidłowej translacji programu w oknie pojawia się komunikat „Translacja bez błędów”. W wierszu poniżej jest wyświetlane zużycie zasobów. W przypadku błędów w programie PLC pojawia się komunikat „Błąd X”, gdzie X określa liczbę błędów. W kolejnych wierszach pojawia się odpowiedni komunikat o błędzie w formacie:

[Numer wiersza]: Opis błędu

3.2.3 Przesłanie programu do urządzenia

Istnieje wiele sposobów przesłania programu PLC do urządzenia.


Bezpośrednie przesłanie programu PLC:

1. Wybrać urządzenie w drzewie projektu
2. Otworzyć menu kontekstowe (nacisnąć prawy przycisk myszy)
3. Wykonać funkcję „Przesłać program PLC do urządzenia”
4. Wybrać plik w oknie dialogowym wyboru plików i nacisnąć „Otwórz”

Przesłanie programu PLC za pomocą edytora PLC (offline):

1. Otworzyć program PLC za pomocą funkcji „Otwórz” (Plik->Otwórz)
2. Połączyć edytor PLC z urządzeniem (PLC->Połącz)
3. Dokonać translacji programu PLC
4. Przesłać program do urządzenia

Przesłanie programu PLC za pomocą edytora PLC (online):

1. Zaznaczyć urządzenie w drzewie projektu
2. Uruchomić edytor PLC
3. Otworzyć program PLC
4. Zaimportować program PLC do widoku online
5. Dokonać translacji programu PLC
6. Przesłać program do urządzenia 



Informacja

SK 1xxE-FDS - ograniczona liczba cykli zapisu

W urządzeniach SK 155E-FDS / SK 175E-FDS jako nośnik danych jest stosowana pamięć flash. Liczba cykli zapisu w pamięci flash jest mocno ograniczona. Dlatego standardowo program jest ładowany tylko do pamięci RAM. Następnie można go uruchomić i przetestować. Jeżeli PLC ma być ponownie uruchomiony, należy ponownie załadować program do urządzenia, aby zainicjować zmienne PLC. Jeżeli program ma być trwale zapisany w urządzeniu, użytkownik musi wykonać funkcję „Przesłać program do urządzenia i zapisać”.

3.2.4 Debugowanie

Ponieważ programy bardzo rzadko działają od razu, NORD PLC oferuje szereg możliwości wyszukiwania błędów. Możliwości te można podzielić na dwie kategorie, które zostaną niżej omówione.

3.2.4.1 Punkty obserwacyjne (Watchpoints)


Najprostszą metodą debugowania jest funkcja Watchpoint. Zapewnia szybką kontrolę zachowania poszczególnych zmiennych. W tym celu można ustawić punkt obserwacyjny w dowolnym miejscu w programie. Gdy PLC wykonuje określoną linię programu, funkcja może zapisać do 5 wartości i wyświetlić je na liście obserwowanych (okno „Lista obserwowanych”). W oknie wprowadzania lub w oknie zmiennych można wybrać 5 obserwowanych wartości za pomocą menu kontekstowego. Jeżeli punkt obserwacyjny został ustawiony w miejscu bez kodu programu, NORD CON wyszukuje poprzedni wiersz kodu. Gdy wykonywany program dojdzie do danego wiersza kodu, wartości zostaną zaktualizowane. Jeżeli punkt obserwacyjny zostanie ominięty w rezultacie wykonania instrukcji skoku (JMP, IF, Switch), wartości nie zostaną zaktualizowane.


Informacja

W aktualnej wersji nie można dodawać zmiennych bloków funkcyjnych do listy obserwowanych!

3.2.4.2 Punkty przerwania (Breakpoints)


Punkty przerwania pozwalają zatrzymać program PLC na żądanej linii programu. Gdy PLC dojdzie do punktu przerwania, zostanie odczytana zawartość AE, Akku1 i Akku2, a także wszystkie zmienne, które zostały wybrane za pomocą punktu menu „Debuguj zmienną” (menu kontekstowe). W programie


PLC można ustawić do 5 punktów przerwania. Funkcję można uruchomić za pomocą ikony  . Program działa do momentu wywołania punktu przerwania. Ponowne naciśnięcie ikony powoduje uruchomienie programu, który działa do momentu osiągnięcia następnego punktu przerwania. Aby



ponownie uruchomić program, należy nacisnąć ikonę  .

3.2.4.3 Pojedyncze kroki (Single Step)


Ta metoda debugowania pozwala wykonywać program PLC wiersz po wierszu w pojedynczych krokach. W każdym kroku są odczytywane z urządzenia PLC wszystkie wybrane zmienne, które są następnie wyświetlane w oknie „Lista obserwowanych”. W oknie wprowadzania lub w oknie zmiennych można wybrać obserwowane wartości za pomocą menu prawego przycisku myszy. Warunkiem debugowania w pojedynczych krokach jest ustawienie co najmniej jednego punktu przerwania przed rozpoczęciem

debugowania. Aby włączyć tryb debugowania, należy nacisnąć ikonę . Gdy program dojdzie do pierwszego punktu przerwania, można debugować kolejne linie programu w pojedynczych krokach za

pomocą ikony . Niektóre linie poleceń ukrywają kilka pojedynczych poleceń. W rezultacie może się zdarzyć, że zostaną wykonane dwa lub więcej pojedynczych kroków, zanim w oknie wprowadzania nastąpi przejście do następnego kroku. Aktualna pozycja jest wskazywana przez małą strzałkę w

lewym oknie edytora PLC. Naciśnięcie ikony  powoduje kontynuację wykonywania programu aż do następnego punktu przerwania. Aby ponownie uruchomić program, należy nacisnąć ikonę .

3.2.5 Konfiguracja PLC

Za pomocą ikony  można otworzyć okno dialogowe konfiguracji PLC. Można tutaj dokonać szereg podstawowych ustawień dla PLC, które zostaną niżej omówione.

Monitorowanie czasu cyklu

Funkcja ta monitoruje maks. czas przetwarzania dla cyklu PLC. Dzięki temu można wychwycić błędnie zaprogramowane ciągłe pętle w programie PLC. W przypadku przekroczenia dopuszczalnego czasu pojawia się błąd E22.4 w przetwornicy częstotliwości.

Dopuszczenie stosowania modułu funkcyjnego ParameterBox

Jeżeli program PLC wykorzystuje do wizualizacji panel ParameterBox, opcja ta musi być aktywowana. W przeciwnym razie podczas uruchamiania przetwornicy częstotliwości odpowiednie bloki funkcyjne wygenerują błąd kompilatora.

Nieprawidłowe dane sterujące

PLC może analizować słowa kontrolne otrzymywane przez systemy magistralowe. Słowa kontrolne przechodzą tylko wtedy, gdy jest ustawiony bit „PZD prawidłowy” (bit 10). Jeżeli PLC ma analizować słowa kontrolne niezgodne z protokołem USS, opcja ta musi być aktywowana. Bit 10 w pierwszym słowie nie jest już sprawdzany.

Gorący start po wystąpieniu błędu

Podczas uruchamiania PLC wszystkie zmienne są zawsze ładowane z wartością „0” lub wartością inicjalizacyjną. Nie ma przy tym znaczenia, czy uruchamianie odbywa się po zatrzymaniu, pobraniu programu czy po błędzie PLC. Opcja ta nie zmienia zawartości zmiennych podczas gorącego startu. Gorący start odbywa się po poleceniu zatrzymania PLC lub błędzie PLC.

Niezatrzymywanie czasu systemowego w punkcie przerwania

Podczas debugowania, gdy PLC znajduje się w punkcie przerwania lub w trybie pracy pojedynczymi krokami, czas systemowy zostaje zatrzymany. Czas systemowy stanowi podstawę dla pracy wszystkich timerów w PLC. Jeżeli podczas debugowania czas systemowy nie powinien być zatrzymywany, należy aktywować tę funkcję.

3.3 Bloki funkcyjne

Bloki funkcyjne to małe programy, które mogą przechowywać swoje parametry stanu w zmiennych wewnętrznych. Z tego względu dla każdego bloku funkcyjnego należy utworzyć oddzielną instancję na liście zmiennych NORD CON. Jeżeli np. jeden timer ma monitorować równolegle 3 czasy, musi zostać trzy razy utworzony na liście zmiennych.

i Informacja

Detekcja zbocza sygnału

Aby bloki funkcyjne mogły rozpoznać zbocze sygnału, konieczne jest dwukrotne wywołanie funkcji z różnymi stanami na wejściu.

3.3.1 CANopen

Za pośrednictwem bloków funkcyjnych PLC może konfigurować, monitorować i wysyłać kanały PDO. PLC może wysyłać lub odbierać przez PDO do 8 bajtów danych procesowych. Dostęp do każdego PDO jest możliwy za pośrednictwem własnego adresu (COB-ID). W PLC można skonfigurować do 20 obiektów PDO. Aby ułatwić obsługę, COB-ID nie jest wprowadzany bezpośrednio. Zamiast tego do FB jest przekazywany adres urządzenia i numer PDO. Wynikowy adres COB-ID jest określany na podstawie Pre-defined Connection Set (CiA DS301). Wynikają z tego następujące możliwe adresy COB-ID dla PLC.

Wysyłanie PDO		Monitorowane PDO	
PDO	COB-ID	PDO	COB-ID
PDO1	200h + adres urządzenia	PDO1	180h + adres urządzenia
PDO2	300h + adres urządzenia	PDO2	280h + adres urządzenia
PDO3	400h + adres urządzenia	PDO3	380h + adres urządzenia
PDO4	500h + adres urządzenia	PDO4	480h + adres urządzenia

Przetwornice częstotliwości NORD wykorzystuje PDO1 do przekazywania danych procesowych, a PDO2 tylko dla wartości zadanej/rzeczywistej 4 i 5.

3.3.1.1 Przegląd

Moduł funkcyjny	Objaśnienie
FB_PDConfig	Konfiguracja PLC
FB_PDOSend	Wysyłanie PDO
FB_PDORceive	Odbiór PDO
FB_NMT	Aktywacja i blokada PDO

3.3.1.2 FB_NMT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Po *Power UP* wszystkie urządzenia CAN znajdują się w stanie magistrali Pre-Operational. W tym stanie nie mogą otrzymywać i wysyłać PDO. Aby PLC mógł komunikować się z innymi urządzeniami na magistrali CAN, muszą być one ustawione w stan Operational. Z reguły odpowiada za to moduł nadrzędny magistrali. Jeżeli nie ma modułu nadrzędnego magistrali, zadanie to może zrealizować FB_NMT. Na stan wszystkich urządzeń podłączonych do magistrali można wpływać przez wejścia **PRE**, **OPE** i **STOP**. Podanie zbocza narastającego na wejście **EXECUTE** powoduje przejście wejść. Funkcja musi być wywoływana tak długo, dopóki wyjście **DONE** lub **ERROR** nie zostanie ustawione na wartość 1.

Jeżeli wyjście **ERROR** zostało ustawione na wartość 1, wtedy albo nie ma zasilania 24V w gnieździe RJ45 CAN przetwornicy albo sterownik CAN przetwornicy jest w stanie *Bus off*. Podanie zbocza opadającego na wejście **EXECUTE** powoduje zresetowanie wszystkich wyjść do wartości 0.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Wykonanie	BOOL	DONE	Wysłanie polecenia NMT	BOOL
PRE	Ustawienie wszystkich urządzeń w stan Pre-Operational	BOOL	ERROR	Błąd w FB	BOOL
OPE	Ustawienie wszystkich urządzeń w stan Operational	BOOL			
STOP	Ustawienie wszystkich urządzeń w stan Stopped	BOOL			

3.3.1.3 FB_PDOConfig

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		

FB jest stosowany do konfiguracji PDO. Za pomocą instancji tej funkcji można skonfigurować wszystkie żądane PDO. Dla każdego PDO należy tylko raz wywołać FB. Można określić do 20 PDO. Każdy PDO posiada własny zestaw parametrów. Przyporządkowanie PDO do innych CANopen FB odbywa się przez numer Messagebox. **TARGETID** zawiera adres urządzenia. W Przetwornice częstotliwości NORD jest on ustawiany w parametrze P515 lub za pomocą przełączników DIP. Żądany numer Messagebox jest wprowadzany pod PDO (patrz Wprowadzenie). **LENGTH** określa długość nadawczą PDO. Za pomocą **DIR** można określić kierunek nadawczy/odbiorczy. Podanie zbocza narastającego na wejście **EXECUTE** powoduje przesłanie danych. Wyjście **DONE** można sprawdzić bezpośrednio po wywołaniu FB. Gdy wyjście **DONE** jest ustawione na wartość 1, został skonfigurowany kanał PDO. Jeżeli **ERROR** = 1, wystąpił problem, dokładna przyczyna znajduje się w **ERRORID**. Podanie zbocza opadającego na wejście **EXECUTE** powoduje zresetowanie wszystkich wyjść do wartości 0.

Wysyłanie PDO		Monitorowane PDO	
PDO	COB-ID	PDO	COB-ID
PDO1	200h + adres urządzenia	PDO1	180h + adres urządzenia
PDO2	300h + adres urządzenia	PDO2	280h + adres urządzenia
PDO3	400h + adres urządzenia	PDO3	380h + adres urządzenia
PDO4	500h + adres urządzenia	PDO4	480h + adres urządzenia
PDO5	180h + adres urządzenia	PDO5	200h + adres urządzenia
PDO6	280h + adres urządzenia	PDO6	300h + adres urządzenia
PDO7	380h + adres urządzenia	PDO7	400h + adres urządzenia
PDO8	480h + adres urządzenia	PDO8	500h + adres urządzenia

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Wykonanie	BOOL	DONE	PDO skonfigurowany	BOOL
NUMBER	Numer Messagebox Zakres wartości = 0 do 19	BYTE	ERROR	Błąd w FB	BOOL
TARGETID	Adres urządzenia Zakres wartości = 1 do 127	BYTE	ERRORID	Kod błędu	INT
PDO	PDO Zakres wartości = 1 do 4	BYTE			
LENGTH	Długość PDO Zakres wartości = 1 do 8	BYTE			
DIR	Wysyłanie lub odbiór Wysyłanie = 1 / odbiór = 0	BOOL			
ERRORID	Objaśnienie				
0	Brak błędu				
1800h	Przekroczony zakres wartości				
1801h	Przekroczony zakres wartości TARGETID				
1802h	Przekroczony zakres wartości PDO				
1803h	Przekroczony zakres wartości LENGTH				

Informacja

Bez podwójnego wykorzystania CAN ID

Nie wolno parametryzować CAN-ID, który jest już używany przez urządzenie!

Związane adresy odbiorcze:

- CAN ID = $0x180 + P515[-01]$ PDO1
- CAN ID = $0x180 + P515[-01]+1$ CAN ID dla enkodera absolutnego
- CAN ID = $0x280 + P515[-01]$ PDO2

Związane adresy nadawcze:

- CAN ID = $0x200 + P515[-01]$ PDO1
- CAN ID = $0x300 + P515[-01]$ PDO2

Przykład w ST:

```
(* Konfiguracja PDO *)
PDOConfig(
  Execute := TRUE,
  (* Konfiguracja Messagebox 1 *)
  Number := 1,
  (* Ustawienie numeru węzła CAN *)
  TargetID := 50,
  (* Wybór PDO (standard dla słowa sterującego PDO1, wartości zadanej1, wartości
zadanej2, wartości zadanej3) *)
  PDO := 1,
  (* Określenie długości danych (standard dla PDO1 równe 8 *)
  LENGTH := 8,
  (* Wysyłanie *)
  Dir := 1);

lub

(* Konfiguracja PDO *)
PDOConfig(
  Execute := TRUE,
  (* Konfiguracja Messagebox 1 *)
  Number := 2,
  (* Ustawienie numeru węzła CAN *)
  TargetID := 50,
  (* Wybór PDO (standard dla wartości zadanej4 PDO2, wartości zadanej5 SK540E) *)
  PDO := 2,
  (* Określenie długości danych (standard dla PDO2 równe 4 *)
  LENGTH := 4,
  (* Wysyłanie *)
  Dir := 1);

lub


(* Konfiguracja PDO *)
PDOConfig(
  Execute := TRUE,
  (* Konfiguracja Messagebox 2 *)
  Number := 2,
  (* Ustawienie numeru węzła CAN *)
  TargetID := 50,
  (* Wybór PDO (standard dla słowa stanu PDO1, wartości rzeczywistej1, wartości
rzeczywistej2, wartości rzeczywistej3) *)
  PDO := 1,
  (* Określenie długości danych (standard dla PDO1 równe 8 *)
  LENGTH := 8,
  (* Odbiór *)
  Dir := 0);
```

3.3.1.4 FB_PDOReceive

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		

FB monitoruje wcześniej skonfigurowany kanał PDO pod kątem przychodzących wiadomości. Monitorowanie rozpoczyna się, gdy wejście **ENABLE** jest ustawione na wartość 1. Po wywołaniu funkcji należy sprawdzić wyjście **NEW**. Po nadejściu nowej wiadomości wyjście przyjmuje wartość 1. Wyjście **NEW** zostanie skasowane po następnym wywołaniu funkcji. **WORD1** do **WORD4** zawierają odebrane dane. Kanał **PDO** można monitorować za pośrednictwem **TIME** pod kątem odbioru cyklicznego. Jeżeli w **TIME** jest wprowadzona wartość od 1 do 32767 ms, w tym okresie musi zostać odebrana wiadomość. W przeciwnym razie FB przechodzi w stan usterki (**ERROR** = 1). Wartość 0 wyłącza funkcję. Timer monitorowania działa w krokach 5 ms. W przypadku błędu **ERROR** przyjmuje wartość 1. **DONE** w tym przypadku wynosi 0. **ERRORID** zawiera odpowiedni kod błędu. Podanie zbocza opadającego na wejście **ENABLE** powoduje zresetowanie **DONE**, **ERROR** i **ERRORID**.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Wykonanie	BOOL	NEW	Odbiór nowego PDO	BOOL
NUMBER	Numer Messagebox Zakres wartości = 0 do 19	BYTE	ERROR	Błąd w FB	BOOL
TIME	Funkcja Watchdog Zakres wartości = 0 do 32767 0 = wył. 1 do 32767 = czas monitorowania	INT	ERRORID	Kod błędu	INT
			WORD1	Odbierane dane słowo 1	INT
			WORD2	Odbierane dane słowo 2	INT
			WORD3	Odbierane dane słowo 3	INT
			WORD4	Odbierane dane słowo 4	INT
ERRORID	Objaśnienie				
0	Brak błędu				
1800h	Przekroczony zakres wartości				
1804h	Nieprawidłowa konfiguracja wybranego Box				
1805h	Brak napięcia 24 V dla sterownika magistrali lub sterownik magistrali znajduje się w stanie „Bus off”				
1807h	Timeout odbioru (funkcja Watchdog)				

 **Informacja**
Czas cyklu PLC

Czas cyklu PLC wynosi 5 ms, tzn. przy wywołaniu funkcji w programie PLC co 5 ms można odczytywać tylko jedną wiadomość CAN. Jeżeli wiadomości będą wysyłane szybciej, mogą zostać nadpisane.

Przykład w ST:

```
IF bFirstTime THEN
  (* Ustawić urządzenia w stan Pre-Operational *)
  NMT(Execute := TRUE, OPE := TRUE);
  IF not NMT.Done THEN
    RETURN;
  END_IF;

  (* Konfiguracja PDO *)
  PDOConfig(
    Execute := TRUE,
    (* Konfiguracja Messagebox 2 *)
    Number := 2,
    (* Ustawienie numeru węzła CAN *)
    TargetID := 50,
    (* Wybór PDO (standard dla słowa stanu PDO1, wartości rzeczywistej1, wartości
    rzeczywistej2, wartości rzeczywistej3) *)
    PDO := 1,
    (* Określenie długości danych (standard dla PDO1 równe 8 *)
    Length := 8,
    (* Odbiór *)
    Dir := 0);
  END_IF;

  (* Odczyt stanu i wartości rzeczywistych *)
  PDOReceive(Enable := TRUE, Number := 2);
  IF PDOReceive.New THEN
    State := PDOReceive.Word1;
    Sollwert1 := PDOReceive.Word2;
    Sollwert2 := PDOReceive.Word3;
    Sollwert3 := PDOReceive.Word4;
  END_IF
```


3.3.1.5 FB_PDOSend

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		

FB pozwala na wysyłanie PDO na wcześniej skonfigurowanym kanale. Możliwe jest wysyłanie jednokrotne lub cykliczne. Dane przeznaczone do wysyłania są wprowadzane do **WORD1 - WORD4**. Wysyłanie PDO jest możliwe niezależnie od stanu CANopen przetwornicy częstotliwości. Wybór wcześniej skonfigurowanego kanału odbywa się za pomocą **NUMBER**. Do **WORD1 - WORD4** są wprowadzane dane przeznaczone do wysyłania. Za pomocą **CYCLE** można wybrać wysyłanie jednorazowe (ustawienie=0) lub wysyłanie cykliczne. Podanie zbocza narastającego na wejście **EXECUTE** powoduje wysłanie PDO. Jeżeli **DONE** = 1 wszystkie dane wejściowe były prawidłowe, a PDO został wysłany. Jeżeli **ERROR** = 1, wystąpił problem. Dokładna przyczyna znajduje się w **ERRORID**. Podanie zbocza opadającego na wejście **EXECUTE** powoduje zresetowanie wszystkich wyjść. Podstawa czasu dla PLC to 5 ms, dotyczy to również wejścia **CYCLE**. Cykle wysyłania można wykonywać tylko dla wielokrotności 5 ms.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Wykonanie	BOOL	DONE	PDO wysłany = 1	BOOL
NUMBER	Numer Messagebox Zakres wartości = 0 do 19	BYTE	ERROR	Błąd w FB	BOOL
CYCLE	Cykl wysyłania Zakres wartości = 0 do 255 0 = wył. 1 do 255 = cykl wysyłania w ms	BYTE	ERRORID	Kod błędu	INT
WORD1	Wysyłane dane słowo 1	INT			
WORD2	Wysyłane dane słowo 2	INT			
WORD3	Wysyłane dane słowo 3	INT			
WORD4	Wysyłane dane słowo 4	INT			
ERRORID	Objaśnienie				
0	Brak błędu				
1800h	Przekroczony zakres wartości				
1804h	Nieprawidłowa konfiguracja wybranego Box				
1805h	Brak napięcia 24 V dla sterownika magistrali lub sterownik magistrali znajduje się w stanie „Bus off”				

Jeżeli **DONE** zostanie ustawione na wartość 1, wiadomość przeznaczona do wysłania została przejęta przez moduł CAN, ale jeszcze nie wysłana. Właściwe wysyłanie odbywa się równolegle w tle. Jeżeli FB ma wysłać kilka wiadomości jedna po drugiej, może się zdarzyć sytuacja, gdy przy nowym wywołaniu poprzednia wiadomość nie została jeszcze wysłana. Można to rozpoznać po braku ustawienia wartości 1 dla sygnałów **DONE** i **ERROR** po wywołaniu **CAL**. W tym przypadku można

powtarzać wywołanie **CAL**, dopóki jeden z obu sygnałów nie zostanie ustawiony na wartość 1. Jeżeli jeden FB ma opisywać kilka różnych CAN-ID, jest to możliwe za pomocą nowej konfiguracji FB. Nie może to jednak nastąpić w tym samym cyklu PLC co wysyłanie. W przeciwnym wypadku istnieje niebezpieczeństwo, że wiadomość przeznaczona do wysłania zostanie usunięta podczas konfiguracji przez FB_PDOConfig.

Przykład w ST:

```
IF bFirstTime THEN
  (* Ustawić urządzenia w stan Pre-Operational *)
  NMT(Execute := TRUE, OPE := TRUE);
  IF not NMT.Done THEN
    RETURN;
  END_IF;

  (* Konfiguracja PDO *)
  PDOConfig(
    Execute := TRUE,
    (* Konfiguracja Messagebox 2 *)
    Number := 2,
    (* Ustawienie numeru węzła CAN *)
    TargetID := 50,
    (* Wybór PDO (standard dla słowa stanu PDO1, wartości rzeczywistej1, wartości
    rzeczywistej2, wartości rzeczywistej3) *)
    PDO := 1,
    (* Określenie długości danych (standard dla PDO1 równe 8 *)
    Length := 8,
    (* Odbiór *)
    Dir := 0);
  END_IF;

  (* Odczyt stanu i wartości rzeczywistych *)
  PDOReceive(Enable := TRUE, Number := 2);
  IF PDOReceive.New THEN
    State := PDOReceive.Word1;
    Sollwert1 := PDOReceive.Word2;
    Sollwert2 := PDOReceive.Word3;
    Sollwert3 := PDOReceive.Word4;
  END_IF
```

3.3.2 Reduktor elektroniczny z funkcją latającej piły

Dla *reduktora elektronicznego* („synchronizacja kątowna”) i podfunkcji *latającej piły* istnieją dwa bloki funkcyjne, które umożliwiają sterowanie tymi funkcjami. Ponadto należy ustawić różne parametry w celu zapewnienia prawidłowego funkcjonowania obu bloków funkcyjnych w przetwornicy częstotliwości Master i Slave. Jest to pokazane w poniższej tabeli na przykładzie SK 540E.

Master FU			Slave FU		
Parametr	Ustawienie	Znaczenie	Parametr	Ustawienie	Znaczenie
P502[-01]	20	Częst. zadana wg rampy częst.	P509	10 *	CANopen Broadcast *
P502[-02]	15	Bież. poz. w Inc. High – Word	P510[-01]	10	CANopen Broadcast
P502[-03]	10	Bież. poz. w Inc. Low – Word	P510[-02]	10	CANopen Broadcast
P503	3	CANopen	P505	0	0,0 Hz
P505	0	0,0 Hz	P515[-02]	P515[-03] _{Master}	Adres Broadcast
P514	5	250 kbd (min. 100 kbd)	P546[-01]	4	Dodawanie częstotliwości
P515[-03]	P515[-02] _{Slave}	Adres Broadcast Master	P546[-02]	24	Zad. poz. Inc. High – Word
			P546[-03]	23	Zad. poz. Inc. Low – Word
			P600	1,2	Regulacja położenia wł.
			Tylko dla FB_Gearing		
			P553[-01]	21	Poz. zad. poz. Low Word
			P553[-02]	22	Poz. zad. poz. High Word

* (P509) nie musi być ustawiony na {10} „CANopen Broadcast”. Master (P502 [-01]) musi być jednak ustawiony na {21} „Częstotliwość bieżąca bez poślizgu”.

Informacja

Położenie rzeczywiste - format transmisji

Położenie rzeczywiste Master musi być przekazywane w formacie „Inkrementy” (Inc).

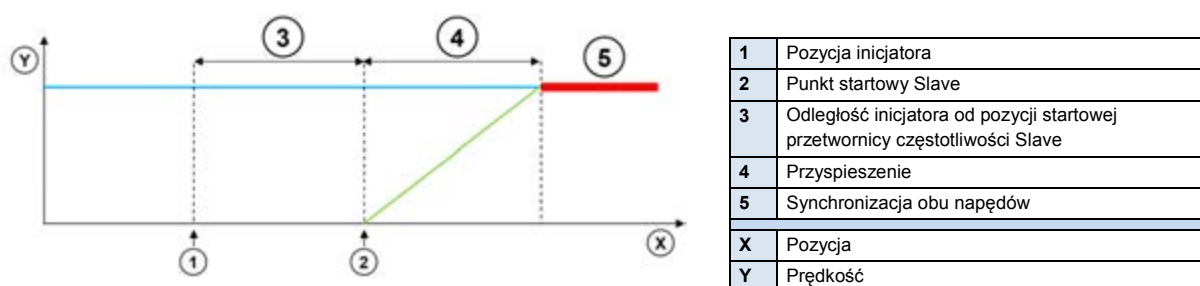
3.3.2.1 Przegląd

Moduł funkcyjny	Objaśnienie
FB_Gearing	FB dla prostej funkcji reduktora
FB_FlyingSaw	FB dla funkcji reduktora z funkcją latającej piły

3.3.2.2 FB_FlyingSaw

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Funkcja latającej piły jest rozszerzeniem funkcji reduktora. Za pomocą tej funkcji jest możliwa dokładna synchronizacja pozycji z pracującym napędem. W przeciwieństwie do FB_Gearing synchronizacja jest względna, tzn. oś Slave przemieszcza się synchronicznie do pozycji Master zajmowanej podczas uruchamiania „latającej piły”. Proces synchronizacji pokazano na poniższym rysunku.



Po uruchomieniu funkcji przetwornica częstotliwości Slave przyspiesza do prędkości osi Master. Rampę przyspieszenia określa krzywa **ACCELERATION**. W przypadku małej prędkości rampa jest bardziej płaska, a w przypadku wysokich prędkości Master rampa jest bardziej stroma dla przetwornicy częstotliwości Slave. Droga rozpędzania jest podana w obrotach (1000 = 1,000 obr), gdy parametr P553 jest zdefiniowany jako pozycja zadana. Jeżeli w parametrze P553 zostanie użyta pozycja zadana INC, droga rozpędzania jest podana w inkrementach.

Jeżeli inicjator zostanie ustawiony przed pozycją napędu Slave w odległości zapisanej w **ACCELERATION**, Slave zostanie dokładnie zsynchronizowany z pozycją zadziałania napędu Master.

FB musi być włączony przez wejście **ENABLE**. Funkcję można uruchomić albo za pomocą wejścia cyfrowego (P420[-xx] = 64, *Uruchomienie latającej piły*) albo za pomocą **EXECUTE**. Przetwornica częstotliwości przyspiesza wtedy do prędkości osi Master. Po osiągnięciu synchronizacji z osią Master wyjście **DONE** przełącza się na wartość 1.

Wyłączenie funkcji reduktora następuje za pomocą wejścia **STOP** lub cyfrowej funkcji wejścia P420[-xx] = 77, *Zatrzymanie latającej piły*, przetwornica częstotliwości hamuje do 0 Hz i zatrzymuje się. Za pomocą wejścia **HOME** można dokonać ustawienia przetwornicy w pozycji absolutnej 0. Po wykonaniu polecenia **HOME** lub **STOP** przyporządkowane wyjście jest aktywne. Funkcję reduktora można ponownie uruchomić przez ponowną aktywację wejścia **EXECUTE** lub wejścia cyfrowego. Za pomocą cyfrowej funkcji wejścia (P420[-xx] = 63, *Wyłączenie synchronizacji*) można zatrzymać funkcję reduktora, a następnie dokonać przemieszczenia do pozycji absolutnej 0.

Jeżeli wykonanie funkcji zostanie przerwane przez funkcję MC_Stop, **ABORT** zostanie ustawione na wartość 1. W przypadku błędu **ERROR** przyjmuje wartość 1, a **ERRORID** zostanie ustawione na kod błędu. Wszystkie trzy wyjścia zostaną resetowane, gdy **ENABLE** przełączy się na wartość 0.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Aktywacja	BOOL	VALID	Osiągnięto zadaną częstotliwość	BOOL
EXECUTE	Uruchomienie synchronizacji	BOOL	DONEHOME	Zakończono ustawienie w pozycji wyjściowej	
STOP	Zatrzymanie synchronizacji	BOOL	DONESTOP	Wykonane polecenie zatrzymania	
HOME	Przemieszczenie na pozycję 0	BOOL	ABORT	Polecenie przerwane	BOOL
ACCELERATION	Droga rozpędzania (1 obr = 1.000)	DINT	ERROR	Błąd w FB	BOOL
			ERRORID	Kod błędu	INT
ERRORID	Objaśnienie				
0	Brak błędu				
1000h	FU nie jest uaktywniona				
1200h	Regulacja położenia nie jest aktywowana				

3.3.2.3 FB_Gearing

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Za pomocą modułu funkcyjnego FB_Gearing można zsynchronizować pozycję i prędkość obrotową przetwornicy częstotliwości w stosunku do przetwornicy Master. Slave, który używa tej funkcji, zawsze podąża za przemieszczeniami przetwornicy Master.

Synchronizacja odbywa się w sposób absolutny, tzn. pozycje Slave i Master są zawsze takie same.

Informacja

Jeżeli Slave przełączy się w tryb reduktora w innej pozycji niż Master, Slave przemieści się do pozycji Master z maks. częstotliwością.

Jeżeli jest zdefiniowane przełożenie, to po ponownym włączeniu zostanie określona nowa pozycja.

Wartość pozycji, do której ma zostać przeprowadzona synchronizacja, i prędkość obrotowa, muszą zostać przesłane kanałem Broadcast. Funkcja jest aktywowana przez wejście **ENABLE**, regulacja położenia i stopień końcowy muszą być aktywne. Stopień końcowy można aktywować np. za pomocą funkcji MC_Power. Gdy **ENABLE** zostanie ustawione na wartość 0, przetwornica częstotliwości hamuje do 0 Hz i zatrzymuje się. Przetwornica ponownie znajdzie się w trybie regulacji położenia. W przypadku aktywacji MC_Stop przetwornica częstotliwości opuszcza tryb reduktora, a wyjście **ABORT** zostanie ustawione na wartość 1. W przypadku błędów w FB **ERROR** przyjmuje wartość 1, a przyczyna błędu znajduje się w **ERRORID**. Ustawiając **ENABLE** na wartość 0, można zresetować **ERROR**, **ERRORID** i **ABORT**.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Synchronizacja aktywna	BOOL	VALID	Funkcja reduktora jest aktywna	BOOL
RELATIVE	Tryb względny (od V2.1)	BOOL	ABORT	Polecenie przerwane	BOOL
			ERROR	Błąd w FB	BOOL
			ERRORID	Kod błędu	INT
ERRORID	Objaśnienie				
0	Brak błędu				
1000h	FU nie jest uaktywniona				
1200h	Regulacja położenia nie jest aktywowana				
1201h	Wartość zadana PLC pozycja High nie jest sparametryzowana				
1202h	Wartość zadana PLC pozycja Low nie jest sparametryzowana				

3.3.3 Motion Control

Biblioteka Motion Control Lib jest oparta na specyfikacji PLCopen „Function blocks for motion control”. Zawiera bloki funkcyjne do sterowania i przemieszczania przetwornicy częstotliwości oraz zapewnia dostęp do jej parametrów. Aby bloki Motion funkcjonowały, należy dokonać szereg ustawień w parametrach urządzenia.

Blok funkcyjny	Wymagane ustawienia
MC_MoveVelocity	<ul style="list-style-type: none"> • P350 = PLC aktywny • P351 = Główna wartość zadana pochodzi z PLC • P553 [-xx] = Częstotliwość zadana • P600 = Regulacja położenia (tryb pozycjonowania) jest wyłączona
MC_MoveAbsolute	<ul style="list-style-type: none"> • P350 = PLC aktywny • P351 = Główna wartość zadana pochodzi z PLC • P600 = Regulacja położenia (tryb pozycjonowania) jest włączona • W parametrze P553 [-xx] (wartości zadane_PLC) pozycja zadana High Word musi być sparametryzowana • W parametrze P553 [-xx] (wartości zadane_PLC) pozycja zadana Low Word musi być sparametryzowana • W parametrze P553 [-xx] (wartości zadane_PLC) częstotliwość zadana musi być sparametryzowana
MC_MoveRelative	
MC_MoveAdditive	
MC_Home	
MC_Power	<ul style="list-style-type: none"> • P350 = PLC aktywny • P351 = Słowo sterujące pochodzi z PLC
MC_Reset	
MC_Stop	

Informacja

Wartości zadane_PLC od 1 do 5 i słowo sterujące PLC można również opisać za pomocą zmiennych procesowych. W przypadku stosowania Motion Control FB w tabeli zmiennych nie powinny być deklarowane odpowiednie zmienne procesowe, ponieważ w przeciwnym wypadku wyjścia Motion Control FB zostaną nadpisane.

Informacja

Detekcja zbocza sygnału

Aby bloki funkcyjne mogły rozpoznać zbocze sygnału, konieczne jest dwukrotne wywołanie funkcji z różnymi stanami na wejściu.

Blok funkcyjny	Objaśnienie
MC_ReadParameter	Dostęp do odczytu parametrów urządzenia
MC_WriteParameter	Dostęp do zapisu parametrów urządzenia
MC_MoveVelocity	Polecenie przemieszczenia w trybie prędkości obrotowej
MC_MoveAbsolute	Polecenie przemieszczenia z absolutną specyfikacją pozycji
MC_MoveRelative	Polecenie przemieszczenia ze względną specyfikacją pozycji
MC_MoveAdditive	Polecenie przemieszczenia z addytywną specyfikacją pozycji
MC_Home	Uruchomienie ustawienia do pozycji wyjściowej
MC_Power	Włączanie/wyłączanie napięcia silnika
MC_ReadStatus	Stan urządzenia
MC_ReadActualPos	Odczyt aktualnej pozycji
MC_Reset	Reset błędu w urządzeniu
MC_Stop	Zatrzymanie wszystkich aktywnych poleceń przemieszczenia

3.3.3.1 MC_Control

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

FB służy do sterowania FU i pozwala wykorzystywać słowo sterujące FU w bardziej szczegółowy sposób niż MC_Power. FU jest sterowana przez wejścia **ENABLE**, **DISABLEVOLTAGE** i **QUICKSTOP**, patrz poniższa tabela.

Moduł wejść			Zachowanie przetwornicy częstotliwości
ENABLE	QUICKSTOP	DISABLEVOLTAGE	
High	Low	Low	Włączenie przetwornicy częstotliwości
Low	Low	Low	Przetwornica częstotliwości hamuje do 0 Hz (P103), a następnie odłącza napięcie silnika.
X	X	High	Przetwornica częstotliwości natychmiast odłącza się od zasilania, silnik zatrzymuje się bez hamowania.
X	High	Low	Przetwornica częstotliwości wykonuje szybkie zatrzymanie (P426), a następnie odłącza napięcie silnika.

Za pomocą wejścia **PARASET** można ustawić aktywny zestaw parametrów.

Gdy wyjście **STATUS** = 1, FU jest włączona, a do silnika jest doprowadzany prąd.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Aktywacja	BOOL	STATUS	Doprowadzanie prądu do silnika	BOOL
DISABLEVOLTAGE	Odłączenie napięcia	BOOL	ERROR	Błąd w FB	BOOL
QUICKSTOP	Szybkie zatrzymanie	BOOL	ERRORID	Kod błędu	INT
PARASET	Aktywny zestaw parametrów Zakres wartości: 0 - 3	BYTE			
ERRORID	Objaśnienie				
0	Brak błędu				
1001h	Funkcja zatrzymania jest aktywna				
1300h	FU znajduje się w nieoczekiwanym stanie				

Przykład w ST:

```
(* Aktywować urządzenie za pomocą Dig3*)
Control.Enable := _5_State_digital_input.2;
(* Zestawy parametrów są definiowane przez Dig1 i Dig2. *)
Control.ParaSet := INT_TO_BYTE(_5_State_digital_input and 2#11);
Control;
(* Czy urządzenie jest aktywowane? *)
if Control.Status then
  (* Czy ma nastąpić przesuw do innej pozycji? *)
  if SaveBit3 <> _5_State_digital_input.3 then
    SaveBit3 := _5_State_digital_input.3;
    if SaveBit3 then
      Move.Position := 500000;
    else
      Move.Position := 0;
    end_if;

    Move(Execute := False);
  end_if;
end_if;

(* Przesuw do pozycji, gdy urządzenie jest aktywne. *)
Move(Execute := Control.Status);
```

3.3.3.2 MC_Control_MS

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność							X

FB służy do sterowania starterem (MS).

Moduł wejść				Zachowanie przetwornicy częstotliwości
ENABLE_RIGHT	ENABLE_LEFT	QUICKSTOP	DISABLEVOLTAGE	
High	Low	Low	Low	Włączenie MS, obroty w prawo
Low	High	Low	Low	Włączenie MS, obroty w lewo
High	High	Low	Low	Wyłączenie MES
Low	Low	Low	Low	MS hamuje do 0 Hz (P103), a następnie odłącza napięcie silnika
X	X	X	High	MS natychmiast odłącza się od zasilania, silnik zatrzymuje się bez hamowania
X	X	High	Low	MS wykonuje szybkie zatrzymanie (P426), a następnie odłącza napięcie silnika

(X = poziom na wejściu nie jest istotny)

Gdy wyjście **STATUS** = 1, MS jest włączony, a do silnika jest doprowadzany prąd.

Jeżeli **OPENBRAKE** jest ustawione na wartość 1, hamulec zostaje zwolniony.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE_RIGHT	Obroty prawe	BOOL	STATUS	Doprowadzanie prądu do silnika	BOOL
ENABLE_LEFT	Obroty lewe	BOOL	ERROR	Błąd w FB	BOOL
DISABLEVOLTAGE	Odłączenie napięcia	BOOL	ERRORID	Kod błędu	INT
QUICKSTOP	Szybkie zatrzymanie	BOOL			
OPENBRAKE	Zwolnienie hamulca	BOOL			
ERRORID	Objaśnienie				
0	Brak błędu				
1001h	Funkcja zatrzymania jest aktywna				
1300h	MS znajduje się w nieoczekiwanym stanie				

3.3.3.3 MC_Home

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność		X	X	X	X	X	

Powoduje uruchomienie przez przetwornicę częstotliwości przesuwu do punktu referencyjnego, gdy **EXECUTE** zmienia wartość z 0 na 1 (zbcze). Przetwornica częstotliwości przemieszcza się z częstotliwością zadaną wprowadzoną w **VELOCITY**. Gdy wejście z sygnałem referencyjnym pozycji (P420 [-xx] = punkt referencyjny) będzie aktywne, nastąpi zmiana kierunku obrotu. W przypadku zbcza opadającego sygnału referencyjnego pozycji jest przejmowana wartość zapisana w **POSITION**. Następnie przetwornica częstotliwości hamuje do 0 Hz, sygnał **DONE** zostanie ustawiony na wartość 1. Podczas całego przesuwu **HOME** wyjście **BUSY** jest aktywne. Gdy wejście **Mode** jest ustawione na wartość **True**, urządzenie po ustawieniu w pozycji wyjściowej zatrzymuje się w środku inicjatora. Nie można stosować wejścia **Position**.

Do zatrzymania procesu (np. przez inny moduł funkcyjny MC) jest wykorzystywany **COMMANDABORTED**.

W przypadku błędu **ERROR** przyjmuje wartość 1. **DONE** w tym przypadku wynosi 0. **ERRORID** zawiera odpowiedni kod błędu.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Aktywacja	BOOL	DONE	Osiągnięto pozycję zadaną	BOOL
POSITION	Pozycja zadana	DINT	COMMAND-ABORTED	Polecenie przerwane	BOOL
VELOCITY	Częstotliwość zadana	INT	ERROR	Błąd w FB	BOOL
MODE	Tryb (od V2.1)	BOOL	ERRORID	Kod błędu	INT
			BUSY	Ustawienie w pozycji wyjściowej aktywne	BOOL
ERRORID	Objaśnienie				
0	Brak błędu				
1000h	FU nie jest uaktywniona				
1200h	Regulacja położenia nie jest aktywowana				
1201h	Nie wprowadzono pozycji High w wartościach zadanych PLC (P553)				
1202h	Nie wprowadzono pozycji Low w wartościach zadanych PLC (P553)				
1D00h	Enkodery absolutne nie są obsługiwane				

3.3.3.4 MC_Home (SK 5xxP)

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X						

Powoduje uruchomienie przez przetwornicę częstotliwości przesuwu do punktu referencyjnego, gdy **EXECUTE** zmienia wartość z 0 na 1 (zbczce). Przetwornica częstotliwości przemieszcza się z częstotliwością zadaną wprowadzoną w **VELOCITY**. Gdy wejście z sygnałem referencyjnym pozycji (P420 [-xx] = punkt referencyjny) będzie aktywne, nastąpi zmiana kierunku obrotu. W przypadku zbczka opadającego sygnału referencyjnego pozycji jest przejmowana wartość zapisana w **POSITION**. Następnie przetwornica częstotliwości hamuje do 0 Hz, sygnał **DONE** zostanie ustawiony na wartość 1. Podczas całego przesuwu **HOME** wyjście **BUSY** jest aktywne.

Do zatrzymania procesu (np. przez inny moduł funkcyjny MC) jest wykorzystywany **COMMANDABORTED**.

W przypadku błędu **ERROR** przyjmuje wartość 1. **DONE** w tym przypadku wynosi 0. **ERRORID** zawiera odpowiedni kod błędu.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Aktywacja	BOOL	DONE	Osiągnięto pozycję zadaną	BOOL
POSITION	Pozycja zadaną	DINT	COMMAND-ABORTED	Polecenie przerwane	BOOL
VELOCITY	Częstotliwość zadaną	INT	ERROR	Błąd w FB	BOOL
MODE	patrz niżej	BYTE	ERRORID	Kod błędu	INT
			BUSY	Przesuw Home aktywny	BOOL
ERRORID	Objaśnienie				
0	Brak błędu				
1000h	FI nie jest uaktywniona				
1200h	Regulacja położenia nie jest aktywowana				
1201h	Nie wprowadzono pozycji High w wartościach zadanych PLC (P553)				
1202h	Nie wprowadzono pozycji Low w wartościach zadanych PLC (P553)				
1D00h	Enkodery absolutne nie są obsługiwane				
1D01h	Przekroczenie lub nieosiągnięcie zakresu wartości wejścia „Mode” (P623)				

Mode

Wartość	Objaśnienie
1..14	Metoda punktów referencyjnych, patrz P623
15	<p>Po osiągnięciu wyłącznika punktu referencyjnego napęd dokonuje nawrotu. Podczas opuszczania wyłącznika punktu referencyjnego (zbocze opadające) wartość pozycji zostanie przejęta jako punkt referencyjny.</p> <p>Punkt referencyjny zwykle znajduje się po stronie wyłącznika punktu referencyjnego, na której został rozpoczęty przesuw do punktu referencyjnego.</p> <p>Uwaga: W przypadku „przejechania” wyłącznika punktu referencyjnego (zbyt wąski wyłącznik, zbyt duża prędkość) podczas opuszczania wyłącznika punktu referencyjnego (zbocze opadające) wartość pozycji zostanie przejęta jako punkt referencyjny. Punkt referencyjny nie znajduje się zatem po stronie wyłącznika punktu referencyjnego, na której został rozpoczęty przesuw do punktu referencyjnego.</p> <p>(P623 = [15] metoda Nord 1)</p>
16	<p>Jak 15, ale przejechanie wyłącznika punktu referencyjnego nie prowadzi do przejęcia wartości pozycji jako punktu referencyjnego. Dopiero po zakończeniu nawrotu zbocze opadające prowadzi do przejęcia wartości pozycji jako punktu referencyjnego.</p> <p>Punkt referencyjny znajduje się zatem na pewno po stronie wyłącznika punktu referencyjnego, na której został rozpoczęty przesuw do punktu referencyjnego.</p> <p>(P623 = [16] metoda Nord 2)</p>
17	<p>W przypadku przejechania wyłącznika punktu referencyjnego podczas przesuwu do punktu referencyjnego (zbocze narastające → zbocze opadające) napęd przejmuje wartość środkową obu pozycji i ustawia ją jako punkt referencyjny. Napęd dokonuje nawrotu i zatrzymuje się w określonym w ten sposób punkcie referencyjnym.</p> <p>(P623 = [17] metoda Nord 3)</p>
18..34	Metoda punktów referencyjnych, patrz P623

3.3.3.5 MC_MoveAbsolute

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Zapisuje wartość zadaną pozycji i prędkości w przetwornicy częstotliwości, gdy **EXECUTE** zmienia wartość z 0 na 1 (zbocze). Częstotliwość zadana **VELOCITY** jest przekazywana zgodnie ze skalowaniem wprowadzonym w MC_MoveVelocity.

POSITION:
MODE = False:

Pozycja zadana wynika z wartości przekazanej do **POSITION**.

MODE = True:

Wartość przekazana do **POSITION** odpowiada indeksowi z parametru P613 zwiększonemu o 1.
Pozycja zapisana w indeksie parametru odpowiada pozycji zadanej.

Przykład:

Mode = True; Position = 12

FB przemieszcza się do pozycji podanej w aktualnym zestawie parametrów P613[-13].

Gdy przetwornica osiągnie pozycję zadaną, **DONE** zostanie ustawione na wartość 1. Wartość **DONE** zostanie skasowana przez zresetowanie **EXECUTE**. Jeżeli wartość **EXECUTE** zostanie skasowana przed osiągnięciem pozycji docelowej, **DONE** zostanie ustawione na wartość 1 dla jednego cyklu. Podczas przemieszczania do pozycji zadanej **BUSY** jest aktywne. Do zatrzymania procesu (np. przez inny moduł funkcyjny MC) jest wykorzystywany **COMMANDABORTED**. W przypadku błędu **ERROR** przyjmuje wartość 1, a **ERRORID** zostanie ustawione na odpowiedni kod błędu. **DONE** w tym przypadku wynosi 0. Podanie zbocza opadającego na wejście **EXECUTE** powoduje zresetowanie wszystkich wyjść do wartości 0.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Aktywacja	BOOL	DONE	Osiągnięto pozycję zadaną	BOOL
POSITION	Pozycja zadana	DINT	BUSY	Pozycja zadana nie została osiągnięta	BOOL
VELOCITY	Częstotliwość zadana	INT	COMMAND-ABORTED	Polecenie przerwane	BOOL
MODE	Tryb źródła pozycji zadanej	BOOL	ERROR	Błąd w FB	BOOL
			ERRORID	Kod błędu	INT
ERRORID	Objaśnienie				
0	Brak błędu				
0x1000	FU nie jest uaktywniona				
0x1200	Regulacja położenia nie jest aktywowana				
0x1201	Nie wprowadzono pozycji High w wartościach zadanych PLC (P553)				
0x1202	Nie wprowadzono pozycji Low w wartościach zadanych PLC (P553)				

Przykład w ST:

```
(* Urządzenie zostanie aktywowane, gdy DIG1 = TRUE *)
Power(Enable := _5_State_digital_input.0);
IF Power.Status THEN
  (* Urządzenie jest aktywowane i przemieszcza się do pozycji 20000 z 50% maks.
  częstotliwości.
  Do tej czynności silnik wymaga enkodera, a regulacja położenia musi być aktywna. *)
  MoveAbs(Execute := _5_State_digital_input.1, Velocity := 16#2000, Position := 20000);
END_IF
```


3.3.3.6 MC_MoveAdditive

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Odpowiada we wszystkich punktach MC_MoveAbsolute z wyjątkiem wejścia **DISTANCE**. Pozycja zadana wynika z dodania aktualnej pozycji zadanej i wartości przekazanej przez **DISTANCE**.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Aktywacja	BOOL	DONE	Osiągnięto pozycję zadaną	BOOL
DISTANCE	Pozycja zadana	DINT	COMMAND-ABORTED	Polecenie przerwane	BOOL
VELOCITY	Częstotliwość zadana	INT	ERROR	Błąd w FB	BOOL
MODE	Tryb źródła pozycji zadanej	BOOL	ERRORID	Kod błędu	INT
			BUSY	Pozycja zadana nie została osiągnięta	BOOL
ERRORID	Objaśnienie				
0	Brak błędu				
1000h	FU nie jest uaktywniona				
1200h	Regulacja położenia nie jest aktywowana				
1201h	Nie wprowadzono pozycji High w wartościach zadanych PLC (P553)				
1202h	Nie wprowadzono pozycji Low w wartościach zadanych PLC (P553)				

3.3.3.7 MC_MoveRelative

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Odpowiada we wszystkich punktach MC_MoveAbsolute z wyjątkiem wejścia **DISTANCE**. Pozycja zadana wynika z dodania aktualnej pozycji rzeczywistej i wartości przekazanej przez **DISTANCE**.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Aktywacja	BOOL	DONE	Osiągnięto pozycję zadaną	BOOL
DISTANCE	Pozycja zadana	DINT	COMMAND-ABORTED	Polecenie przerwane	BOOL
VELOCITY	Częstotliwość zadana	INT	ERROR	Błąd w FB	BOOL
MODE	Tryb źródła pozycji zadanej	BOOL	ERRORID	Kod błędu	INT
			BUSY	Pozycja zadana nie została osiągnięta	BOOL
ERRORID	Objaśnienie				
0	Brak błędu				
1000h	FU nie jest uaktywniona				
1200h	Regulacja położenia nie jest aktywowana				
1201h	Nie wprowadzono pozycji High w wartościach zadanych PLC (P553)				
1202h	Nie wprowadzono pozycji Low w wartościach zadanych PLC (P553)				

3.3.3.8 MC_MoveVelocity

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Dokonuje ustawienia częstotliwości zadanej dla przetwornicy częstotliwości, gdy **EXECUTE** zmienia wartość z 0 na 1 (zbcze). Gdy przetwornica częstotliwości osiągnie częstotliwość zadaną, **INVELOCITY** zostanie ustawione na wartość 1. Gdy FU przyspiesza do częstotliwości zadanej, wyjście **BUSY** jest aktywne. Jeżeli **EXECUTE** zostało już ustawione na wartość 0, **INVELOCITY** zostanie ustawione na wartość 1 dla jednego cyklu. Do zatrzymania procesu (np. przez inny moduł funkcyjny MC) jest wykorzystywany **COMMANDABORTED**.

Podanie zbocza opadającego na wejście **EXECUTE** powoduje zresetowanie wszystkich wyjść do wartości 0.

Wartość **VELOCITY** jest skalowana zgodnie z następującą formułą:

$$\mathbf{VELOCITY} = (\text{częstotliwość zadana (Hz)} \times 0x4000) / P105$$

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Aktywacja	BOOL	INVELOCITY	Osiągnięto zadaną częstotliwość	BOOL
VELOCITY	Częstotliwość zadana	INT	BUSY	Częstotliwość zadana nie została jeszcze osiągnięta	BOOL
			COMMAND-ABORTED	Polecenie przerwane	BOOL
			ERROR	Błąd w FB	BOOL
			ERRORID	Kod błędu	INT
ERRORID	Objaśnienie				
0	Brak błędu				
1000h	FU nie jest uaktywniona				
1100h	FU nie znajduje się w trybie prędkości obrotowej (regulacja położenia aktywna)				
1101h	Częstotliwość zadana nie jest sparametryzowana (P553)				

Przykład AWL:

```
CAL Power
CAL Move

LD TRUE
ST Power.Enable

(* Ustawić 20 Hz (maks. 50 Hz) *)
LD DINT#20
MUL 16#4000
DIV 50

DINT_TO_INT
ST Move.Velocity

LD Power.Status
ST Move.Execute
```

Przykład w ST:

```
(* Urządzenie gotowe do pracy, gdy jest ustawione DIG1 *)
Power(Enable := _5_State_digital_input.0);
IF Power.Status THEN
  (* Aktywować urządzenie z 50% maks. częstotliwości, gdy jest ustawione DIG2 *)
  MoveVelocity(Execute := _5_State_digital_input.1, Velocity := 16#2000);
END_IF
```

3.3.3.9 MC_Power

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Za pomocą tej funkcji można włączyć lub wyłączyć stopień końcowy urządzenia. Jeżeli wejście **ENABLE** zostanie ustawione na wartość 1, następuje aktywacja stopnia końcowego. Ma to miejsce w przypadku, gdy urządzenie znajduje się w stanie „Blokada włączenia” lub „Gotowe do włączenia”. Jeżeli urządzenie znajduje się w stanie „Usterka” lub „Aktywna reakcja na usterkę”, należy najpierw usunąć usterkę i potwierdzić. Dopiero wtedy może nastąpić aktywacja za pośrednictwem tego bloku. Jeżeli urządzenie znajduje się w stanie „Niegotowe do włączenia”, włączenie nie jest możliwe. We wszystkich przypadkach FB przechodzi w stan usterki, a **ENABLE** musi być ustawione na wartość 0, aby potwierdzić błąd.

Jeżeli wejście **ENABLE** zostanie ustawione na wartość 0, następuje wyłączenie urządzenia. Jeżeli odbywa się to przy pracującym silniku, zostanie on najpierw zatrzymany do 0 Hz za pomocą rampy ustawionej w parametrze P103.

Wyjście **STATUS** ma wartość 1, gdy stopień końcowy urządzenia jest włączony, w przeciwnym wypadku ma wartość 0.

ERROR i **ERRORID** zostaną zresetowane, gdy **ENABLE** przełączy się na wartość 0.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Aktywacja	BOOL	STATUS	Doprowadzanie prądu do silnika	BOOL
			ERROR	Błąd w FB	BOOL
			ERRORID	Kod błędu	INT
ERRORID	Objaśnienie				
0	Brak błędu				
1001h	Funkcja zatrzymania jest aktywna				
1300h	Urządzenie nie znajduje się w stanie „Gotowe do włączenia” lub „Blokada włączenia”				

Przykład w AWL:

```
CAL Power
CAL Move

LD TRUE
ST Power.Enable

(* Ustawić 20 Hz (maks. 50 Hz) *)
LD DINT#20
MUL 16#4000
DIV 50

DINT_TO_INT
ST Move.Velocity

LD Power.Status
ST Move.Execute
```

Przykład w ST:

```
(* Aktywować Power Block *)
Power(Enable := TRUE);
IF Power.Status THEN
  (* Urządzenie jest gotowe do włączenia *)
END_IF
```

3.3.3.10 MC_ReadActualPos

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Dostarcza w sposób ciągły aktualną pozycję rzeczywistą przetwornicy częstotliwości, gdy **ENABLE** jest ustawione na wartość 1. Gdy na wyjściu występuje prawidłowa pozycja rzeczywista, **VALID** zostanie ustawione na wartość „prawidłowy”. W przypadku błędu **ERROR** przyjmuje wartość 1, **VALID** w tym przypadku wynosi 0.

Skalowanie pozycji: 1 obrót silnika = 1000

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Aktywacja	BOOL	VALID	Wyjście jest prawidłowe	BOOL
			ERROR	Błąd w FB	BOOL
			POSITION	Aktualna pozycja rzeczywista FU	DINT

Przykład w ST:

```

ReadActualPos(Enable := TRUE);
IF ReadActualPos.Valid THEN
    Pos := ReadActualPos.Position;
END_IF

```

3.3.3.11 MC_ReadParameter

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Odczytuje cyklicznie parametr z urządzenia, gdy **ENABLE** jest ustawione na wartość 1. Odczytany parametr zostanie zapisany w Value i obowiązuje, gdy **DONE** jest ustawione na wartość 1. Podczas operacji odczytu wyjście **BUSY** jest ustawione na wartość 1. Jeżeli **ENABLE** pozostanie ustawione na wartości 1, parametr jest odczytywany cyklicznie. W każdej chwili można zmienić numer parametru i indeks przy aktywnym **ENABLE**. Jednak trudno określić, kiedy została odczytana nowa wartość, ponieważ sygnał **DONE** przez cały czas jest równy 1. W takim przypadku zaleca się ustawienie sygnału **ENABLE** na wartość 0 dla jednego cyklu, ponieważ sygnał **DONE** zostanie wtedy zresetowany. Indeks parametru wynika z indeksu w dokumentacji minus 1. Np. P700 indeks 3 („Przyczyna blokady włączenia”) jest sprawdzany za pomocą indeksu parametru 2. W przypadku błędu **ERROR** przyjmuje wartość 1. **DONE** w tym przypadku wynosi 0, **ERRORID** zawiera kod błędu. Jeżeli sygnał **ENABLE** jest ustawiony na wartość 0, zostaną skasowane wszystkie sygnały i **ERRORID**.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Aktywacja	BOOL	DONE	Value jest prawidłowy	BOOL
PARAMETERNUMBER	Numer parametru	INT	ERROR	Operacja odczytu nie powiodła się	BOOL
PARAMETERINDEX	Indeks parametru	INT	BUSY	Proces nie jest zakończony	BOOL
			ERRORID	Kod błędu	INT
			VALUE	Odczytany parametr	DINT
ERRORID	Objaśnienie				
0	Niedopuszczalny numer parametru				
3	Nieprawidłowy indeks parametru				
4	Brak matrycy				
201	Nieprawidłowy element zadania w ostatnio otrzymanym zadaniu				
202	Nie można odwzorować wewnętrznego identyfikatora odpowiedzi				

Przykład w ST:

```
(* Moduł Motion FB_ReadParameter *)
ReadParam(Enable := TRUE, Parameternumber := 102, ParameterIndex := 0);
IF ReadParam.Done THEN
  Value := ReadParam.Value;
  ReadParam(Enable := FALSE);
END_IF
```


3.3.3.12 MC_ReadStatus

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Odczytuje stan urządzenia. Status maszyny jest oparty na specyfikacji PLCopen „Function blocks for motion control”. Dopóki **ENABLE** jest ustawione na wartość 1, jest odczytywany stan.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Aktywacja	BOOL	VALID	Wyjście jest prawidłowe	BOOL
			ERROR	Błąd w FB	BOOL
			ERRORSTOP	Błąd w urządzeniu	BOOL
			DISABLED	Stopień końcowy urządzenia jest wyłączony	BOOL
			STOPPING	Polecenie zatrzymania jest aktywne	BOOL
			DISCRETEMOTION	Aktywny jeden z trzech FB pozycjonowania	BOOL
			CONTINUOUSMOTION	MC_Velocity jest aktywny	BOOL
			HOMING	MC_Home jest aktywny	BOOL
			STANDSTILL	Urządzenie nie posiada aktywnego polecenia przemieszczenia. Prędkość obrotowa wynosi 0 obr/min i stopień końcowy jest włączony.	BOOL

Przykład w ST:

```

ReadStatus(Enable := TRUE);
IF ReadStatus.Valid THEN
  fError := ReadStatus.ErrorStop;
  fDisable := ReadStatus.Disabled;
  fStopping := ReadStatus.Stopping;
  fInMotion := ReadStatus.DiscreteMotion;
  fInVelocity := ReadStatus.ContinuousMotion;
  fInHome := ReadStatus.Homing;
  fStandStill := ReadStatus.StandStill;
end_if

```

3.3.3.13 MC_Reset

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Resetowanie błędu w urządzeniu (potwierdzenie usterki), przy podaniu zbocza narastającego na wejście **EXECUTE**. W przypadku błędu **ERROR** zostanie ustawione na wartość 1, a przyczyna błędu wprowadzona do **ERRORID**. Podanie zbocza opadającego na wejście **EXECUTE** powoduje zresetowanie wszystkich błędów.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Start	BOOL	DONE	Zresetowany błąd urządzenia	BOOL
			ERROR	Błąd w FB	BOOL
			ERRORID	Kod błędu	INT
			BUSY	Proces resetowania jest jeszcze aktywny	BOOL
ERRORID	Objaśnienie				
0	Brak błędu				
1001h	Funkcja zatrzymania jest aktywna				
1700h	Nie można wykonać resetu błędu, przyczyna błędu nie jest jeszcze usunięta				

Przykład w ST:

```

Reset(Execute := TRUE);
IF Reset.Done THEN
  (* Błąd został zresetowany *)
  Reset(Execute := FALSE);
ELSIF Reset.Error THEN
  (* Nie można wykonać resetu błędu, przyczyna błędu nie jest jeszcze usunięta *)
  Reset(Execute := FALSE);
END_IF

```

3.3.3.14 MC_Stop

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

W przypadku zbrocza narastającego (0 na 1) urządzenie zostanie ustawione w stan **STANDINGSTILL**. Wszystkie aktywne funkcje Motion zostaną przerwane. Urządzenie hamuje do 0 Hz i wyłącza stopień końcowy. Dopóki polecenie zatrzymania jest aktywne (**EXECUTE** = 1), wszystkie inne Motion FB są zablokowane. Wyjście **BUSY** staje się aktywne po podaniu zbrocza narastającego na wejście **EXECUTE** i pozostaje takie, dopóki na wejście **EXECUTE** nie zostanie podane zbrocze opadające.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Start	BOOL	DONE	Polecenie jest wykonane	BOOL
			BUSY	Polecenie jest aktywne	BOOL

3.3.3.15 MC_WriteParameter_16 / MC_WriteParameter_32

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Zapisuje parametr 16/32 bity w urządzeniu, gdy **EXECUTE** zmienia wartość z 0 na 1 (zbczce). Parametr został zapisany, gdy **DONE** jest ustawione na wartość 1. Podczas operacji odczytu wyjście **BUSY** jest ustawione na wartość 1. W przypadku błędu **ERROR** zostanie ustawione na wartość 1, a **ERRORID** na kod błędu. Sygnały **DONE**, **ERROR**, **ERRORID** zachowują swoje wartości, dopóki **EXECUTE** ponownie nie zmieni wartości na 0. W przypadku zmiany wartości sygnału **EXECUTE** na 0 proces zapisu nie zostanie przerwany. Tylko sygnał **DONE** zachowuje swoją wartość tylko na 1 cykl PLC.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Aktywacja	BOOL	DONE	Value jest prawidłowy	BOOL
PARAMETERNUMBER	Numer parametru	INT	BUSY	Operacja zapisu jest aktywna	BOOL
PARAMETERINDEX	Indeks parametru	INT	ERROR	Operacja odczytu nie powiodła się	BOOL
VALUE	Zapisywana wartość	INT	ERRORID	Kod błędu	INT
RAMONLY	Zapisanie wartości tylko w pamięci RAM (od wersji V2.1)	BOOL			
ERRORID	Objaśnienie				
0	Niedopuszczalny numer parametru				
1	Nie można zmienić wartości parametru				
2	Przekroczona dolna i górna wartość graniczna parametru				
3	Nieprawidłowy indeks parametru				
4	Brak matrycy				
5	Niedopuszczalny typ danych				
6	Tylko reset (można zapisać tylko 0)				
7	Nie można zmienić elementu opisującego				
201	Nieprawidłowy element zadania w ostatnio otrzymanym zadaniu				
202	Nie można odwzorować wewnętrznego identyfikatora odpowiedzi				

Przykład w ST:

```

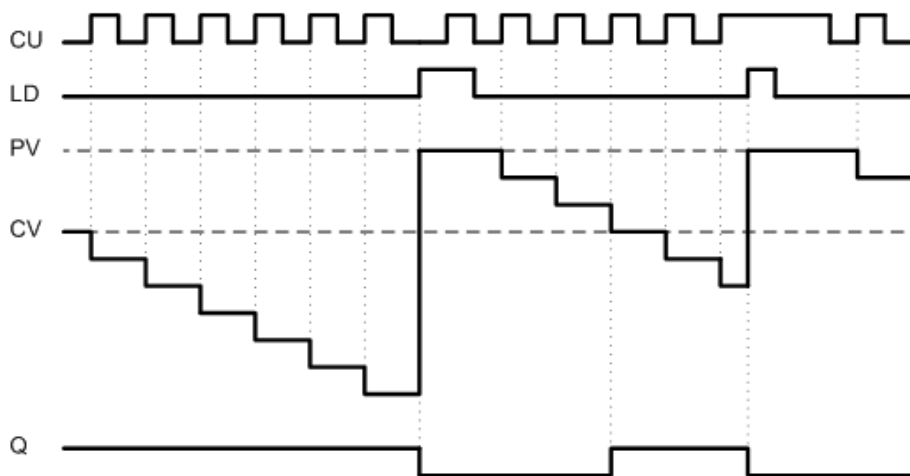
WriteParam16(Execute := TRUE, ParameterNumber := 102, ParameterIndex := 0, Value := 300);
IF WriteParam16.Done THEN
    WriteParam16(Execute := FALSE);
END_IF;
    
```

3.3.4 Standard

3.3.4.1 Licznik odliczający w dół CTD

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Podanie zbocza narastającego na wejście **CD** powoduje zmniejszenie wartości licznika bloku funkcyjnego **CV** o jeden, dopóki wartość **CV** jest większa od -32 768. Gdy wartość **CV** jest mniejsza lub równa 0, wyjście **Q** zachowuje wartość TRUE. Za pomocą **LD** można ustawić licznik **CV** na wartość zapisaną w **PV**.



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
CD	Wejście licznika	BOOL	Q	TRUE, gdy CV ≤ 0	BOOL
LD	Ładowanie wartości początkowej	BOOL	CV	Aktualny stan licznika	INT
PV	Wartość początkowa	INT			

Przykład w AWL:

```
LD VarBOOL1
ST CTDInst.CD
LD VarBOOL2
ST CTDInst.LD
LD VarINT1
ST CTDInst.PV
CAL CTDInst
LD CTDInst.Q
ST VarBOOL3
LD CTDInst.CV
ST VarINT2
```

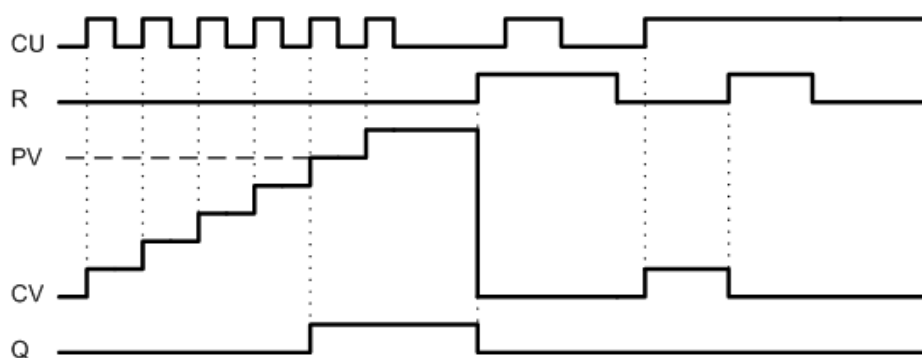
Przykład w ST:

```
CTDInst(CD := VarBOOL1, LD := VarBOOL2, PV := VarINT1);
VarBOOL3 := CTDInst.Q;
VarINT2 := CTDInst.CV;
```

3.3.4.2 Licznik odliczający do góry CTU

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Podanie zbocza narastającego na wejście **CU** powoduje zwiększenie wartości licznika bloku funkcyjnego **CV** o jeden. **CV** może zliczać do wartości 32767. Gdy wartość **CV** jest większa lub równa **PV**, wyjście **Q** zachowuje wartość TRUE. Za pomocą **R** można zresetować licznik **CV** do wartości zero.



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
CU	Wejście licznika	BOOL	Q	TRUE, gdy CV >= 0	BOOL
R	Reset stanu licznika	BOOL	CV	Aktualny stan licznika	INT
PV	Wartość początkowa	INT			

Przykład w AWL:

```
LD VarBOOL1
ST CTUInst.CU
LD VarBOOL2
ST CTUInst.R
LD VarINT1
ST CTUInst.PV
CAL CTUInst(CU := VarBOOL1, R := VarBOOL2, PV := VarINT1)
LD CTUInst.Q
ST VarBOOL3
LD CTUInst.CV
ST VarINT2
```

Przykład w ST:

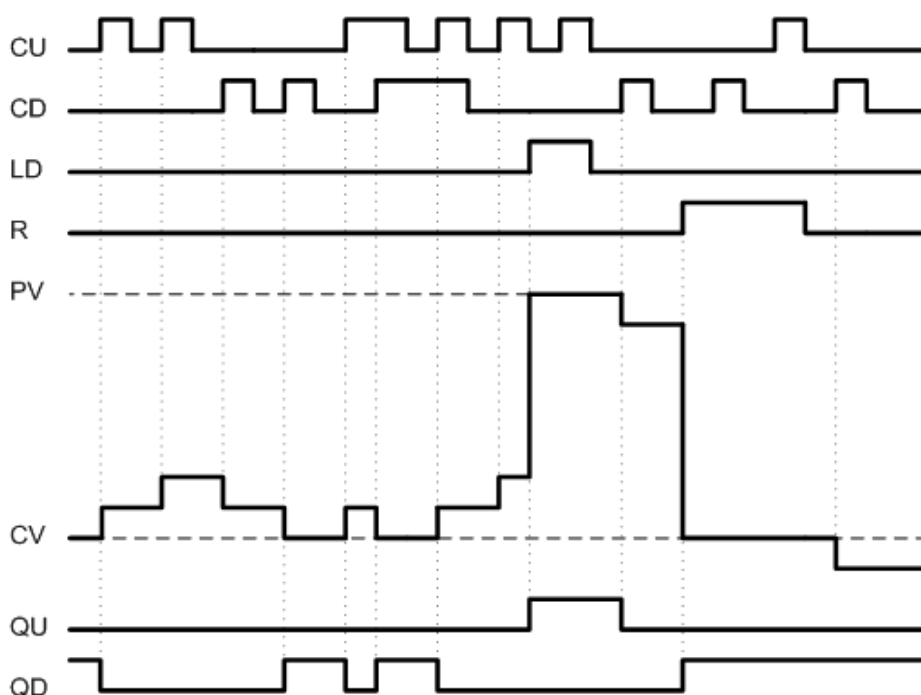
```
CTUInst(CU := VarBOOL1, R := VarBOOL2, PV := VarINT1);
VarBOOL3 := CTUInst.Q;
VarINT2 := CTUInst.CV;
```

3.3.4.3 Licznik odliczający do góry i w dół CTUD

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Podanie zbocza narastającego na wejście **CU** powoduje zwiększenie wartości licznika **CV** o jeden, dopóki wartość **CV** jest mniejsza od 32767. Podanie zbocza narastającego na wejście **CD** powoduje zmniejszenie wartości licznika **CV** o jeden, dopóki wartość **CV** jest większa od -32 768. Za pomocą **R** można zresetować licznik **CV** do wartości zero. Za pomocą **LD** można skopiować wartość zapisaną w **PV** do **CV**.

R ma pierwszeństwo względem **LD**, **CU** i **CV**. **PV** można zmienić w każdej chwili, **QU** zawsze odnosi się do aktualnie ustawionej wartości.



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
CU	Odliczanie do góry	BOOL	QU	TRUE, gdy $CV \geq PV$	BOOL
CD	Odliczanie w dół	BOOL	QD	TRUE, gdy $CV \leq 0$	BOOL
R	Reset stanu licznika	BOOL	CV	Aktualny stan licznika	INT
LD	Ładowanie wartości początkowej	BOOL			
PV	Wartość początkowa	INT			

Przykład w AWL:

```
LD VarBOOL1
ST CTUDInst.CU
LD VarBOOL3
ST CTUDInst.R
LD VarBool4
ST CTUDInst.LD
LD VarINT1
ST CTUInst.PV
CAL CTUDInst
LD CTUDInst.QU
ST VarBOOL5
LD CTUDInst.QD
ST VarBOOL5
LD CTUInst.CV
ST VarINT2
```

Przykład w ST:

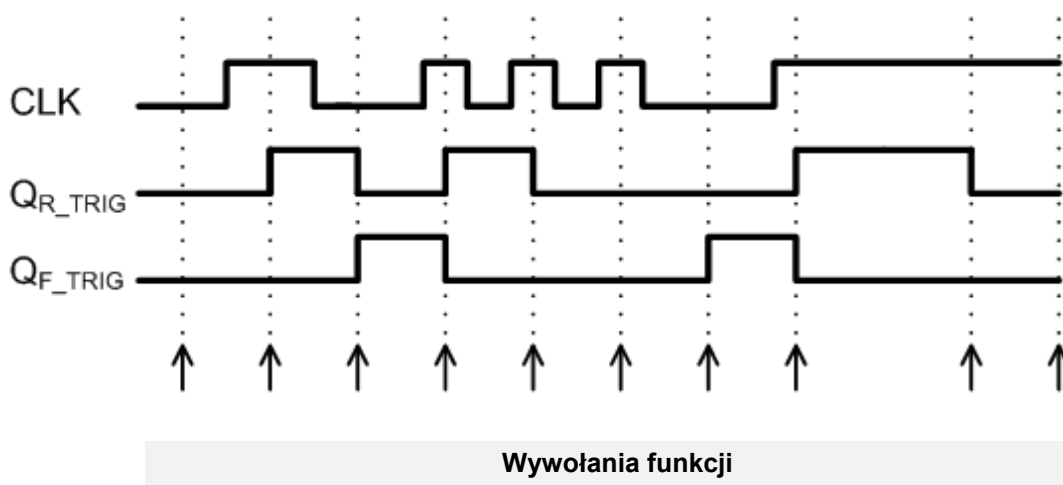
```
CTUDInst(CU:=VarBOOL1, R:=VarBOOL3, LD:=VarBOOL4, PV:=VarINT1);
VarBOOL5 := CTUDInst.QU;
VarBOOL5 := CTUDInst.QD;
VarINT2 := CTUDInst.CV;
```


3.3.4.4 R_TRIG i F_TRIG

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Obie funkcje służą do detekcji zbocza. Jeżeli na **CLK** zostanie wykryte zbocze, **Q** przyjmuje wartość TRUE do następnego wywołania funkcji, a następnie ponownie wartość FALSE. Tylko po podaniu nowego zbocza **Q** może ponownie przyjąć wartość TRUE dla jednego cyklu.

- R_TRIG = zbocze narastające
- F_TRIG = zbocze opadające



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
CLK	Ustawienie	BOOL	Q	Wyjście	BOOL

Przykład w AWL:

```
LD VarBOOL1
ST RTRIGInst.CLK
CAL RTRIGInst
LD RTRIGInst.Q
ST VarBOOL2
```

Przykład w ST:

```
RTRIGInst(CLK:= VarBOOL1);
VarBOOL2 := RTRIGInst.Q;
```

 Informacja

Wyjście funkcji zmienia się tylko wtedy, gdy funkcja zostanie wywołana. Z tego powodu wskazane jest ciągle wywoływanie detekcji zbocza za pomocą cyklu PLC.

3.3.4.5 RS Flip Flop

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Funkcja bistabilna, wyjście **Q1** jest ustawiane przez **S** i ponownie kasowane przez **R1**. Jeżeli **R1** i **S** równocześnie przyjmują wartość TRUE, **R1** posiada priorytet.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
S	Ustawienie	BOOL	Q1	Wyjście	BOOL
R1	Reset	BOOL			

Przykład w AWL:

```
LD VarBOOL1
ST RSInst.S
LD VarBOOL2
ST RSInst.R1
CAL RSInst
LD RSInst.Q1
ST VarBOOL3
```

Przykład w ST:

```
RSInst(S:= VarBOOL1 , R1:=VarBOOL2);
VarBOOL3 := RSInst.Q1;
```

3.3.4.6 SR Flip Flop

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Funkcja bistabilna, wyjście **Q1** jest ustawiane przez **S1** i ponownie kasowane przez **R**. Jeżeli **R1** i **S** równocześnie przyjmują wartość TRUE, **S1** posiada priorytet.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
S1	Ustawienie	BOOL	Q1	Wyjście	BOOL
R	Reset	BOOL			

Przykład w AWL:

```
LD VarBOOL1
ST SRInst.S1
LD VarBOOL2
ST SRInst.R
CAL RSInst
LD SRInst.Q1
ST VarBOOL3
```

Przykład w ST:

```
SRInst(S1:= VarBOOL1 , R:=VarBOOL2);
VarBOOL3 := SRInst.Q1;
```

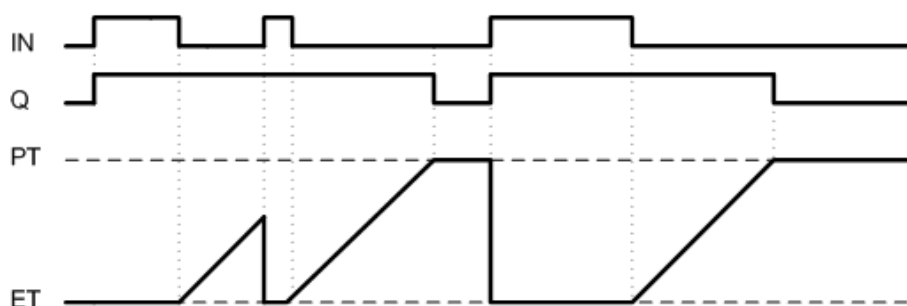
3.3.4.7 Opóźnienie wyłączenia TOF

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Jeżeli **IN** = TRUE, **Q** zostanie ustawione na wartość TRUE. Gdy **IN** przyjmuje wartość FALSE, timer uruchamia się. Dopóki timer działa (**ET** < **PT**), **Q** pozostaje ustawione na wartości TRUE. Gdy (**ET** = **PT**), timer zatrzymuje się, a **Q** przyjmuje wartość FALSE. Po podaniu nowego zbocza narastającego na wejście **IN** timer **ET** zostanie ponownie ustawiony na wartość zero.

W celu uproszczonego wprowadzania można używać literałów, np.

- LD TIME#50s20ms = 50,020 sekund
- LD TIME#1d30m = 1 dzień i 30 minut



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
IN	Timer aktywny	BOOL	Q	TRUE β (ET < PT)	BOOL
PT	Czas trwania	DINT	ET	Aktualny stan timera	DINT

Przykład w AWL:

```
LD VarBOOL1
ST TOFInst.IN
LD DINT#5000
ST TOFInst.PT
CAL TOFInst
LD TOFInst.Q
ST VarBOOL2
```

Przykład w ST:

```
TOFInst(IN := VarBOOL1, PT:= T#5s);
VarBOOL2 := TOFInst.Q;
```

Informacja

Timer ET

Czas ET biegnie niezależnie od cyklu PLC. Uruchomienie timera za pomocą **IN** i ustawienie wyjścia **Q** zostanie wykonane tylko przez wywołanie funkcji „CAL”. Wywołanie funkcji odbywa się w cyklu PLC, ale w przypadku dłuższych programów PLC może trwać dłużej niż 5 ms, co może spowodować wystąpienie fluktuacji w czasie.

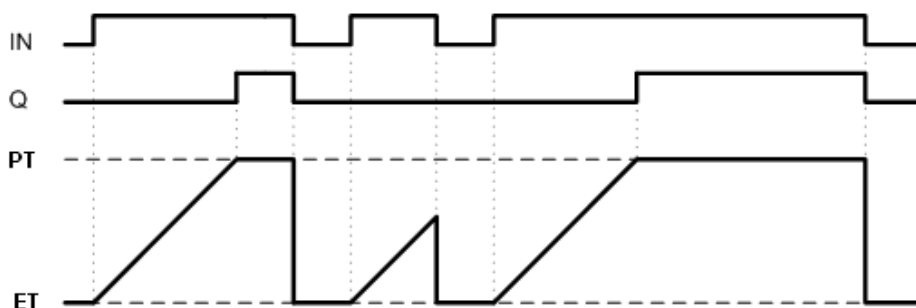
3.3.4.8 Opóźnienie włączenia TON

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Jeżeli zostanie ustawione **IN** = TRUE, timer uruchamia się. Gdy **ET** = PT, **Q** zostanie ustawione na wartość TRUE, a timer zatrzymuje się. **Q** zachowuje wartość TRUE, dopóki **IN** także ma wartość TRUE. Po podaniu nowego zbocza narastającego na wejście **IN** timer ponownie uruchamia się od wartości zero. **PT** można zmienić podczas pracy timera. Czas trwania w **PT** jest wprowadzany w milisekundach. Możliwe jest opóźnienie czasowe w zakresie od 5 ms do 24,8 dnia. Ponieważ podstawa czasu dla PLC wynosi 5 ms, minimalne opóźnienie czasowe wynosi również 5 ms.

W celu uproszczonego wprowadzania można używać literałów, np.

- LD TIME#50s20ms = 50,020 sekund
- LD TIME#1d30m = 1 dzień i 30 minut



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
IN	Timer aktywny	BOOL	Q	TRUE & (IN=TRUE & ET=PT)	BOOL
PT	Czas trwania	DINT	ET	Aktualny stan timera	DINT

Przykład w AWL:

```
LD VarBOOL1
ST TONInst.IN
LD DINT#5000
ST TONInst.PT
CAL TONInst
LD TONInst.Q
ST VarBOOL2
```

Przykład w ST:

```
TONInst(IN := VarBOOL1, PT:= T#5s);
VarBOOL2 := TONInst.Q;
```

Informacja

Timer ET

Czas ET biegnie niezależnie od cyklu PLC. Uruchomienie timera za pomocą IN i ustawienie wyjścia Q zostanie wykonane tylko przez wywołanie funkcji „CAL”. Wywołanie funkcji odbywa się w cyklu PLC, ale w przypadku dłuższych programów PLC może trwać dłużej niż 5 ms, co może spowodować wystąpienie fluktuacji w czasie.

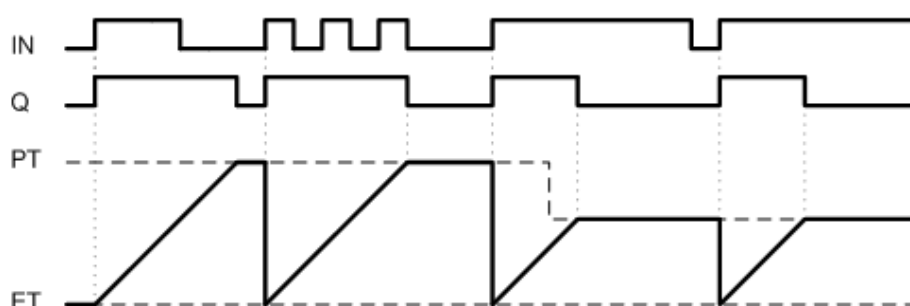
3.3.4.9 Impuls czasowy TP

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Po podaniu zbocza narastającego na wejście **IN** timer zostanie uruchomiony od wartości 0. Timer odlicza do wartości wprowadzonej w **PT**, a następnie zatrzymuje się. Tego procesu nie można przerwać. **PT** można zmienić podczas odliczania. Wyjście **Q** ma wartość TRUE, dopóki timer **ET** jest mniejszy niż **PT**. Gdy **ET = PT** i na wejściu **IN** zostanie wykryte narastające zbocze, timer zostanie ponownie uruchomiony od wartości 0.

W celu uproszczonego wprowadzania można używać literałów, np.

- LD TIME#50s20ms = 50,020 sekund
- LD TIME#1d30m = 1 dzień i 30 minut



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
IN	Timer aktywny	BOOL	Q	TRUE & (ET<PT)	BOOL
PT	Czas trwania	DINT	ET	Aktualny stan timera	DINT

Przykład w AWL:

```
LD VarBOOL1
ST TPInst.IN
LD DINT#5000
ST TPInst.PT
CAL TPInst
LD TPInst.Q
ST VarBOOL2
```

Przykład w ST:

```
TPInst(IN := VarBOOL1, PT:= T#5s);
VarBOOL2 := TPInst.Q;
```

Informacja

Timer ET

Czas ET biegnie niezależnie od cyklu PLC. Uruchomienie timera za pomocą **IN** i ustawienie wyjścia **Q** zostanie wykonane tylko przez wywołanie funkcji „CAL”. Wywołanie funkcji odbywa się w cyklu PLC, ale w przypadku dłuższych programów PLC może trwać dłużej niż 5 ms, co może spowodować wystąpienie fluktuacji w czasie.

3.3.5 Dostęp do obszarów pamięci przetwornicy częstotliwości

Gdy konieczne jest tymczasowe przechowywanie dużych ilości danych, ich przesyłanie do innych urządzeń lub odbiór z innych urządzeń, należy wykorzystać moduły FB_WriteTrace i FB_ReadTrace.

3.3.5.1 FB_ReadTrace

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		

Za pomocą tego FB można bezpośrednio odczytywać różne obszary pamięci FU.

Jeżeli FB wykryje zbocze narastające na wejściu **ENABLE**, zostaną przejęte wszystkie parametry występujące na wejściu. **STARTINDEX** i **MEMORY** zaznaczają lokalizację pamięci, która ma zostać odczytana. W przypadku pomyślniej operacji odczytu wyjście **VALID** przyjmuje wartość 1, a w **VALUE** zostaje zapisana odczytana wartość.

Jeżeli FB zostanie teraz kilkakrotnie wywołany i wejście **ENABLE** pozostanie ustawione na wartości 1, podczas każdego wywołania odczytywany adres pamięci jest zwiększany o 1, a zawartość nowej komórki pamięci jest natychmiast kopiowana do wyjścia **VALUE**.

Aktualny indeks pamięci dla następnego dostępu można odczytać na wyjściu **ACTINDEX**. Po osiągnięciu końca pamięci wyjście **READY** przyjmuje wartość 1 i operacja odczytu zostaje zatrzymana.

Można odczytywać wartości w formacie INT lub DINT. W przypadku wartości w formacie INT na wyjściu **VALUE** należy przeanalizować tylko część Low. Przyporządkowanie odbywa się przez wejście **SIZE**, 0 oznacza format INT, a 1 format DINT.

Przyporządkowanie obszarów pamięci odbywa się przez wejście MEMORY.

MEMORY = 1 à P613[0-251] odpowiada wartościom 504 INT lub 252 DINT

MEMORY = 0 à P900[0-247] do P906[0-111] odpowiada wartościom 3200 INT lub 1600 DINT

Nie można przerwać FB za pomocą innych bloków.

Podanie zbocza opadającego na wejście ENABLE powoduje ustawienie wszystkich wyjść na wartość 0 i zakończenie działania FB.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Wykonanie	BOOL	VALID	Pomyślna operacja odczytu	BOOL
SIZE	Format pamięci	BOOL	READY	Cała pamięć jest odczytana	BOOL
MEMORY	Wybór obszaru pamięci	BYTE	ERROR	Błąd w FB	BOOL
STARTINDEX	Wskazuje opisywaną komórkę pamięci	INT	ERRORID	Kod błędu	INT
			ACTINDEX	Aktualny indeks pamięci, z którego odbędzie się odczyt w następnym cyklu	INT
			VALUE	Odczytana wartość	DINT
ERRORID	Objaśnienie				
0	Brak błędu				
1A00h	Zakres wartości STARTINDEX został przekroczony				
1A01h	Zakres wartości MEMORY został przekroczony				

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Wykonanie	BOOL	VALID	Pomyślna operacja zapisu	BOOL
SIZE	Format pamięci	BOOL	FULL	Cała pamięć jest pełna	BOOL
OVERWRITE	Pamięć można nadpisać	BOOL	ERROR	Błąd w FB	BOOL
MEMORY	Wybór obszaru pamięci	BYTE	ERRORID	Kod błędu	INT
STARTINDEX	Wskazuje opisywaną komórkę pamięci	INT	ACTINDEX	Aktualny indeks pamięci, do którego odbędzie się zapis w następnym cyklu	DINT
VALUE	Zapisywana wartość	DINT			
ERRORID	Objaśnienie				
0	Brak błędu				
1A00h	Zakres wartości STARTINDEX został przekroczony				
1A01h	Zakres wartości MEMORY został przekroczony				

Informacja


Uwaga! Obszar pamięci w ustawieniu MEMORY = 0 jest również używany przez funkcję Scope. Stosowanie funkcji Scope powoduje nadpisanie zapisanych wartości!

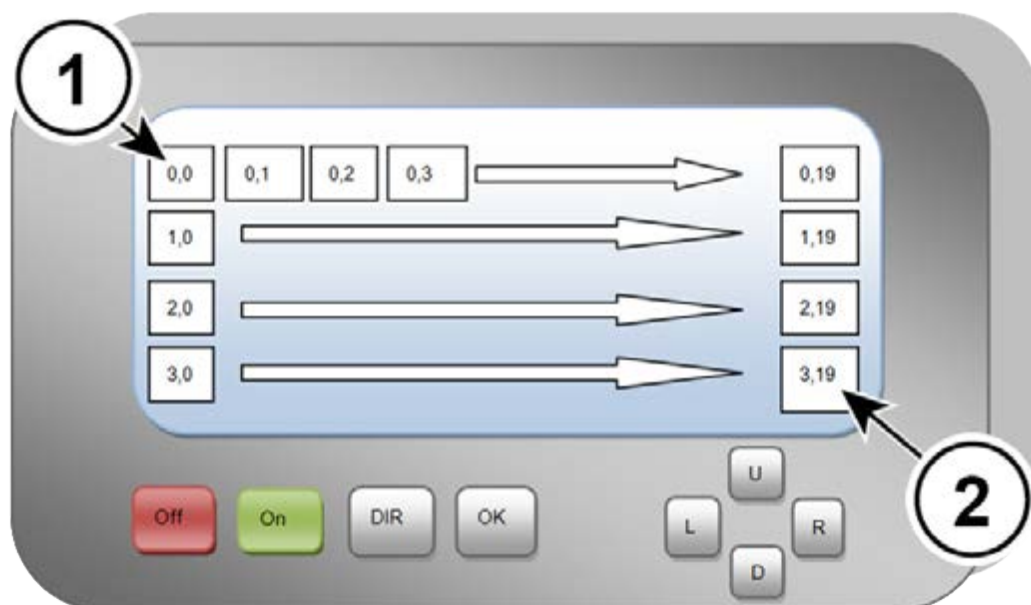
3.3.6 Wizualizacja panelu ParameterBox

Panel ParameterBox pozwala wykorzystać cały wyświetlacz do prezentacji informacji. W tym celu należy przełączyć ParameterBox w tryb wizualizacji. Jest to możliwe od wersji oprogramowania sprzętowego V4.3 panelu ParameterBox (parametr P1308) i odbywa się w następujący sposób:

- W punkcie menu „Wyświetlacz” ustawić parametr P1003 na „Wyświetlacz PLC”
- Za pomocą prawego lub lewego przycisku myszy przejść do wyświetlania wartości roboczych
- Wyświetlacz PLC jest teraz aktywny w panelu P-Box i pozostaje aktywny w sposób trwały

W trybie wizualizacji panelu P-Box zawartość wyświetlacza można opisać za pomocą dwóch niżej objaśnionych FB. Wcześniej w oknie dialogowym konfiguracji PLC musi być aktywowany punkt

„Dopuszczenie modułów funkcyjnych ParameterBox” (przycisk ). Za pomocą wartości procesowej „Parameterbox_key_state” można dodatkowo sprawdzić stan klawiatury panelu Box. Umożliwia to wprowadzanie danych do programu PLC. Poniższy rysunek przedstawia budowę wyświetlacza i pozycję przycisków dla panelu ParameterBox.



1	Pierwszy znak	(0,0 → wiersz = 0, kolumna = 0)
2	Ostatni znak	(3,19 → wiersz = 3, kolumna = 19)

3.3.6.1 Przegląd wizualizacji

Moduł funkcyjny	Objaśnienie
FB_STRINGToPBox	Kopiuje łańcuch znaków do panelu P-Box
FB_DINTToPBox	Kopiuje wartość DINT do panelu P-Box

3.3.6.2 FB_DINTToPBOX

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Moduł funkcyjny zamienia wartość DINT na łańcuch znaków ASCII i kopiuje go do panelu ParameterBox. Dane wyjściowe mogą być przedstawione w formacie dziesiętnym, binarnym lub szesnastkowym, wybór formatu odbywa się za pomocą **MODE**. **ROW** i **COLUMN** są stosowane do ustawienia pozycji początkowej łańcucha znaków na wyświetlaczu P-Box. Parametr **LENGTH** określa długość łańcucha w znakach. Jeżeli w **MODE** wybrano format dziesiętny, parametr **POINT** określa przecinek w przedstawianej liczbie. **POINT** określa liczbę znaków na prawo od przecinka. W przypadku ustawienia 0 funkcja **POINT** jest wyłączona. Jeżeli liczba zawiera więcej znaków niż dopuszczalna długość i nie ma przecinka, znak „#” określa przepełnienie. Jeżeli liczba zawiera przecinek, w razie potrzeby można usunąć wszystkie liczby znajdujące się za przecinkiem. Jeżeli w **MODE** wybrano format szesnastkowy i binarny, zawsze są wyświetlane najmniej znaczące bity, gdy ustawiona długość jest zbyt mała. Dopóki **ENABLE** jest ustawione na wartość 1, wszystkie zmiany na wejściach są wprowadzane natychmiast. Jeżeli **VALID** przyjmuje wartość 1, łańcuch znaków został prawidłowo przesłany. W przypadku błędu **ERROR** przyjmuje wartość 1. **VALID** w tym przypadku wynosi 0. **ERRORID** zawiera odpowiedni kod błędu. Podanie zbrocza opadającego na wejście **ENABLE** powoduje zresetowanie **VALID**, **ERROR** i **ERRORID**.

Przykłady:

Ustawienie	Przedstawiana liczba	Wyświetlacz P-Box
Length = 5 Point = 0	12345	12345
Length = 5 Point = 0	-12345	#####
Length = 10 Point = 3	123456789	123456,789
Length = 8 Point = 3	123456789	123456,7

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Przesłanie łańcucha znaków	BOOL	VALID	Ciąg znaków przesłany	BOOL
MODE	Format przedstawienia 0 = dziesiętny 1 = binarny 2 = szesnastkowy Zakres wartości = 0 do 2	BYTE	ERROR	Błąd w FB	BOOL
ROW	Wiersz wyświetlacza Zakres wartości = 0 do 3	BYTE	ERRORID	Kod błędu	INT
COLUMN	Kolumna wyświetlacza Zakres wartości = 0 do 19	BYTE			
POINT	Pozycja przecinka Zakres wartości = 0 do 10 0 = funkcja jest wyłączona	BYTE			
LENGTH	Długość wyjścia Zakres wartości = 1 do 11	BYTE			
VALUE	Liczba wyjściowa	DINT			
ERRORID	Objaśnienie				
0	Brak błędu				
1500h	Ciąg znaków nadpisuje obszar pamięci matrycy P-Box				
1501h	Na wejściu LINE został przekroczony zakres wartości				
1502h	Na wejściu ROW został przekroczony zakres wartości				
1504h	Na wejściu POINT został przekroczony zakres wartości				
1505h	Na wejściu LENGTH został przekroczony zakres wartości				
1506h	Na wejściu MODE został przekroczony zakres wartości				

Przykład w ST:

```
(* Inicjalizacja *)
if FirstTime then
  StringToPBox.ROW := 1;
  StringToPBox.Column := 16;
  FirstTime := False;
end_if;

(* Sprawdzanie aktualnej pozycji *)
ActPos(Enable := TRUE);
if ActPos.Valid then
  (* Wyświetlanie pozycji w panelu PBox (PBox P1003 = Wyświetlacz PLC ) *)
  DintToPBox.Value := ActPos.Position;
  DintToPBox.Column := 9;
  DintToPBox.LENGTH := 10;
  DintToPBox(Enable := True);
end_if;

(* Włączenie lub wyłączenie urządzenia przez DIG1 *)
Power(Enable := _5_State_digital_input.0);
if OldState <> Power.Status then
  OldState := Power.Status;
  (* Czy urządzenie jest włączone? *)
  if Power.Status then
    StringToPBox(Enable := False, Text := TextOn);
  else
    StringToPBox(Enable := False, Text := TextOff);
  end_if;

  StringToPBox(Enable := TRUE);
else
  StringToPBox;
end_if;
```

3.3.6.3 FB_STRINGToPBOX

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Moduł funkcyjny kopiuje łańcuch znaków do matrycy pamięci P-Box. **ROW** i **COLUMN** są stosowane do ustawienia pozycji początkowej łańcucha znaków na wyświetlaczu P-Box. Parametr **TEXT** przekazuje żądany łańcuch znaków do bloku funkcyjnego, nazwę łańcucha można pobrać z tabeli zmiennych. Dopóki **ENABLE** jest ustawione na wartość 1, wszystkie zmiany na wejściach są wprowadzane natychmiast. Po ustawieniu wejścia **CLEAR** cała zawartość wyświetlacza zostanie zastąpiona pustymi znakami przed zapisaniem wybranego łańcucha znaków. Jeżeli **VALID** przyjmuje wartość 1, łańcuch znaków został prawidłowo przesłany. W przypadku błędu **ERROR** przyjmuje wartość 1. **VALID** w tym przypadku wynosi 0. **ERRORID** zawiera odpowiedni kod błędu. Podanie zbocza opadającego na wejście **ENABLE** powoduje zresetowanie **VALID**, **ERROR** i **ERRORID**.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Przesłanie łańcucha znaków	BOOL	VALID	Ciąg znaków przesłany	BOOL
CLEAR	Kasowanie wyświetlacza	BOOL	ERROR	Błąd w FB	BOOL
ROW	Wiersz wyświetlacza Zakres wartości = 0 do 3	BYTE	ERRORID	Kod błędu	INT
COLUMN	Kolumna wyświetlacza Zakres wartości = 0 do 19	BYTE			
TEXT	Wyświetlany tekst	STRING			
ERRORID	Objaśnienie				
0	Brak błędu				
1500h	Ciąg znaków nadpisuje obszar pamięci matrycy P-Box				
1501h	Na wejściu ROW został przekroczony zakres wartości				
1502h	Na wejściu COLUMN został przekroczony zakres wartości				
1503h	Wybrany numer łańcucha znaków nie istnieje				
1506h	W konfiguracji PLC opcja „Dopuszczenie modułów funkcyjnych ParameterBox” nie jest aktywna.				

Przykład w ST:

```
(* Inicjalizacja *)
if FirstTime then
  StringToPBox.ROW := 1;
  StringToPBox.Column := 16;
  FirstTime := False;
end_if;

(* Sprawdzanie aktualnej pozycji *)
ActPos(Enable := TRUE);
if ActPos.Valid then
  (* Wyświetlanie pozycji w panelu PBox (PBox P1003 = Wyświetlacz PLC ) *)
  DintToPBox.Value := ActPos.Position;
  DintToPBox.Column := 9;
  DintToPBox.LENGTH := 10;
  DintToPBox(Enable := True);
end_if;

(* Włączenie lub wyłączenie urządzenia przez DIG1 *)
Power(Enable := _5_State_digital_input.0);
if OldState <> Power.Status then
  OldState := Power.Status;
  (* Czy urządzenie jest włączone? *)
  if Power.Status then
    StringToPBox(Enable := False, Text := TextOn);
  else
    StringToPBox(Enable := False, Text := TextOff);
  end_if;

  StringToPBox(Enable := TRUE);
else
  StringToPBox;
end_if;
```


3.3.7 FB_Capture (rejestracja szybkich zdarzeń)

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		

Czas cyklu PLC wynosi 5 ms, cykl ten niekiedy jest zbyt duży, aby zarejestrować bardzo szybkie zdarzenia zewnętrzne. Za pomocą FB Capture możliwa jest rejestracja określonych wielkości fizycznych na zboczach wejść FI. Monitorowanie wejść odbywa się w cyklu 1 ms. Zapisane w ten sposób wartości mogą zostać później odczytane przez PLC.

Podanie zbocza narastającego na wejście **EXECUTE** powoduje odczytanie wszystkich wejść i włączenie funkcji Capture. Za pomocą wejścia **INPUT** można wybrać monitorowane wejście FI. Za pomocą **EDGE** można wybrać rodzaj zbocza i zachowanie modułu.

EDGE = 0 Podanie pierwszego zbocza narastającego powoduje zapisanie wybranej wartości do **OUTPUT1** i ustawienie **DONE1** na wartość 1. Podanie następnego zbocza narastającego powoduje zapisanie do **OUTPUT2** i ustawienie **DONE2** na wartość 1. Następnie FB zostanie wyłączony.

EDGE = 1 Zachowanie jak w przypadku **EDGE = 0**, z tą różnicą, że działanie powoduje podanie zbocza opadającego.

EDGE = 2 Podanie pierwszego zbocza narastającego powoduje zapisanie wybranej wartości do **OUTPUT1** i ustawienie **DONE1** na wartość 1. Podanie następnego zbocza opadającego powoduje zapisanie do **OUTPUT2** i ustawienie **DONE2** na wartość 1. Następnie FB zostanie wyłączony.

EDGE = 3 Zachowanie jak w przypadku **EDGE = 2**, z tą różnicą, że działanie powoduje najpierw podanie zbocza opadającego, a następnie zbocza narastającego.

Jeżeli wejście **CONTINUOUS** zostanie ustawione na wartość 1, dla **EDGE** jest istotne tylko ustawienie 0 i 1. FB kontynuuje działanie w sposób ciągły i zawsze zapisuje ostatnie zdarzenie wywołujące do **OUTPUT1**. **DONE1** pozostaje aktywne od pierwszego zdarzenia. **DONE2** i **OUTPUT2** nie są stosowane.

Wyjście **BUSY** pozostaje aktywne, dopóki nie wystąpią oba zdarzenia Capture (**DONE1** i **DONE2**).

Działanie modułu może zostać w każdej chwili zakończone przez podanie zbocza opadającego na wejście **EXECUTE**. Wszystkie wyjścia zachowują swoje wartości. Podanie zbocza narastającego na wejście **EXECUTE** powoduje najpierw skasowanie wszystkich wyjść, a następnie uruchomienie modułu.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Wykonanie	BOOL	DONE1	Wartość w OUTPUT1 jest prawidłowa	BOOL
CONTINUOUS	Jednokrotne wykonanie lub praca ciągła	BOOL	DONE2	Wartość w OUTPUT2 jest prawidłowa	BOOL
INPUT	SK54xE Monitorowane wejście 0 = wejście 1 ---- 7 = wejście 8 SK52xE, SK53xE, SK2xxE, SK2xx-EFDS Monitorowane wejście 0 = wejście 1 ---- 3 = wejście 4	BYTE	BUSY	FB oczekuje na zdarzenia Capture	BOOL
EDGE	Zbocze wywołujące	BYTE	ERROR	Błąd w FB	BOOL
SOURCE	Zapisywana wielkość 0 = pozycja w obrotach 1 = częstotliwość rzeczywista 2 = moment	BYTE	ERRORID	Kod błędu	INT
			OUTPUT1	Wartość dla 1. zdarzenia Capture	DINT
			OUTPUT2	Wartość dla 2. zdarzenia Capture	DINT
ERRORID	Objaśnienie				
0	Brak błędu				
1900h	Zakres wartości INPUT został przekroczony				
1901h	Zakres wartości EDGE został przekroczony				
1902h	Zakres wartości SOURCE został przekroczony				
1903h	Więcej niż dwie instancje są aktywne				

Przykład w ST:

```

Power(ENABLE := TRUE);
IF Power.STATUS THEN
    Move(EXECUTE := TRUE, POSITION := Pos, VELOCITY := 16#2000);
    (* Capture FB oczekuje sygnału High na DIG1. W przypadku detekcji sygnału FB zapisuje aktualną
    pozycję. Za pomocą właściwości „OUTPUT1” można sprawdzić wartość. *)
    Capture(EXECUTE := TRUE, INPUT := 0);

    IF Capture.DONE1 THEN
        Pos := Capture.OUTPUT1;
        Move(EXECUTE := FALSE);
    END_IF;
END_IF;
    
```

Informacja

W programie PLC może istnieć kilka instancji tego FB. Ale tylko dwie instancje mogą być aktywne w tym samym czasie!

3.3.8 FB_DinCounter

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	ab V2.3	ab V3.1	ab V2.1	X	ab V1.1	

FB służy do zliczania impulsów przez wejścia cyfrowe. Są zliczane wszystkie zbocza (Low – High i High – Low). Minimalna szerokość impulsu wynosi 250 μ s.

FB jest aktywowany za pomocą ENABLE. Podanie zbocza narastającego powoduje przejście PV, UD, DIN i MODE oraz skasowanie wszystkich wyjść.

UD definiuje kierunek zliczania

- 0 = większe liczby
- 1 = mniejsze liczby

Do PV można wprowadzić wartość licznika. Ma to różny efekt w zależności od ustawienia wejścia MODE.

MODE

- 0 = przepelnienie, licznik działa jak licznik ciągły. Przepelnienie może wystąpić w kierunku dodatnim i ujemnym. Podczas uruchamiania funkcji zostanie ustawione CV = PV. W tym trybie BUSY zawsze wynosi 1, a Q zawsze wynosi 0.
- 1 = bez przepelnienia
 - Zliczanie przyrostowe: CV uruchamia się przy ustawieniu 0, BUSY = 1, i działa do CV=>PV. Następnie BUSY przyjmuje wartość 0, a Q wartość 1. Proces zliczania zatrzymuje się.
 - Zliczanie zmniejszające: CV uruchamia się przy PV i działa do CV<=0. W tym czasie BUSY = 1 i przyjmuje wartość 0 po osiągnięciu końca licznika. Q przyjmuje wartość 1.
 - Ponowne uruchomienie licznika odbywa się przez podanie nowego zbocza na wejście ENABLE

DIN definiuje wejście pomiarowe. Liczba wejść zależy od FU (do 4 sztuk).

- Wejście 1 = 0
- Wejście 2 = 1
- Wejście 3 = 2
- Wejście 4 = 3

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Aktywacja	BOOL	Q	Zliczanie zakończone	BOOL
UD	Kierunek zliczania 0 = większe liczby 1 = mniejsze liczby	BOOL	BUSY	Licznik działa	BOOL
PV	Wartość licznika	INT	ERROR	Błąd w FB	BOOL
MODE	Tryb	BYTE	ERRORID	Kod błędu	INT
DIN	Wejście pomiarowe	BYTE	CV	Wartość licznika	INT
			CF	Częstotliwość zliczania (rozdzielczość 0,1)	INT
ERRORID	Objaśnienie				
0	Brak błędu				
0x1E00	Wejście cyfrowe jest już używane przez inny licznik				
0x1E01	Wejście cyfrowe nie istnieje				
0x1E02	Przekroczony zakres wartości MODE				

3.3.9 FB_FunctionCurve

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	

Moduł funkcyjny zapewnia sterowanie charakterystyką. Do bloku funkcyjnego można przesłać zdefiniowane punkty, za pomocą których emuluje funkcję. Wyjście zachowuje się zgodnie z zapisaną charakterystyką. Między poszczególnymi punktami jest przeprowadzana interpolacja liniowa. Punkty interpolacji są definiowane za pomocą wartości X i Y. Wartości X są zawsze typu **INT**, a wartości Y mogą być typu **INT** lub **DINT** w zależności od wielkości największego węzła interpolacji. Typ **DINT** zajmuje więcej miejsca w pamięci. Węzły interpolacji są wprowadzane w oknie zmiennych w kolumnie „Wartość Init”. Jeżeli na wejściu **ENABLE** została wykryta wartość TRUE, na podstawie wartości wejściowej **INVALUE** jest obliczana odpowiednia wartość wyjściowa **OUTVALUE**. **VALID** = TRUE sygnalizuje, że wartość wyjściowa **OUTVALUE** jest prawidłowa. Dopóki **VALID** = FALSE, wyjście **OUTVALUE** ma wartość 0. Jeżeli wartość wejściowa **INVALUE** wykracza poza górny lub dolny koniec charakterystyki, na wyjściu pozostaje pierwsza lub ostatnia wartość wyjściowa charakterystyki, dopóki **INVALUE** nie powróci do obszaru charakterystyki. W przypadku wyjścia poza górny lub dolny obszar charakterystyki odpowiednie wyjście **MINLIMIT** lub **MAXLIMIT** zostanie ustawione na wartość TRUE. **ERROR** przyjmuje wartość TRUE, gdy wartości odciętych (wartości X) charakterystyki nie zwiększają się lub nie jest zainicjowana tabela. Odpowiedni błąd jest wyprowadzany przez **ERRORID**, a wartość wyjściowa przyjmuje wartość 0. Błąd jest resetowany, gdy **ENABLE** = FALSE.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Wykonanie	BOOL	VALID	Wartość wyjściowa jest prawidłowa	BOOL
INVALUE	Wartość wejściowa (x)	INT	ERROR	Błąd w FB	BOOL
			ERRORID	Kod błędu	INT
			MAXLIMIT	Osiągnięto maksymalny limit	BOOL
			MINLIMIT	Osiągnięto minimalny limit	BOOL
			OUTVALUE	Wartość wyjściowa (y)	DINT
ERRORID	Objaśnienie				
0	Brak błędu				
1400h	Wartości odciętych (wartości X) charakterystyki nie zawsze wzrastają				
1401h	Brak inicjalizacji charakterystyki				

3.3.10 FB_PIDT1

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

P-I-DT1 to swobodnie parametryzowalny dyskretny regulator. Jeżeli poszczególne człony P, I lub DT1 nie są potrzebne, odpowiedni parametr przyjmuje wartość 0. Człon T1 współpracuje wyłącznie z członem D. Dlatego nie można parametryzować regulatora PT1. Ze względu na wewnętrzne ograniczenie pamięci parametry regulacji są ograniczone do następujących zakresów:

Dopuszczalny zakres wartości parametrów regulacji			
Parametr	Zakres wartości	Skalowanie	Wynikowy zakres wartości
P (Kp)	0 – 32767	1/100	0,00 – 32,767
I (Ki)	0 – 10240	1/100	0,00 – 10,240
D (Kd)	0 – 32767	1/1000	0,000 – 3,2767
T1 (ms)	0 – 32767	1/1000	0,000 – 3,2767
Maks.	-32768 – 32767		
Min.	-32768 – 32767		

Jeżeli wejście **ENABLE** jest ustawione na wartość TRUE, regulator rozpoczyna obliczanie. Parametry regulacji są przejmowane tylko po podaniu zbocza narastającego na wejście **ENABLE**. Gdy **ENABLE** ma wartość TRUE, zmiana parametrów regulacji nie powoduje żadnych skutków. Jeżeli **ENABLE** zostanie ustawione na wartość FALSE, na wyjściu zostaje zachowana ostatnia wartość.

Bit wyjściowy **VALID** jest ustawiony, dopóki wartość wyjściowa Q utrzymuje się w granicach Min. i Maks, a wejście **ENABLE** ma wartość TRUE.

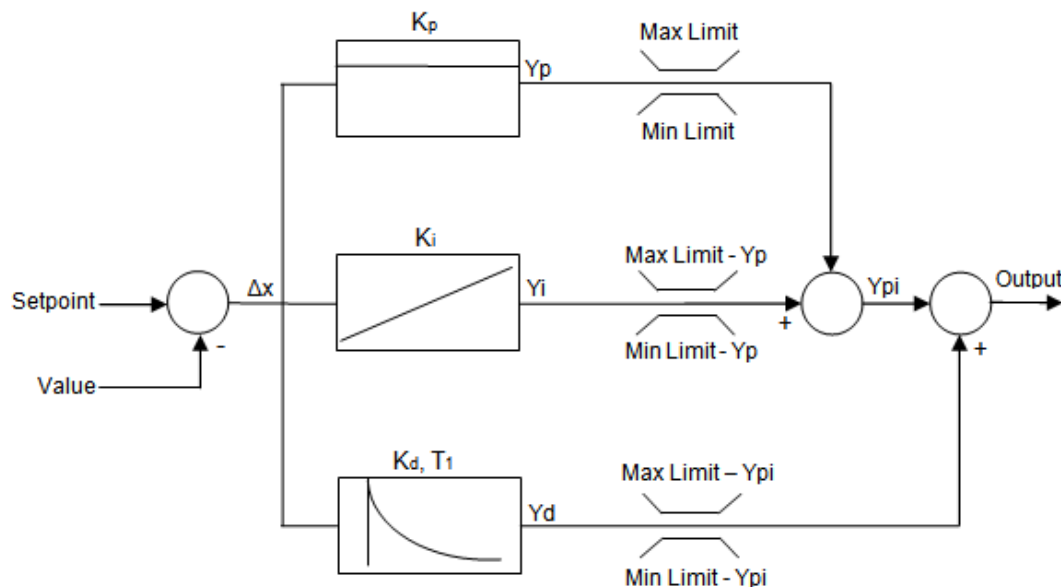
Gdy wystąpi błąd, zostanie ustawiony **ERROR**. Bit **VALID** przyjmuje wtedy wartość FALSE, a przyczynę błędu przedstawia **ERRORID** (patrz poniższa tabela).

Jeżeli bit **RESET** zostanie ustawiony na wartość TRUE, zawartość członu całującego i różniczkującego zostanie ustawiona na wartość 0. Jeżeli wejście **ENABLE** jest ustawione na wartość FALSE, wyjście **OUTPUT** również zostanie ustawione na wartość 0. Jeżeli wejście **ENABLE** jest ustawione na wartość TRUE, na wyjście **OUTPUT** oddziałuje wyłącznie człon P.

Jeżeli wartość wyjściowa **OUTPUT** wykracza poza maksymalne lub minimalne wartości wyjściowe, zostanie ustawiony odpowiedni bit **MAXLIMIT** lub **MINLIMIT**, a bit **VALID** zostanie ustawiony na wartość FALSE.

Informacja

Jeżeli cały program nie może zostać wykonany w jednym cyklu PLC, regulator oblicza wartość wyjściową po raz drugi z dotychczasowymi wartościami próbkowania. Pozwala to osiągnąć stałą częstotliwość próbkowania. Z tego względu konieczne jest, aby polecenie CAL było wykonywane dla regulatora PIDT1 w każdym cyklu PLC i tylko na końcu programu PLC!



VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
ENABLE	Wykonanie	BOOL	VALID	Wartość wyjściowa jest prawidłowa	BOOL
RESET	Resetowanie wartości wyjściowych	BOOL	ERROR	Błąd w FB	BOOL
P	Człon P (K_p)	INT	ERRORID	Kod błędu	INT
I	Człon I (K_i)	INT	MAXLIMIT	Osiągnięto maksymalny limit	BOOL
D	Człon D (K_d)	INT	MINLIMIT	Osiągnięto minimalny limit	BOOL
T1	Człon T1 w ms	INT	OUTPUT	Wartość wyjściowa	INT
MAX	Maksymalna wartość wyjściowa	INT			
MIN	Minimalna wartość wyjściowa	INT			
SETPOINT	Wartość zadana	INT			
VALUE	Wartość rzeczywista	INT			
ERRORID	Objaśnienie				
0	Brak błędu				
1600h	Człon P wykracza poza zakres wartości				
1601h	Człon I wykracza poza zakres wartości				
1602h	Człon D wykracza poza zakres wartości				
1603h	Człon T1 wykracza poza zakres wartości				

3.3.11 FB_ResetPostion

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	ab V2.3	ab V3.1	ab V2.1	X	ab V1.2	

Podanie zbocza na wejście **EXECUTE** powoduje ustawienie aktualnej pozycji na wartość wprowadzoną w Position. W enkoderach absolutnych aktualną pozycję można zresetować tylko do wartości 0. Wartość znajdująca się w Position nie jest używana.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Wykonanie	BOOL			
Position	Pozycja	DINT			

3.3.12 FB_Weigh

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	od V2.3	od V3.1	od V2.1	X	od V1.2	

Moduł ten służy do określania średniego momentu obrotowego podczas ruchu ze stałą prędkością obrotową. Na podstawie tej wartości można np. określić wielkości fizyczne takie jak przemieszczany ciężar.

Podanie zbocza narastającego na wejście **EXECUTE** powoduje uruchomienie FB. Za pomocą zbocza są przejmowane wszystkie wejścia na FB. FU porusza się z prędkością obrotową ustawioną w **SPEED**. Po upływie czasu ustawionego w **STARTTIME** rozpoczyna się pomiar. Czas pomiaru jest zdefiniowany w **MEASURETIME**. Po upływie czasu pomiaru FU zatrzymuje się. Gdy wejście **REVERSE** = 1, proces pomiaru uruchamia się ponownie, ale z odwróconą prędkością obrotową. W przeciwnym wypadku pomiar zostanie zakończony, wyjście **DONE** przyjmuje wartość 1, a **VALUE** zawiera wynik pomiaru.

Dopóki trwa proces pomiaru, **BUSY** jest aktywne.

Skalowanie wyniku pomiaru **VALUE** wynosi 1 = 0,01% znamionowego momentu obrotowego silnika.

Wywołanie innego Motion FB zatrzymuje funkcję pomiaru, a wyjście **ABORT** przyjmuje wartość 1.

Wszystkie wyjścia FB są resetowane przez podane zbocza narastającego na wejście **EXECUTE**.

VAR_INPUT			VAR_OUTPUT		
Wejście	Objaśnienie	Typ	Wyjście	Objaśnienie	Typ
EXECUTE	Wykonanie	BOOL	DONE	Pomiar zakończony	BOOL
REVERSE	Zmiana kierunku obrotu	BOOL	BUSY	Pomiar trwa	BOOL
STARTTIME	Czas do rozpoczęcia pomiaru w ms	INT	ABORT	Pomiar przerwany	BOOL
MEASURETIME	Czas pomiaru w ms	INT	ERROR	Błąd w FB	BOOL
SPEED	Prędkość pomiaru w % (znormalizowana do maksymalnej częstotliwości, 16#4000 odpowiada 100%)	INT	ERRORID	Kod błędu	INT
			VALUE	Wyniki pomiaru	INT
ERRORID	Objaśnienie				
0	Brak błędu				
0x1000	FU nie jest włączona				
0x1101	Częstotliwość zadana nie jest sparametryzowana jako wartość zadana (P553)				
0x1C00	Zakres wartości STARTTIME został przekroczony				
0x1C01	Zakres wartości MEASURETIME został przekroczony				
0x1C02	Wzajemna tolerancja wartości pomiarowych, jest większa niż 1/8				

Przykład w ST:

```

(* Aktywować urządzenie *)
Power(Enable := TRUE);
(* Czy urządzenie jest aktywowane? *)
if Power.Status then
    (* Określić czas startu 2000 ms *)
    Weigh.STARTTIME := 2000;
    (* Określić czas pomiaru 1000 ms *)
    Weigh.MEASURETIME := 1000;
    (* Określić prędkość na 25% maksymalnej prędkości *)
    Weigh.SPEED := 16#1000;
end_if;

Weigh(EXECUTE := Power.Status);
(* Czy ważenie zostało zakończone? *)
if Weigh.done then
    Value := Weigh.Value;
end_if;
    
```

Informacja

W programie PLC jest dopuszczalna tylko jedna instancja tego FB!

3.4 Operatory

3.4.1 Operatory arytmetyczne

i Informacja

Niektóre z poniższych operatorów mogą zawierać dodatkowe polecenia. Muszą być one umieszczone w nawiasach za operatorem. Za nawiasem otwierającym musi znajdować się spacja. Nawias zamykający należy umieścić w osobnej linii programu.

```
LD Var1
ADD( Var2
SUB Var3
)
```

3.4.1.1 ABS

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych			X	X			

Tworzy z Akku wartość absolutną.

Przykład w AWL:

```
LD -10 (* Ładuje wartość -10 *)
ABS (* Akku = 10 *)
ST Value1 (* Zapisuje wartość 10 w Value1 *)
```

Przykład w ST:

```
Value1 := ABS(-10); (* Wynik wynosi 10 *)
```

3.4.1.2 ADD i ADD(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Dodaje zmienne i stałe wraz z odpowiednim znakiem. Pierwsza dodawana wartość znajduje się w Akku, a druga jest ładowana za pomocą polecenia ADD lub znajduje się w nawiasie. Do polecenia ADD można dołączyć wiele zmiennych lub stałych. W operacji dodawania z nawiasami Akku dodaje się do wyniku wyrażenia w nawiasie. Możliwych jest do 6 poziomów nawiasów. Dodawane wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 10
ADD 204 (* Dodanie dwóch stałych *)
ST Value
LD 170 (* Dodanie jednej stałej i 2 zmiennych. *)
ADD Var1, Var2 (* 170dec + Var1 + Var2 *)
ST Value
LD Var1
ADD( Var2
SUB Var3 (* Var1 + ( Var2 - Var3 ) *)
)
ST Value
```

Przykład w ST:

```
Ergebnis := 10 + 30; (* Wynik wynosi 40 *)
Ergebnis := 10 + Var1 + Var2;
```

3.4.1.3 DIV i DIV(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Dzieli Akku przez operand. W przypadku dzielenia przez zero do Akku jest wprowadzany maksymalny możliwy wynik, np. przy dzieleniu przez wartości INT jest to wartość 0x7FFF lub gdy dzielnik jest ujemny, jest to wartość 0x8000. W operacji dzielenia z nawiasami Akku dzieli się przez wynik wyrażenia w nawiasie. Możliwych jest do 6 poziomów nawiasów. Dzielone wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 10
DIV 3 (* Dzielenie dwóch stałych *)
ST iValue (* Wynik wynosi 9 *)
LD 170 (* Dzielenie jednej stałej i 2 zmiennych. *)
DIV Var1, Var2 (* (170dec : Var1) : Var2 *)
ST Value
LD Var1 (* Dzielenie Var1 przez wyrażenie w nawiasach *)
DIV( Var2
SUB Var3
) (* Var1 : ( Var2 - Var3 ) *)
ST Value
```

Przykład w ST:

```
Ergebnis := 30 / 10; (* Wynik wynosi 3 *)
Ergebnis := 30 / Var1 / Var2;
```

3.4.1.4 LIMIT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Polecenie ogranicza wartość znajdującą się w Akku do przesłanych wartości min. i maks. Wartości. W przypadku przekroczenia wartości maksymalnej do Akku zostanie wprowadzona wartość maks., a gdy wartość jest mniejsza od minimalnej zostanie wprowadzona wartość min. Jeżeli wartość znajduje się między wartościami granicznymi, wtedy nie zmienia się.

Przykład w AWL:

```
LD 10 (* Ładuje wartość 10 do Akku *)
LIMIT 20, 30 (* Wartość jest porównywana z wartościami granicznymi 20 i 30. *)
(* Wartość w Akku jest mniejsza, Akku zostanie nadpisane przez 20*)
ST iValue (* Zapisuje wartość 20 w Valuel *)
```

Przykład w ST:

```
Ergebnis := Limit(10, 20, 30); (* Wynik wynosi 20 *)
```

3.4.1.5 MAX

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Polecenie określa maksymalną wartość dwóch zmiennych lub stałych. W tym celu aktualna zawartość Akku jest porównywana z wartością przesłaną w poleceniu MAX. Po wykonaniu polecenia większa z dwóch wartości znajduje się w Akku. Obie wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 100 (* Załadować 100 do Akku *)
MAX 200 (* Porównać z wartością 200 *)
ST iValue (* Zapisać 200 do Value2 (ponieważ jest to większa wartość) *)
```

Przykład w ST:

```
Ergebnis := Max(100, 200); (* Wynik wynosi 200 *)
```

3.4.1.6 MIN

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Polecenie określa minimalną wartość dwóch zmiennych lub stałych. W tym celu aktualna zawartość Akku jest porównywana z wartością przesłaną w poleceniu MIN. Po wykonaniu polecenia mniejsza z dwóch wartości znajduje się w Akku. Obie wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 100 (* Załadować 100 do Akku *)
MIN 200 (* Porównać z wartością 200 *)
ST Value2 (* Zapisać 100 w Value2 (ponieważ jest to mniejsza wartość) *)
```

Przykład w ST:

```
Ergebnis := Min(100, 200); (* Zapisać 100 w Value2 (ponieważ jest to mniejsza wartość) *)
```

3.4.1.7 MOD i MOD(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Akku dzieli się przez jedną lub więcej zmiennych lub stałych, reszta z dzielenia jako wynik znajduje się w Akku. W operacji modulo z nawiasami Akku dzieli się przez wynik wyrażenia w nawiasie i z tego tworzy się modulo. Możliwych jest do 6 poziomów nawiasów.

Przykład w AWL:

```
LD 25 (* Załadować dzielną *)
MOD 20 (* Dzielenie 25/20 à modulo = 5 *)
ST Var1 (* Zapisanie wyniku 5 w Var1 *)
LD 25 (* Załadować dzielną *)
MOD( Var1 (* Wynik = 25/(Var1 + 10) à modulo do Akku *)
ADD 10
)
ST Var3 (* Zapisanie wyniku 10 w Var3 *)
```

Przykład w ST:

```
Ergebnis := 25 MOD 20; (* Zapisanie wyniku 5 w Var1 *)
Ergebnis := 25 MOD (Var1 + 10); (* Wynik = 25/(Var1 + 10) à modulo do Akku *)
```

3.4.1.8 MUL i MUL(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Mnożenie Akku przez jedną lub więcej zmiennych lub stałych. W operacji mnożenia z nawiasami Akku mnoży się przez wynik wyrażenia w nawiasie. Możliwych jest do 6 poziomów nawiasów. Obie wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 25 (* Załadować mnożnik *)
MUL Var1, Var2 (* 25 * Var1 * Var2 *)
ST Var2 (* Zapisanie wyniku *)
LD 25 (* Załadować mnożnik *)
MUL( Var1 (* Wynik = 25*(Var1 + Var2) *)
ADD Var2
)
ST Var3 (* Zapisanie wyniku jako zmiennej Var3 *)
```

Przykład w ST:

```
Ergebnis := 25 * Var1 * Var2;
Ergebnis := 25 * (Var1 + Var2);
```

3.4.1.9 MUX

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Za pomocą indeksu, który znajduje się przed poleceniem w Akku, można wybrać różne stałe lub zmienne. Dostęp do pierwszej wartości jest możliwy za pomocą indeksu 0. Wybrana wartość zostanie załadowana do Akku. Liczba wartości jest ograniczona tylko przez pamięć programową.

Przykład w AWL:

```
LD 1 (* Wybrać żądany element *)
MUX 10,20,30,40,Value1 (* Polecenie MUX z 4 stałymi i jedną zmienną *)
ST Value (* Zapisanie wyniku = 20 *)
```

Przykład w ST:

```
Ergebnis := Mux(1, 10, 20, 30, 40, Value1) (* Zapisanie wyniku = 20 *)
```

3.4.1.10 SUB i SUB(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Odejmuje od Akku jedną lub więcej zmiennych lub stałych. W operacji odejmowania z nawiasami odejmuje od Akku wynik wyrażenia w nawiasie. Możliwych jest do 6 poziomów nawiasów. Odejmowane wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 10
SUB Var1 (* Wynik = 10 - Var1 *)
ST Ergebnis
LD 20
SUB Var1, Var2, 30 (* Wynik = 20 - Var1 - Var2 - 30 *)
ST Ergebnis
LD 20
SUB( 6 (* Odjąć od 20 zawartość nawiasów *)
AND 2
) (* Wynik = 20 - (6 AND 2) *)
ST Ergebnis (* Wynik = 18 *)
```

Przykład w ST:

```
Ergebnis := 10 - Value1;
```


3.4.2 Zaawansowane operatory matematyczne

i Informacja

Wymienione tutaj operatory wymagają bardzo intensywnych obliczeń. Może to prowadzić do znacznie dłuższego czasu wykonywania programu PLC.

3.4.2.1 COS, ACOS, SIN, ASIN, TAN, ATAN

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		

	BOOL	BYTE	INT	DINT
Typ danych				X

Obliczanie odpowiedniej funkcji matematycznej. Obliczana wartość wyrażona w minutach kątowych musi być dostępna w Akku. Skalowanie odpowiada 1 = 1000.

Przeliczenie: Kąt w radianach = (kąt w stopniach * PI / 180)*1000
 np. kąt 90° zostanie przeliczony w następujący sposób à 90° * 3.14 / 180) *1000 = 1571

$$AE = \sin\left(\frac{AE}{1000}\right) \cdot 1000 \quad AE = \cos\left(\frac{AE}{1000}\right) \cdot 1000 \quad AE = \tan\left(\frac{AE}{1000}\right) \cdot 1000$$

Przykład w AWL:

```
LD 1234
SIN
ST Ergebnis (* Wynik = 943 *)
```

Przykład w ST:

```
Ergebnis := COS(1234); (* Wynik = 330 *)
Ergebnis := ACOS(330); (* Wynik = 1234 *)
Ergebnis := SIN(1234); (* Wynik = 943 *)
Ergebnis := ASIN(943); (* Wynik = 1231 *)
Ergebnis := TAN(999); (* Wynik = 1553 *)
Ergebnis := ATAN(1553); (* Wynik = 998 *)
```

3.4.2.2 EXP

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		
	BOOL	BYTE	INT	DINT			
Typ danych				X			

Tworzy z Akku funkcję wykładniczą o podstawie równej liczbie Eulera (2,718). Można podać 3 miejsca po przecinku, tzn. 1,002 musi być wprowadzone jako 1002.

$$AE = e^{\left(\frac{AE}{1000}\right)} \cdot 1000$$

Przykład w AWL:

```
LD 1000
EXP
ST Ergebnis (* Wynik = 2718 *)
```

Przykład w ST:

```
Ergebnis := EXP(1000); (* Wynik = 2718 *)
```

3.4.2.3 LN

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		
	BOOL	BYTE	INT	DINT			
Typ danych				X			

Logarytm o podstawie równej e (2,718). Można podać 3 miejsca po przecinku, tzn. 1,000 musi być wprowadzone jako 1000.

$$AE = \ln\left(\frac{AE}{1000}\right) \cdot 1000$$

Przykład w AWL:

```
LD 1234
LN
ST Ergebnis
```

Przykład w ST:

```
Ergebnis := LN(1234); (* Wynik = 210 *)
```

3.4.2.4 LOG

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		
	BOOL	BYTE	INT	DINT			
Typ danych				X			

Tworzy z Akku logarytm o podstawie równej 10. Można podać 3 miejsca po przecinku, tzn. 1,000 musi być wprowadzone jako 1000.

$$AE = \log_{10} \left(\frac{AE}{1000} \right) \cdot 1000$$

Przykład w AWL:

```
LD 1234
LOG
ST Ergebnis (* Wynik = 91 *)
```

Przykład w ST:

```
Ergebnis := LOG(1234); (* Wynik = 91 *)
```

3.4.2.5 SQRT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X		
	BOOL	BYTE	INT	DINT			
Typ danych				X			

Tworzy z Akku pierwiastek kwadratowy. Można podać 3 miejsca po przecinku, tzn. 1,000 musi być wprowadzone jako 1000.

$$AE = \sqrt{\left(\frac{AE}{1000} \right)} \cdot 1000$$

Przykład w AWL:

```
LD 1234
SQRT
ST Ergebnis (* Wynik = 1110 *)
```

Przykład w ST:

```
Ergebnis := SQRT(1234); (* Wynik = 1110 *)
```

3.4.3 Operatory bitowe

3.4.3.1 AND i AND(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X	X	X	X			

Operacja bitowa I między AE/Akku i jedną lub dwiema zmiennymi lub stałymi. Operacja bitowa I(...) między AE/Akku i AE/Akku, które zostało wcześniej utworzone w nawiasach. Możliwych jest do 6 poziomów nawiasów. Wszystkie wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 170
AND 204 (* Operacja AND między 2 stałymi *)
(* Akku = 136 (patrz przykład w tabeli) *)

LD 170 (* Operacja między jedną stałą i 2 zmiennymi.*)
AND Var1, Var2 (* Akku = 170dec AND Var1 AND Var2 *)

LD Var1
AND ( Var2 (* AE/Akku = Var1 AND ( Var2 OR Var3 ) *)
OR Var3
)
```

Przykład w ST:

```
Ergebnis := 170 AND 204; (* Wynik = 136dec *)
```

Var2	Var1	Wynik
0	0	0
0	1	0
1	0	0
1	1	1

Przykład: 170dec (1010 1010bin) AND 204dec (1100 1100bin) = (1000 1000bin) 136dec

3.4.3.2 ANDN i ANDN(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X	X	X	X			

Operacja bitowa I między AE/Akku i zanegowanym operandem. Operacja bitowa I (...) między AE/Akku i zanegowanym wynikiem w nawiasach. Możliwych jest do 6 poziomów nawiasów. Łączone wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 2#0000_1111
ANDN 2#0011_1010 (* Operacja ANDN między 2 stałymi *)
(* Akku = 2#1111_0101 *)

LD 170 (* Operacja między jedną stałą i 2 zmiennymi. *)
ANDN Var1, Var2 (* Akku = 170d ANDN Var1 ANDN Var2 *)

LD Var1
ANDN ( Var2 (* AE/Akku = Var1 ANDN ( Var2 OR Var3 ) *)
OR Var3
)
```

Var2	Var1	Wynik
0	0	1
0	1	1
1	0	1
1	1	0

Przykład: 170dec (1010 1010bin) AND 204dec (1100 1100bin) = (1000 1000bin) 136dec

3.4.3.3 NOT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Typ danych	X	X	X	X

Negacja bitowa Akku.

Przykład w AWL:

```
LD BYTE#10 (* Załadować do AKKU wartość 10dec w formacie Byte *)
NOT (* Wartość jest sprowadzana na poziom bitowy (0000 1010), *)
(* Negacja bitowa (1111 0101) i ponowna konwersja do *)
(* wartości dziesiętnej, wynik = 245dec *)
ST Var3 (* Zapisanie wyniku jako zmiennej Var3 *)
```

Przykład w ST:

```
Ergebnis := not BYTE#10; (* Wynik = 245dec *)
```

3.4.3.4 OR i OR(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X	X	X	X			

Operacja bitowa LUB między AE/Akku i jedną lub dwiema zmiennymi lub stałymi. Operacja bitowa LUB(...) między AE/Akku i AE/Akku, które zostało wcześniej utworzone w nawiasach. Możliwych jest do 6 poziomów nawiasów. Wszystkie wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 170
OR 204 (* Operacja OR między 2 stałymi *)

LD 170 (* Operacja między jedną stałą i 2 zmiennymi. *)
OR Var1, Var2 (* Akku = 170d OR Var1OR Var2 *)

LD Var1
OR ( Var2 (* AE/Akku = Var1 OR ( Var2 AND Var3 ) *)
AND Var3
)
```

Przykład w ST:

```
Ergebnis := 170 or 204; (* Wynik = 238 *)
```

Var2	Var1	Wynik
0	0	0
0	1	1
1	0	1
1	1	1

3.4.3.5 ORN i ORN(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Typ danych	X	X	X	X

Operacja bitowa LUB między AE/Akku i zanegowanym operandem. Operacja bitowa LUB (...) między AE/Akku i zanegowanym wynikiem w nawiasach. Możliwych jest do 6 poziomów nawiasów. Łączone wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 2#0000_1111
ORN 2#0011_1010 (* Operacja ORN między 2 stałymi *)
(* Akku = 2#1100_0000 *)

LD 170 (* Operacja między jedną stałą i 2 zmiennymi. *)
ORN Var1, Var2 (* Akku = 170d ORN Var1 ORN Var2 *)

LD Var1
ORN ( Var2 (* AE/Akku = Var1 ORN ( Var2 OR Var3 ) *)
OR Var3
)
```

Przykład w ST:

```
Ergebnis := 2#0000_1111 ORN 2#0011_1010; (* Wynik = 2#1100_0000 *)
```

Var2	Var1	Wynik
0	0	1
0	1	0
1	0	0
1	1	0

3.4.3.6 ROL

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Obrót bitowy Akku w lewo. Zawartość Akku zostanie przesunięta w lewo n razy, przy czym lewy bit zostanie ponownie wprowadzony po prawej stronie.

Przykład w AWL:

```
LD 175      (* Ładuje wartość 1010_1111*)
ROL 2      (* Zawartość Akku jest obracana 2 razy w lewo *)
ST Value1 (* Zapisuje wartość 1011_1110 *)
```

Przykład w ST:

```
Ergebnis := ROL(BYTE#175, 2); (* Wynik = 2#1011_1110 *)
Ergebnis := ROL(INT#175, 2); (* Wynik = 16#C02B *)
```

3.4.3.7 ROR

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Obrót bitowy Akku w prawo. Zawartość Akku zostanie przesunięta w prawo n razy, przy czym prawy bit zostanie ponownie wprowadzony po lewej stronie.

Przykład w AWL:

```
LD 175      (* Ładuje wartość 1010_1111*)
ROR 2      (* Zawartość Akku jest obracana 2 razy w prawo *)
ST Value1 (* Zapisuje wartość 1110_1011 *)
```

Przykład w ST:

```
Ergebnis := ROR(BYTE#175, 2); (* Wynik = 2#1110_1011 *)
```

3.4.3.8 SiR

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X						

Ustawianie i resetowanie zmiennej boolowskiej, gdy poprzedni wynik operacji (AE) miał wartość TRUE.

Przykład w AWL:

```
LD TRUE (* Ładuje do AE wartość TRUE *)
S Var1 (* VAR1 jest ustawiane na wartość TRUE *)
R Var1 (* VAR1 jest ustawiane na wartość FALSE *)
```

Przykład w ST:

```
Ergebnis := TRUE;
Ergebnis := FALSE;
```

3.4.3.9 SHL

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Przesunięcie bitowe Akku w lewo Zawartość Akku zostanie przesunięta w lewo n razy, wypchnięte bity zostaną utracone.

Przykład w AWL:

```
LD 175 (* Ładuje wartość 1010_1111*)
SHL 2 (* Zawartość Akku jest przesuwana 2 razy w lewo *)
ST Value1 (* Zapisuje wartość 1011_1100 ab *)
```

Przykład w ST:

```
Ergebnis := SHL(BYTE#175, 2); (* Wynik = 2#1011_1100 *)
Ergebnis := SHL(INT#175, 2); (* Wynik = 16#2BC *)
```

3.4.3.10 SHR

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Przesunięcie bitowe Akku w prawo Zawartość Akku zostanie przesunięta w prawo n razy, wypchnięte bity zostaną utracone.

Przykład w AWL:

```
LD 175      (* Ładuje wartość 1010_1111*)
SHR 2      (* Zawartość Akku jest przesuwana 2 razy w prawo *)
ST Value1 (* Zapisuje wartość 0010_1011 ab *)
```

Przykład w ST:

```
Ergebnis := SHR(BYTE#175, 2); (* Wynik = 2#0010_1011 *)
```

3.4.3.11 XOR i XOR(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Typ danych	X			

Operacja bitowa „LUB wykluczające” między AE/Akku i jedną lub dwiema zmiennymi lub stałymi. Pierwsza wartość znajduje się w AE/Akku, a druga jest ładowana za pomocą polecenia lub znajduje się w nawiasie. Łączone wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 2#0000_1111
XOR 2#0011_1010 (* Operacja XOR między 2 stałymi *)
                (* Akku = 2#0011_0101 *)

LD 170          (* Operacja między jedną stałą i 2 zmiennymi. *)
XOR Var1, Var2 (* Akku = 170d XOR Var1 XOR Var2 *)

LD Var1
XOR ( Var2      (* AE/Akku = Var1 XOR ( Var2 OR Var3 ) *)
OR Var3
)
```

Przykład w ST:

```
Ergebnis := 2#0000_1111 XOR 2#0011_1010; (* Wynik = 2#0011_0101 *)
```

Var2	Var1	Wynik
0	0	0
0	1	1
1	0	1
1	1	0

3.4.3.12 XORN i XORN(

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X						

Operacja bitowa LUB wykluczające między AE/Akku i zanegowanym operandem. Operacja bitowa LUB wykluczające (...) między AE/Akku i zanegowanym wynikiem w nawiasach. Możliwych jest do 6 poziomów nawiasów. Łączone wartości muszą należeć do tego samego typu zmiennych.

Przykład w AWL:

```
LD 2#0000_1111
XORN 2#0011_1010 (* Operacja XORN między 2 stałymi *)
(* Akku = 2#1100_1010 *)

LD 170 (* Operacja między jedną stałą i 2 zmiennymi. *)
XORN Var1, Var2 (* Akku = 170d XORN Var1 XORN Var2 *)

LD Var1
XORN ( Var2 (* AE/Akku = Var1 XORN ( Var2 OR Var3 ) *)
OR Var3
)
```

Przykład w ST:

```
Ergebnis := 2#0000_1111 XORN 2#0011_1010; (* Wynik = 2#1100_1010 *)
```

Var2	Var1	Wynik
0	0	1
0	1	0
1	0	0
1	1	1

3.4.4 Operatory ładowania i zapisu

3.4.4.1 LD

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X	X	X	X			

Ładuje stałą lub zmienną do AE lub Akku.

Przykład w AWL:

```
LD 10 (* Ładuje 10 jako BYTE *)
LD -1000 (* Ładuje -1000 jako INT *)
LD Value1 (* Ładuje zmienną Value1 *)
```

3.4.4.2 LDN

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X						

Ładuje zanegowaną zmienną boolowską do AE.

Przykład w AWL:

```
LDN Value1 (* Value1 = TRUE à AE = FALSE *)
ST Value2 (* Zapisanie do Value2 = FALSE *)
```

3.4.4.3 ST

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X	X	X	X			

Zapisuje zawartość AE/Akku do zmiennej. Zapisywana zmienna musi być zgodna z wcześniej załadowanym i przetwarzanym typem danych.

Przykład w AWL:

```
LD 100 (* Ładuje wartość 1010_1111 *)
ST Value1 (* Zawartość Akku 100 jest zapisywana w Value1 *)
```

3.4.4.4 STN

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X						

Zapisuje zawartość AE do zmiennej i neguje ją. Zapisywana zmienna musi być zgodna z wcześniej załadowanym i przetwarzanym typem danych.

Przykład w AWL:

```
LD Value1 (* Value1 = TRUE à AE = TRUE *)
STN Value2 (* Zapisanie do Value2 = FALSE *)
```

3.4.5 Operatory porównywania

3.4.5.1 EQ

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Porównuje zawartość Akku ze zmienną lub stałą. Jeżeli wartości są równe, AE zostanie ustawione na wartość TRUE.

Przykład w AWL:

```
LD Value1 (* Value = 5 *)
EQ 10 (* AE = Czy 5 jest równe 10 ? *)
JMPC NextStep (* AE = FALSE à program nie wykonuje skoku *)
ADD 1
NextStep:
ST Value1
```

Przykład w ST:

```
(* Czy Value = 10 *)
if Value = 10 then
    Value2 := 5;
end_if;
```

3.4.5.2 GE

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Porównuje zawartość Akku ze zmienną lub stałą. Jeżeli wartość w Akku jest większa lub równa zmiennej lub stałej, AE zostanie ustawione na wartość TRUE.

Przykład w AWL:

```
LD Value1 (* Value1 = 5 *)
GE 10 (* Czy 5 jest większe lub równe 10? *)
JMPC NextStep (* AE = FALSE à program nie wykonuje skoku *)
ADD 1

NextStep:
ST Value1
```

Przykład w ST:

```
(* Czy 5 jest większe lub równe 10? *)
if Value >= 10 then
    Value := Value - 1
end_if;
```


3.4.5.3 GT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Porównuje zawartość Akku ze zmienną lub stałą. Jeżeli wartość w Akku jest większa od zmiennej lub stałej, AE zostanie ustawione na wartość TRUE.

Przykład w AWL:

```
LD Value1 (* Value1 = 12 *)
GT 8 (* Czy 12 jest większe od 8? *)
JMPC NextStep (* AE = TRUE - program wykonuje skok *)
ADD 1
NextStep:
ST Value1
```

Przykład w ST:

```
(* Czy 12 jest większe od 8? *)
if Value > 8 then
  Value := 0;
end_if;
```

3.4.5.4 LE

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X	X	X			

Porównuje zawartość Akku ze zmienną lub stałą. Jeżeli wartość w Akku jest mniejsza lub równa zmiennej lub stałej, AE zostanie ustawione na wartość TRUE.

Przykład w AWL:

```
LD Value1 (* Value1 = 5 *)
LE 10 (* Czy 5 jest mniejsze lub równe 10? *)
JMPC NextStep:
ST Value1
```

Przykład w ST:

```
(* Czy Value jest mniejsze lub równe 10?*)
if Value <= 10 then
  Value := 11;
end_if;
```

3.4.5.5 LT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Typ danych		X	X	X

Porównuje zawartość Akku ze zmienną lub stałą. Jeżeli wartość w Akku jest mniejsza od zmiennej lub stałej, AE zostanie ustawione na wartość TRUE.

Przykład w AWL:

```
LD Value1 (* Value1 = 12 *)
LT 8 (* Czy 12 jest mniejsze od 8 ? *)
JMPC NextStep (* AE = FALSE à program nie wykonuje skoku *)
ADD 1
NextStep:
ST Value1
```

Przykład w ST:

```
(* Czy Value jest mniejsze od 0? *)
if Value < 0 then
    Value := 0;
end_if;
```

3.4.5.6 NE

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Typ danych		X	X	X

Porównuje zawartość Akku ze zmienną lub stałą. Jeżeli wartość w Akku jest nierówna zmiennej lub stałej, AE zostanie ustawione na wartość TRUE.

Przykład w AWL:

```
LD Value1 (* Value1 = 5 *)
NE 10 (*Czy 5 jest nierówne 10 ?*)
JMPC NextStep (* AE = TRUE à program wykonuje skok *)
ADD 1
NextStep:
ST Value1
```

Przykład w ST:

```
if Value <> 5 then
    Value := 5;
end_if;
```

3.5 Wartości procesowe

Wszystkie analogowe i cyfrowe wejścia i wyjścia lub wartość zadana i rzeczywista magistrali mogą być odczytywane i przetwarzane przez PLC lub ustawiane przez PLC (dla wartości wyjściowych). Dostęp do poszczególnych wartości odbywa się za pośrednictwem wymienionych niżej wartości procesowych. Dla wszystkich wartości wyjściowych należy zaprogramować wyjście (np. wyjścia cyfrowe lub wartość zadaną PLC) w taki sposób, aby PLC był źródłem zdarzenia. Wszystkie dane procesowe są wczytywane przez PLC z urządzenia na początku każdego nowego cyklu i zapisywane do urządzenia dopiero na końcu programu PLC! Poniższe tabele przedstawiają wszystkie wartości, do których funkcja PLC ma bezpośredni dostęp. Dostęp do wszystkich innych wartości procesowych jest konieczny za pośrednictwem bloków funkcyjnych MC_ReadParameter lub MC_WriteParameter.

3.5.1 Wejścia i wyjścia

Są tutaj zebrane wszystkie wartości procesowe, które opisują interfejs WE/WY urządzenia.

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_0_Set_digital_output	Ustawianie wyjść cyfrowych	Bit 0: Mfr1 Bit 1: Mfr2 Bit 2: DOUT 1 Bit 3: DOUT 2 Bit 4: DOUT 1 CU5-MLT Bit 5: DOUT 2 CU5-MLT Bit 6: DOUT 3 CU5-MLT Bit 7: DOUT 4 CU5-MLT Bit 8: Fun. cyfr. AOUT Bit 9: Wolny Bit 10: BusIO Bit0 Bit 11: BusIO Bit1 Bit 12: BusIO Bit2 Bit 13: BusIO Bit3 Bit 14: BusIO Bit4 Bit 15: BusIO Bit5	UINT	R/W	SK 5xxP
_0_Set_digital_output	Ustawianie wyjść cyfrowych	Bit 0: Mfr1 Bit 1: Mfr2 Bit 2: DOUT1 Bit 3: DOUT2 Bit 4: Fun. cyfr. AOUT Bit 5: DOUT3 (Din7) Bit 6: Słowo stanu Bit 10 Bit 7: Słowo stanu Bit 13 Bit 8: BusIO Bit0 Bit 9: BusIO Bit1 Bit 10: BusIO Bit2 Bit 11: BusIO Bit3 Bit 12: BusIO Bit4 Bit 13: BusIO Bit5 Bit 14: BusIO Bit6 Bit 15: BusIO Bit7	UINT	R/W	SK 54xE
_0_Set_digital_output	Ustawianie wyjść cyfrowych	Bit 0: Mfr1 Bit 1: Mfr2	UINT	R/W	SK 52xE SK 53xE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
		Bit 2: DOUT1 Bit 3: DOUT2 Bit 4: Fun. cyfr. AOUT Bit 5: Wolny Bit 6: Słowo stanu Bit 10 Bit 7: Słowo stanu Bit 13 Bit 8: BusIO Bit0 Bit 9: BusIO Bit1 Bit 10: BusIO Bit2 Bit 11: BusIO Bit3 Bit 12: BusIO Bit4 Bit 13: BusIO Bit5 Bit 14: BusIO Bit6 Bit 15: BusIO Bit7			
_0_Set_digital_output	Ustawianie wyjść cyfrowych	Bit 0: DOUT1 Bit 1: BusIO Bit0 Bit 2: BusIO Bit1 Bit 3: BusIO Bit2 Bit 4: BusIO Bit3 Bit 5: BusIO Bit4 Bit 6: BusIO Bit5 Bit 7: BusIO Bit6 Bit 8: BusIO Bit7 Bit 9: Bus PZD Bit 10 Bit 10: Bus PZD Bit 13 Bit 11: DOUT2	UINT	R/W	SK 2xxE SK 2xxE-FDS
_0_Set_digital_output	Ustawianie wyjść cyfrowych	Bit 0: DOUT1 Bit 1: DOUT2 Bit 2: BusIO Bit0 Bit 3: BusIO Bit1 Bit 4: BusIO Bit2 Bit 5: BusIO Bit3 Bit 6: BusIO Bit4 Bit 7: BusIO Bit5 Bit 8: BusIO Bit6 Bit 9: BusIO Bit7 Bit 10: Bus PZD Bit 10 Bit 11: Bus PZD Bit 13	UINT	R/W	SK 180E SK 190E
_0_Set_digital_output	Ustawianie wyjść cyfrowych	Bit 0: DOUT1 Bit 1: DOUT2 Bit 2: DOUT_BRAKE Bit 3: DOUT_BUS1 Bit 4: DOUT_BUS2	UINT	R/W	SK 155E-FDS SK 175E-FDS
_1_Set_analog_output	Ustawianie wyjścia analogowego FI	10,0V = 100	BYTE	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE
_2_Set_external_analog_out1	Ustawianie wyjścia analogowego 1. IOE	10,0V = 100	BYTE	R/W	SK 5xxP SK 54xE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
					SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_3_Set_external_analog_out2	Ustawianie wyjścia analogowego 2. IOE	10,0V = 100	BYTE	R/W	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_4_State_digital_output	Stan wyjść cyfrowych	Bit 0: Mfr1 Bit 1: Mfr2 Bit 2: DOUT 1 Bit 3: DOUT 2 Bit 4: DOUT 1 CU5-MLT Bit 5: DOUT 2 CU5-MLT Bit 6: DOUT 3 CU5-MLT Bit 7: DOUT 4 CU5-MLT Bit 8: Fun. cyfr. AOUT Bit 9: Wolny Bit 10: DOUT1 IOE1 Bit 11: DOUT2 IOE1 Bit 12: DOUT1 IOE2 Bit 13: DOUT2 IOE2 Bit 14: Wolny Bit 15: Wolny	INT	R	SK 5xxP
_4_State_digital_output	Stan wyjść cyfrowych	Bit 0: Mfr1 Bit 1: Mfr2 Bit 2: DOUT1 Bit 3: DOUT2 Bit 4: Fun. cyfr. AOUT Bit 5: DOUT3 (Din7) Bit 6: Słowo stanu Bit 8 Bit 7: Słowo stanu Bit 9 Bit 8: BusIO Bit0 Bit 9: BusIO Bit1 Bit 10: BusIO Bit2 Bit 11: BusIO Bit3 Bit 12: BusIO Bit4 Bit 13: BusIO Bit5 Bit 14: BusIO Bit6 Bit 15: BusIO Bit7	INT	R	SK 54xE
_4_State_digital_output	Stan wyjść cyfrowych	P711	BYTE	R	SK 52xE SK 53xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_4_State_digital_output	Stan wyjść cyfrowych	Bit 0: DOUT1	BYTE	R	SK 155E-FDS

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
		Bit 1: DOUT2 Bit 2: DOUT_BRAKE Bit 3: DOUT_BUS1 Bit 4: DOUT_BUS2			SK 175E-FDS
_5_State_Digital_input	Stan wejść cyfrowych	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: DIN4 Bit 4: DIN5 Bit 5: DIN6 Bit 6: DIN1 CU5-MLT Bit 7: DIN2 CU5-MLT Bit 8: DIN3 CU5-MLT Bit 9: DIN4 CU5-MLT Bit 10: Wolny Bit 11: Wolny Bit 12: Funkcja cyfrowa AIN1 Bit 8: Funkcja cyfrowa AIN2	INT	R	SK 5xxP
_5_State_Digital_input	Stan wejść cyfrowych	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: DIN4 Bit 4: DIN5 Bit 5: DIN6 Bit 6: DIN7 Bit 7: Funkcja cyfrowa AIN1 Bit 8: Funkcja cyfrowa AIN2	INT	R	SK 54xE
_5_State_Digital_input	Stan wejść cyfrowych	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: DIN4 Bit 4: DIN5 Bit 5: DIN6 Bit 6: DIN7	INT	R	SK 52xE SK 53xE
_5_State_Digital_input	Stan wejść cyfrowych	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: DIN4 Bit 4: Wolny Bit 5: Termistor PTC Bit 6: Wolny Bit 7: Wolny Bit 8: DIN1 IOE 1 Bit 9: DIN2 IOE 1 Bit 10: DIN3 IOE 1 Bit 11: DIN4 IOE 1	INT	R	SK 2xxE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
		Bit 12: DIN1 IOE 2 Bit 13: DIN2 IOE 2 Bit 14: DIN3 IOE 2 Bit 15: DIN4 IOE 2			
_5_State_Digital_input	Stan wejść cyfrowych	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: AIN1 Bit 4: AIN2 Bit 5: Termistor PTC Bit 6: Wolny Bit 7: Wolny Bit 8: DIN1 IOE 1 Bit 9: DIN2 IOE 1 Bit 10: DIN3 IOE 1 Bit 11: DIN4 IOE 1 Bit 12: DIN1 IOE 2 Bit 13: DIN2 IOE 2 Bit 14: DIN3 IOE 2 Bit 15: DIN4 IOE 2	INT	R	SK 180E SK 190E
_5_State_Digital_input	Stan wejść cyfrowych	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: TF (termistor PTC) Bit 4: DIN-BUS1 (ASi1) Bit 5: DIN-BUS2 (ASi2) Bit 6: DIN-BUS3 (ASi3) Bit 7: DIN-BUS4 (ASi4) Bit 8: BDDI1 (ASIO3) Bit 9: BDDI2 (ASIO4) Bit 10: STO	INT	R	SK 155E-FDS SK 175E-FDS
_5_State_Digital_input	Stan wejść cyfrowych	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: DIN4 Bit 4: DIN5 Bit 5: DIN6/AIN1 Bit 6: DIN7/AIN2 Bit 7: Termistor PTC Bit 8: DIN1 IOE 1 Bit 9: DIN2 IOE 1 Bit 10: DIN3 IOE 1 Bit 11: DIN4 IOE 1 Bit 12: DIN1 IOE 2 Bit 13: DIN2 IOE 2 Bit 14: DIN3 IOE 2 Bit 15: DIN4 IOE 2	INT	R	SK 2xxE-FDS
_6_Delay_digital_inputs	Stan wejść cyfrowych zgodnie z P475	Bit 0: DIN1 Bit 1: DIN2	INT	R	SK 5xxP SK 54xE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
		Bit 2: DIN3 Bit 3: DIN4 Bit 4: DIN5 Bit 5: DIN6 Bit 6: DIN7 Bit 7: Funkcja cyfrowa AIN1 Bit 8: Funkcja cyfrowa AIN2			
_6_Delay_digital_inputs	Stan wejść cyfrowych zgodnie z P475	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: DIN4 Bit 4: DIN5 Bit 5: DIN6 Bit 6: DIN7	INT	R	SK 52xE SK 53xE
_6_Delay_digital_inputs	Stan wejść cyfrowych zgodnie z P475	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: AIN1 Bit 4: AIN2 Bit 5: Termistor PTC Bit 6: Wolny Bit 7: Wolny Bit 8: DIN1 IOE 1 Bit 9: DIN2 IOE 1 Bit 10: DIN3 IOE 1 Bit 11: DIN4 IOE 1 Bit 12: DIN1 IOE 2 Bit 13: DIN2 IOE 2 Bit 14: DIN3 IOE 2 Bit 15: DIN4 IOE 2	INT	R	SK 2xxE SK 180E SK 190E
_6_Delay_digital_inputs	Stan wejść cyfrowych zgodnie z P475	Bit 0: DIN1 Bit 1: DIN2 Bit 2: DIN3 Bit 3: DIN4 Bit 4: DIN5 Bit 5: DIN6/AIN1 Bit 6: DIN7/AIN2 Bit 7: Termistor PTC Bit 8: DIN1 IOE 1 Bit 9: DIN2 IOE 1 Bit 10: DIN3 IOE 1 Bit 11: DIN4 IOE 1 Bit 12: DIN1 IOE 2 Bit 13: DIN2 IOE 2 Bit 14: DIN3 IOE 2 Bit 15: DIN4 IOE 2	INT	R	SK 2xxE-FDS
_7_Analog_input1	Wartość wejścia analogowego 1 (AIN1)	10,00V = 1000	INT	R	Wszystkie

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_8_Analog_input2	Wartość wejścia analogowego 2 (AIN2)	10,00V = 1000	INT	R	Wszystkie
_9_Analog_input3	Wartość funkcji analogowej DIN2	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 155E-FDS SK 175E-FDS
_10_Analog_input4	Wartość funkcji analogowej DIN3	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 155E-FDS SK 175E-FDS
_11_External_analog_input1	Wartość wejścia analogowego 1 (1.IOE)	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_12_External_analog_input2	Wartość wejścia analogowego 2 (1.IOE)	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_13_External_analog_input3	Wartość wejścia analogowego 1 (2.IOE)	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_14_External_analog_input4	Wartość wejścia analogowego 2 (2.IOE)	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_15_State_analog_output	Stan wyjścia analogowego	10,0V = 100	BYTE	R	SK 5xxP SK 54xE
_16_State_ext_analog_out1	Stan wyjścia analogowego (1. IOE)	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_17_State_ext_analog_out2	Stan wyjścia analogowego (2. IOE)	10,00V = 1000	INT	R	SK 5xxP SK 54xE SK 2xxE SK 180E

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
					SK 190E
_18_Dip_switch_state	Stan przełączników DIP	Bit 0: DIP1 Bit 1: DIP2 Bit 2: DIP3 Bit 3: DIP4 Bit 4: DIP_I1 Bit 5: DIP_I2 Bit 6: DIP_I3 Bit 7: DIP_I4	INT	R	SK 155E-FDS SK 175E-FDS

3.5.2 Wartości zadane i rzeczywiste PLC

Wymienione tutaj wartości procesowe stanowią interfejs między PLC i urządzeniem. Funkcja wartości zadanych PLC jest określona w (P553).

i Informacja

Wartość procesowa PLC_control_word nadpisuje blok funkcyjny MC_Power. Wartości zadane PLC nadpisują bloki funkcyjne MC_Move.... i MC_Home.

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_20_PLC_control_word	PLC słowo sterujące	Odpowiada profilowi USS	INT	R/W	Wszystkie
_21_PLC_set_val1	PLC wartość zadana 1	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_22_PLC_set_val2	PLC wartość zadana 2	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_23_PLC_set_val3	PLC wartość zadana 3	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_24_PLC_set_val4	PLC wartość zadana 4	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_25_PLC_set_val5	PLC wartość zadana 5	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_26_PLC_additional_co	PLC dodatkowe słowo	Odpowiada profilowi	INT	R/W	SK 5xxP

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
ntrol_word1	sterujące 1	USS			SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_27_PLC_additional_control_word2	PLC dodatkowe słowo sterujące 2	Odpowiada profilowi USS	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_28_PLC_status_word	PLC słowo stanu	Odpowiada profilowi USS	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_29_PLC_act_val1	PLC wartość rzeczywista 1	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_30_PLC_act_val2	PLC wartość rzeczywista 2	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_31_PLC_act_val3	PLC wartość rzeczywista 3	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_32_PLC_act_val4	PLC wartość rzeczywista 4	100% = 4000h	INT	R/W	SK 5xxP SK 54xE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
					SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_33_PLC_act_val5	PLC wartość rzeczywista 5	100% = 4000h	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_34_PLC_Busmaster_Control_word	Słowo sterujące funkcji wiodącej (funkcja modułu nadrzędnego magistrali) przez PLC	Odpowiada profilowi USS	INT	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_35_PLC_32Bit_set_val1	32bit PLC wartość zadana - P553[1] = Low Part wartości 32bit - P553[2] = High Part wartości 32bit	-	LONG	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_36_PLC_32Bit_act_val1	32bit PLC wartość rzeczywista - PLC wartość rzeczywista 1 = Low Part wartości 32bit - PLC wartość rzeczywista 2 = High Part wartości 32bit	-	LONG	R/W	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_37_PLC_status_bits	Wirtualne wyjścia stanu PLC	Bit 0: PLC-DOUT1 Bit 1: PLC-DOUT2	INT	R/W	SK 155E-FDS SK 175E-FDS
_38_PLC_control_bits	Wirtualne wyjścia sterujące PLC	Bit 0: PLC-DIN1 Bit 1: PLC-DIN2 Bit 2: PLC-DIN3 Bit 3: PLC-DIN4 Bit 4: PLC-DIN5 Bit 5: PLC-DIN6 Bit 6: PLC-DIN7 Bit 7: PLC-DIN8	INT	R/W	SK 155E-FDS SK 175E-FDS

3.5.3 Wartości zadane i rzeczywiste magistrali

Wartości procesowe odzwierciedlają wszystkie wartości zadane i rzeczywiste, które są przesyłane do urządzenia za pośrednictwem różnych systemów magistralowych.

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_40_Inverter_status	FI słowo stanu	Odpowiada profilowi USS	INT	R	Wszystkie
_41_Inverter_act_val1	FI wartość rzeczywista 1	100% = 4000h	INT	R	Wszystkie
_42_Inverter_act_val2	FI wartość rzeczywista 2	100% = 4000h	INT	R	Wszystkie
_43_Inverter_act_val3	FI wartość rzeczywista 3	100% = 4000h	INT	R	Wszystkie
_44_Inverter_act_val4	FI wartość rzeczywista 4	100% = 4000h	INT	R	SK 5xxP SK 54xE
_45_Inverter_act_val5	FI wartość rzeczywista 5	100% = 4000h	INT	R	SK 5xxP SK 54xE
_46_Inverter_lead_val1	Funkcja Broadcast Master Wartość wiodąca 1	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_47_Inverter_lead_val2	Funkcja Broadcast Master Wartość wiodąca 2	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_48_Inverter_lead_val3	Funkcja Broadcast Master Wartość wiodąca 3	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_49_Inverter_lead_val4	Funkcja Broadcast Master Wartość wiodąca 4	100% = 4000h	INT	R	SK 5xxP SK 54xE
_50_Inverter_lead_val5	Funkcja Broadcast Master Wartość wiodąca 5	100% = 4000h	INT	R	SK 5xxP SK 54xE
_51_Inverter_control_w ord	Wynikowe słowo sterujące magistrali	Odpowiada profilowi USS	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
					SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_52_Inverter_set_val1	Wynikowa główna wartość zadana 1 magistrali	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_53_Inverter_set_val2	Wynikowa główna wartość zadana 2 magistrali	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_54_Inverter_set_val3	Wynikowa główna wartość zadana 3 magistrali	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_55_Inverter_set_val4	Wynikowa główna wartość zadana 4 magistrali	100% = 4000h	INT	R	SK 5xxP SK 54xE
_56_Inverter_set_val5	Wynikowa główna wartość zadana 5 magistrali	100% = 4000h	INT	R	SK 5xxP SK 54xE
_57_Broadcast_set_val1	Broadcast Slave: Pomocnicza wartość zadana 1	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_58_Broadcast_set_val2	Broadcast Slave: Pomocnicza wartość zadana 2	100% = 4000h	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_59_Broadcast_set_val	Broadcast Slave:	100% = 4000h	INT	R	SK 5xxP

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
3	Pomocnicza wartość zadana 3				SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_60_Broadcast_set_val4	Broadcast Slave: Pomocnicza wartość zadana 4	100% = 4000h	INT	R	SK 5xxP SK 54xE
_61_Broadcast_set_val5	Broadcast Slave: Pomocnicza wartość zadana 5	100% = 4000h	INT	R	SK 5xxP SK 54xE
_62_Inverter_32Bit_set_val1	32bit wynikowa główna wartość zadana 1 magistrali	- Low Part w P546[1] - High Part w P546[2]	LONG	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_63_Inverter_32Bit_act_val1	FI 32bit wartość rzeczywista 1	- Low Part w P543[1] - High Part w P543[2]	LONG	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_64_Inverter_32Bit_lead_val1	32bit wartość wiodąca 1	- Low Part w P502[1] - High Part w P502[2]	LONG	R	SK 5xxP SK 54xE SK 2xxE SK 180E SK 190E
_65_Broadcast_32Bit_set_val1	32bit Broadcast Slave pomocnicza wartość zadana 1	- Low Part w P543[1] - High Part w P543[2]	LONG	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_66_BusIO_input_bits	Przychodzące dane Bus I/O	- Bit0 – 7 = Bus I/O In Bit 0 – 7 - Bit 8 = Znacznik 1 - Bit 9 = Znacznik 2 - Bit 10 = Bit8 ze słowa sterującego magistrali - Bit 11 = Bit9 ze słowa sterującego magistrali	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_67_BusIO_output_bits	Wychodzące dane Bus I/O	Bit0 = Bus / AS-i Dig Out1 Bit1 = Bus / AS-i Dig	INT	R	SK 5xxP SK 54xE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
		Out2 Bit2 = Bus / AS-i Dig Out3 Bit3 = Bus / AS-i Dig Out4 Bit4 = Bus / 1.IOE Dig Out1 Bit5 = Bus / 1.IOE Dig Out2 Bit6 = Bus / 2.IOE Dig Out1 Bit7 = Bus / 2.IOE Dig Out2 Bit8 = Bit 10 Bus słowo stanu Bit9 = Bit 11 Bus słowo stanu			
_67_BusIO_output_bits	Wychodzące dane Bus I/O	Bit0 = Bus / AS-i Dig Out1 Bit1 = Bus / AS-i Dig Out2 Bit2 = Bus / AS-i Dig Out3 Bit3 = Bus / AS-i Dig Out4 Bit4 = AS-i aktuator 1 Bit5 = AS-i aktuator 2 Bit6 = Znacznik 1 Bit7 = Znacznik 2 Bit8 = Bit 10 Bus słowo stanu Bit9 = Bit 11 Bus słowo stanu	INT	R	SK 53xE SK 52xE
_67_BusIO_output_bits	Wychodzące dane Bus I/O	Bit0 = Bus / AS-i Dig Out1 Bit1 = Bus / AS-i Dig Out2 Bit2 = Bus / AS-i Dig Out3 Bit3 = Bus / AS-i Dig Out4 Bit4 = Bus / IOE Dig Out1 Bit5 = Bus / IOE Dig Out2 Bit6 = Bus / 2nd IOE Dig Out1 Bit7 = Bus / 2nd IOE Dig Out2 Bit8 = Bit 10 Bus słowo stanu Bit9 = Bit 11 Bus słowo stanu	INT	R	SK 2xxE

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_67_BusIO_output_bits	Wychodzące dane Bus I/O	Bit0 = Bus / AS-i Dig Out1 Bit1 = Bus / AS-i Dig Out2 Bit2 = Bus / AS-i Dig Out3 Bit3 = Bus / AS-i Dig Out4 Bit4 = Bus / AS-i Dig Out5 Bit5 = Bus / AS-i Dig Out6 Bit6 = Bus / 2nd IOE Dig Out1 Bit7 = Bus / 2nd IOE Dig Out2 Bit8 = Bit 10 Bus słowo stanu Bit9 = Bit 11 Bus słowo stanu	INT	R	SK 2xxE-FDS

3.5.4 ControlBox i ParameterBox

Za pomocą wymienionych wartości procesowych można uzyskać dostęp do paneli obsługi. Umożliwia to realizację prostych aplikacji HMI.

Informacja

Aby „key_states” były wyświetlane w sterowniku PLC, panele ControlBox i ParameterBox muszą znajdować się w trybie wyświetlania PLC. W przeciwnym wypadku zostanie wyświetlona tylko wartość „0”.

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_70_Set_controlbox_show_val	Wartość wyświetlana dla panelu ControlBox	Wartość wyświetlana = bit 29 – bit 0 Pozycja przecinka = bit 31 – bit 30	DINT	R/W	Wszystkie
_71_Controlbox_key_state	Stan klawiatury panelu ControlBox	Bit 0: ON Bit 1: OFF Bit 2: DIR Bit 3: UP Bit 4: DOWN Bit 5: Enter	BYTE	R	Wszystkie
_72_Parameterbox_key_state	Stan klawiatury panelu ParameterBox	Bit 0: ON Bit 1: OFF Bit 2: DIR Bit 3: UP Bit 4: DOWN Bit 5: Enter Bit 6: Right Bit 7: Left	BYTE	R	Wszystkie

3.5.5 Parametry informacyjne

Są tutaj wymienione najważniejsze wartości rzeczywiste urządzenia.

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_80_Current_fault	Aktualny numer usterki	Błąd 10.0= 100	BYTE	R	Wszystkie
_81_Current_warning	Aktualne ostrzeżenie	Ostrzeżenie 10.0 = 100	BYTE	R	Wszystkie
_82_Current_reason_FI_blocked	Aktualna przyczyna stanu Blokada włączenia	Problem 10.0 = 100	BYTE	R	Wszystkie
_83_Input_voltage	Aktualne napięcie zasilające	100 V = 100	INT	R	Wszystkie
_84_Current_frequenz	Aktualna częstotliwość	10Hz = 100	INT	R	Wszystkie
_85_Current_set_point_frequency1	Aktualna częstotliwość zadana ze źródła wartości zadanych	10Hz = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_86_Current_set_point_frequency2	Aktualna częstotliwość zadana przetwornicy	10Hz = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_87_Current_set_point_frequency3	Aktualna częstotliwość zadana za rampą	10Hz = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_88_Current_Speed	Aktualna obliczona prędkość obrotowa	100 obr/min = 100	INT	R	Wszystkie
_89_Actual_current	Aktualny prąd wyjściowy	10.0A = 100	INT	R	Wszystkie
_90_Actual_torque_current	Aktualny prąd momentu	10.0A = 100	INT	R	Wszystkie
_91_Current_voltage	Aktualne napięcie	100V = 100	INT	R	Wszystkie

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_92_Dc_link_voltage	Aktualne napięcie obwodu pośredniego	100V = 100	INT	R	S SK 5xxP K 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_93_Actual_field_current	Aktualny prąd pola	10.0A = 100	INT	R	Wszystkie
_94_Voltage_d	Aktualna składowa napięcia oś d	100V = 100	INT	R	Wszystkie
_95_Voltage_q	Aktualna składowa napięcia oś q	100V = 100	INT	R	Wszystkie
_96_Current_cos_phi	Aktualny cos(fi)	0.80 = 80	BYTE	R	Wszystkie
_97_Torque	Aktualny moment obrotowy	100% = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_98_Field	Aktualne pole	100% = 100	BYTE	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_99_Apparent_power	Aktualna moc pozorna	1,00kW = 100	INT	R	Wszystkie
_100_Mechanical_power	Aktualna moc mechaniczna	1,00kW = 100	INT	R	Wszystkie
_101_Speed_encoder	Aktualna zmierzona prędkość obrotowa	100 obr/min = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE
_102_Usage_rate_motor	Aktualne obciążenie silnika (wartość chwilowa)	100% = 100	INT	R	Wszystkie

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_103_Usage_rate_motor_I2t	Aktualne obciążenie silnika I2t	100% = 100	INT	R	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_104_Usage_rate_brake_resistor	Aktualne obciążenie rezystora hamowania	100% = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_105_Head_sink_temp	Aktualna temperatura radiatora	100°C = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_106_Inside_temp	Aktualna temperatura wnętrza	100°C = 100	INT	R	SK 5xxP SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_107_Motor_temp	Aktualna temperatura silnika	100°C = 100	INT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_108_Actual_net_frequency	Aktualna częstotliwość sieciowa	10Hz = 100	INT	R	SK 155E-FDS SK 175E-FDS
_109_Mains_phase_sequence	Aktualna kolejność faz sieci	0=CW, 1=CCW	BYTE	R	SK 155E-FDS SK 175E-FDS
_141_Pos_Sensor_Inc	Pozycja enkodera przyrostowego	0,001 obrotu	DINT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_142_Pos_Sensor_Abs	Pozycja enkodera absolutnego	0,001 obrotu	DINT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_143_Pos_Sensor_Uni	Pozycja enkodera uniwersalnego	0,001 obrotu	DINT	R	SK 5xxP SK 54xE
_144_Pos_Sensor_HTL	Pozycja enkodera HTL	0,001 obrotu	DINT	R	SK 5xxP SK 54xE
_145_Actual_pos	Pozycja rzeczywista	0,001 obrotu	DINT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_146_Actual_ref_pos	Aktualna pozycja zadana	0,001 obrotu	DINT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_147_Actual_pos_diff	Różnica pozycji między wartością zadaną i rzeczywistą	0,001 obrotu	DINT	R	SK 5xxP SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E

3.5.6 Błędy PLC

Błędy urządzenia E23.0 do E24.7 można ustawić z programu PLC za pomocą User Error Flags.

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_110_ErrorFlags	Generuje błąd użytkownika w urządzeniu	Bit 0: E 23.0 Bit 1: E 23.1 Bit 2: E 23.2 Bit 3: E 23.3 Bit 4: E 23.4 Bit 5: E 23.5 Bit 6: E 23.6 Bit 7: E 23.7	BYTE	R/W	Wszystkie
_111_ErrorFlags_ext	Generuje błąd użytkownika w urządzeniu	Bit 0: E 24.0 Bit 1: E 24.1 Bit 2: E 24.2 Bit 3: E 24.3 Bit 4: E 24.4 Bit 5: E 24.5 Bit 6: E 24.6 Bit 7: E 24.7	BYTE	R/W	Wszystkie

3.5.7 Parametry PLC

Te grupy danych procesowych można wykorzystać do bezpośredniego dostępu do parametrów PLC P355, P356 i P360.

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
_115_PLC_P355_1	PLC INT parametr P355 [-01]	-	INT	R	Wszystkie
_116_PLC_P355_2	PLC INT parametr P355 [-02]	-	INT	R	Wszystkie
_117_PLC_P355_3	PLC INT parametr P355 [-03]	-	INT	R	Wszystkie
_118_PLC_P355_4	PLC INT parametr P355 [-04]	-	INT	R	Wszystkie
_119_PLC_P355_5	PLC INT parametr P355 [-05]	-	INT	R	Wszystkie
_120_PLC_P355_6	PLC INT parametr P355 [-06]	-	INT	R	Wszystkie
_121_PLC_P355_7	PLC INT parametr P355 [-07]	-	INT	R	Wszystkie
_122_PLC_P355_8	PLC INT parametr P355 [-08]	-	INT	R	Wszystkie
_123_PLC_P355_9	PLC INT parametr P355 [-09]	-	INT	R	Wszystkie
_124_PLC_P355_10	PLC INT parametr P355 [-10]	-	INT	R	Wszystkie
_125_PLC_P356_1	PLC LONG parametr P356 [-01]	-	DINT	R	Wszystkie
_126_PLC_P356_2	PLC LONG parametr P356 [-02]	-	DINT	R	Wszystkie
_127_PLC_P356_3	PLC LONG parametr P356 [-03]	-	DINT	R	Wszystkie
_128_PLC_P356_4	PLC LONG parametr P356 [-04]	-	DINT	R	Wszystkie
_129_PLC_P356_5	PLC LONG parametr P356 [-05]	-	DINT	R	Wszystkie
_130_PLC_P360_1	PLC parametr wyświetlania P360[-01]	-	DINT	R/W	Wszystkie
_131_PLC_P360_2	PLC parametr wyświetlania P360[-02]	-	DINT	R/W	Wszystkie
_132_PLC_P360_3	PLC parametr wyświetlania P360[-03]	-	DINT	R/W	Wszystkie
_133_PLC_P360_4	PLC parametr wyświetlania P360[-04]	-	DINT	R/W	Wszystkie
_134_PLC_P360_5	PLC parametr	-	DINT	R/W	Wszystkie

Nazwa	Funkcja	Normowanie	Typ	Dostęp	Urządzenie
	wyświetlania P360[-05]				
_135_PLC_Scope_Int_1	PLC Scope wartość wyświetlana 1	-	INT	R/W	Wszystkie
_136_PLC_Scope_Int_2	PLC Scope wartość wyświetlana 2	-	INT	R/W	Wszystkie
_137_PLC_Scope_Int_3	PLC Scope wartość wyświetlana 3	-	INT	R/W	Wszystkie
_138_PLC_Scope_Int_4	PLC Scope wartość wyświetlana 4	-	INT	R/W	Wszystkie
_139_PLC_Scope_Bool_1	PLC Scope wartość wyświetlana 5	-	INT	R/W	Wszystkie
_140_PLC_Scope_Bool_2	PLC Scope wartość wyświetlana 6	-	INT	R/W	Wszystkie

3.6 Języki

3.6.1 Lista instrukcji (AWL / IL)

3.6.1.1 Informacje ogólne

Typy danych

PLC obsługuje niżej podane typy danych.

Nazwa	Wymagane miejsce w pamięci	Zakres wartości
BOOL	1 bit	0 do 1
BYTE	1 bajt	0 do 255
INT	2 bajty	-32768 do 32767
DINT	4 bajty	-2.147.483.648 do 2.147.483.647
LABEL_ADDRESS	2 bajty	Znacznik skoku

Literały

W celu większej przejrzystości można wprowadzać stałe wszystkich typów danych w różnych formatach wyświetlania. Poniższa tabela zawiera przegląd wszystkich możliwych wariantów.

Literał	Przykład	Liczba wyświetlana dziesiętnie
Bool	FALSE	0
	TRUE	1
	BOOL#0	0
	BOOL#1	1
Binarny (podstawa 2)	2#01011111	95
	2#0011_0011	51
	BYTE#2#00001111	15
	BYTE#2#0001_1111	31
Ósemkowy (podstawa 8)	8#0571	377
	8#05_71	377
	BYTE#8#10	8
	BYTE#8#111	73
	BYTE#8#1_11	73
Szesnastkowy (podstawa 16)	16#FFFF	-1
	16#0001_FFFF	131071
	INT#16#1000	4096
	DINT#16#0010_2030	1056816
Całkowity (podstawa 10)	10	10
	-10	-10
	10_000	10000
	INT#12	12
	DINT#-100000	-100000
Czas	TIME#10s50ms	10,050 sekund
	T#5s500ms	5,5 sekundy
	TIME#5.2s	5,2 sekundy
	TIME#5D10H15M	5dni+10godzin+15minut
	T#1D2H30M20S	1dzień+2godziny+30minut+20sekund

Komentarze

Aby zapewnić czytelność programu PLC, zaleca się umieścić wyjaśnienia w poszczególnych segmentach programu. Komentarze są oznaczone w programie użytkowym znakami „(” na początku komentarza i „)” na końcu komentarza zgodnie z poniższymi przykładami.

```
(* Komentarz dotyczący bloku programu *)  
LD 100 (* Komentarz za poleceniem *)  
ADD 20
```

Znacznik skoku

Za pomocą operatorów JMP, JMPC i JMPCN można pominąć całe części programu. Adresem docelowym jest znacznik skoku. Może zawierać wszystkie litery, cyfry od 0 do 9 i podkreślenia, z wyjątkiem umlautów i znaku „ß”, inne znaki nie są dopuszczalne. Na końcu znacznika skoku znajduje się dwukropek. Może występować samodzielnie. W tym samym wierszu za znacznikiem skoku może również znajdować się inne polecenie.

Możliwe warianty mogą wyglądać w następujący sposób:

Przykład:

```
Sprungmarke:  
LD 20  
  
Das_Ist_eine_Sprungmarke:  
ADD 10  
  
MainLoop: LD 1000
```

Innym wariantem jest przesłanie znacznika skoku jako zmiennej. Zmienna musi być zdefiniowana w tabeli zmiennych jako typ LABEL_ADDRESS, aby do zmiennej można było załadować znaczniki skoku. Można to wykorzystać do łatwego tworzenia maszyn stanów, patrz niżej.

Przykład:

```
LD FirstTime  
JMPC AfterFirstTime  
(* Na początku należy zainicjować adres etykiety. *)  
LD Address_1  
ST Address_Var  
LD TRUE  
ST FirstTime  
AfterFirstTime:  
JMP Address_Var  
Address_1:  
LD Address_2  
ST Address_Var  
JMP Ende  
Address_2:  
LD Address_3  
ST Address_Var  
JMP Ende  
Address_3:  
LD Address_1  
ST Address_Var  
Ende:
```

Wywołania funkcji

Edytor obsługuje jeden sposób wywołań funkcji. W poniższych wariantach funkcja CTD jest wywoływana przez instancję I_CTD. Wyniki są zapisywane w zmiennych. Znaczenie używanych funkcji objaśniono w dalszej części instrukcji.

Przykład:

```
LD 10000
ST I_CTD.PV
LD LoadNewVar
ST I_CTD.LD
LD TRUE
ST I_CTD.CD
CAL I_CTD
LD I_CTD.Q
ST ResultVar
LD I_CTD.CV
ST CurrentCountVar
```

Dostęp do bitów zmiennych

Aby uzyskać dostęp do jednego bitu zmiennej lub zmiennej procesowej, możliwa jest uproszczona notacja.

Polecenie	Znaczenie
LD Var1.0	Ładuje bit 0 zmiennej Var1 do AE
ST Var1.7	Zapisuje AE w bicie 7 zmiennej Var1
EQ Var1.4	Porównuje AE z bitem 4 zmiennej Var1

3.6.2 Tekst strukturalny (ST)

Tekst strukturalny składa się z szeregu instrukcji, które można wykonywać warunkowo (IF..THEN..ELSE) lub w pętlach (WHILE..DO), jak w językach wysokiego poziomu.

Przykład:

```
IF value < 7 THEN
  WHILE value < 8 DO
    value := value + 1;
  END_WHILE;
END_IF;
```

3.6.2.1 Informacje ogólne

Typy danych w ST

PLC obsługuje niżej podane typy danych.

Nazwa	Wymagane miejsce w pamięci	Zakres wartości
BOOL	1 bit	0 do 1
BYTE	1 bajt	0 do 255
INT	2 bajty	-32768 do 32767
DINT	4 bajty	-2 147 483 648 do 2 147 483 647

Informacja

W przypadku liczb celowe jest podanie typu danych, aby utworzyć efektywny program PLC, np.:
VarInt := INT#-32768, VarDINT := DINT#-2147483648.

Operator przypisania

Po lewej stronie przypisania znajduje się operand (zmienna, adres), do którego zostanie przypisana wartość wyrażenia po prawej stronie za pomocą operatora przypisania „:=”.

Przykład:

```
Var1 := Var2 * 10;
```

Po wykonaniu operacji w Var1 znajdzie się dziesięciokrotność Var2.

Wywołanie bloków funkcyjnych w ST

Wywołanie bloku funkcyjnego w ST odbywa się przez zapisanie nazwy instancji bloku funkcyjnego, a następnie przypisanie żądanych wartości parametrów w nawiasach. W poniższym przykładzie odbywa się wywołanie timera z przypisaniem dla jego parametrów IN i PT. Następnie zmienna wynikowa Q zostanie przypisana do zmiennej A.

Dostęp do zmiennej wynikowej jest możliwy, podobnie jak w AWL, za pomocą nazwy bloku funkcyjnego, punktu i nazwy zmiennej.

Przykład:

```
Timer(IN := TRUE, PT := 300);
A := Timer.Q;
```

Analiza wyrażeń

Analiza wyrażenia odbywa się przez wykonywanie operatorów zgodnie z określonymi regułami wiązania. Najpierw wykonywany jest operator o największej sile wiązania, następnie operator o następnej w kolejności sile wiązania itd., aż zostaną wykonane wszystkie operatory. Operatory o takiej samej sile wiązania są wykonywane od lewej do prawej.

Poniżej znajduje się tabela operatorów ST w kolejności ich siły wiązania:

Operacja	Symbol	Siła wiązania
Nawiasy	(Wyrażenie)	Najsilniejsze wiązanie
Wywołanie funkcji	Nazwa funkcji (lista parametrów)	
Negacja dopełnienia	NOT	
Mnożenie Dzielenie Modulo AND	* / MOD AND	
Dodawanie Odejmowanie OR XOR	+ - OR XOR	
Porównanie Równość Nierówność	<,>,<=,>= = <>	Najslabsze wiązanie

3.6.2.2 Instrukcje

Return

Za pomocą instrukcji RETURN można przeskoczyć na koniec programu, np. w zależności od warunku.

IF

Za pomocą instrukcji IF można sprawdzić warunek i wykonać instrukcje w zależności od tego warunku.

Składnia:

```
IF <Boolscher_Ausdruck1> THEN
  <IF_Anweisungen>
ELSIF <Boolscher_Ausdruck2> THEN
  <ELSIF_Anweisungen1>
ELSIF <Boolscher_Ausdruck n> THEN
  <ELSIF_Anweisungen n-1>
ELSE
  <ELSE_Anweisungen>}
END_IF;
```

Część w nawiasach klamrowych {} jest opcjonalna.

Jeżeli <Wyrażenie_boolowskie1> przyjmie wartość TRUE, są wykonywane tylko <Instrukcje_IF> i żadne inne instrukcje. W przeciwnym wypadku wyrażenia boolowskie są analizowane kolejno, rozpoczynając od <Wyrażenia_boolowskiego2>, dopóki jedno z wyrażen nie przyjmie wartości TRUE. Następnie są analizowane tylko instrukcje następujące po wyrażeniu boolowskim i przed następnym ELSE lub ELSIF. Gdy żadne z wyrażen boolowskich nie przyjmie wartości TRUE, są analizowane wyłącznie <Instrukcje_ELSE>.

Przykład:

```
IF temp < 17 THEN
  Bool1 := TRUE;
ELSE
  Bool2 := FALSE;
END_IF;
```

CASE

Za pomocą instrukcji CASE można połączyć kilka instrukcji warunkowych o tej samej zmiennej warunkowej w jeden konstrukt.

Składnia:

```
CASE <Var1> OF
  <Wert 1>: <Anweisung 1>
  <Wert 2>: <Anweisung 2>
  <Wert3, Wert4, Wert5: <Anweisung 3>
  <Wert6 .. Wert10 : <Anweisung 4>
  ...
  <Wert n>: <Anweisung n>
ELSE <ELSE-Anweisung>
END_CASE;
```

Instrukcja CASE jest wykonywana zgodnie z następującym schematem:

- Gdy zmienna w <Var1> ma wartość <Wartość i>, jest wykonywana instrukcja <Instrukcja i>.
- Jeżeli <Var 1> nie ma żadnej z podanych wartości, jest wykonywana <Instrukcja ELSE>.
- Jeżeli ta sama instrukcja ma być wykonana dla kilku wartości zmiennych, można zapisać kolejno te wartości oddzielone przecinkami jako warunek wspólnej instrukcji.
- Jeżeli ta sama instrukcja ma być wykonana dla jednego zakresu wartości zmiennych, można zapisać kolejno wartość początkową i końcową oddzielone dwukropkiem jako warunek wspólnej instrukcji.

Przykład:

```
CASE INT1 OF
  1, 5:
    BOOL1 := TRUE;
    BOOL3 := FALSE;
  2:
    BOOL2 := FALSE;
    BOOL3 := TRUE;
  10..20:
    BOOL1 := TRUE;
    BOOL3:= TRUE;
  ELSE
    BOOL1 := NOT BOOL1;
    BOOL2 := BOOL1 OR BOOL2;
END_CASE;
```

Pętla FOR

Za pomocą pętli FOR można zaprogramować powtarzające się operacje.

Składnia:

```
FOR <INT_Var> := <INIT_WERT> TO <END_WERT> {BY <Schrittgröße>} DO  
  <Anweisungen>  
END_FOR;
```

Część w nawiasach klamrowych {} jest opcjonalna. <Instrukcje> są wykonywane dopóty, dopóki licznik <INT_Var> nie będzie większy od <WARTOŚCI_END>. Warunek ten zostanie sprawdzony przed wykonaniem <Instrukcji>, aby <Instrukcje> nigdy nie były wykonywane, gdy <WARTOŚĆ_INIT> jest większa od <WARTOŚCI_END>. Gdy <Instrukcje> zostały wykonane, <INT_Var> zostanie zwiększone o <Wielkość kroku>. Wielkość kroku może być dowolną wartością całkowitą. Jeżeli nie jest zadana, zostanie ustawiona wartość 1. Pętla musi się zakończyć, ponieważ <INT_Var> będzie się tylko zwiększać.

Przykład:

```
FOR Zaehler :=1 TO 5 BY 1 DO  
  Var1 := Var1 * 2;  
END_FOR;
```

Pętla REPEAT

Pętla REPEAT różni się od pętli WHILE tym, że warunek przerwania jest sprawdzany dopiero po wykonaniu pętli. W rezultacie pętla zostanie wykonana co najmniej raz, niezależnie od warunków przerwania.

Składnia:

```
REPEAT  
  <Anweisungen>  
UNTIL <Boolescher Ausdruck>  
END_REPEAT;
```

<Instrukcje> są wykonywane do momentu, aż <Wyrażenie boolowskie> przyjmie wartość TRUE. Gdy <Wyrażenie boolowskie> przyjmie wartość TRUE już podczas pierwszej analizy, <Instrukcje> zostaną wykonane dokładnie jeden raz. Gdy <Wyrażenie boolowskie> nigdy nie przyjmie wartości TRUE, <Instrukcje> będą powtarzane bez końca, co spowoduje błąd czasu wykonywania.

Informacja

Programista musi sam zadbać, aby nie powstała nieskończona pętla, zmieniając warunek w instrukcji pętli, np. ustawiając licznik odliczający do góry lub w dół.

Przykład:

```
REPEAT  
  Var1 := Var1 * 2;  
  Zaehler := Zaehler - 1;  
UNTIL  
  Zaehler = 0  
END_REPEAT
```

Pętla WHILE

Pętli WHILE można używać podobnie jak pętli FOR, z tą różnicą, że warunkiem przerwania może być dowolne wyrażenie boolowskie. Oznacza to, że jest określony warunek, który powoduje wykonanie pętli, o ile jest spełniony.

Składnia:

```
WHILE <Boolescher Ausdruck> DO  
  <Anweisungen>  
END_WHILE;
```

<Instrukcje> są wykonywane do momentu, aż <Wyrażenie boolowskie> przyjmie wartość FALSE. Gdy <Wyrażenie boolowskie> przyjmie wartość FALSE już podczas pierwszego wykonania, <Instrukcje> zostaną wykonane dokładnie jeden raz. Gdy <Wyrażenie boolowskie> nigdy nie przyjmie wartości FALSE, <Instrukcje> będą powtarzane bez końca, co spowoduje błąd czasu wykonywania.

Informacja

Programista musi sam zadbać, aby nie powstała nieskończona pętla, zmieniając warunek w instrukcji pętli, np. ustawiając licznik odliczający do góry lub w dół.

Przykład:

```
WHILE Zaehler >0 DO  
  Var1 := Var1 * 2;  
  Zaehler := Zaehler - 1;  
END_WHILE
```

Exit

Jeżeli instrukcja EXIT występuje w pętli FOR, WHILE lub REPEAT, zostanie zakończona najbardziej wewnętrzna pętla, niezależnie od warunku przerwania.

3.7 Skoki

3.7.1 JMP

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Bezwarunkowy skok do znacznika skoku.

Przykład w AWL:

```
JMP NextStep (* Bezwarunkowy skok do NextStep *)
ADD 1

NextStep:
ST Value1
```

3.7.2 JMPC

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Warunkowy skok (Jump Conditional) do znacznika skoku. Gdy AE = TRUE, instrukcja JMPC przeskakuje do podanego znacznika skoku.

Przykład w AWL:

```
LD 10
JMPC NextStep (* AE = TRUE à program wykonuje skok *)
ADD 1

NextStep:
ST Value1
```

3.7.3 JMPCN

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

Warunkowy skok (Jump Conditional) do znacznika skoku. JMPCN wykonuje skok, gdy rejestr AE = FALSE. W przeciwnym wypadku program wykona następną instrukcję.

Przykład w AWL:

```
LD 10
JMPCN NextStep (* AE = TRUE à program nie wykonuje skoku *)
ADD 1

NextStep:
ST Value1
```

3.8 Konwersja typów

3.8.1 BOOL_TO_BYTE

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych	X						

Dokonuje konwersji typu danych AE z BOOL na BYTE. Jeżeli AE jest równe FALSE, Akku zostanie przekonwertowane na 0. Jeżeli AE jest równe TRUE, Akku zostanie przekonwertowane na 1.

Przykład w AWL:

```
LD TRUE
BOOL_TO_BYTE (* AE = 1 *)
```

Przykład w ST:

```
Ergebnis := BOOL_TO_BYTE(TRUE); (* Wynik = 1 *)
```

3.8.2 BYTE_TO_BOOL

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X					

Dokonuje konwersji typu danych z BYTE na BOOL. Dopóki BYTE nie jest równy zero, wynikiem konwersji jest zawsze TRUE.

Przykład w AWL:

```
LD 10
BYTE_TO_BOOL (* AE = TRUE *)
```

Przykład w ST:

```
Ergebnis := BYTE_TO_BOOL(10); (* Wynik = TRUE *)
```

3.8.3 BYTE_TO_INT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych		X					

Dokonuje konwersji typu danych z BYTE na INT. BYTE zostanie skopiowany do części Low wartości INT, a część High wartości INT zostanie ustawiona na wartość 0.

Przykład w AWL:

```
LD 10
BYTE_TO_INT (* Akku = 10 *)
```

Przykład w ST:

```
Ergebnis := BYTE_TO_INT(10); (* Wynik = 10 *)
```

3.8.4 DINT_TO_INT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT			
Typ danych				X			

Dokonuje konwersji typu danych z DINT na INT. Część High wartości DINT nie zostanie przejęta.

Przykład w AWL:

```
LD 200000
DINT_TO_INT (* Akku = 3392 *)

LD DINT# -5000
DINT_TO_INT (* Akku = -5000 *)

LD DINT# -50010
DINT_TO_INT (* Akku = 15526 *)
```

Przykład w ST:

```
Ergebnis := DINT_TO_INT(200000); (* Wynik = 3392 *)
Ergebnis := DINT_TO_INT(-5000); (* Wynik = -5000 *)
Ergebnis := DINT_TO_INT(-50010); (* Wynik = 15526 *)
```

3.8.5 INT_TO_BYTE

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Typ danych			X	

Dokonuje konwersji typu danych z INT na BYTE. Część High wartości INT nie zostanie przejęta. Znaki liczby zostaną utracone, ponieważ typ BYTE nie posiada znaku.

Przykład w AWL:

```
LD 16#5008
INT_TO_BYTE (* Akku = 8 *)
```

Przykład w ST:

```
Ergebnis := INT_TO_BYTE(16#5008); (* Wynik = 8 *)
```

3.8.6 INT_TO_DINT

	SK 5xxP	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Dostępność	X	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Typ danych			X	

Dokonuje konwersji typu danych z INT na DINT. INT zostanie skopiowany do części Low wartości DINT, a część High wartości DINT zostanie ustawiona na wartość 0.

Przykład w AWL:

```
LD 10
INT_TO_DINT (* Akku = 10 *)
```

Przykład w ST:

```
Ergebnis := INT_TO_DINT(10); (* Wynik = 10 *)
```


3.9 Komunikaty o usterkach PLC

Usterki powodują wyłączenie urządzenia, aby zapobiec jego uszkodzeniu. W przypadku pojawienia się komunikatów o usterkach PLC praca PLC zostanie zatrzymana, a PLC przechodzi w stan „PLC-Error”. W przypadku innych komunikatów o usterkach PLC nadal działa. Po potwierdzeniu błędu PLC uruchamia się automatycznie.

W przypadku pojawienia się błędu PLC User Fault 23.X PLC nadal działa!

Wyświetlony kod na panelu SimpleBox		Usterka Opis tekstowy na panelu ParameterBox	Przyczyna Środek zaradczy
Grupa	Szczegóły w P700[-01] / P701		
E022	22.0	Brak programu PLC	PLC został uruchomiony, ale FU nie zawiera programu PLC. - Załadować program PLC do urządzenia
	22.1	Wadliwy program PLC	Kontrola sumy kontrolnej za pomocą programu PLC spowodowała błąd. - Ponownie uruchomić urządzenie (Power ON) i spróbować ponownie - Alternatywnie ponownie załadować program
	22.2	Nieprawidłowy adres skoku	Błąd programu, postępować jak w przypadku błędu 22.1
	22.3	Przepełnienie stosu	Podczas wykonywania programu zostało otwartych więcej niż 6 poziomów nawiasów. - Sprawdzić program pod kątem błędów czasu wykonywania
	22.4	Przekroczono maks. czas cyklu PLC	Podany maks. czas cyklu programu PLC został przekroczony. - Dopasować czas cyklu lub sprawdzić program
	22.5	Nieznany kod polecenia	Nie można wykonać kodu polecenia występującego w programie, ponieważ jest nieznany. - Błąd programu, postępować jak w przypadku błędu 22.1 - Wersja PLC i wersja NORD CON nie są ze sobą zgodne
	22.6	Dostęp zapisu PLC	Podczas pracy programu PLC została zmieniona treść programu.
	22.9	Błąd zbiorczy PLC	Nie można dokładnie określić przyczyny błędu. - Postępować jak w przypadku błędu 22.1
E023	23.0	PLC User Fault 1	Błąd może być wywołany przez program PLC, aby zasygnalizować na zewnątrz problemy w wykonaniu programu PLC. Wywołanie odbywa się przez opis zmiennej procesowej „ErrorFlags”.
	23.1	PLC User Fault 2	
	23.2	PLC User Fault 3	

4 Parametry

Parametry urządzenia istotne dla funkcjonalności PLC są szczegółowo opisane w instrukcji odpowiedniej przetwornicy częstotliwości lub startera silnika.

5 Załącznik

5.1 Wskazówki serwisowe i dotyczące uruchamiania

W przypadku problemów, np. podczas uruchamiania, należy skontaktować się z naszym serwisem:

☎ +49 4532 289-2125

Nasz serwis jest dostępny przez całą dobę (24 h / 7 dni) i może udzielić najlepszej pomocy, gdy użytkownik przygotuje następujące informacje o urządzeniu i jego akcesoriach:

- Oznaczenie typu
- Numer seryjny
- Wersja oprogramowania sprzętowego

5.2 Dokumenty i oprogramowanie

Dokumenty i oprogramowanie można pobrać z naszej strony internetowej www.nord.com.

Dodatkowo obowiązujące i inne dokumenty

Dokumentacja	Zawartość
BU_0155	Instrukcja startera silnika, dystrybutora polowego NORDAC <i>LINK SK 180E / SK 190E</i>
BU_0180	Instrukcja przetwornicy częstotliwości NORDAC <i>BASE SK 180E / SK 190E</i>
BU_0200	Instrukcja przetwornicy częstotliwości NORDAC <i>FLEX SK 200E .. SK 235E</i>
BU_0250	Instrukcja przetwornicy częstotliwości, dystrybutora polowego NORDAC <i>LINK SK 250E-FDS .. SK 280E-FDS</i>
BU_0500	Instrukcja przetwornicy częstotliwości NORDAC <i>PRO SK 500E .. SK 535E</i>
BU_0505	Instrukcja przetwornicy częstotliwości NORDAC <i>PRO SK 540E .. SK 545E</i>
BU_0600	Instrukcja przetwornicy częstotliwości NORDAC <i>PRO SK 500P .. SK 550P</i>
BU_0000	Instrukcja obsługi programu NORDCON
BU_0040	Instrukcja obsługi paneli ParameterBox firmy NORD

Oprogramowanie

Oprogramowanie	Opis
NORDCON	Oprogramowanie do parametryzacji i diagnostyki

5.3 Skróty

- **AE** Aktualny wynik
- **AIN** Wejście analogowe
- **AOUT** Wyjście analogowe
- **AWL** Lista instrukcji (także IL)
- **COB-ID** Communication Object Identifier
- **DI / DIN** Wejście cyfrowe
- **DO / DOUT** Wyjście cyfrowe
- **E/A lub I/O** Wejście/wyjście
- **EEPROM** Pamięć trwała
- **EMC** Kompatybilność elektromagnetyczna
- **FB** Blok funkcyjny
- **FU** Przetwornica częstotliwości
- **HSW** Główna wartość zadana
- **IL** Lista instrukcji (patrz AWL)
- **ISD** Prąd polowy (sterowanie wektorem prądu)
- **LED** Dioda świecąca
- **MC** Motion Control
- **NSW** Pomocnicza wartość zadana
- **P** Parametr zależny od zestawu parametrów, tzn. parametr, któremu w każdym z 4 zestawów parametrów urządzenia można przypisać różne funkcje lub wartości.
- **P-BOX** ParameterBox
- **PDO** Obiekt danych procesu
- **PLC** PLC (sterownik programowalny)
- **S** Parametr systemowy, tzn. parametr, który jest dostępny tylko wtedy, gdy w parametrze **P003** jest wprowadzony prawidłowy kod systemowy.
- **SW** Wersja oprogramowania (patrz parametr **P707**)
- **STW** Słowo sterujące
- **ZSW** Słowo stanu

Spis haseł

D	Exit.....	148
Dokumenty	EXP	98
Dodatkowo obowiązujące	F_TRIG.....	65
O	FB_FunctionCurve	85
Oprogramowanie	FB_PIDT1.....	86
P	FB_ResetPostion	88
PLC.....	FB_Capture	81
ABS	FB_DinCounter.....	83
ACOS	FB_DINTToPBOX	76
ADD.....	FB_FlyingSaw	36
ADD(.....	FB_Gearing	38
Analiza wyrażeń	FB_NMT	27
AND.....	FB_PDConfig.....	28
AND(.....	FB_PDORceive	31
ANDN	FB_PDOSend.....	33
ANDN(.....	FB_ReadTrace	71
ASIN.....	FB_STRINGToPBOX.....	79
ATAN.....	FB_Weigh.....	89
Błędy	FB_WriteTrace	73
Bloki funkcyjne	GE	112
BOOL_TO_BYTE.....	GT.....	113
BYTE_TO_BOOL.....	IF	145
BYTE_TO_INT.....	INT_TO_BYTE	152
CASE	INT_TO_DINT	152
ControlBox	Języki	139
ControlBox i ParameterBox	JMP	149
COS	JMPC.....	149
CTD.....	JMPCN	149
CTU.....	Komentarze	141
CTUD	Komunikacja CANopen	16
Debugowanie	Komunikaty o usterkach.....	153
DINT_TO_INT	Konfiguracja	25
DIV	Konwersja typów	150
DIV(.....	Ładowanie, zapisywanie i drukowanie	17
Dostęp do bitów zmiennych	LD.....	110
Edytor.....	LDN	110
EQ.....	LE	113
	LIMIT	93

Lista instrukcji (AWL / IL)	139	Operatory porównywania	112
Literaly.....	139	OR	103
LN.....	98	OR(.....	103
LOG.....	99	ORN.....	104
LT	114	ORN(.....	104
MAX	94	Pamięć	13
MC_MoveAbsolute.....	47	ParameterBox	15
MC_WriteParameter_16	60	Parametry.....	137
MC_WriteParameter_32	60	Parametry informacyjne	132
MC_Control.....	41	Pętla FOR.....	147
MC_Control_MS	43	Pętla REPEAT	147
MC_Home	44	Pętla WHILE.....	148
SK5xxP	45	Pojedynczy krok	24
MC_MoveAdditive	49	Przegląd wizualizacji	75
MC_MoveRelative.....	50	Przesłanie programu do urządzenia	22
MC_MoveVelocity	51	Przetwarzanie danych przez akumulator ..	14
MC_Power	53	Przetwarzanie wartości zadanej.....	14
MC_ReadActualPos.....	55	Punkty obserwacyjne	23
MC_ReadParameter	56	Punkty przerwania.....	23
MC_ReadStatus.....	57	R	106
MC_Reset	58	R_TRIG	65
MC_Stop	59	Reduktor elektroniczny z funkcją latającej piły	15, 35
MIN.....	94	Regulator procesu	16
MOD.....	95	Return.....	145
MOD(.....	95	ROL	105
Motion Control Lib	15	ROR.....	105
MUL.....	95	RS Flip Flop.....	66
MUL(.....	95	S	106
MUX	96	SHL.....	106
NE	114	SHR.....	107
NOT.....	102	SIN.....	97
Obraz procesu	13	Skoki.....	149
Okno komunikatów	21	Specyfikacja	12
Okno wprowadzania	20	SQRT	99
Okno wyświetlania punktu kontrolnego i punktu przerwania	21	SR Flip Flop.....	67
Operator przypisania.....	143	ST	111
Operatory	91	Standardowe bloki funkcyjne	61
Operatory arytmetyczne.....	91	STN	111
Operatory bitowe.....	100	SUB	96
Operatory ładowania i zapisu	110	SUB(.....	96

TAN.....	97	XOR.....	108
Tekst strukturalny (ST).....	143	XOR(.....	108
TOF.....	68	XORN.....	109
TON.....	69	XORN(.....	109
TP.....	70	Zaawansowane operatory matematyczne	97
Typy danych.....	139	Zadanie programowe.....	14
Typy danych w ST.....	143	Zakres funkcji.....	15
Wartości procesowe.....	115	Zmienne i deklaracja FB.....	19
Wartości zadane i rzeczywiste.....	123	Znacznik skoku.....	141
Wartości zadane i rzeczywiste magistrali	126	W	
Wejścia i wyjścia.....	115	Wykwalifikowany elektryk.....	10
Wizualizacja.....	15	Wykwalifikowany personel.....	10
Wizualizacja panelu ParameterBox.....	75	Z	
Wywołania funkcji.....	142	Zasady bezpieczeństwa.....	10
Wywołanie bloków funkcyjnych w ST.....	144	Zastosowanie zgodne z przeznaczeniem.....	9

NORD DRIVESYSTEMS Group

Headquarters and Technology Centre
in Bargteheide, close to Hamburg

Innovative drive solutions
for more than 100 branches of industry

Mechanical products
parallel shaft, helical gear, bevel gear and worm gear units

Electrical products
IE2/IE3/IE4 motors

Electronic products
centralised and decentralised frequency inverters,
motor starters and field distribution systems

7 state-of-the-art production plants
for all drive components

Subsidiaries and sales partners
in 98 countries on 5 continents
provide local stocks, assembly, production,
technical support and customer service

More than 4,000 employees throughout the world
create customer oriented solutions

www.nord.com/locator

Headquarters:

Getriebebau NORD GmbH & Co. KG
Getriebebau-Nord-Straße 1
22941 Bargteheide, Germany
T: +49 (0) 4532 / 289-0
F: +49 (0) 4532 / 289-22 53
info@nord.com, www.nord.com

Member of the NORD DRIVESYSTEMS Group

