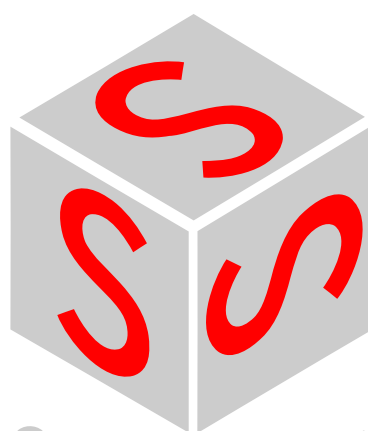


Podręcznik użytkownika Programowanie sterowników PLC w systemie

CoDeSys 2.3



S m a r t
Software
Solutions

Copyright © 1994, 1997, 1999, 2001, 2002, 2003, 2005, 2006 by 3S – Smart Software Solutions GmbH
Wszelkie prawa zastrzeżone.

Informujemy, że podjęto wszelkie środki, by zapewnić poprawność i kompletność niniejszej dokumentacji. Jednak mimo największej staranności w tekście mogą występować pomyłki, których nie da się całkowicie wyeliminować. Wydawca podręcznika z góry dziękuje za wszystkie Państwa uwagi i spostrzeżenia w tym zakresie.

Znaki towarowe

Intel jest zarejestrowanym znakiem towarowym, a 80286, 80386, 80486, Pentium są znakami towarowymi Intel Corporation.

Microsoft, MS i MS-DOS są zarejestrowanymi znakami towarowymi, Windows oraz Intellisense są znakami towarowymi Microsoft Corporation.

Wydawca:

3S – Smart Software Solutions GmbH
Memminger Straße 151
D-87439 Kempten
Tel. +49/ 831/ 5 40 31 - 0
Faks +49/ 831/ 5 40 31 – 50

Stan na dzień: 29.09.2006

Wersja dokumentacji 3.8, dla CoDeSys V2.3.7.0 i wersji późniejszych

Spis treści

CoDeSys 2.3	i
<hr/>	
1 Krótka informacja o CoDeSys	1-1
1.1 Co to jest CoDeSys	1-1
1.2 Przegląd funkcji systemu CoDeSys	1-1
1.3 Przegląd dokumentacji użytkownika systemu CoDeSys	1-3
2 Co jest czym w CoDeSys	2-1
2.1 Składniki projektu... ..	2-1
2.2 Języki... ..	2-9
2.2.1 Lista instrukcji (IL)... ..	2-9
2.2.2 Tekst zhierarchizowany (ST)... ..	2-11
2.2.3 Język przebiegu (SFC)... ..	2-17
2.2.4 Schemat funkcji (FBD)... ..	2-22
2.2.5 Graficzny edytor schematów funkcji (CFC)... ..	2-22
2.2.6 Schemat styków (LD)... ..	2-23
2.3 Debugowanie, funkcjonalności online... ..	2-24
2.4 Norma... ..	2-27
3	3-28
<hr/>	
4 Tworzenie niewielkiego programu	4-1
4.1 Sterownik sygnalizacji świetlnej... ..	4-1
4.2 Wizualizacja sygnalizacji świetlnej... ..	4-11
5	5-15
<hr/>	
6 Szczegółowy opis składników	6-1
6.1 Okno główne... ..	6-1
6.2 Opcje projektu... ..	6-3
6.3 Zarządzanie projektami... ..	6-22
6.4 Zarządzanie obiektami... ..	6-57
6.5 Ogólne funkcje edycji... ..	6-65
6.6 Ogólne funkcje online... ..	6-71
6.7 Okno... ..	6-89
6.8 Pomoc... ..	6-90
7	7-92
<hr/>	
8 Edytory	8-1
8.1 Poniższe uwagi dotyczą wszystkich edytorów	8-1
8.2 Edytor deklaracji... ..	8-3
8.2.1 Praca z użyciem edytora deklaracji	8-3
8.2.2 Edytory deklaracji w trybie online... ..	8-11
8.2.3 Dyrektywy pragma w edytorze deklaracji... ..	8-12
8.3 Edytory tekstowych języków programowania	8-20
8.3.1 Praca w edytorach tekstowych	8-20
8.3.2 Edytor list instrukcji	8-23

8.3.3	Edytor tekstu strukturalnego.....	8-24
8.4	Edytory graficznych języków programowania.....	8-25
8.4.1	Praca w edytorach graficznych	8-25
8.4.2	Edytor schematów funkcji.....	8-29
8.4.3	Edytor schematów styków.....	8-35
8.4.4	Edytor języka przebiegu.....	8-41
8.4.5	Graficzny edytor schematów funkcji (CFC).....	8-49
9	Zasoby	9-1
9.1	Przegląd zasobów	9-1
9.2	Zmienne globalne, konfiguracja zmiennych, szablon dokumentu	9-2
9.2.1	Zmienne globalne.....	9-3
9.2.2	Konfiguracja zmiennych.....	9-7
9.2.3	Szablon dokumentu	9-8
9.3	Konfiguracja alarmów	9-9
9.3.1	Przegląd	9-9
9.3.2	System alarmów, definicje	9-10
9.3.3	Klasy alarmów	9-11
9.3.4	Grupa alarmów.....	9-14
9.3.5	Zapisywanie alarmów.....	9-15
9.3.6	Menu Dodatki: Ustawienia	9-16
9.4	Zarządzanie biblioteką.....	9-17
9.5	Dziennik.....	9-19
9.6	Konfiguracja sterownika	9-21
9.6.1	Przegląd	9-21
9.6.2	Używanie konfiguratora sterownika w CoDeSys.....	9-23
9.6.3	Ustawienia podstawowe w konfiguracji sterownika.....	9-25
9.6.4	Okno dialogowe parametrów specyficzne dla aplikacji	9-26
9.6.5	Konfiguracja modułu wejścia/wyjścia.....	9-26
9.6.6	Konfiguracja kanału.....	9-29
9.6.7	Konfiguracja modułów Profibus.....	9-30
9.6.8	Konfiguracja modułów CANopen.....	9-38
9.6.9	Konfiguracja urządzenia CanDevice (CANopen Slave)	9-43
9.6.10	Konfiguracja modułów DeviceNet	9-47
9.6.11	Konfiguracja sterownika w trybie online	9-52
9.6.12	Skan/status/diagnoza urządzenia z systemu docelowego	9-52
9.7	Konfiguracja zadań.....	9-53
9.7.1	Obsługa konfiguratora zadań	9-54
9.7.2	Zdarzenia systemowe	9-57
9.7.3	Konfiguracja zadań w trybie online	9-58
9.8	Menedżer podglądu zmiennych i receptur.....	9-60
9.8.1	Przegląd	9-60
9.8.2	Menedżer podglądu zmiennych i receptur w trybie offline	9-60
9.8.3	Menedżer podglądu zmiennych i receptur w trybie online	9-62
9.9	Zapis śledzenia.....	9-63
9.9.1	Przegląd i konfiguracja	9-63
9.9.2	Uruchamianie zapisu śledzenia	9-65
9.9.3	Czytanie zapisu śledzenia.....	9-66
9.9.4	„Dodatki” „Zapisz wartości śledzenia”	9-68
9.9.5	„Dodatki” „Zewnętrzne konfiguracje śledzenia”	9-69
9.10	Obszar roboczy.....	9-70
9.11	Menedżer parametrów.....	9-70
9.11.1	Włączanie menedżera parametrów.....	9-71
9.11.2	Edytor menedżera parametrów, przegląd	9-71

9.11.3	Listy parametrów: Typy i atrybuty	9-72
9.11.4	Zarządzanie listami parametrów	9-74
9.11.5	Edycja listy parametrów	9-76
9.11.6	Menedżer parametrów w trybie online	9-77
9.11.7	Eksport / import list parametrów	9-78
9.12	Ustawienia systemu docelowego	9-78
9.13	Przeglądarka PLC.....	9-80
9.13.1	Uwagi ogólne o obsłudze przeglądarki PLC	9-80
9.13.2	Wpisywanie poleceń w przeglądarce PLC	9-80
9.13.3	Stosowanie makr przy wpisywaniu poleceń w przeglądarce PLC	9-82
9.13.4	Dalsze opcje przeglądarki PLC	9-83
9.14	Narzędzia	9-83
9.14.1	Właściwości istniejących skrótów (Właściwości obiektu)	9-84
9.14.2	Zarządzanie skrótami.....	9-86
9.14.3	Najważniejsze pytania dotyczące narzędzi.....	9-88
10 Zarządzanie wersjami ENI		10-1
10.1.1	Co to jest ENI.....	10-1
10.1.2	Założenia pracy z bazą danych projektów ENI	10-1
10.1.3	Praca w CoDeSys za pomocą bazy danych projektu	10-2
10.1.4	Kategorie w zakresie bazy danych projektu.....	10-2
11 Komunikacja DDE		11-1
11.1	Interfejs DDE systemu programowania CoDeSys.....	11-1
11.2	Komunikacja DDE za pomocą serwera GatewayDDE.....	11-2
12 Zarządzanie licencjami w CoDeSys		12-1
12.1	3S Licensing Manager.....	12-1
12.1.1	Tworzenie biblioteki wymagającej licencji.....	12-1
13 ZAŁĄCZNIK		13-1
Załącznik A Operatory IEC i dodatkowe funkcje poszerzające normę		13-1
13.1	Operatory arytmetyczne.....	13-1
13.2	Operatory bitstring.....	13-4
13.3	Operatory bit-shift.....	13-6
13.4	Operatory wyboru.....	13-8
13.5	Operatory porównawcze.....	13-11
13.6	Operatory adresu.....	13-13
13.7	Operator wywołania.....	13-14
13.8	Konwersje typów.....	13-14
13.9	Operatory numeryczne.....	13-20
13.10	Operator inicjalizacji	13-24
Załącznik B Argumenty w CoDeSys		13-27
13.11	Stałe.....	13-27
13.12	Zmienne.....	13-29
13.13	Adresy.....	13-31
13.14	Funkcje.....	13-32
Załącznik C		13-34

Załącznik D	Typy danych w CoDeSys	13-35
13.15	Standardowe typy danych	13-35
13.16	Zdefiniowane typy danych	13-37
Załącznik E	Biblioteki CoDeSys	13-45
13.17	Biblioteka Standard.lib	13-45
13.17.1	Funkcje String...	13-45
13.17.2	Bistabilne bloki funkcji...	13-48
13.17.3	Rozpoznawanie krawędzi...	13-50
13.17.4	Licznik...	13-51
13.17.5	Timer...	13-53
13.18	Biblioteka Util.lib	13-56
13.18.1	Konwersja BCD...	13-56
13.18.2	Funkcje bit/bajt...	13-56
13.18.3	Matematyczne funkcje pomocnicze...	13-57
13.18.4	Regulator...	13-59
13.18.5	Generatory sygnału...	13-62
13.18.6	Manipulatory funkcji...	13-64
13.18.7	Przetwarzanie wartości analogowych...	13-66
13.19	Biblioteka AnalyzationNew.lib	13-66
13.20	Biblioteki systemowe CoDeSys	13-67
Załącznik F	Przegląd: Operatory i moduły biblioteki	13-69
13.21	Operatory w CoDeSys	13-69
13.22	Moduły biblioteki Standard.lib	13-72
13.23	Moduły biblioteki Util.lib	13-73
Załącznik G	Linie poleceń/plik poleceń	13-75
13.24	Linie poleceń.....	13-75
13.25	Plik poleceń (Cmdfile).....	13-76
Załącznik H	Siemens, importowanie	13-83
13.26	Importuj plik symboli &SEQ	13-83
13.27	Importuj plik S5	13-84
13.28	Konwersja S5 zgodnie z IEC 1131-3	13-84
Załącznik I	Okna dialogowe ustawień systemu docelowego	13-89
13.29	Ustawienia kategorii Platforma docelowa	13-90
13.29.1	System docelowy „Intel 386 compatible”, platforma docelowa.....	13-90
13.29.2	System docelowy Motorola 68K, kategoria Platforma docelowa.....	13-91
13.29.3	System docelowy Infineon C16x, kategoria Platforma docelowa.....	13-92
13.29.4	Systemy docelowe Intel StrongARM i Power PC, kategoria Platforma docelowa 13-93	
13.29.5	System docelowy MIPS III ISA, kategoria Platforma docelowa	13-94
13.29.6	System docelowy Hitachi SH, kategoria Platforma docelowa	13-95
13.29.7	System docelowy 8051 compatible, kategoria Platforma docelowa.....	13-96
13.29.8	System docelowy TriCore, kategoria Platforma docelowa	13-96
13.30	Ustawienia kategorii Podział pamięci	13-97
13.31	Ustawienia kategorii Ogólne	13-99
13.32	Ustawienia kategorii Funkcje sieciowe	13-101
13.33	Ustawienia w kategorii Wizualizacja	13-102

<u>Załącznik J</u>	<u>Obsługa za pomocą klawiatury</u>	<u>13-105</u>
13.34	Obsługa za pomocą klawiatury.....	13-105
13.35	Kombinacje klawiszy	13-105
<u>Załącznik K</u>		<u>13-109</u>
<u>Załącznik L</u>	<u>Zalecenia odnośnie nadawania identyfikatorów</u>	<u>13-111</u>
13.36	Nadawanie identyfikatorów w trakcie deklaracji	13-111
13.37	Identyfikator zmiennych (nazw zmiennych).....	13-111
13.38	Identyfikator zdefiniowanych przez użytkownika typów danych (DUT)	13-113
13.39	Identyfikator funkcji, bloków funkcji, programów (POU)	13-113
13.40	Identyfikator wizualizacji.....	13-114
<u>Załącznik M</u>	<u>Błędy i ostrzeżenia w zakresie kompilacji</u>	<u>13-115</u>
13.41	Ostrzeżenia... ..	13-115
13.42	Błąd kompilacji... ..	13-120
<u>14 Indeks</u>		<u>I</u>

1 Krótka informacja o CoDeSys

1.1 Co to jest CoDeSys

CoDeSys jest skrótem utworzonym od Controller Development System, a oznaczającym środowisko programistyczne do projektowania sterowników.

Programista sterowników PLC może dzięki systemowi CoDeSys w bardzo łatwy sposób poznać szerokie zastosowania języków standardu IEC. Wykorzystywane w systemie edytory oraz funkcje debuggowania stanowią zaawansowanie środowiska programistyczne typowe dla języków wyższego rzędu (jak np. Visual C++).

1.2 Przegląd funkcji systemu CoDeSys

Jak wygląda struktura projektu?

Projekt, który zawiera się w programie sterownika, jest przechowywany w pliku z nazwą projektu. Projekt zawiera różnego rodzaju obiekty: moduły, definicje typów danych, elementy prezentacji (wizualizacja) i zasoby.

Pierwszy moduł, utworzony w nowym projekcie, nosi automatycznie nazwę **PLC_PRG**. Stamtąd uruchamiane są procesy (analogicznie jak funkcja main w programie C) oraz wywołania innych modułów (programów, bloków funkcji i funkcji).

Jeśli została zdefiniowana konfiguracja zadań (zasoby), nie jest wtedy wymagane tworzenie programu o nazwie PLC_PRG. Więcej informacji na ten temat można znaleźć w rozdziale 9.7 Konfiguracja zadań.

Organizer zawiera strukturę wszystkich obiektów w projekcie.

Jak utworzyć projekt?

W pierwszej kolejności należy wprowadzić i ewentualnie dopasować dla używanego sterownika **ustawienia systemu docelowego**.

Następnie należy **skonfigurować sterownik**, by móc dokonać sprawdzenia poprawności używanych w projekcie adresów wejścia i wyjścia.

Po wykonaniu tej czynności można tworzyć **moduły** i programować w dowolnych **językach**.

Po zakończeniu programowania można **skompilować** projekt i usunąć wykryte błędy.

Jak przetestować projekt?

Jeśli wszystkie błędy zostały usunięte, można aktywować **symulację**, zalogować się do symulacji sterownika i „załadować” projekt do sterownika. Teraz program pracuje w **trybie online**.

Można otworzyć okno z **konfiguracją sterownika** i sprawdzić, czy proces przebiega prawidłowo. W tym celu ustaw wejścia ręcznie i obserwuj, czy na wyjściach pojawiają się żądane wartości. Ponadto można obserwować w modułach, jak zmieniają się wartości zmiennych lokalnych. W **Menadżerze podglądu zmiennych i receptur** można konfigurować rekordy, których wartości chcesz kontrolować.

Debugowanie

Jeśli w kodzie programu występują błędy, można wstawić **punkty wstrzymania** (breakpoint). Punkt wstrzymania powoduje przerwanie wykonywania programu, dając możliwość śledzenia wartości wszystkich zmiennych projektowych w danym momencie. Dzięki przetwarzaniu krokowemu (**krok pojedynczy**) można sprawdzić poprawność logiczną programu.

Inne funkcje online

Inne funkcje debugowania:

Użytkownik może dla zmiennych projektowych oraz wejść/wyjść **ustawić określone wartości**.

Funkcja **kontroli przebiegu** umożliwia sprawdzenie, które wiersze programu zostały wykonane.

W **dzienniku** zapisywane są chronologicznie informacje o operacjach oraz czynnościach użytkownika, a także wewnętrzne operacje podczas sesji online.

Funkcja **zapisu śledzenia** oferuje możliwość śledzenia i przedstawienia graficznego przebiegu zmiennych w dowolnie długim okresie. Funkcję można aktywować w ustawieniach systemu docelowego.

W ustawieniach systemu docelowego jest dostępna również opcjonalna **przeładowarka PLC**, za pomocą której można wysyłać kwerendy określonych informacji ze sterownika.

Gotowy i przetestowany projekt można **załadować do urządzenia** i również tam przetestować. Dostępne są takie same funkcje online jak w przypadku symulacji.

Inne możliwości systemu CoDeSys

Cały projekt można w każdej chwili **udokumentować, wyeksportować** do pliku tekstowego lub **kompilować na inne języki**.

Komunikację w CoDeSys obsługują interfejsy symboli, DDE oraz COM. Serwer Gateway plus serwer OPC i serwer DDE są składnikami podstawowej instalacji systemu CoDeSys.

Za pomocą odpowiedniego pakietu **ustawień systemu docelowego**, które można załadować z pliku docelowego (Target Support Package), istnieje możliwość używania tego samego projektu CoDeSys na różnych systemach docelowych.

W zależności od ustawień systemu docelowego do transmisji danych z innymi sterownikami w sieci można opcjonalnie używać **zmiennych sieciowych globalnych** i **Menadżera parametrów (katalog obiektów)**.

ENI: Interfejs „Engineering Interface” zapewnia dostęp poprzez samodzielny serwer ENI do zewnętrznej bazy danych, w której przechowywane są moduły CoDeSys ew. pliki kompilacji. Są one również dostępne dla innych klientów serwera ENI, dzięki czemu jest możliwe np. tworzenie projektów CoDeSys w zespołach projektowych, oferowanie wspólnych danych dla różnych narzędzi między systemami CoDeSys oraz zarządzanie wersjami.

Narzędzia: Mechanizm narzędzi służy do tego, by podłączyć do CoDeSys specyficzne dla systemu docelowego pliki exe. Ponadto można zdefiniować, które pliki mają być wczytane do sterownika. Powiązania narzędzi dla systemu docelowego można zdefiniować w pliku docelowym lub w projekcie indywidualnie poprzez wstawienie w drzewie zasobów. Dostępność funkcji narzędzi jest zależna od systemu docelowego.

Wizualizacja CoDeSys może być używana w systemie **CoDeSys HMI** jako zwyczajna platforma obsługi lub zależnie od systemu docelowego jako **wizualizacja sieciowa** i/lub **wizualizacja elementu docelowego**. Ostatnie umożliwiają wywołanie wizualizacji przedstawiającej dane bieżącego sterownika poprzez Internet lub bezpośrednio na monitorze komputera. Więcej informacji na temat wizualizacji CoDeSys można znaleźć w osobnym podręczniku użytkownika.

Biblioteki tworzone w CoDeSys można zaopatrzyć w **informację o licencji**, jeśli przyszłe korzystanie z nich ma być uwarunkowane uzyskaniem licencji.

1.3 Przegląd dokumentacji użytkownika systemu CoDeSys

Moduł	Zawartość dokumentacji	Nazwa pliku
System programowania CoDeSys	ten podręcznik oraz Pomoc Online w menu Pomoc systemu programowania Pierwsze kroki w systemie programowania CoDeSys	CoDeSys_V23_D.pdf Erste Schritte mit CoDeSys V23.pdf
Serwer Gateway	Koncepcja, instalacja i interfejs użytkownika; oraz Pomoc Online dla interfejsu użytkownika poprzez menu Gateway, otwieranego kliknięciem myszy ikony Gateway na pasku narzędzi (tylko w języku angielskim)	Gateway Manual.pdf
Serwer OPC	Serwer OPC V2.0, instalacja i korzystanie	OPC_20_How_to_use_D.pdf
Wizualizacja CoDeSys	Podręcznik dla wizualizacji CoDeSys i CoDeSys HMI, wizualizacji elementu docelowego i wizualizacji sieciowej	CoDeSys_Visu_V23_D.pdf
SoftMotion	Podręcznik obsługi, częściowy opis bibliotek SoftMotion	CoDeSys_SoftMotion_V23_D.pdf
Biblioteki	Standard.lib i Util.lib są opisane w tym podręczniku. Dla każdej biblioteki systemowej CoDeSys istnieje osobny dokument zawierający nazwę biblioteki. (Biblioteki SoftMotion: patrz dokumentacja SoftMotion.)	<nazwa biblioteki>.pdf CoDeSys_V23_D.pdf
Serwer ENI	Instalacja i konfiguracja serwera ENI w aspekcie zarządzania projektem CoDeSys w zewnętrznej bazie danych. Więcej informacji nt. konfiguracji CoDeSys znajduje się w tym podręczniku. Informacje nt. ENI Admin, ENI Control i ENI Explorer są dostępne w Pomocy Online.	EniServerQuickstart_D.pdf CoDeSys_V23_D.pdf

2 Co jest czym w CoDeSys

2.1 Składniki projektu...

Projekt

Projekt zawiera wszystkie obiekty programu sterownika. Projekt jest przechowywany w pliku pod nazwą projektu. Do projektu należą następujące obiekty:

moduły, typy danych, wizualizacje, zasoby i biblioteki.

Moduł

Funkcje, bloki funkcji i programy są to moduły, dla których można wykonywać określone działania.

Każdy moduł składa się z części deklaracyjnej i kodu. Kod jest pisany w jednym z języków programowania IEC: IL, ST, SFC, FBD, LD lub CFC.

CoDeSys obsługuje wszystkie standardowe moduły IEC. Chcąc używać tych modułów w projekcie, należy podłączyć do projektu bibliotekę standard.lib.

Moduły mogą wywoływać inne moduły. Rekursje nie są jednak dozwolone.

Funkcja

Funkcją jest moduł, który po wykonaniu zwraca dokładnie jedną daną (może być także wieloskładnikowa, jak np. pola lub struktury) . Wywołanie funkcji w językach tekstowych może występować w wyrażeniach jako operator.

Podczas deklarowania funkcji należy pamiętać o tym, że funkcja musi zawierać typ; tzn. wpisując nazwę funkcji należy po dwukropku wstawić jej typ. Warto również zapoznać się z zaleceniami dot. nadawania nazw, Załącznik L.

Prawidłowa deklaracja funkcji przedstawia się następująco:

```
FUNCTION Fct: INT
```

Poza tym dla funkcji należy przypisać wynik; tzn. nazwa funkcji jest używana jak zmienna wyjściowa.

Deklaracja funkcji rozpoczyna się słowem kluczowym FUNCTION.

W języku SFC wywołanie funkcji można odbywać się wyłącznie w obrębie akcji danego kroku lub w obrębie tranzycji.

W ST wywołanie funkcji może być używane jako argument w wyrażeniach.

Wskazówka: CoDeSys dopuszcza używanie **zmiennych globalnych** wewnątrz funkcji. Jest to celowe odstępstwo od normy IEC61131-3, według której wartość zwracana przez funkcję może być zmieniana wyłącznie przez parametry wejściowe. Różnica między funkcjami i programami polega mianowicie na tym, że funkcje zwracają dokładnie tylko jedną wartość i że ich parametry oraz wartości zwracane są przekazywane w stosie.

Przykłady wywołania opisanej wyżej funkcji:

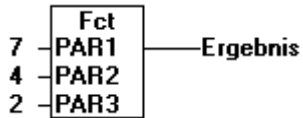
w IL:

```
LD 7
Fct 2,4
ST wynik
```

w ST:

```
wynik := Fct(7, 2, 4);
```

w FBD:



Uwaga: Jeśli zmienna lokalna w funkcji zadeklarowana jest jako RETAIN, nie wywołuje to żadnego skutku. Zmienna nie zostanie zapisana w obszarze retain!

Wskazówka: Jeśli w projekcie zostanie zdefiniowana funkcja o nazwie CheckBounds można z jej pomocą automatycznie sprawdzić przekroczenie obszaru matrycy (patrz Załącznik D, Typy danych).

Jeśli w projekcie zostaną zdefiniowane funkcje CheckDivByte, CheckDivWord, CheckDivDWord i CheckDivReal można używając operatora DIV sprawdzić wartość dzielnika, przykładowo po to, by uniknąć dzielenia przez 0 (patrz Załącznik A, Operatory IEC).

Jeśli zostaną zdefiniowane funkcje CheckRangeSigned i CheckRangeUnsigned można z ich pomocą automatycznie wysłać kwerendy przekroczenia obszarów dla zmiennych, które zostały zadeklarowane z typami podzакresu (patrz Załącznik D, Typy danych).
Wszystkie te nazwy funkcji są zarezerwowane ze względu na opisane tu możliwości zastosowania.

Moduł funkcji (blok funkcji)

Moduł funkcji – nazywany także blokiem funkcji – jest to moduł, który podczas wykonywania dostarcza jedną lub kilka wartości. Blok funkcji nie dostarcza wartości zwracanych w przeciwieństwie do funkcji.

Deklaracja bloku funkcji rozpoczyna się słowem kluczowym FUNCTION_BLOCK.

Warto również zapoznać się z zaleceniami dot. nadawania nazw, Załącznik L.

Można tworzyć duplikacje, inaczej zwane instancjami (kopie) bloku funkcji.

Przykład bloku funkcji z dwiema zmiennymi wejściowymi i dwiema zmiennymi wyjściowymi w IL. Pierwsze wyjście jest produktem obydwu wprowadzeń, drugie wynika z porównania równości:

```

0001 FUNCTION_BLOCK FUB
0002 VAR_INPUT
0003   PAR1:INT;
0004   PAR2:INT;
0005 END_VAR
0006 VAR_OUTPUT
0007   MULERG:INT;
0008   VERGL:BOOL;
0009 END_VAR
0010
0001 LD   PAR1
0002 MUL  PAR2
0003 ST   MULERG
0004
0005 LD   PAR1
0006 EQ   PAR2
0007 ST   VERGL
0008
  
```

Instancje bloków funkcji

Można tworzyć duplikacje, inaczej zwane instancjami (kopie) bloku funkcji.

Każda instancja posiada osobną identyfikację (nazwę instancji) oraz strukturę danych, która zawiera wejścia, wyjścia i wewnętrzne zmienne. Instancje deklaruje się podobnie jak zmienne lokalnie lub globalnie, podając jako typ identyfikatora nazwę bloku funkcji.

Warto również zapoznać się z zaleceniami dot. nadawania nazw, Załącznik L.

Przykład instancji o nazwie INSTANCJA bloku funkcji FUB:

```
INSTANCJA: FUB;
```

Wywołania bloków funkcji odbywają się zawsze poprzez powyższe instancje.

Poza instancją bloku funkcji możliwy jest dostęp tylko do parametrów wejścia i wyjścia, ale już nie do jego zmiennych wewnętrznych.

Deklaracje bloków funkcji i programy mogą zawierać deklaracje instancji. Deklaracje instancji w funkcjach są niedozwolone.

Dostęp do instancji bloku funkcji jest ograniczony do modułu, w którym została ona umieszczona, chyba że została ona zadeklarowana globalnie.

Nazwa instancji dla instancji bloku funkcji może być wykorzystana jako wejście dla funkcji lub bloku funkcji.

Wskazówka: Wszystkie wartości są przechowywane pomiędzy kolejnymi chwilami wykonania bloku funkcji. Stąd też wywołania bloku funkcji z tymi samymi argumentami nie zawsze dostarczają tych samych wartości wyjściowych!

Wskazówka: Jeśli blok funkcji zawiera chociaż jedną zmienną retain, kompletna instancja jest zapisana w obszarze retain.

Wywołanie bloku funkcji

Do zmiennych wejścia i wyjścia bloku funkcji można odwoływać się z innego modułu tworząc instancję bloku funkcji i podając w następującej składni żadaną zmienną:

```
<nazwa instancji>.<nazwa zmiennej>
```

Przypisanie parametrów przy wywołaniu:

Ustawienie parametrów wejścia i/lub wyjścia przy wywołaniu odbywa się poprzez wprowadzenie w nawiasach wartości tych parametrów bezpośrednio za nazwą instancji bloku funkcji (obowiązuje w przypadku języków tekstowych IL i ST). Przypisanie wartości dla parametrów wejścia odbywa się za pomocą wprowadzenia „:=” analogicznie jak dla inicjalizacji zmiennych w miejscu deklaracji, a dla parametrów wyjścia za pomocą „=>”.

Jeśli instancja wstawiana jest przy użyciu funkcji pomocy przy wprowadzaniu (<F2>) z opcją **Z argumentami** w oknie implementacji modułu ST lub IL, wtedy jest ona automatycznie wyświetlana w tej składni wraz z parametrami. Parametry nie muszą być jednak koniecznie przyporządkowane.

Przykład:

FBINST jest zmienną lokalną typu bloku funkcji, który zawiera zmienną wejściową xx i zmienną wyjściową yy. Podczas wywołania FBINST asystentem deklaracji przy wprowadzaniu zostanie ona wstawiona do programu ST w taki sposób: FBINST1(xx:= , yy=>);

Zmienne wejściowe/wyjściowe przy wywołaniu:

Należy pamiętać, że **zmienne wejściowe/wyjściowe (VAR_IN_OUT)** bloku funkcji są przesyłane jako **wskazniki**. Z tego względu przy wywołaniach nie można przypisywać im żadnych stałych, ani też uzyskać dostępu z zewnątrz z prawem do odczytu lub zapisu.

Przykład

Wywołanie zmiennej inout 1 typu VAR_IN_OUT bloku funkcji fubo w module ST:

```
VAR
  inst:fubo;
  var1:int;
END VAR
varI:=2;
inst(inout1:=var1);
```

Składniki projektu...

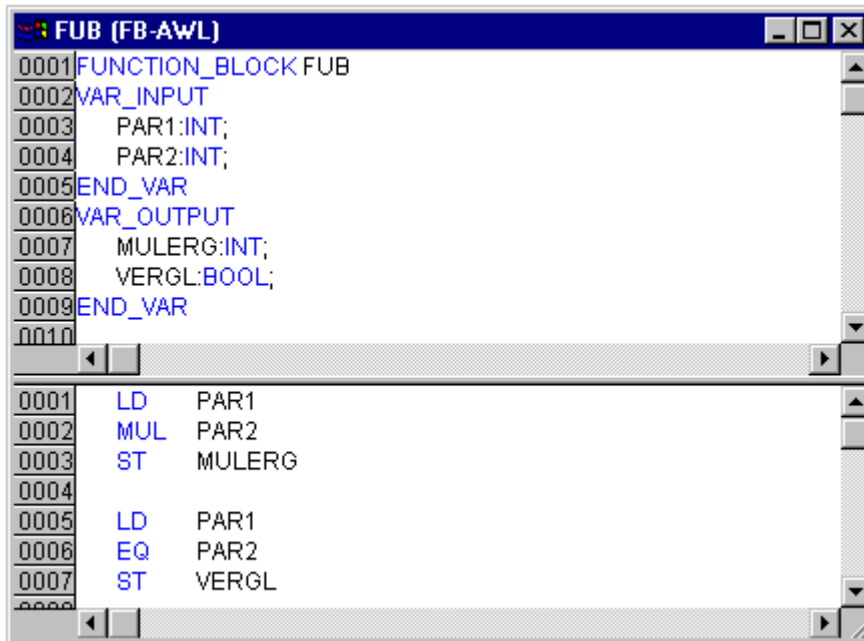
Niedozwolone jest: inst(inout1:=2); albo inst.inout1:=2;

Przykłady wywołania bloku funkcji FUB:

patrz wyżej, punkt „Blok funkcji”

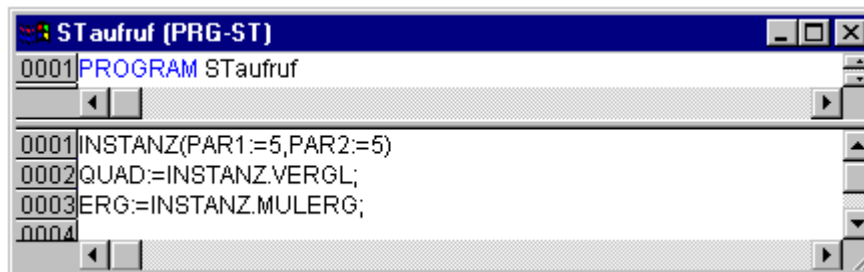
Wynik mnożenia jest przechowywany w zmiennej ERG, wynik porównywania w QUAD. Utworzona została instancja bloku FUB o nazwie INSTANCJA.

Wywołanie instancji bloku funkcji w IL odbywa się następująco:



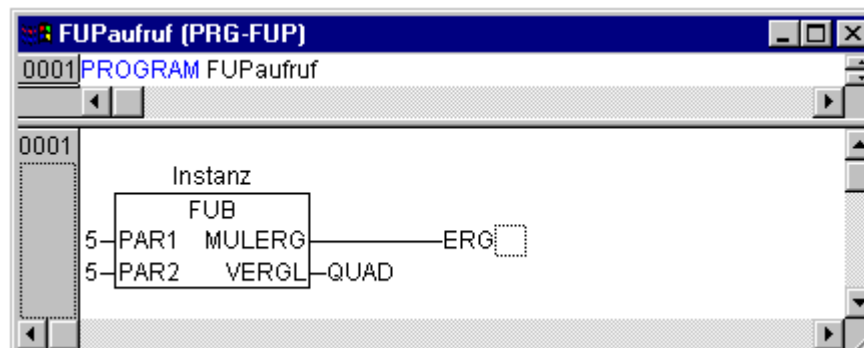
```
0001 FUNCTION_BLOCK FUB
0002 VAR_INPUT
0003     PAR1:INT;
0004     PAR2:INT;
0005 END_VAR
0006 VAR_OUTPUT
0007     MULERG:INT;
0008     VERGL:BOOL;
0009 END_VAR
0010
0001 LD  PAR1
0002 MUL PAR2
0003 ST  MULERG
0004
0005 LD  PAR1
0006 EQ  PAR2
0007 ST  VERGL
0008
```

Wywołanie instancji bloku funkcji w ST odbywa się następująco (część z deklaracją jak dla IL):



```
0001 PROGRAM STaufuf
0002
0001 INSTANZ(PAR1:=5,PAR2:=5)
0002 QUAD:=INSTANZ.VERGL;
0003 ERG:=INSTANZ.MULERG;
0004
```

Wywołanie instancji bloku funkcji w FBD odbywa się następująco (część z deklaracją jak dla IL):



W języku SFC wywołania bloków funkcji mogą odbywać się tylko krokowo.

Program

Program jest to moduł, który podczas wykonywania dostarcza jedną lub kilka wartości. Programy są jawne globalnie w całym projekcie. Wszystkie wartości są zachowywane pomiędzy kolejnymi chwilami wykonania programu.

Deklaracja programu rozpoczyna się słowem kluczowym PROGRAM i kończy słowem kluczowym END_PROGRAM. Warto również zapoznać się z zaleceniami dot. nadawania nazw, Załącznik L.

Przykład programu:



Programy mogą wywoływać programy lub bloki funkcji. Wywołanie programu w funkcji jest niedozwolone. Nie można również tworzyć instancji programów.

W przypadku gdy program jest wywoływany przez moduł i przy tym zmieniają się wartości programu, wówczas zmiany te są przechowywane między kolejnymi chwilami wywołania programu, nawet wtedy jeśli ten sam program zostanie wywołany przez inny moduł.

Zupełnie inaczej wygląda to przy wywołaniu bloku funkcji. Tam zmieniają się tylko wartości w danej instancji bloku funkcyjnego. Zmiany te odgrywają rolę również tylko wtedy, gdy wywoływana jest ta sama instancja.

Chcąc ustawić parametry wejścia i/lub wyjścia przy wywołaniu programu, czyli przypisać zmiennym wejścia/wyjścia określone wartości, wprowadź w nawiasach – bezpośrednio za nazwą programu – wartości tych parametrów (kod musi być napisany w językach tekstowych IL lub ST). Przypisanie odbywa się za pomocą wprowadzenia „:=” analogicznie jak dla inicjalizacji zmiennych w miejscu deklaracji.

Jeśli program wstawiany jest przy użyciu funkcji pomocy przy wprowadzaniu (<F2>) z opcją **Z argumentami** w oknie implementacji modułu ST lub IL, wtedy jest on automatycznie wyświetlany w tej składni wraz z parametrami. Parametry nie muszą być jednak koniecznie przyporządkowane.

Przykłady wywołań programu:

W programie PRGexample2 zadeklarowano zmienną wejściową in_var i zmienną wyjściową out_var każda typu INT. Zmienna zadeklarowana lokalnie nazywa się erg, również typu INT:

W IL:

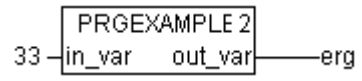
```
CAL PRGexample2
LD PRGexample2.out_var
ST   ERG
lub z podaniem parametrów bezpośrednio (pomoc przy wprowadzaniu „Z argumentami”,
patrz wyżej):
CAL PRGexample2(in_var:=33, out_var=>erg)
```

W ST:

```
PRGexample;
Erg := PRGexample2.out_var;
lub z podaniem parametrów bezpośrednio (pomoc przy wprowadzaniu „Z argumentami”,
patrz wyżej):
PRGexample2(in_var:=33, out_var=>erg);
```

W FBD:

Składniki projektu...



Przykładowa sekwencja wywołania PLC_PRG:

Por. program PRGbeispiel na ilustracji na początku tego rozdziału:

```
LD 0
ST PRGbeispiel.PAR (* ustawieniem domyślnym dla PAR jest 0 *)
CAL AWLaufruf (* ERG w AWLaufruf daje wynik 1 *)
CAL STaufruf (* ERG w STaufruf daje wynik 2 *)
CAL FUPaufruf (* ERG w FUPaufruf daje wynik 3 *)
```

Jeśli zmienna PAR programu PRGbeispiel zostanie najpierw zainicjalizowana z programu głównego wartością 0, a następnie będą wykonywane po kolei powyższe wywołania programów, wówczas wynik Erg w programach będzie miał wartości 1, 2 i 3. Jeśli kolejność wywołań zostanie zmieniona, zmieniają się również odpowiednio wartości danych parametrów wyniku.

PLC_PRG

Istnieje możliwość, ale nie jest to konieczne, użycia tzw. zadań do sterowania przetwarzaniem projektu (konfiguracja zadań). Jeśli jednak nie istnieje żadna konfiguracja zadań, projekt musi wówczas zawierać moduł PLC_PRG. PLC_PRG jest modulem typu program generowanym automatycznie podczas tworzenia nowego projektu jednorazowo w „Projekt” „Wstaw obiekt”. PLC_PRG jest wywoływany tylko jeden raz na każdy cykl sterujący.

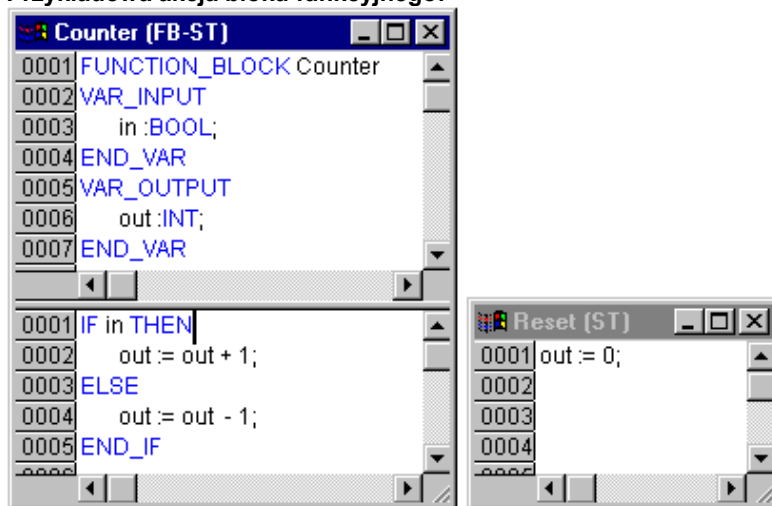
Jeśli jest dostępna konfiguracja zadań, wówczas projekt nie musi zawierać modułu PLC_PRG, gdyż kolejność wykonywania jest zależna od przyporządkowania zadań.

Uwaga: Nie należy usuwać ani nie zmieniać nazwy modułu PLC_PRG (przy założeniu, że w projekcie nie jest używana konfiguracja zadań. PLC_PRG jest zasadniczo programem głównym w programie pojedynczego zadania.

Akcja

Dla bloków funkcji i programów można definiować i dodawać akcje („Projekt” „Dodaj akcję”). Akcja reprezentuje dalszą implementację, którą w całości można utworzyć w innym języku jako „normalną” implementację. Każda akcja posiada nazwę.

Akcja pracuje z danymi bloku funkcji ew. programu, do którego należy. Akcja używa takich samych zmiennych wejścia/wyjścia i zmiennych lokalnych, jak „normalna” implementacja.

Przykładowa akcja bloku funkcyjnego:

W tym przykładzie wywołanie bloku funkcji Counter powoduje zwiększenie lub zmniejszenie wartości zmiennej wyjściowej out w zależności od zmiennej wejściowej in. Wywołanie akcji „Reset” bloku funkcji ustawia zmienną wyjściową out na zero. W obu przypadkach opisywana jest ta sama zmienna out.

Wywołanie akcji:

Akcja jest wywoływana składnią <nazwa programu>.<nazwa akcji> albo <nazwa instancji>.<nazwa akcji>. Należy pamiętać o zasadzie pisowni w FBD (patrz przykład niżej). Jeśli akcja ma być

wywoływana w obrębie własnego modułu, wówczas w edytorach tekstu używa się tylko nazwy akcji, a w graficznych edytorach wywołania bloku funkcji bez podawania instancji.

Przykłady wywołania opisanej wyżej akcji z innego modułu:

Deklaracja dla wszystkich przykładów:

```
PROGRAM PLC_PRG
VAR
  inst : counter;
END_VAR
```

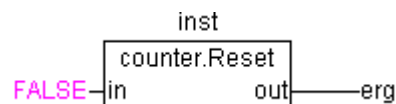
Wywołanie akcji „Reset” w innym module zaprogramowanym w **IL**:

```
CAL inst.Reset(in := FALSE)
LD  inst.out
ST  ERG
```

Wywołanie w innym module zaprogramowanym w **ST**:

```
inst.Reset(in := FALSE);
Erg := inst.out;
```

Wywołanie w innym module zaprogramowanym w **FBD**:



Wskazówka: W modułach w języku przebiegu akcje odgrywają znaczącą rolę.

Zasoby

Zasobów używa się do konfiguracji i organizacji projektu oraz do śledzenia wartości zmiennych:

- **Zmienne globalne**, które można zastosować w całym projekcie albo sieci
- **Biblioteki**, które można podłączać do projektu za pomocą menedżera bibliotek
- **Dziennik** do rejestrowania aktywności w trybie online
- **Konfiguracja alarmów** do konfigurowania obsługi alarmów w projekcie
- **Konfiguracja sterownika** do konfigurowania urządzenia
- **Konfiguracja zadań** do sterowania programem za pomocą zadań
- **Menadżer podglądu zmiennych i receptur** do wyświetlania i ustawiania wartości domyślnych dla zmiennych
- **Ustawienia systemu docelowego** do wybierania i również konfiguracji końcowej systemu docelowego
- **Obszar roboczy** zawiera obraz opcji projektu

W zależności od wybranego systemu docelowego albo ustawień systemu docelowego w CoDeSys dostępne są również następujące zasoby:

- **Menadżer parametrów** do wymiany danych między sterownikami podłączonymi w sieci
- **Przeglądarka PLC** jako monitor sterownika
- **Zapis śledzenia** do graficznego przedstawienia wartości zmiennych
- **Narzędzia** do wywoływania zewnętrznych aplikacji
- **SoftMotion** funkcjonalność (licencjonowana) z edytorem CNC (lista programów CNC) i edytorem CAM (krzywki tarczowe)

Por. rozdział 9, Zasoby.

Biblioteki

W projekcie można podłączyć szereg bibliotek, których moduły, typy danych i zmienne globalne można używać tak samo, jak zdefiniowane przez użytkownika. Biblioteki `standard.lib` i `util.lib` są dostępne w standardowej wersji programu.

Por. rozdział 9.3, Zarządzanie biblioteką, „Zarządzanie biblioteką”.

Typy danych

Obok standardowych typów danych użytkownik może definiować własne typy danych. Można tworzyć struktury, typy wyliczania i referencje.

Por. Załącznik D, Typy danych.

Wizualizacja

CoDeSys umożliwia tworzenie wizualizacji przedstawiających zmienne projektowe na obrazie graficznym. Za pomocą wizualizacji można rysować w trybie offline elementy geometryczne. Te z kolei w trybie online zmieniają w zależności od wartości określonych zmiennych swoje kształty/kolory/napisy.

Wizualizacja CoDeSys może być również używana w systemie CoDeSys HMI jako jedyna platforma obsługi lub zależnie od systemu docelowego również jako wizualizacja sieciowa lub wizualizacja elementu docelowego obsługiwana przez Internet lub na systemie docelowym.

Patrz także dokument „Wizualizacja CoDeSys”.

2.2 Języki...

Obsługiwane języki programowania

CoDeSys obsługuje wszystkie języki programowania opisane w normie IEC-61131:

Języki tekstowe:

- Lista instrukcji (IL)
- Tekst zhierarchizowany (ST)

Języki graficzne:

- Język przebiegu (SFC)
- Schemat styków (LD)
- Schemat funkcji (FBD)
- Dodatkowo dostępny jest na bazie schematu funkcji graficzny edytor schematów bloków funkcji (CFC).

2.2.1 Lista instrukcji (IL)...

Lista instrukcji (IL) składa się z ciągu instrukcji. Każda instrukcja zaczyna się w nowym wierszu i zawiera operator i, w zależności od rodzaju operacji, jeden lub kilka argumentów oddzielonych przecinkami.

Przed instrukcją może znajdować się identyfikator *znacznik* z dwukropkiem (:) na końcu. Służy on do oznaczania instrukcji i może być przykładowo używany jako punkt docelowy skoku.

Komentarz można dodać na końcu wiersza. Między instrukcjami można wstawiać puste wiersze.

Przykład:

```
LD 17
ST lint (* komentarz *)
```

```

GE 5
JMPC next
LD idword
EQ istruct.sdword
STN test
next:

```

Modyfikatory i operatory w IL

W języku IL można używać następujących operatorów i modyfikatorów.

Modyfikatory:

- C z JMP, CAL, RET: Instrukcja zostanie wykonana tylko wtedy, gdy wynik poprzedniego wyrażenia jest TRUE.
- N z JMPC, CALC, Instrukcja zostanie wykonana tylko wtedy, gdy wynik poprzedniego wyrażenia jest FALSE.
- N inaczej: Negacja argumentu (nie dla akumulatora).

Poniżej znajduje się tabela zawierająca opis wszystkich operatorów języka IL wraz z odpowiadającymi im modyfikatorami:

Operator	Modyfikatory	Znaczenie
LD	N	Ustawia bieżący wynik równy argumentowi
ST	N	Zapisuje bieżący wynik w miejscu wstawienia argumentu
S		Ustawia argument boolean na TRUE, jeśli bieżący wynik jest TRUE
R		Ustawia argument boolowski na FALSE, jeśli bieżący wynik jest FALSE
AND	N,(Operator bitowy AND
OR	N,(Operator bitowy OR
XOR	N,(Operator bitowy specjalny OR
ADD	(Dodawanie
SUB	(Odejmowanie
MUL	(Mnożenie
DIV	(Dzielenie
GT	(>
GE	(>=
EQ	(=
NE	(<>
LE	(<=
LT	(<
JMP	CN	Skok do znacznika
CAL	CN	Wywołuje program lub blok funkcji
RET	CN	Opuszcza moduł i powraca do elementu, który spowodował wywołanie.
)		Analizuje zawieszoną operację

Wykaz wszystkich operatorów IEC znajduje się w załączniku.

Przykład programu IL wykorzystującego kilka modyfikatorów:

```
LD      TRUE      (* ładuje TRUE do akumulatora *)
ANDN    BOOL1     (* wykonuje AND z zanegowaną wartością zmiennej BOOL1 *)
JMPC    znacznik  (* jeśli wynik był TRUE, przeskakuje do znacznik „znacznik” *)
```

```
LDN     BOOL2     (* zachowuje zanegowaną wartość dla *)
ST      ERG       (* BOOL2 w ERG *)
```

Znacznik:

```
LD      BOOL2     (* zachowuje wartość dla *)
ST      ERG       (* BOOL2 w ERG *)
```

W języku IL można również wstawiać nawiasy po operacji. Jako argument będzie traktowana wtedy wartość nawiasów.

Przykład:

```
LD  2
MUL 2
ADD 3
Erg
```

W tym przykładzie wartość Erg wynosi 7. Jeśli jednak zostaną wstawione nawiasy

```
LD  2
MUL (2
ADD 3
)
ST Erg
```

wartość Erg wynosi już 10, gdyż operacja MUL zostanie wykonana dopiero wtedy, gdy osiągnięty zostanie nawias zamykający „)”; z obliczenia argumentem dla MUL będzie wtedy 5.

2.2.2 Tekst zhierarchizowany (ST)...

Tekst zhierarchizowany składa się z szeregu instrukcji, które tak jak w językach wyższego rzędu mogą być wykonywane warunkowo (IF..THEN..ELSE) lub w pętlach (WHILE..DO).

Przykład:

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

Wyrażenia

Wyrażenie jest konstrukcją, która po wykonaniu zwraca obliczoną wartość.

Wyrażenia składają się z operatorów i argumentów. Argument może być stałą, zmienną, wywołaniem funkcji lub innym wyrażeniem.

Obliczanie wyrażeń

Obliczanie wyrażenia odbywa się poprzez działania na operatorach wg określonych *zasad ważności*. Operator z największym priorytetem jest przetwarzany w pierwszej kolejności, dalej operator z kolejnym co do ważności priorytetem, itd. aż wszystkie operatory zostaną przetworzone.

Języki...

Przy przetwarzaniu operatorów z takim samym priorytetem obowiązuje kolejność z lewej do prawej.

Poniżej przedstawiono tabelę z operatorami ST uporządkowanymi wg priorytetu:

Operacja	Symbol	Priorytet
Ujęcie w nawiasy	(wyrażenie)	największy priorytet
Wywołanie funkcji	nazwa funkcji (lista parametrów)	
Potęgowanie	EXPT	
Negowanie uzupełnianie	- NOT	
Iloczyn Iloraz Modulo	* / MOD	
Dodawanie Odejmowanie	+ -	
Porównanie	<,>,<=,>=	
Znak równości Znak większości lub mniejszości	= <>	
Iloczyn logiczny AND	AND	
Suma modulo 2 XOR	XOR	
Suma logiczna OR	OR	Największy priorytet

Następująca tabela przedstawia uporządkowane instrukcje w języku ST wraz z przykładami:

Rodzaj instrukcji	Przykład
Przyporządkowanie	A:=B; CV := CV + 1; C:=SIN(X);
Wywołanie bloku funkcji i użycie wyjścia bloku funkcji	CMD_TMR(IN := %IX5, PT := 300); A:=CMD_TMR.Q
RETURN	RETURN;
IF	D:=B*B; IF D<0.0 THEN C:=A; ELSIF D=0.0 THEN C:=B; ELSE C:=D; END_IF;
CASE	CASE INT1 OF 1: BOOL1 := TRUE; 2: BOOL2 := TRUE; ELSE BOOL1 := FALSE; BOOL2 := FALSE; END_CASE;

Rodzaj instrukcji	Przykład
FOR	J:=101; FOR I:=1 TO 100 BY 2 DO IF ARR[I] = 70 THEN J:=I; EXIT; END_IF; END_FOR;
WHILE	J:=1; WHILE J<= 100 AND ARR[J] <> 70 DO J:=J+2; END_WHILE;
REPEAT	J:=-1; REPEAT J:=J+2; UNTIL J= 101 OR ARR[J] = 70 END_REPEAT;
EXIT	EXIT;
Pusta instrukcja	;

Operator przyporządkowania

Po lewej stronie przyporządkowania znajduje się argument (zmienna, adres), dla którego z prawej strony zostanie przyporządkowana wartość wyrażenia za pomocą operatora przyporządkowania :=

Przykład:

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

Po wykonaniu tego wiersza zmienna Var1 będzie miała dziesięciokrotną wartość Var2.

Wywołanie bloków funkcji w języku ST

Blok funkcji w ST wywołuje się poprzez wpisanie nazwy jego instancji i podaniem w nawiasach żądanych wartości dla parametrów. Następujący przykład przedstawia wywołanie timera z przyporządkowaniami dla jego parametrów IN i PT. Następnie zmienna Q zawierająca wynik zostanie przyporządkowana do zmiennej A.

Zmienną zwracającą wynik wywołuje się jak w języku IL poprzez podanie nazwy bloku funkcji, następnie kropki i nazwy zmiennej:

```
CMD_TMR(IN := %IX5, PT := 300);  
A:=CMD_TMR.Q
```

Instrukcja RETURN

Instrukcji RETURN używa się w celu opuszczenia modułu, np. zależnie od warunku.

Instrukcja CASE

W instrukcji CASE można połączyć w jednej konstrukcji wiele instrukcji warunkowych z tymi samymi zmiennymi warunkowymi.

Składnia:

CASE <Var1>

OF

```
<wartość 1>: <instrukcja 1>  
<wartość 2>: <instrukcja 2>  
<wartość3, wartość4, wartość5: <instrukcja 3>  
<wartość6 .. wartość10 : <instrukcja 4>
```

...

<wartość n>: <instrukcja n>

ELSE <instrukcja ELSE>

END_CASE;

Instrukcja CASE pracuje wg następującego schematu:

- Jeśli zmienna w <Var1> ma <wartość i>, wówczas wykonuje się <instrukcja i>.
- Jeśli zmienna <Var 1> nie ma żadnej z zadeklarowanych wartości, wówczas wykonuje się <instrukcja ELSE>.
- Chcąc wykonać tę samą instrukcję warunkową dla kilku wartości zmiennych, wartości te należy wpisać jedna za drugą oddzielając przecinkami i dalej powiązać wspólną instrukcją warunkową.
- Jeśli ta sama instrukcja ma być wykonana dla określonego zakresu wartości zmiennych, wystarczy podać wartość początkową i końcową oddzielone dwiema kropkami i dalej powiązać wspólną instrukcją warunkową.

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;
    
```

Instrukcja IF

Instrukcja IF bada pewien warunek i w zależności od wyniku wykonuje odpowiednie instrukcje.

Składnia:

IF <wyrażenie_boolean1>

THEN

<instrukcje_IF>

{ELSIF <wyrażenie_boolean2>

THEN

<instrukcje1_ELSIF>

.

.

ELSIF <wyrażenie_boolean n>

THEN

<instrukcje_ELSIF n-1>

ELSE

<instrukcje_ELSE>

}

END_IF;

Element w nawiasach klamrowych {} jest opcjonalny.

Jeśli <wyrażenie_boolean1> daje w wyniku TRUE, wówczas wykonywane będą tylko <instrukcje_IF> i żadne z pozostałych instrukcji.

Jeśli warunek nie jest spełniony, wykonywane będą po kolei wyrażenia boolean zaczynające się od <wyrażenie_boolean2> tak długo, dopóki któreś z nich da wynik TRUE. Następnie wykonywane będą tylko instrukcje zawarte między tym wyrażeniem boolean (spełniającym warunek) a następną instrukcją ELSE lub ELSIF.

Jeśli żadne z wyrażen boolean nie daje wyniku TRUE, wówczas będą wykonywane wyłącznie <instrukcje_ELSE>.

Przykład:

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

W tym przykładzie program włącza ogrzewanie, gdy temperatura spadnie poniżej 17 stopni, w innym razie ogrzewanie jest wyłączone.

Pętla FOR

Za pomocą pętli FOR można zaprogramować powtarzanie przebiegów.

Składnia:

```
INT_Var :INT;
FOR <INT_Var> := <INIT_WART>
TO <END_WART>
{BY <wielkość kroku>}
```

DO

```
<instrukcje>
END_FOR;
```

Element w nawiasach klamrowych {} jest opcjonalny.

<Instrukcje> są wykonywane tak długo, jak długo licznik <INT_Var> nie będzie większy niż <END_WART>. Warunek ten jest sprawdzany przed wykonaniem się <instrukcji>; w ten sposób <instrukcje> nigdy nie są wykonywane, jeśli <INIT_WART> jest większa niż <END_WART>.

Zawsze, gdy <instrukcje> zostaną wykonane, wartość <INT_Var> zwiększa się o <wielkość kroku>. Wielkość kroku może przyjmować każdą wartość typu integer. Jeśli nie została ona podana, wtedy wynosi ona domyślnie 1. Pętla musi określać warunek zakończenia dla powtarzania, gdyż <INT_Var> za każdym razem zwiększa się.

Przykład:

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

Przyjmijmy, że zmienna Var1 ma wartość początkową 1, wtedy po wykonaniu pętli FOR będzie miała wartość 32.

Wskazówka: Wartość <END_WART> nie może być równa wartości granicznej licznika <INT_VAR>. Jeśli np. zmienna Licznik jest typu SINT, wartość <END_WART> nie może być 127, gdyż oznacza to pętlę nieskończoną.

Pętla WHILE

Pętli WHILE można używać tak jak pętli FOR z tą różnicą, że warunkiem kontrolującym jej zakończenie może być dowolne wyrażenie boolean. Oznacza to, że jeśli podany warunek jest spełniony, wówczas pętla zostanie wykonana.

Składnia:

WHILE <wyrażenie boolean>

DO

<instrukcje>

END_WHILE;

<Instrukcje> są powtarzane tak długo w pętli, jak długo <wyrażenie boolean> daje wynik TRUE. Jeśli <wyrażenie boolean> już przy pierwszym wykonaniu daje wynik FALSE, wówczas <instrukcje> nie zostaną ani razu wykonane. Jeśli <wyrażenie_boolean> nigdy nie przyjmuje wartości FALSE, wówczas <instrukcje> są wykonywane bez końca, przez co powstaje błąd czasu przebiegu.

Wskazówka: Programista musi pamiętać o tym, aby przy zmianie warunku w części instrukcji nie spowodować powstania pętli nieskończonej, np. powiększanie lub pomniejszanie licznika.

Przykład:

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

Pętle WHILE i REPEAT mają w pewnym sensie tę przewagę nad pętlą FOR, gdyż przed wykonaniem pętli nie trzeba znać liczby jej przebiegów. Najczęściej w praktyce wykorzystywane są tylko te dwa rodzaje pętli. Jeśli jednak znana jest liczba przebiegów pętli, należy korzystać z pętli FOR, gdyż ta nie powoduje powstawania pętli nieskończonych.

Pętla REPEAT

Pętla REPEAT odróżnia się od pętli WHILE tym, że warunek zakończenia wykonywania pętli sprawdzany jest dopiero po jej wykonaniu. Skutek tego jest taki, że pętla musi zostać co najmniej jeden raz wykonana niezależnie od warunku jej zakończenia.

Składnia:

REPEAT

<instrukcje>

UNTIL <wyrażenie boolean>

END_REPEAT;

<Instrukcje> są wykonywane tak długo, jak długo <wyrażenie boolean> daje wynik TRUE.

Jeśli <wyrażenie boolean> już przy pierwszym wykonaniu daje wynik TRUE, wówczas <instrukcje> zostaną wykonane tylko jeden raz. Jeśli <wyrażenie_boolean> nigdy nie przyjmuje wartości TRUE, wówczas <instrukcje> są wykonywane bez końca, przez co powstaje błąd czasu przebiegu.

Wskazówka: Programista musi pamiętać o tym, aby przy zmianie warunku w części instrukcji nie spowodować powstania pętli nieskończonej, np. powiększanie lub pomniejszanie licznika.

Przykład:

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

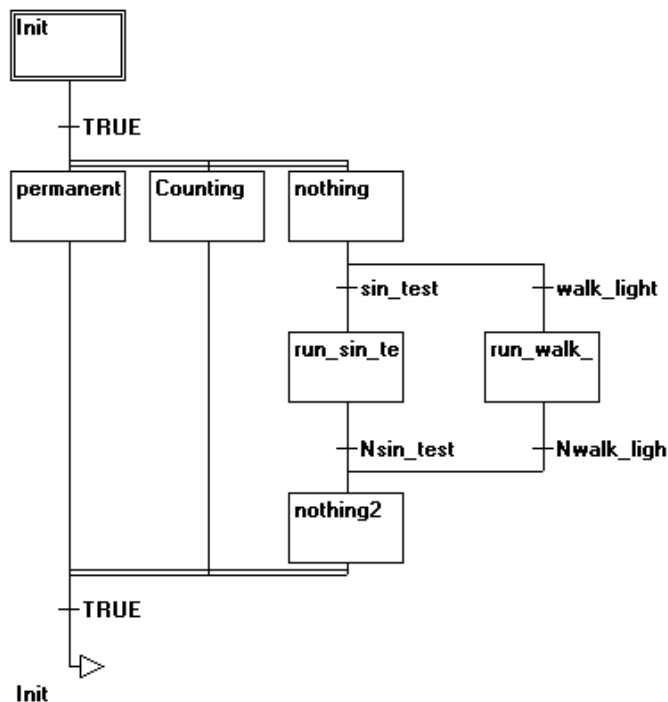
Instrukcja EXIT

Instrukcja EXIT występująca w pętli FOR, WHILE lub REPEAT skutkuje zakończeniem wykonywania najbardziej wewnętrznej pętli niezależnie od warunku zakończenia.

2.2.3 Język przebiegu (SFC)...

Język przebiegu jest jednym z języków zorientowanych graficznie, opisujących czasowy przebieg różnych czynności wewnątrz programu. W tym celu język wykorzystuje grafy zawierające kroki, dla których przypisane są określone akcje, i tranzycje, które sterują przebiegiem kroków.

Przykład sieci opisanej w języku przebiegu:



Krok

Napisany w języku przebiegu moduł, składa się z ciągu kroków połączonych ze sobą za pomocą ukierunkowanych warunków (tranzycji).

Istnieją dwa rodzaje kroków.

- Najprostsza konstrukcja składa się z jednej akcji i jednej flagi, która pokazuje, czy krok jest aktywny. Jeśli dla danego kroku została zaimplementowana akcja, wówczas w prawym górnym rogu prostokąta przedstawiającego krok pojawi się mały trójkąt.
- Krok IEC składa się z flagi oraz jednej lub kilku przypisanych akcji albo zmiennych boolean. Powiązane akcje pojawiają się z prawej strony kroku.

Akcja

Jedna akcja może zawierać ciąg instrukcji w IL lub w ST, dowolną ilość sieci w FBD lub w LD albo znowu strukturę przebiegu (SFC).

W krokach uproszczonych akcja jest powiązana zawsze z należącym do niej krokiem. By edytować akcję kliknij dwukrotnie krok, który do niej należy, lub zaznacz krok i wybierz polecenie menu „Dodatki” „Powiększ akcję/przejście”. Ponadto dla każdego kroku można przypisać jedną akcję wejściową i/lub wyjściową.

Akcje występujące w krokach IEC należą w strukturze organizera obiektów bezpośrednio do odpowiedniego modułu SFC i można je stamtąd załadować do programu edytora podwójnym

kliknięciem lub naciśnięciem klawisza <Enter>. Nowe akcje tworzy się za pomocą polecenia „Projekt” „Dodaj akcję”. W jednym kroku IEC może być zawartych maksymalnie dziewięć akcji.

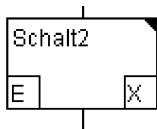
Akcja wejściowa i akcja wyjściowa

Dla akcji zawartej w danym kroku można dodatkowo utworzyć akcję wejściową i akcję wyjściową. Akcja wejściowa wykonywana jest tylko jeden raz, bezpośrednio po aktywowaniu kroku. Akcja wyjściowa wykonywana jest również tylko jeden raz, zanim krok zostanie dezaktywowany.

Krok z akcją wejściową jest zaznaczony literą „E” w lewym dolnym rogu, akcja wyjściowa jest zaznaczana literą „X” w prawym dolnym rogu.

Akcja wejściowa i wyjściowa może być implementowana w dowolnym języku. By edytować akcję wejściową lub wyjściową, kliknij dwukrotnie krok w odpowiednim narożniku.

Przykład kroku z akcją wejściową i wyjściową:



Tranzycja / Warunek tranzycji

Między krokami znajdują się tzw. tranzycje.

Warunek tranzycji musi przyjmować wartość TRUE lub FALSE. Zatem może on zawierać zmienną boolean, adres boolean lub stałą boolean. Może on również zawierać ciąg instrukcji dających wynik boolean napisanych w składni ST (np. $(i \leq 100) \text{ AND } b$) lub w dowolnym języku (patrz „Dodatki” „Powiększ akcję/przejsięcie”). Tranzycja nie może jednak zawierać programów, bloków funkcji lub przyporządkowań!

W celu analizy wyrażeń tranzycji można zdefiniować flagę SFCErrAnalyzationTable.

W edytorze SFC warunek tranzycji można napisać bezpośrednio do znacznika tranzycji lub otworzyć w tym celu osobne okno edytora. Warunek występujący w edytorze (patrz rozdział 8.4.4, „Dodatki” „Powiększ akcję/tranzycję”) ma pierwszeństwo!

Wskazówka: Poza omawianymi tu tranzycjami można używać również trybu krokowego, by przejść do następnego kroku, patrz SFCtip i SFCtipmode.

Bieżący krok

Po wywołaniu modułu SFC wykonywana jest najpierw akcja należąca do kroku inicjującego (z podwójną ramką). Krok, którego akcja jest aktualnie wykonywana, określa się jako „bieżący”. W trybie online bieżące kroki są wyświetlane na niebiesko.

W każdym cyklu wykonywane są najpierw akcje, należące do bieżących kroków. Po ich wykonaniu bieżącymi krokami będą kroki znajdujące się bezpośrednio za nimi, o ile warunki tranzycji dla tych kroków mają wartość TRUE. A aktualnie bieżące kroki zostaną wtedy wykonane w następnym cyklu.

Wskazówka: Jeśli bieżący krok zawiera akcję wyjściową, również i ta zostanie wykonana dopiero w następnym cyklu, pod warunkiem że następująca po niej tranzycja przyjmie wartość TRUE.

Krok IEC

Obok uproszczonych kroków w języku SFC można używać kroków zgodnych z normą IEC.

By używać kroków IEC, należy w projekcie podłączyć specjalną bibliotekę SFC **lecsfc.lib**.

Dla jednego kroku IEC może być przypisanych maksymalnie dziewięć akcji. Akcje IEC nie są, jak w przypadku uproszczonych kroków, na stałe przyporządkowane dla kroku jako akcje wejściowe, krokowe lub wyjściowe, lecz występują oddzielnie od kroków i mogą być wielokrotnie używane w obszarze modułu, w którym się znajdują. W tym celu należy je skojarzyć z żądanymi krokami za pomocą polecenia „Dodatki” „Powiązanie akcji”.

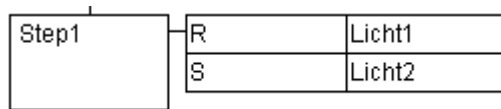
Poza akcjami do kroków można również przyporządkować zmienne boolean.

Aktywowaniem i dezaktywowaniem akcji i zmiennych boolean można sterować za pomocą tzw. kwalifikatorów. Możliwe są przy tym opóźnienia czasowe. Ponieważ akcja może być ciągle jeszcze aktywna podczas wykonywania następnego kroku, np. przez kwalifikator S (Set), wówczas przy takim stanie można osiągnąć procesy równoległe.

Skojarzona zmienna boolean jest ustawiana lub resetowana każdym wywołaniem modułu SFC. To znaczy, że za każdym razem przypisywana jest jej od nowa albo wartość TRUE, albo FALSE.

Akcje powiązane należące do kroku IEC są pokazywane z prawej strony kroku w dwuczęściowym prostokącie. Lewe pole zawiera kwalifikator, ewentualnie ze stałymi czasowymi, a prawe nazwę akcji lub zmiennej boolean.

Przykładowy krok IEC z dwiema akcjami:



Dla lepszej czytelności procesów wszystkie bieżące akcje w trybie online są wyświetlane na niebiesko (tak jak bieżące kroki). Po każdym cyklu odbywa się sprawdzenie, które akcje są aktywne.

Należy przy tym pamiętać również o ograniczeniu przy stosowaniu kwalifikatorów czasowych dla akcji używanych wielokrotnie w tym samym cyklu (patrz „Kwalifikatory”)!

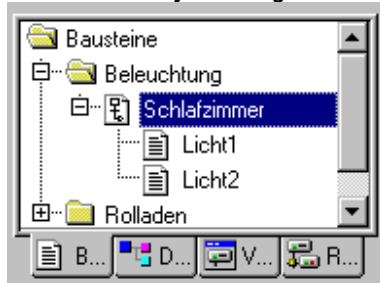
Wskazówka: Jeśli akcja została dezaktywowana, zostanie ona jeszcze raz wykonana. Oznacza to, że każda akcja wykonywana jest przynajmniej dwa razy (także akcja z kwalifikatorem P).

Przetwarzanie akcji przy wywołaniu odbywa się w kolejności alfabetycznej, przy czym najpierw przetwarzane są akcje dezaktywowane, a następnie znowu wszystkie bieżące akcje.

Nowy krok, który został wstawiony do projektu, będzie krokiem IEC, jeśli zostało użyte polecenie menu „Dodatki” „Zastosuj kroki IEC”.

W Organizерze obiektów poszczególne akcje są przyporządkowane bezpośrednio do swoich modułów SFC. Nowe akcje tworzy się za pomocą polecenia „Projekt” „Dodaj akcję”.

Moduł SFC z akcjami w Organizерze obiektów:



Kwalifikatory

Do powiązywania akcji z krokami IEC służą następujące kwalifikatory:

N	Non-stored	akcja jest tak długo aktywna, jak długo aktywny jest krok
R	overriding Reset	akcja dezaktywowana
S	Set (Stored)	akcja jest aktywowana i pozostaje aktywna do następnego resetu
L	time Limited	akcja jest aktywna w określonym czasie, jednak maksymalnie dopóki aktywny jest krok
D	time Delayed	akcja jest aktywna po upływie określonego czasu, o ile krok jest nadal jeszcze aktywny i dalej dopóki krok jest aktywny
P	Pulse	akcja jest wykonywana dokładnie jeden raz, jeśli krok jest aktywny

Języki...

SD	Stored and time Delayed	akcja jest aktywowana po upływie określonego czasu i pozostaje aktywna do następnego resetu
DS	Delayed and Stored	akcja jest aktywowana po upływie określonego czasu, o ile krok jest jeszcze aktywny i pozostaje aktywna do następnego resetu
SL	Stored and time Limited	akcja jest aktywna w określonym czasie

Kwalifikatory L, D, SD, DS i SL wymagają podania czasu w formacie stałych czasowych TIME, np. L T#5s.

Wskazówka: Jeśli akcja została dezaktywowana, zostanie ona jeszcze raz wykonana. Oznacza to, że każda akcja wykonywana jest przynajmniej dwa razy (także akcja z kwalifikatorem P).

Wskazówka: Przetwarzanie akcji przy wywołaniu odbywa się w kolejności alfabetycznej, przy czym najpierw przetwarzane są akcje dezaktywowane, a następnie znowu wszystkie bieżące akcje.

Uwaga: Jeśli ta sama akcja zostanie użyta w dwóch bezpośrednio następujących po sobie krokach z kwalifikatorami, które sterują przebiegiem czasowym, wówczas kwalifikator czasowy może być nieskuteczny przy wykonywaniu drugiego kroku. Aby temu zapobiec, należy dodać krok pośredni, który spowoduje ponowną inicjalizację stanu akcji w dodatkowym cyklu.

Zmienne pośrednie w języku SFC

W języku SFC można używać zmiennych deklarowanych pośrednio (flagi) do tworzenia kwerend statusu i akcji oraz czasu trwania kroków. Takie flagi wstawia się zawsze na początku cyklu. Dla kroków i akcji IEC są one dostępne w standardowej bibliotece iecsfc.lib (struktury SFCStepType i SFCActionType), dla kroków uproszczonych są one zaimplementowane bezpośrednio w CoDeSys:

Kwerenda statusu kroku lub akcji z wykorzystaniem zmiennej boolean:

- W przypadku kroków IEC: <StepName>.x wzgl. <StepName>._x: <StepName>.x wskazuje aktualny status aktywacji. <StepName>._x wskazuje status aktywacji dla następnego cyklu. Jeśli <StepName>.x=TRUE, krok zostanie wykonany w aktualnym cyklu.
- Jeśli <StepName>._x=TRUE i <StepName>.x=FALSE, wtedy krok zostanie wykonany w następnym cyklu, tzn. <StepName>._x zostanie na początku cyklu skopiowany na <StepName>.x.
- W przypadku kroków uproszczonych: <StepName> wzgl. _<StepName>: Jeśli <StepName> ma wartość TRUE, krok zostanie wykonany w aktualnym cyklu. Jeśli _<StepName> ma wartość TRUE, krok zostanie wykonany w następnym cyklu, tzn. <StepName> zostanie na początku cyklu skopiowany na <StepName>.
- W przypadku czynności IEC: <Aktionsname>.x otrzyma wartość TRUE natychmiast po uaktywnieniu czynności. (<Aktionsname>._x służy wyłącznie do celów wewnętrznych, nie do sprawdzania statusu).

Sprawdzanie czasu trwania kroku na podstawie zmiennych TIME:

Poniższe zmienne pośrednie podają aktualny czas trwania od momentu uaktywnienia kroku, w odniesieniu do którego w atrybutach kroku skonfigurowano minimalny czas trwania:

- W przypadku kroków IEC: <StepName>.t (<StepName>._t nie umożliwia użycia z zewnątrz)
- W przypadku kroków uproszczonych: _time<StepName>. Tę zmienną pośrednią należy jednakże również zadeklarować w tym celu bezpośrednio jako zmienną TIME; np. "_timeStep1 : TIME;"
- W przypadku czynności IEC: wewnętrzne zmienne czasowe nie są używane

Te „znaczniki” statusu mogą być stosowane w przypadku każdej czynności i każdego przejścia modułu AS. Można również uzyskać do nich dostęp z poziomu innego programu:

Przykład: boolvar1:=sfc.step1.x;

step1.x oznacza tu zmienną boolean, która prezentuje status kroku IECstep1 w module sfc1.

Znaczniki AS

Do sterowania przebiegiem w języku przebiegu można użyć znaczników, generowanych automatycznie podczas przetwarzania projektu. Należy w tym celu zadeklarować odpowiednie zmienne globalne lub lokalne jako zmienne wejściowe lub wyjściowe. Przykład: Jeśli w języku przebiegu (AS) krok jest dłuższy, niż wskazano w jego atrybutach, ustawiany jest znacznik, dostępny za pośrednictwem zmiennej o nazwie SFCErrror (SFCErrror ma wartość TRUE). Możliwe są znaczniki zmiennych:

SFCEnableLimit: Jest to specjalna zmienna typu BOOL. Jeśli ma ona wartość TRUE, przekroczenia czasu w przypadku kroków rejestrowane są w SFCErrror. W przeciwnym razie przekroczenia czasu będą ignorowane. Zastosowanie to może być np. przydatne podczas uruchamiania lub w trybie ręcznym.

SFCInitt: Gdy ta zmienna boolean ma wartość TRUE, język przebiegu resetowany jest do poziomu kroku Initt. Inne znaczniki AS są również resetowane (inicjalizacja). Dopóki zmienna ma wartość TRUE, włączony (aktywny) jest krok Initt, który jednakże nie jest wykonywany. Dopiero w momencie ustawienia dla SFCInitt wartości FALSE, moduł jest dalej przetwarzany w zwykły sposób.

SFCReset: Ta zmienna typu BOOL zachowuje się podobnie jak SFCInitt. W odróżnieniu od poprzedniej krok Initt przetwarzany jest jednak również po inicjalizacji. W ten sposób można np. w kroku Initt ustawić wartość FALSE dla znacznika SFCReset.

Należy pamiętać: Począwszy od wersji kompilatora 2.3.7.0 można za pomocą SFCReset resetować również czynności boolean, przyporządkowane do kroków IEC, co nie było możliwe wcześniej.

SFCQuitError: Dopóki ta zmienna boolean ma wartość TRUE, przetwarzanie wykresów AS jest zatrzymane, resetowane jest przy tym ew. przekroczenie czasu w zmiennych SFCErrror. Po ustawieniu dla zmiennej ponownie wartości FALSE, resetowane są dotychczasowe czasy w aktywnych krokach. Warunkiem jest zadeklarowanie znacznika SFCErrror, który rejestruje przekroczenie czasu.

SFCPause: Dopóki zmienna ta ma wartość TRUE, przetwarzanie wykresu AS jest zatrzymane.

SFCErrror: Zmienna ta ma wartość TRUE, gdy w wykresie AS nastąpi przekroczenie czasu. Jeśli w programie, po pierwszym przekroczeniu czasu, wystąpi następne przekroczenie, nie jest ono już rejestrowane, jeśli zmienna SFCErrror nie została uprzednio ponownie zresetowana. Deklaracja SFCErrror stanowi warunek funkcjonowania innych zmiennych znaczników, służących do kontroli przebiegu czasowego (SFCErrrorStep, SFCErrrorPOU, SFCQuitError, SFCErrrorAnalyzationTable).

SFCTrans: Ta zmienna boolean ma wartość TRUE, jeśli zostanie włączone przejście.

SFCErrrorStep: Jest to zmienna typu STRING. Jeśli SFCErrror zarejestruje w wykresie AS przekroczenie czasu, w zmiennej tej zapisywana jest nazwa kroku, który spowodował przekroczenie czasu. Warunkiem jest zadeklarowanie zmiennej SFCErrror, która rejestruje przekroczenie czasu.

SFCErrrorPOU: Ta zmienna typu STRING zawiera, w przypadku przekroczenia czasu, nazwę modułu, w którym wystąpiło przekroczenie czasu. Warunkiem jest zadeklarowanie zmiennej SFCErrror, która rejestruje przekroczenie czasu.

SFCCurrentStep: Jest to zmienna typu STRING. W zmiennej tej zapisywana jest nazwa kroku, który jest aktywny niezależnie od monitorowania czasu. W przypadku rozgałęzienia równoległego krok ten zapisywany jest w zewnętrznej prawej gałęzi.

SFCErrrorAnalyzationTable: Ta zmienna typu ARRAY [0..n] OF ExpressionResult wykazuje w przypadku każdej zmiennej złożonego wyrażenia przejścia, która prowadzi do stanu FALSE przejścia, a tym samym do przekroczenia czasu w kroku poprzedzającym, następujące informacje: nazwa, adres, komentarz, aktualna wartość.

Jest to możliwe w przypadku maks. 16 zmiennych, tzn. obszar tablicy wynosi maks. 0–15.

Struktura ExpressionResult, jak również używane pośrednio moduły analizy znajdują się w bibliotece AnalyzationNew.lib. Moduły analizy mogą być używane również bezpośrednio w modułach, które nie zostały zaprogramowane w SFC.

Warunkiem analizy wyrażenia przejścia jest zarejestrowanie przekroczenia czasu w kroku poprzedzającym. Dlatego też musi być zaimplementowane monitorowanie czasu oraz zadeklarowana w module zmienna SFCErrror (por. wyżej).

SFCTip, SFCTipMode: Te zmienne typu BOOL umożliwiają tryb impulsowy SFC. Jeśli zostanie on włączony przez SFCTipMode=TRUE, przełączenie do następnego kroku może nastąpić tylko poprzez ustawienie wartości TRUE dla SFCTip. Dopóki dla trybu SFCTip ustawiona jest wartość FALSE, dalsze przełączanie może odbywać się dodatkowo także poprzez przejścia.

Wskazówka: Należy pamiętać również o zmiennych pośrednich, opisanych w poprzednim rozdziale, których można użyć do sprawdzania statusu i czasu trwania kroków oraz czynności.

Gałąź alternatywna

Dwie lub więcej gałęzi w AS można zdefiniować jako odgałęzienia alternatywne. Każda gałąź alternatywna musi rozpoczynać i kończyć się przejściem. Odgałęzienia alternatywne mogą zawierać odgałęzienia równoległe oraz dalsze odgałęzienia alternatywne. Odgałęzienie alternatywne rozpoczyna się od poziomej linii (początek alternatywy) i kończy się poziomą linią (koniec alternatywy) lub też skokiem.

Jeśli krok, poprzedzający linię początku alternatywy, jest aktywny, wtedy pierwsze przejście każdego odgałęzienia alternatywnego analizowane jest od lewej do prawej. Otwierane jest pierwsze przejście z lewej, którego warunek przejścia ma wartość TRUE, i uaktywniane są kroki kolejne (patrz aktywny krok).

Gałąź równoległa

Dwa lub więcej odgałęzień w AS można zdefiniować jako odgałęzienia równoległe. Każda gałąź równoległa musi rozpoczynać i kończyć się krokiem. Odgałęzienia równoległe mogą zawierać odgałęzienia alternatywne oraz dalsze odgałęzienia równoległe. Odgałęzienie równoległe rozpoczyna się od podwójnej linii (początek równoległy) i kończy się podwójną linią (koniec równoległy) lub też skokiem. Można je oznaczyć znacznikiem skoku.

Jeśli krok poprzedzający linię równoległego początku jest aktywny, i warunek przejścia po tym kroku ma wartość TRUE, wtedy aktywne są pierwsze kroki wszystkich odgałęzień równoległych (patrz aktywny krok). Wszystkie te gałęzie są teraz przetwarzane równoległe. Krok po linii końca równoległego jest aktywny, gdy wszystkie kroki poprzedzające są aktywne i warunek przejścia przed tym krokiem dostarcza wartość TRUE.

Skok

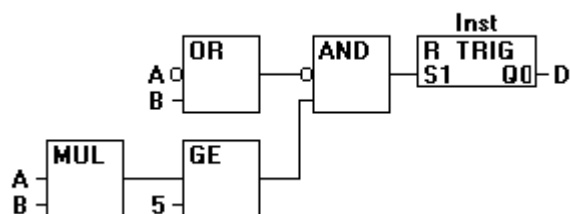
Skok to połączenie z krokiem, którego nazwa podana jest pod symbolem skoku. Skoki są niezbędne, ponieważ niedozwolone jest tworzenie połączeń prowadzących do góry lub krzyżujących się.

2.2.4 Schemat funkcji (FBD)...

Schemat funkcji to ukierunkowany graficznie język programowania. Działa on w oparciu o listę sieci, przy czym każda sieć posiada strukturę, przedstawiającą zawsze jakieś wyrażenie logiczne wzgl. arytmetyczne, wywołanie bloku funkcji, skok lub instrukcję powrotu.

W graficznym edytorze schematów funkcji nie są używane sieci.

Przykład sieci w schemacie funkcji:



2.2.5 Graficzny edytor schematów funkcji (CFC)...

Graficzny edytor schematów funkcji działa nie jak schemat funkcji, w oparciu o sieci, lecz w oparciu o dowolnie rozmieszczane elementy. To umożliwia tworzenie np. sprzężeń zwrotnych.

Przykład implementacji w graficznym edytorze schematów funkcji:



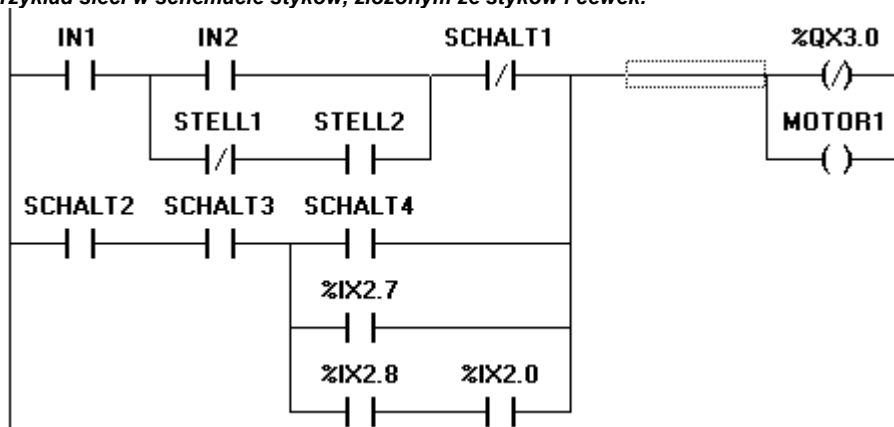
2.2.6 Schemat styków (LD)...

Schemat styków to ukierunkowany graficznie język programowania, zbliżony zasadą działania do połączenia elektrycznego.

Schemat styków nadaje się z jednej strony do konstruowania zespołów przełączających, z drugiej zaś można dzięki niemu tworzyć sieci podobnie jak w FBD. Dlatego też schemat styków można dobrze wykorzystać do sterowania wywoływaniem innych modułów. Schemat styków składa się z ciągu sieci. Sieć ograniczana jest z lewej i z prawej strony przez lewy i prawy, pionowy przewód elektryczny. Pomiędzy znajduje się schemat ideowy złożony ze styków, cewek i linii połączeń.

Każda sieć składa się z lewej strony z ciągu styków, przekazujących z lewej strony na prawą stan „WŁ.” lub „WYŁ.”, stany te odpowiadają wartościom logicznym TRUE i FALSE. Do każdego styku należy jedna zmienna boolean. Jeśli zmienna ta ma wartość TRUE, stan przekazywany jest za pomocą linii połączeń z lewej strony na prawą, w przeciwnym razie prawe połączenie ma wartość WYŁ.

Przykład sieci w schemacie styków, złożonym ze styków i cewek:



Styk

Każda sieć w schemacie styków składa się po lewej stronie z sieci styków (styki są przedstawiane za pomocą dwóch równoległych linii: | |), które przekazują z lewej do prawej stan „wł.” lub „wył.”.

Stany te odpowiadają wartościom boolean TRUE i FALSE. Do każdego styku należy jedna zmienna boolean. Jeśli zmienna ta ma wartość TRUE, stan przekazywany jest za pomocą linii połączeń z lewej strony na prawą, w przeciwnym razie prawe połączenie ma wartość „wył.”.

Styki mogą być połączone równolegle, wtedy *jedno* z odgałęzień równoległych musi przekazywać wartość „wł.”, aby odgałęzienie równoległe przekazywało wartość „wł.”, lub też styki są połączone szeregowo, wtedy *wszystkie* styki muszą przekazywać stan „wł.”, aby ostatni styk przekazywał również stan „wł.”. Odpowiada to więc elektrycznemu połączeniu równoległemu wzgl. szeregowemu.

Styk może być również zanegowany, co jest rozpoznawalne po ukośniku w symbolu styku: |/. Jeśli zmienna ma wartość FALSE, wtedy przekazywana jest wartość linii.

Cewka

Po prawej stronie sieci w schemacie styków znajduje się dowolna liczba tzw. cewek, symbolizowanych przez nawiasy: (). Mogą one być łączone tylko równolegle. Cewka przekazuje wartość połączeń od lewej do prawej, i kopiuje ją do odpowiedniej zmiennej boolean. Na linii wejścia może być ustawiona wartość WŁ. (odpowiadająca zmiennej boolean TRUE) lub też wartość WYŁ. (odpowiadająca FALSE).

Styki i cewki mogą być również negowane. Jeśli cewka jest zanegowana (co jest rozpoznawalne po ukośniku w symbolu cewki: (/)), wtedy kopiuje ona wartość zanegowaną do odpowiedniej zmiennej boolean. Jeśli styk jest zanegowany, wtedy przełącza on tylko, gdy odpowiednia zmienna boolean ma wartość FALSE.

Bloki funkcji w schemacie styków

Oprócz styków i cewek można wprowadzać również bloki funkcji oraz programy, które muszą mieć w sieci wejście i wyjście o wartościach boolean i które mogą być używane w tych samych miejscach, co styki, tzn. po lewej stronie sieci schematu styków.

Cewki Set/Reset

Cewki można również zdefiniować jako cewki Set lub Reset. Cewka Set (rozpoznawalna po „S” w symbolu cewki: (S)) nigdy nie zastępuje wartości TRUE w przyporządkowanej zmiennej boolean. To znaczy, że jeśli dla zmiennej ustawiono raz wartość TRUE, pozostanie ona bez zmian.

Cewka Reset (rozpoznawalna po „R” w symbolu cewki: (R)) nigdy nie zastępuje wartości FALSE w przyporządkowanej zmiennej boolean. Jeśli dla zmiennej ustawiono raz wartość FALSE, pozostanie ona bez zmian.

Schemat styków jako schemat funkcji

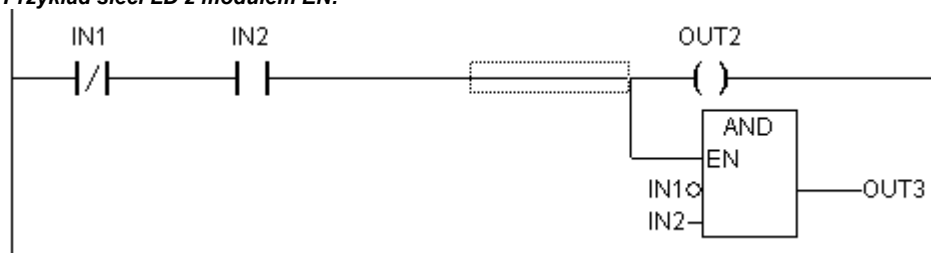
Podczas pracy z zastosowaniem schematu styków często zdarza się, że pojawia się chęć wykorzystania wyniku połączenia styków do sterowania innymi modułami. Można wtedy z jednej strony zapisać wynik za pomocą cewek w zmiennej globalnej, wykorzystywanej w innym miejscu. Można również zaimplementować ew. wywołanie bezpośrednio do sieci schematu styków. W tym celu należy wprowadzić moduł z wejściem EN.

Takie moduły to zwykłe argumenty, funkcje, programy oraz bloki funkcji, które posiadają dodatkowe wejście, opisane jako EN. Wejście EN jest zawsze wejściem typu BOOLE i ma następujące znaczenie: moduł z wejściem EN wykorzystywany jest wtedy, gdy EN ma wartość TRUE.

Moduł EN jest podłączany równolegle do cewek, przy czym wejście EN łączone jest z linią połączenia pomiędzy stykami a cewkami. Jeśli za pośrednictwem tej linii przekazywana jest informacja WŁ., wtedy moduł ten przetwarzany jest całkiem normalnie.

Zaczynając od takiego modułu EN można tworzyć sieci podobnie jak w schemacie funkcji.

Przykład sieci LD z modulem EN:



2.3 Debugowanie, funkcjonalności online...

Debugowanie

Funkcja debugowania programu CoDeSys ułatwia wyszukiwanie błędów.

Aby umożliwić debugowanie, należy wykonać polecenie „Projekt” „Opcje” i w wyświetlonym oknie dialogowym wybrać w menu **Opcje kompilacji** punkt **Debugowanie**.

Punkt wstrzymania

Punkt wstrzymania to miejsce programu, w którym przetwarzanie jest zatrzymywane. Umożliwia to obserwację zmiennych w określonym punkcie programu.

Punkty wstrzymania można umieszczać we wszystkich edytorach. W edytorach tekstowych punkty wstrzymania ustawiane są na numerach wierszy, w schematach funkcji i styków na numerach sieci, w CFC na modułach, a w AS na krokach. Nie można ustawić punktów wstrzymania w instancjach bloków funkcji.

Uwaga: System czasu przebiegu CoDeSys SP 32 Bit Full dezaktywuje funkcję śledzenia danego zadania, gdy tylko przetwarzanie programu zatrzyma się w punkcie wstrzymania.

Krok pojedynczy

krok pojedynczy oznacza:

- w IL: Wykonywanie programu do następnego CAL, LD lub polecenia JMP.
- w ST: Wykonanie następnej instrukcji.
- w FBD, LD: Wykonanie następnej sieci.
- w AS: Wykonanie czynności do następnego kroku.
- w CFC: Wykonanie następnego modułu (Box) w programie CFC.

Dzięki przetwarzaniu krokowemu można sprawdzić poprawność logiczną programu.

Cykl pojedynczy

Jeśli wybrano cykl pojedynczy, przetwarzanie zatrzymywane jest po każdym cyklu.

Zmianianie wartości online

Podczas pracy systemu można jeden raz ustawić określoną wartość zmiennych (zapisywanie wartości) lub też zastępować je na nowo określoną wartością po upływie każdego cyklu (wymuszanie). Wartość zmiennej można zmienić również w trybie online, klikając ją dwukrotnie. Zmienne boolean zmieniają się wtedy z TRUE na FALSE wzgl. odwrotnie, w przypadku wszystkich innych wyświetlane jest okno dialogowe **Zapisz zmienną xy**, w którym można edytować aktualną wartość zmiennej.

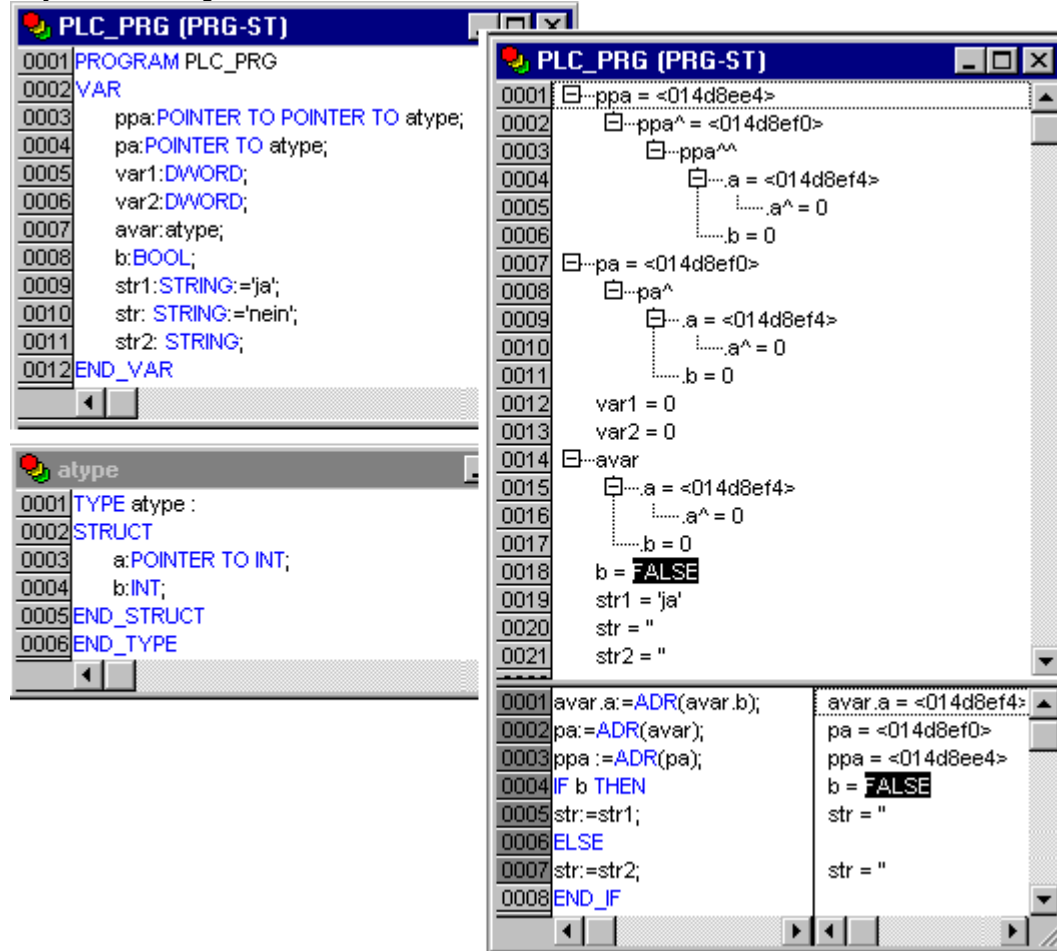
Monitoring

W trybie online ze sterownika na bieżąco są odczytywane i prezentowane aktualne wartości wszystkich zmiennych, jakie są widoczne na ekranie. Widok ten dostępny jest w edytorze deklaracji oraz w edytorze programów, ponadto zaś aktualne wartości zmiennych można prezentować w administratorze list kontrolnych i instrukcji oraz w wizualizacji. Jeśli monitorowane mają być zmienne z instancji bloku funkcji, należy najpierw otworzyć odpowiednią instancję. Podczas monitorowania zmiennych VAR_IN_OUT prezentowana jest wartość dereferencyjna.

Podczas monitorowania wskaźników w części deklaracyjnej, prezentowany jest zarówno wskaźnik, jak też wartość dereferencyjna. W części programowej prezentowany jest tylko wskaźnik:

```
+ --pointervar = '<'pointervalue'>'
```

Przykład monitoringu wskaźników:



Wskaźniki (POINTER) w wartościach dereferencyjnych są również wyświetlane w odpowiedni sposób. Pojedyncze kliknięcie symbolu krzyżyka lub dwukrotne kliknięcie wiersza powoduje zwinięcie wzgl. rozwinięcie widoku.

W implementacjach wyświetlana jest wartość wskaźnika. Dla dereferencji wyświetlana jest jednakże wartość dereferencyjna.

Monitoring składników tablic (ARRAY): Dodatkowo do składników tablic, indeksowanych za pośrednictwem stałej, wyświetlane są również składniki, indeksowane za pośrednictwem zmiennej:

```
anarray[1] = 5
anarray[i] = 1
```

Jeśli indeks składa się z wyrażenia (np. [i+j] lub [i+1]), składnika nie można wyświetlić.

Należy pamiętać: Gdy zostanie osiągnięta liczba zmiennych, jakie można maksymalnie monitorować, w przypadku każdej dalszej zmiennej, zamiast aktualnej wartości wyświetlany jest tekst „Zbyt wiele zmiennych monitoringu”.

Symulacja

W przypadku symulacji utworzony program sterujący nie jest przetwarzany w sterowniku, lecz w komputerze, na którym działa również program CoDeSys. Dostępne są wszystkie funkcje online. Dzięki temu użytkownik ma możliwość przetestowania logicznej poprawności programu bez samego urządzenia sterującego.

Wskazówka: Moduły z bibliotek zewnętrznych nie działają w symulacji.

Dziennik

W dzienniku zapisywane są w kolejności chronologicznej czynności wykonywane przez użytkownika, procesy wewnętrzne, zmiany statusu i stany wyjątkowe podczas trybu online. Umożliwia to nadzorowanie oraz śledzenie błędów.

2.4 Norma...

Norma IEC 61131-3 to międzynarodowy standard języków programowania w sterownikach z programowalną pamięcią.

Języki programowania, stosowane w programie CoDeSys, są zgodne z wymaganiami tej normy.

Zgodnie z tym standardem program składa się z następujących elementów:

- Struktury
- Moduły
- Zmienne globalne

Ogólne elementy języka zostały opisane w rozdziałach: Identyfikator, Adresy, Typy, Komentarze oraz Stałe.

Przetwarzanie programu CoDeSys rozpoczyna się od specjalnego modułu PLC_PRG. Moduł PLC_PRG może wywoływać inne moduły.

4 Tworzenie niewielkiego programu

4.1 Sterownik sygnalizacji świetlnej...

Wprowadzenie

Poniżej przedstawiono sposób napisania małego, przykładowego programu. Dotyczy on niewielkiej instalacji sygnalizacji świetlnej, która ma sterować dwoma sygnalizatorami na skrzyżowaniu. Obydwa sygnalizatory zmieniają fazowo światła czerwone/zielone, a w celu uniknięcia wypadków należy wyposażyć je pomiędzy fazami jeszcze w światło żółte wzgl. fazę przełączania żółte/czerwone. Te ostatnie światła będą więc działać dłużej niż pierwsze.

Na tym przykładzie można zobaczyć, w jaki sposób można przedstawiać programy zależne od czasu za pomocą środków językowych normy IEC61131-3, w jaki sposób za pomocą CoDeSys można edytować różne języki normy i jak można je bezproblemowo łączyć. Dodatkowo można nauczyć się symulacji programu CoDeSys.

Tworzenie modułów

Sam początek jest łatwy: należy uruchomić program CoDeSys i wybrać polecenie menu „**Plik**” „**Nowy**”.

W wyświetlonym oknie dialogowym „Ustawienia systemu docelowego” można wybrać system docelowy lub też ustawienie „Zaden”. Ta ostatnia opcja odpowiada ustawieniu trybu symulacji, co wystarczy dla konstruowanego tu modułu. Należy potwierdzić wybór za pomocą przycisku OK. Zostanie wyświetlone okno dialogowe „Nowy moduł”, z wprowadzonym już domyślnie wpisem PLC_PRG. Nazwę tę należy zachować, zaś rodzaj modułu zawsze powinien stanowić program; w każdym projekcie niezbędny jest program o tej nazwie. W naszym przypadku jako język tego modułu należy wybrać graficzny edytor schematów funkcji (CFC).

Znaczenie modułów zostanie objaśnione w dalszej części, na początek należy utworzyć jeszcze trzy dodatkowe moduły za pomocą polecenia „Projekt” „Wstaw obiekt” z paska menu lub za pośrednictwem menu podręcznego (kliknięcie prawym przyciskiem myszy w Organizeryze obiektów): Program w języku AS (język przebiegu) o nazwie PRZEBIEG, blok funkcji w języku FBD (schemat funkcji) o nazwie SYGNALIZATOR, oraz moduł CZEKAJ, również typu blok funkcji, który zostanie zaprogramowany jako lista instrukcji (IL).

Sposób działania SYGNALIZATORA

W module SYGNALIZATOR do świateł sygnalizatora zostaną przyporządkowane poszczególne fazy sygnalizacji, tzn. czerwona lampka będzie świecić w przypadku fazy czerwonej oraz żółtej/czerwonej, zaś żółta lampka w przypadku fazy żółtej oraz żółtej/czerwonej itp.

Sposób działania modułu CZEKAJ

W module CZEKAJ zostanie zaprogramowany prosty timer, w którym wprowadzony zostanie czas trwania fazy w milisekundach, i który po upływie tego czasu przekazuje w wyniku wartość TRUE.

Sposób działania modułu PRZEBIEG

W module PRZEBIEG wszystkie te elementy zostaną połączone, dzięki czemu prawidłowe światło sygnalizacji zapali się we właściwym czasie i będzie świecić się przez żądany okres czasu.

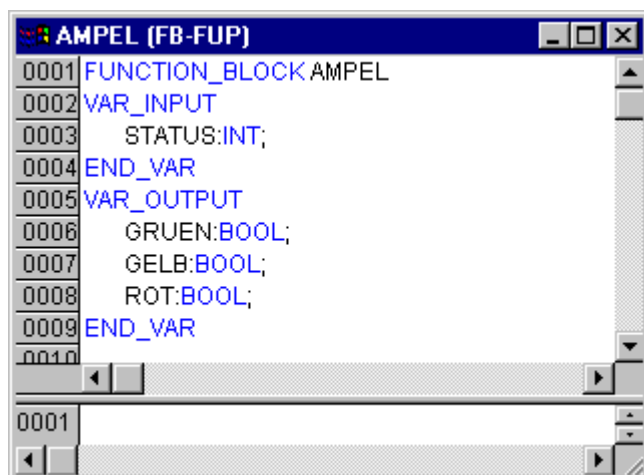
Sposób działania programu PLC_PRG

W programie PLC_PRG wejściowy sygnał uruchamiający sprzęgany jest z przebiegiem faz sygnalizacji i jako wyjścia udostępniane są „instrukcje kolorów” dla poszczególnych lamp obu sygnalizatorów.

Deklaracja „SYGNALIZATORA”

Jako pierwszy należy przygotować moduł SYGNALIZATOR. W edytorze deklaracji należy jako zmienną wejściową (pomiędzy słowami kluczowymi VAR_INPUT oraz END_VAR) zadeklarować zmienną typu INT o nazwie STATUS. STATUS będzie posiadać cztery możliwe stany, a mianowicie po jednym dla fazy sygnalizacji: zielone, żółte, żółte-czerwone i czerwone.

Sygnalizator ma zgodnie z tym trzy pozycje wyjściowe, a mianowicie: CZERWONE, ZOLTE, ZIELONE (znaki specjalne nie są w zmiennych dopuszczalne). Po zadeklarowaniu tych trzech zmiennych część deklaracyjna bloku funkcji SYGNALIZATOR ma postać:

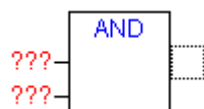


Szkielet „SYGNALIZATORA”

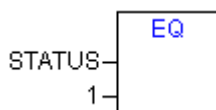
Teraz należy ustalić wartości zmiennych wyjściowych na podstawie wpisu STATUS w module. W tym celu należy przejść do szkieletu modułu. Kliknąć pole po lewej stronie obok pierwszej sieci (szare pole o numerze 0001). Wybrano pierwszą sieć. Teraz należy wybrać punkt menu

„Wstaw” „Moduł”.

W pierwszej sieci zostanie wstawione okno dialogowe z argumentem AND oraz dwoma wejściami:

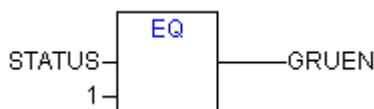


Należy kliknąć za pomocą myszy tekst AND, aby został zaznaczony, a następnie zmienić ten tekst w EQ. Należy wybrać za każdym razem trzy znaki zapytania dla obydwu wejść i zastąpić je przez „STATUS” wzgl. „1”. Wynikiem będzie następująca sieć:



Następnie należy kliknąć miejsce za oknem dialogowym EQ. Wybrane zostanie wyjście operacji EQ. Wybierz polecenie

„Wstaw” „Przyporządkowanie”. Trzy znaki zapytania ??? należy zmienić na ZIELONE. W ten sposób została utworzona sieć o następującej postaci:

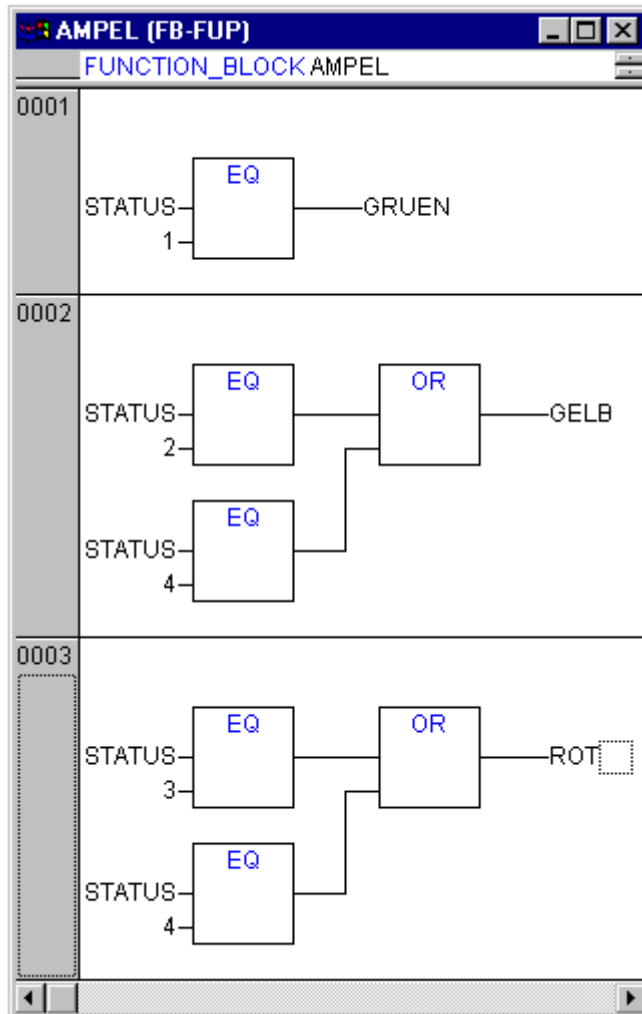


STATUS porównywany jest z 1, wynik przyporządkowany jest do wartości ZIELONE. Sieć przełącza się więc na ZIELONE, gdy zdefiniowana wartość statusu wyniesie 1.

W przypadku pozostałych kolorów sygnalizacji konieczne będą dwie dodatkowe sieci. Pierwszą z nich należy utworzyć za pośrednictwem polecenia „Wstaw” „Sieć (za)”, a następnie utworzyć moduł EQ

4 - Tworzenie niewielkiego programu

w opisany sposób. W przypadku wybrania wyjścia, należy wybrać polecenie „Wstaw” „Moduł” i zastąpić w nim określenie „AND” przez „OR”. Następnie należy ponownie wybrać wyjście modułu OR i przypisać je za pomocą polecenia „Wstaw” „Przyporządkowanie” do elementu ZOLTE. Następnie należy wybrać drugie wejście modułu OR, klikając myszą poziomą kreskę obok trzech znaków zapytania, aby zostało ono zaznaczone za pomocą kropkowanego prostokąta, i za pomocą polecenia „Wstaw” „Moduł” wstawić kolejny moduł EQ w opisany już sposób. Na koniec sieć powinna wyglądać w przedstawiony poniżej sposób:



Trzecią sieć najlepiej jest utworzyć kopiując i edytując drugą. W tym celu należy kliknąć pole sieci 002 i wybrać polecenia „Edycja” „Kopiuj” oraz „Edycja” „Wklej”. Skopiowana sieć zostanie wstawiona pod numerem 002 i otrzyma numer „003”. Następnie należy wyedytować wejścia i wyjścia zgodnie przedstawionym powyżej rysunkiem, klikając każdorazowo istniejące wpisy i wprowadzając żądaną wartość.

Tym samym pierwszy moduł jest już gotowy. SYGNALIZATOR steruje żądanym w danej chwili kolorem sygnalizacji, w zależności od wpisu wartości STATUS.

Podłączanie biblioteki standard.lib

W przypadku timera w module CZEKAJ niezbędny jest moduł z biblioteki standardowej. Za pomocą polecenia menu „Okno” „Zarządzanie biblioteką” należy otworzyć zarządzanie biblioteką. Należy wybrać polecenie „Wstaw” „Inna biblioteka”. Zostanie wyświetlone okno dialogowe, umożliwiające otwieranie plików. Z listy dostępnych bibliotek można wybrać standard.lib.

Deklaracja „CZEKAJ”

Należy teraz powrócić do modułu CZEKAJ. Powinien to być taki timer, za pomocą którego można odtworzyć długość każdej fazy sygnalizacji. Moduł zawiera w postaci zmiennej wejściowej zmienną

CZAS typu TIME i w wyniku zwraca wartość boolean, którą chcemy nazwać OK i która po upływie żądanego czasu powinna mieć wartość TRUE. Wartość tę ustawiamy wstępnie jako FALSE, poprzez umieszczenie jej na końcu deklaracji (ale przed średnikiem) „:= FALSE”.

Do wymienionych wyżej celów niezbędny jest moduł TP, pulasator. Ma on dwa wejścia (IN, PT) i dwa wyjścia (Q, ET). TP działa teraz w następujący sposób:

Dopóki ma on status IN FALSE, ET ma ustawienie 0, zaś Q FALSE. Gdy tylko IN przekaże wartość TRUE, czas w wyjściu ET może być liczony nawet w milisekundach. Gdy ET osiągnie wartość PT, zliczanie ET jest przerywane. Natomiast Q ma wartość TRUE, dopóki ET jest mniejsze od PT. Gdy tylko zostanie osiągnięta wartość PT, Q zwraca ponownie FALSE.

Aby móc przekształcić moduł TP na moduł CZEKAJ, należy utworzyć lokalną instancję TP. W tym celu należy zadeklarować lokalną zmienną ZAB (czas upłynął) typu TP (pomiędzy słowami kluczowymi VAR, END_VAR).

Część deklaracyjna CZEKAJ wygląda więc tak:

```

WARTEN [FB-AWL]
0001 FUNCTION_BLOCK WARTEN
0002 VAR_INPUT
0003     ZEIT:TIME;
0004 END_VAR
0005 VAR_OUTPUT
0006     OK:BOOL:=FALSE;
0007 END_VAR
0008 VAR
0009     ZAB:TP;
0010 END_VAR
0011

```

Szkielet „CZEKAJ”

W celu zastosowania żądanego timera, szkielet modułu należy zaprogramować w następujący sposób:

```

WARTEN [FB-AWL]
0001 FUNCTION_BLOCK WARTEN
0002
0003
0004 LD     ZAB.Q
0005 JMP    marke
0006
0007 CAL   ZAB(IN:=FALSE)
0008 LD   ZEIT
0009 ST   ZAB.PT
0010 CAL   ZAB(IN:=TRUE)
0011 JMP   ende
0012
0013 marke:
0014 CAL   ZAB
0015 ende:
0016 LDN   ZAB.Q
0017 ST   OK
0018 RET
0019

```

Najpierw wykonywane jest sprawdzenie, czy Q ma ustawioną wartość TRUE (a więc czy trwa odliczanie), w takim przypadku w przyporządkowaniu ZAB nie należy nic zmieniać, lecz wywołać blok funkcji ZAB bez wprowadzania (aby sprawdzić, czy czas już upłynął).

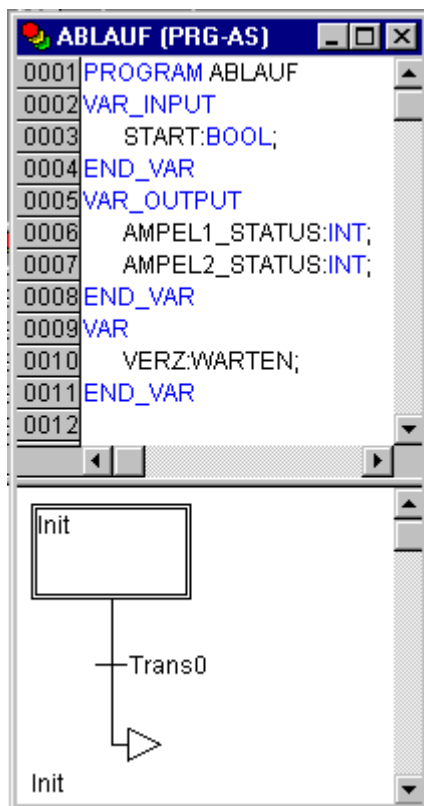
W przeciwnym razie dla zmiennej IN w ZAB należy ustawić wartość FALSE, a tym samym ET na 0 i Q na FALSE. W ten sposób został ustawiony żądany stan początkowy wszystkich zmiennych. Teraz należy zapisać wymagany czas ze zmiennej CZAS do zmiennej PT i wywołać ZAB z parametrem IN:=TRUE. W bloku funkcji ZAB zmienna ET jest zliczana, aż osiągnie wartość CZAS, wtedy Q otrzymuje wartość FALSE.

Zanegowana wartość Q zapisywana jest po każdym przebiegu CZEKAJ w OK. Gdy tylko Q uzyska wartość FALSE, dostarcza również OK TRUE.

Timer jest tym samym gotowy. Teraz należy połączyć ze sobą w programie PRZEBIEG oba bloki funkcji CZEKAJ i SYGNALIZATOR, dzięki czemu w żądany sposób sterowana będzie kolejność faz sygnalizacji.

„PRZEBIEG” pierwszego stopnia rozbudowy

Najpierw należy zadeklarować potrzebne zmienne. Są to: jedna zmienna wejściowa START typu BOOL, dwie zmienne wyjściowe AMPEL1_STATUS i AMPEL2_STATUS typu INT oraz jedna typu CZEKAJ (OPOZ jak opóźnienie). Program PRZEBIEG ma teraz taką postać:



Tworzenie wykresu języka AS

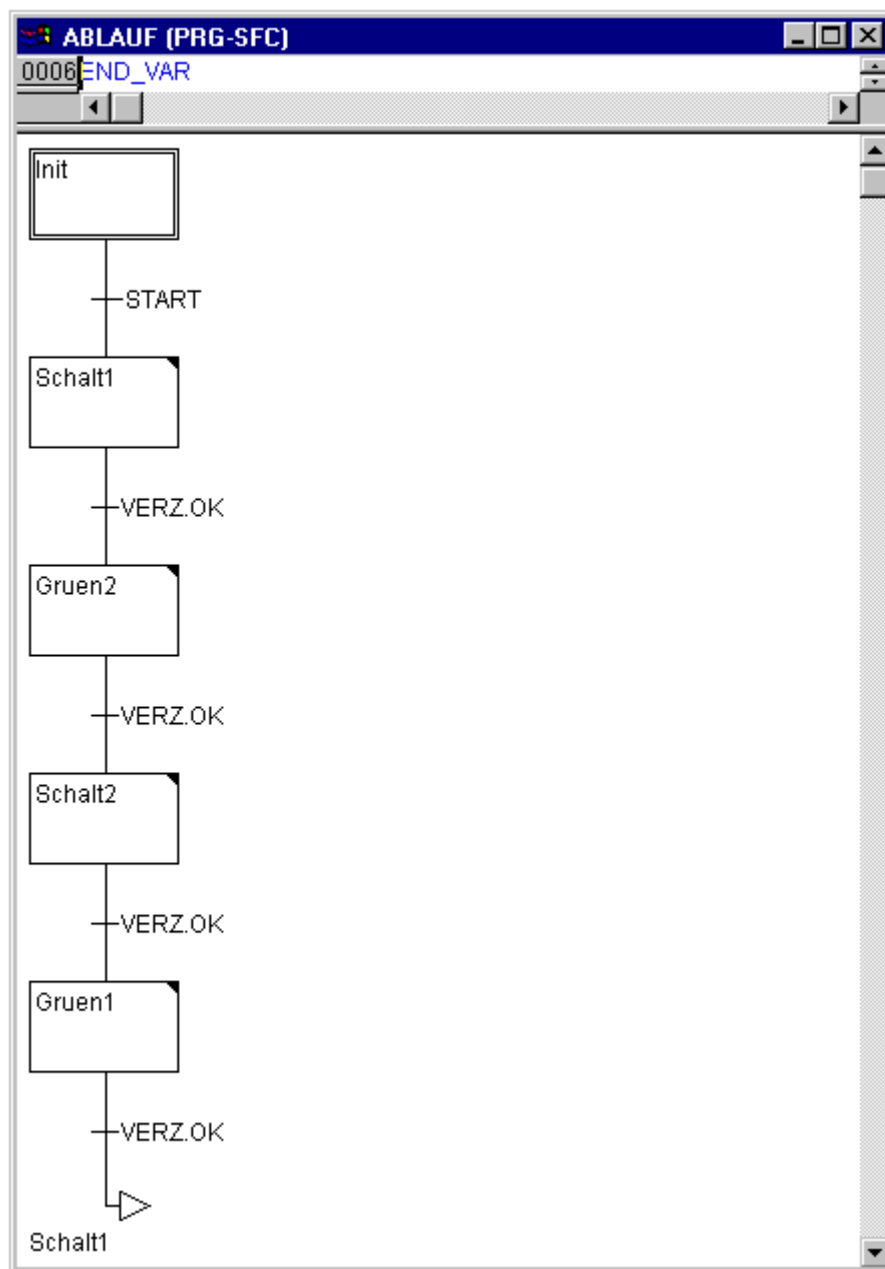
Wykres początkowy modułu w AS składa się zawsze z czynności „Init” następującego za nią przejścia „Trans0” oraz ze skoku powrotnego do Init. Należy go nieco rozszerzyć.

Przed zaprogramowaniem poszczególnych czynności i przejść należy najpierw ustalić strukturę wykresu. Po pierwsze do każdej fazy sygnalizacji potrzebny jest jeden krok. Należy go wstawić zaznaczając Trans0 i wybierając polecenie menu „Wstaw” „Przejście-krok (za)”. Czynność tę należy powtórzyć jeszcze trzy razy.

Klikając bezpośrednio nazwę przejścia lub kroku, można je zaznaczyć, a następnie zmienić. Pierwsze przejście po procedurze Init należy nazwać „START”, a wszystkie pozostałe „OPOZ.OK.”.

Przełączenie pierwszego przejścia następuje, gdy START ma wartość TRUE, wszystkich pozostałych natomiast wtedy, gdy OPOZ w OK przekazuje wartość TRUE, w więc gdy już upłynął wprowadzony czas.

Kroki zawierają (od góry do dołu) nazwy Przelacz1, Zielone2, Przelacz2, Zielone1, przy czym Init oczywiście zachowuje swą nazwę. „Przelacz” powinno oznaczać za każdym razem fazę żółtą, przy Zielone1 świeci się zielone światło SYGNALIZATORA1, w przypadku Zielone2 - SYGNALIZATORA2. Na koniec należy jeszcze zmienić adres skoku powrotnego z Init na Przelacz1. Jeśli wszystkie instrukcje zostały wykonane poprawnie, wykres powinien wyglądać w następujący sposób:



Teraz konieczne jest zaprogramowanie poszczególnych kroków. Klikając dwukrotnie pole danego kroku, można otworzyć okno dialogowe umożliwiające utworzenie nowej czynności. W tym przypadku jako język używana będzie IL (lista instrukcji).

Czynności i warunki przejścia

W czynności kroku **Init** inicjowane są zmienne, STATUS SYGNALIZATORA1 powinien mieć wartość 1 (zielone). Status SYGNALIZATORA2 powinien mieć wartość 3 (czerwone). Czynność Init ma wtedy następującą postać:

4 - Tworzenie niewielkiego programu

```
Aktion Init (AWL)
0001 LD 1
0002 ST AMPEL1_STATUS
0003 LD 3
0004 ST AMPEL2_STATUS
```

W przypadku **Przelaczn1** STATUS SYGNALIZATORA1 zmienia się na 2 (żółte), a status SYGNALIZATORA2 na 4 (żółte-czerwone). Ponadto ustalany jest też czas opóźnienia rzędu 2000 milisekund. Czynność ma wtedy taką postać:

```
Aktion Schalt1 (AWL)
0001 LD 2
0002 ST AMPEL1_STATUS
0003 LD 4
0004 ST AMPEL2_STATUS
0005 CAL VERZ(ZEIT:=t#2s)
```

W przypadku **Zielone2** SYGNALIZATOR1 jest czerwony (STATUS:=3), SYGNALIZATOR2 zielony (STATUS:=1), a ustawiony czas opóźnienia wynosi 5000 milisekund:

```
Aktion Gruen2 (AWL)
0001 LD 3
0002 ST AMPEL1_STATUS
0003 LD 1
0004 ST AMPEL2_STATUS
0005 CAL VERZ(ZEIT:=t#5s)
```

W przypadku **Przelaczn2** STATUS SYGNALIZATORA1 zmienia się na 4 (żółte-czerwone), a status SYGNALIZATORA2 na 2 (żółte). Ustalany jest też czas opóźnienia rzędu 2000 milisekund:

```
Aktion Schalt2 (AWL)
0001 LD 4
0002 ST AMPEL1_STATUS
0003 LD 2
0004 ST AMPEL2_STATUS
0005 CAL VERZ(ZEIT:=t#2s)
```

W przypadku **Zielone1** SYGNALIZATOR1 świeci na zielono (STATUS:=1), SYGNALIZATOR2 na czerwono (STATUS:=3), a ustawiony czas opóźnienia wynosi 5000 milisekund:

```
Aktion Gruen1 (AWL)
0001 LD 1
0002 ST AMPEL1_STATUS
0003 LD 3
0004 ST AMPEL2_STATUS
0005 CAL VERZ(ZEIT:=t#5s)
```

Na tym kończy się pierwsza faza rozbudowy przykładowego programu.

Aby przeprowadzić pierwszy test modułu PRZEBIEG w trybie symulacji, należy wykonać następujące czynności:

1. Otworzyć moduł PLC_PRG. Tu uruchamiany jest każdy projekt. Aby uruchomić w sposób prowizoryczny zaprogramowany już moduł PRZEBIEG, należy wstawić moduł, zaznaczyć w module opcję „AND” i zastąpić ją przez „PRZEBIEG”. Na początku wyjścia i wejścia należy pozostawić jako nieprzyporządkowane.

2. Kompilować projekt za pomocą poleceń „Projekt” „Kompiluj”. W oknie komunikatów poniżej okna roboczego powinien zostać wyświetlony komunikat „0 błędów, 0 ostrzeżeń”. Następnie należy wykonać polecenie „Online” „Zaloguj”, aby zalogować się do trybu symulacji (opcja „Online” „Symulacja” powinna być już włączona). Uruchomić program za pomocą poleceń „Online” „Start”. Otworzyć moduł PRZEBIEG, klikając dwukrotnie wpis „PRZEBIEG” w Organizерze obiektów. Program jest teraz wprowadzicie uruchomiony, jednakże do uruchomienia przebiegu sygnalizatora niezbędne jest jeszcze, aby zmienna START otrzymała wartość TRUE. Później otrzyma ona ją z PLC_PRG, w tym przypadku jednakże trzeba ją ustawić bezpośrednio w module. W tym celu należy w części deklaracyjnej modułu PRZEBIEG kliknąć dwukrotnie wiersz, w którym zdefiniowana jest zmienna START (START=FALSE). Wtedy za zmiennymi zostanie wyświetlona w kolorze turkusowym opcja „<:=TRUE>”. Następnie należy wybrać polecenie „Online” „Zapisz wartości”, aby ustawić tę wartość dla zmiennej. W wyniku tego zmienna START zostanie wyświetlona na wykresie przebiegu w kolorze niebieskim i można rozpoznać przetwarzanie poszczególnych kroków po niebieskim zaznaczeniu kroku aktualnego w danej chwili.

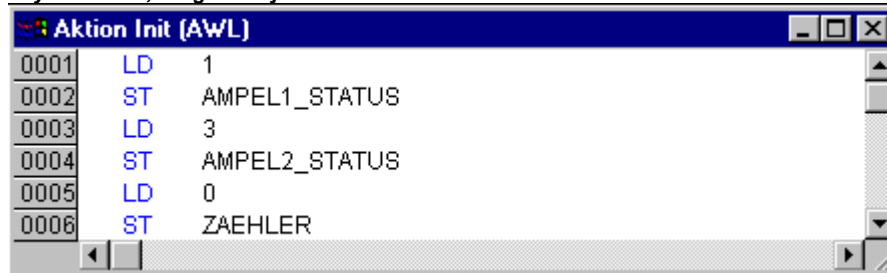
Tak można wykonać mały test pośredni. Następnie w celu opuszczenia trybu symulacji należy wykonać polecenia „Online” „Wyloguj” i można kontynuować programowanie.

PRZEBIEG drugiego stopnia rozbudowy

Aby w wykonanym wykresie znalazło się przynajmniej jedno odgańlenie alternatywne i aby możliwe było wyłączanie nocą projektowanej sygnalizacji świetlnej, do programu należy zaimplementować licznik, wyłączający instalację po określonej liczbie cykli sygnalizacji.

Na początku potrzebna jest więc nowa zmienna LICZNIK typu INT. Należy ją zadeklarować podobnie jak PRZEBIEG w części deklaracyjnej i zainicjować w InIt za pomocą wartości 0.

Czynność InIt, druga wersja

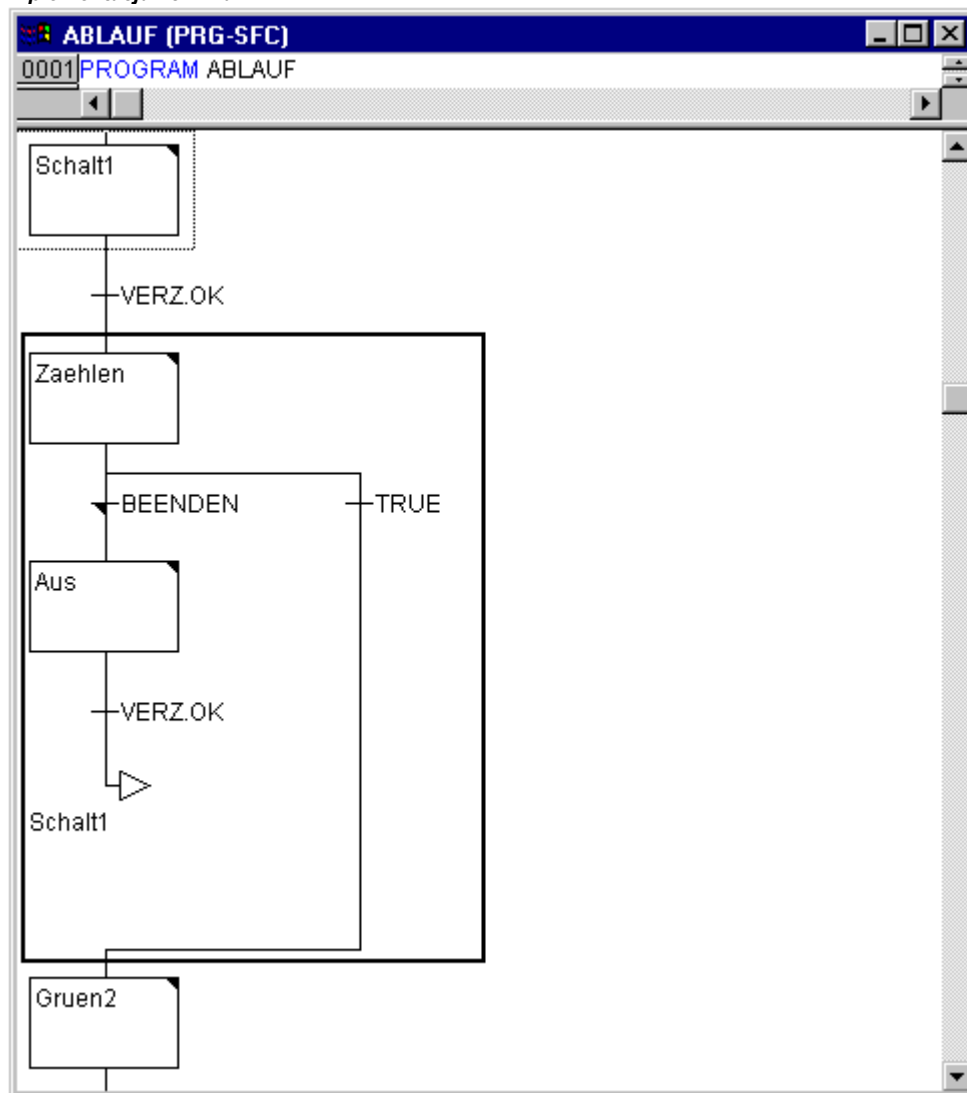


Adres	Operacja	Wartość / Zmienna
0001	LD	1
0002	ST	AMPEL1_STATUS
0003	LD	3
0004	ST	AMPEL2_STATUS
0005	LD	0
0006	ST	ZAEHLER

Następnie należy zaznaczyć przejście za krokiem Przelacz1 i wstawić za nim krok oraz przejście. Teraz należy zaznaczyć nowe przejście i wstawić odgańlenie alternatywne na lewo od niego. Za lewym przejściem należy wstawić krok oraz przejście. Następnie za nowym przejściem należy wstawić skok do kroku Przelacz1.

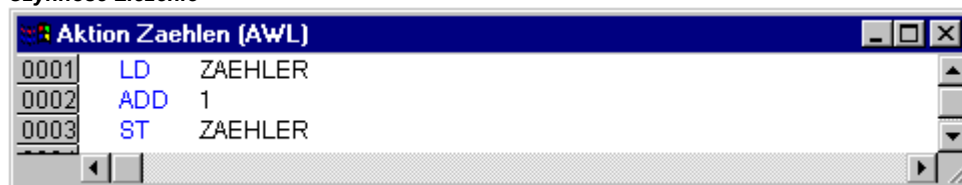
Powstałym elementom należy nadać następujące nazwy: Najwyższy z nowych kroków powinien nazywać się „Liczenie”, zaś najniższy „Wy.”. Przejścia nazywają się (o góry do dołu i z lewej do prawej) ZAKONCZ, TRUE i OPOZ.OK. Nazwa celu skoku „Step” zostanie zmieniona na „Przelacz1”. Nowo powstały element powinien wyglądać tak jak fragment oznaczony tu czarną ramką:

Implementacja licznika



Należy więc zaimplementować dwie nowe czynności i nowy warunek przejścia. Podczas kroku Liczenie LICZNIK zwiększa po prostu swą wartość o jeden:

Czynność Liczenie



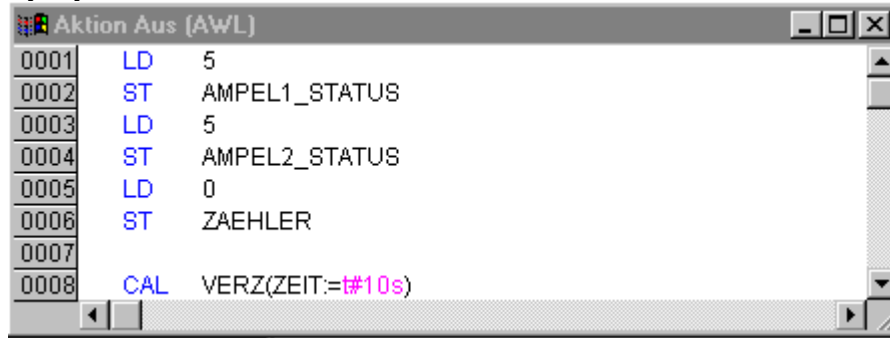
Przejście ZAKONCZ sprawdza, czy wartość licznika jest większa, niż określona liczba, np. 7:

ZAKONCZ przejście



W przypadku „Wy.” status obydwu sygnalizatorów ustawiany jest na wartość 5 (WY), (przy czym dla statusu tego można wybrać dowolną inną liczbę, różną od 1, 2, 3 i 4) licznik jest resetowany do wartości 0 i ustalany jest czas opóźnienia rzędu 10 sekund:

Wy. czynność



```

0001 LD 5
0002 ST AMPEL1_STATUS
0003 LD 5
0004 ST AMPEL2_STATUS
0005 LD 0
0006 ST ZAEHLER
0007
0008 CAL VERZ(ZEIT:=#10s)

```

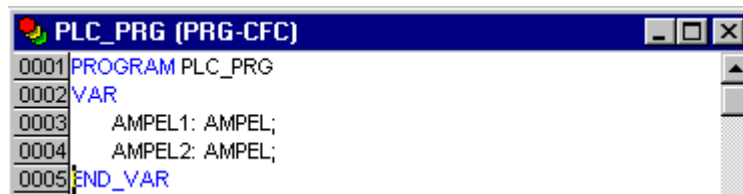
Wynik

W przypadku projektowanej instalacji sygnalizacyjnej po siedmiu cyklach następuje noc, sygnalizator wyłącza się na dziesięć sekund, potem następuje ponownie dzień, sygnalizacja się włącza i cykl rozpoczyna się od nowa. Przed przystąpieniem do tworzenia modułu PLC_PRG działanie to można również sprawdzić w opisany powyżej sposób w trybie symulacji.

PLC_PRG

W module PRZEBIEG został zdefiniowany i skorelowany przebieg czasowy faz obu sygnalizatorów. Ponieważ jednak sygnalizacja ma stanowić moduł systemu magistrali, np. CAN, należy teraz w module PLC_PRG ustawić odpowiednie zmienne wejściowe i wyjściowe na potrzeby komunikacji za pośrednictwem magistrali. Sygnalizacja ma być włączana za pomocą włącznika WŁ. i ma przesyłać w przypadku każdego kroku w module PRZEBIEG każdej z sześciu lamp (każdy sygnalizator: czerwone, zielone, żółte) odpowiednią „instrukcję sygnalizacji”. W przypadku tych sześciu wyjść i jednego wejścia należy następnie, jeszcze przed utworzeniem programu w edytorze, zadeklarować odpowiednie zmienne boolean i przypisać je od razu do odpowiednich adresów IEC.

W edytorze deklaracji PLC_PRG należy najpierw zadeklarować zmienne Sygnalizator1 i Sygnalizator2 typu Sygnalizator:



```

0001 PROGRAM PLC_PRG
0002 VAR
0003     AMPEL1: AMPEL;
0004     AMPEL2: AMPEL;
0005 END_VAR

```

Podczas każdego kroku w module PRZEBIEG przekazują one wartości logiczne dla każdej z sześciu lamp do wskazanych wyżej sześciu wyjść. Przewidzianych do tego celu sześciu zmiennych wyjściowych nie należy jednak jeszcze deklarować w obrębie modułu PLC_PRG, lecz jako dostępne globalnie w całym projekcie zmienne w obrębie zasobów „Zmienne globalne”. To samo dotyczy boolean zmiennej wejściowej WE., za pomocą której można ustawić wartość TRUE dla zmiennej START w module PRZEBIEG. Także zmiennej WŁ. przypisany zostanie adres IEC.

Należy więc wybrać zakładkę Zasoby i otworzyć listę Zmienne globalne.

Należy wykonać następujące deklaracje:



```

0001 VAR_GLOBAL
0002     EIN AT %IX0.0: BOOL;
0003     A1_gruen AT %QX0.0: BOOL;
0004     A1_gelb AT %QX0.1: BOOL;
0005     A1_rot AT %QX0.2: BOOL;
0006     A2_gruen AT %QX0.3: BOOL;
0007     A2_gelb AT %QX0.4: BOOL;
0008     A2_rot AT %QX0.5: BOOL;
0009 END_VAR

```

4 - Tworzenie niewielkiego programu

Za nazwą zmiennej (np. WE.) po AT następuje adres IEC, rozpoczynający się od znaku procent. I oznacza przy tym wejście, Q oznacza wyjście, X (stosowane w tym przykładzie) oznacza bajt, zaś 0.0 (0.1, 0.2 itp.) to wywołania poszczególnych bitów modułu. W przykładzie tym nie będzie wykonywana niezbędna konfiguracja sterownika, ponieważ zależy ona od tego, jakie jest dostępne wyposażenie plików konfiguracyjnych. W razie potrzeby należy przejść do rozdziału 9.6, Zasoby, Konfiguracja sterownika.

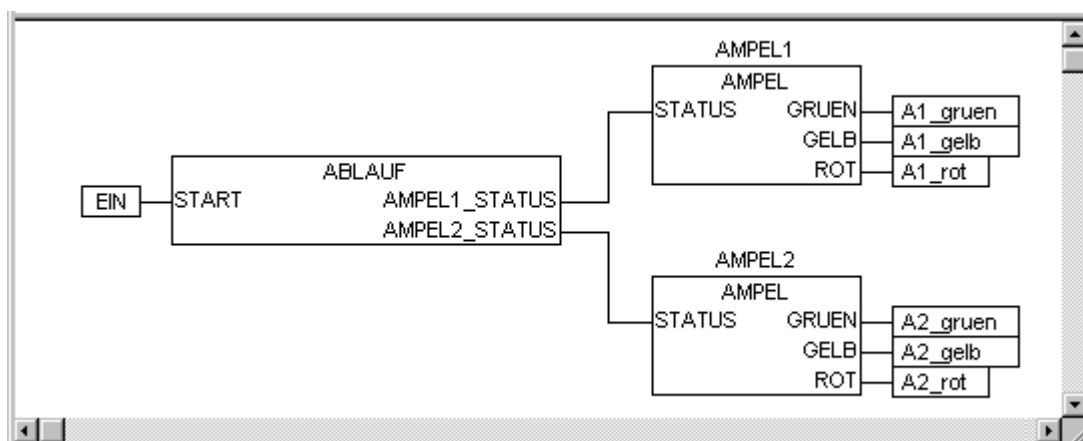
Kolejny etap stanowi wykonanie gotowego modułu PLC_PRG.

W tym celu należy przejść do okna edytora. Wybrany został graficzny edytor schematów funkcji CFC, dlatego też pod paskiem menu wyświetlany jest pasek symboli CFC wraz z dostępnymi elementami.

Należy kliknąć prawym przyciskiem myszy w oknie edytora i wybrać element **Moduł**. Następnie należy kliknąć tekst AND i zamiast niego wpisać „PRZEBIEG”. W wyniku tego zostanie wyświetlony moduł PRZEBIEG wraz ze zdefiniowanymi już zmiennymi wejściowymi i wyjściowymi. Teraz należy dodać dwa dodatkowe moduły, nazywając je SYGNALIZATOR. Sygnalizator to blok funkcji, dlatego nad modułem pojawią się trzy czerwone znaki zapytania, które należy zastąpić zmiennymi SYGNALIZATOR1 i SYGNALIZATOR2, zadeklarowanymi lokalnie powyżej. Następnie zaś należy ustawić element typu **Wejście**, który zostanie nazwany WE oraz sześć elementów typu **Wyjście**, które należy oznaczyć w zaprezentowany sposób nazwami WY1_zielone, WY1_zolte, WY1_czerwone, WY2_zielone, WY2_zolte, WY2_czerwone.

Zostały w ten sposób umieszczone wszystkie elementy programu i można połączyć ich wejścia i wyjścia, klikając przyciskiem myszy krótką linię przy wejściu lub wyjściu elementu i przeciągając ją do wejścia lub wyjściażądanego elementu.

Przykładowy program powinien mieć na koniec następującą postać:




Symulacja sygnalizatora

Teraz należy przetestować program. W tym celu konieczna jest ponowna kompilacja („Projekt” „Kompiluj wszystko”), wczytanie („Online” „Zaloguj”) i uruchomienie. W celu uruchomienia należy wykonać polecenie „Online” „Start” i ustawić dla zmiennej WE. wartość TRUE, np. poprzez dwukrotne kliknięcie w PLC_PRG wpisu „Wł.” w oknie wejściowym w edytorze CFC. W wyniku tego zmienna zostanie wyświetlona z zaznaczoną wartością <TRUE>. Aby ustawić tę wartość, należy następnie wybrać kombinację klawiszy <Ctrl><F7> lub wykonać polecenie „Online” „Zapisz”. Zmienna START w module PRZEBIEG, która w pierwszym stopniu rozbudowy programu wymagała ręcznego ustawienia wartości TRUE, otrzymuje teraz tę wartość ze zmiennej WE. z PLC_PRG. Tym samym rozpoczyna się cykl sygnalizacji. Okno modułu PLC_PRG zmieniło się w okno monitora. Dwukrotne kliknięcie symbolu plus w edytorze deklaracji powoduje rozwinięcie widoku zmiennych i można obserwować wartości poszczególnych zmiennych.

4.2 Wizualizacja sygnalizacji świetlnej...

Wizualizacja w programie CoDeSys umożliwia szybkie i proste przedstawienie zmiennych projektu w postaci graficznej. W dalszej części zostaną narysowane dla sygnalizacji świetlnej dwa sygnalizatory i włącznik, które będą obrazować proces przełączania.

Tworzenie nowej wizualizacji

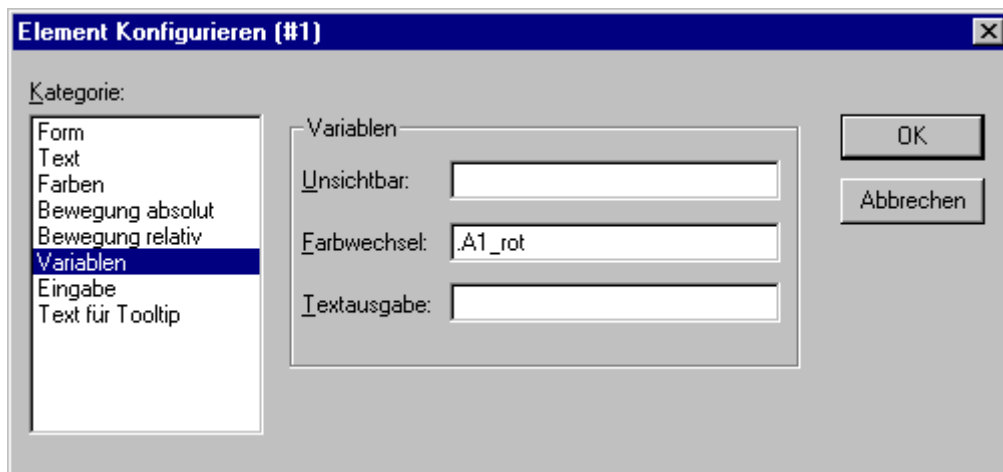
W celu utworzenia wizualizacji należy najpierw wybrać w Organizeryze obiektów obszar **Wizualizacja**. Należy w tym celu kliknąć przy dolnej krawędzi okna po lewej stronie, w którym znajduje się wpis **Moduły**, kartę o symbolu  i o nazwie **Wizualizacja**. Po wybraniu polecenia „Projekt” „Wklej obiekt” otworzy się okno dialogowe.

Należy wpisać w nim dowolną nazwę. W przypadku potwierdzenia okna dialogowego za pomocą polecenia **OK** zostanie otwarte okno, w którym można utworzyć nową wizualizację.

Wstawianie elementu do wizualizacji

W przypadku przykładowej wizualizacji sygnalizatora należy postępować wg poniższego schematu:

- Wybrać polecenie „Wstaw” „Elipsa” i spróbować narysować niezbyt duży okrąg (□2cm). Następnie należy kliknąć pole edycji i trzymając naciśnięty lewy przycisk myszy zmienić długość okręgu.
- Następnie należy dwukrotnie kliknąć okrąg. Zostanie wyświetlone okno edycji elementów wizualizacji.
- Teraz należy wybrać kategorię Zmienne i wpisać w polu zmiany kolorów tekst „WY1_czerwone” lub „.WY1_czerwone”. Oznacza to, że zmienna globalna WY1_czerwone powoduje zmianę koloru, gdy otrzyma wartość TRUE. Kropka przed nazwą zmiennej wskazuje, że chodzi o zmienną globalną, jej stosowanie nie jest jednakże konieczne:



- Następnie zaś należy wybrać kategorie Kolory i kliknąć przycisk Wewnątrz w obszarze Kolor. Należy wybrać w miarę możliwości neutralny kolor, np. czarny.
- Teraz należy kliknąć przycisk Wewnątrz w obszarze Kolor alarmu i wybrać kolor czerwony, który najbardziej odpowiada czerwonemu światłu sygnalizatora.



Powstały okrąg będzie wypełniony w stanie normalnym kolorem czarnym, a gdy zmienna CZERWONE z SYGNALIZATORA1 otrzyma wartość TRUE, kolor zmieni się na czerwony. W ten sposób zostało wykonane pierwsze światło pierwszego sygnalizatora.

Kolejne światła sygnalizatora

Należy użyć poleceń „Edycja” „Kopiuj” (<Ctrl> + <C>), a następnie „Edycja” „Wklej” (<Ctrl> + <V>). W ten sposób powstaną dwa kolejne okręgi tej samej wielkości, znajdujące się ponad pierwszym. Okręgi te można przesunąć, klikając je i przesuując w żądane miejsce przy naciśniętym lewym przycisku myszy. Odpowiednia pozycja to w tym przypadku ustawienie w rzędzie jeden nad drugim w lewej części okna edytora. Podwójnym kliknięciem jednego z obydwu okręgów można otworzyć ponownie okno dialogowe konfiguracji. W polu **Zmiana koloru** danego okręgu należy wprowadzić następujące zmienne:

dla środkowego okręgu: WY1_zolte

dla dolnego okręgu: WY1_zielone

W dalszej kolejności należy w kategorii **Kolory** oraz w obszarze **Kolor alarmu** wybrać dla okręgów odpowiednie kolory (żółty wzgl. zielony).

Obudowa sygnalizatora

Teraz należy wybrać polecenie „Wstaw” „Prostokąt” i tak jak okrąg wstawić prostokąt, mieszczący w sobie te trzy okręgi. Dla prostokąta należy również wybrać w miarę możliwości neutralny kolor, a następnie użyć polecenia „Dodatki” „Przenieś do tyłu”, aby okręgi stały się ponownie widoczne.

Jeśli tryb symulacji nie został jeszcze włączony, można go uaktywnić za pomocą polecenia „Online” „Symulacja”.

Po uruchomieniu symulacji za pomocą poleceń „Online” „Zaloguj” i „Online” „Start” można teraz śledzić zmianę kolorów w pierwszym sygnalizatorze.

Drugi sygnalizator

Najłatwiejszy sposób stworzenia drugiego sygnalizatora polega na skopiowaniu wszystkich składników pierwszego sygnalizatora. Należy w tym celu zaznaczyć wszystkie elementy pierwszego sygnalizatora i skopiować je (podobnie jak przedtem światła pierwszego sygnalizatora) za pomocą poleceń „Edycja” „Kopiuj” i „Edycja” „Wklej”. W odpowiednich oknach dialogowych wizualizacji należy wtedy zmienić tylko tekst „WY1” na „WY2” i wizualizacja drugiego sygnalizatora jest gotowa.

Włącznik

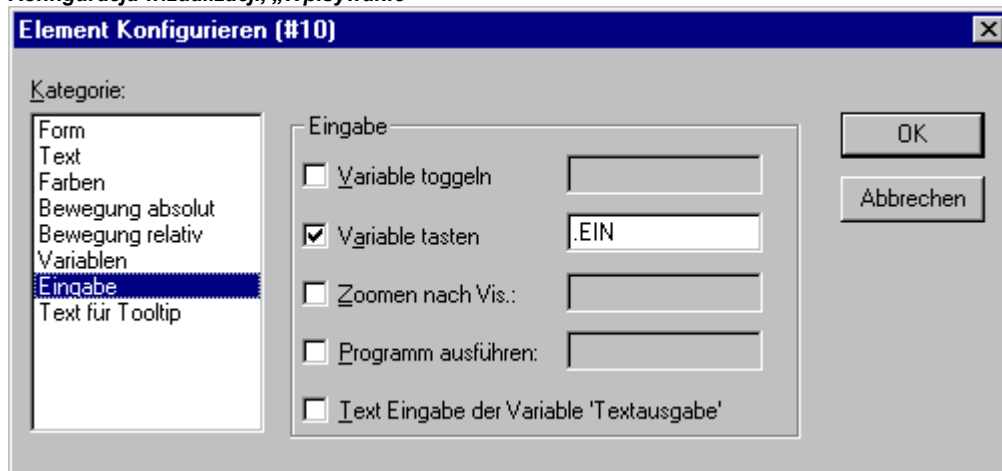
Należy wstawić prostokąt i nadać mu, podobnie jak w przypadku sygnalizatorów, dowolny kolor, a w opcjach **Zmienne** ustawić dla **zmiany kolorów** ".WŁ". W kategorii Tekst w opcji **Zawartość** należy w polu wprowadzania wpisać „WŁ”:



Aby za pomocą kliknięcia myszy móc nadawać zmiennej WŁ wartość TRUE, należy w kategorii **Wpisywanie** włączyć opcję Sondowanie zmiennej, a za nią wprowadzić zmienną .WŁ. Sondowanie

zmiennej oznacza, że w przypadku kliknięcia elementu wizualizacji zmienna `.Wł` otrzyma wartość `TRUE`, lecz po zwolnieniu przycisku ponownie powróci do stanu `FALSE` (w ten sposób można stworzyć prosty włącznik programu sygnalizacji).

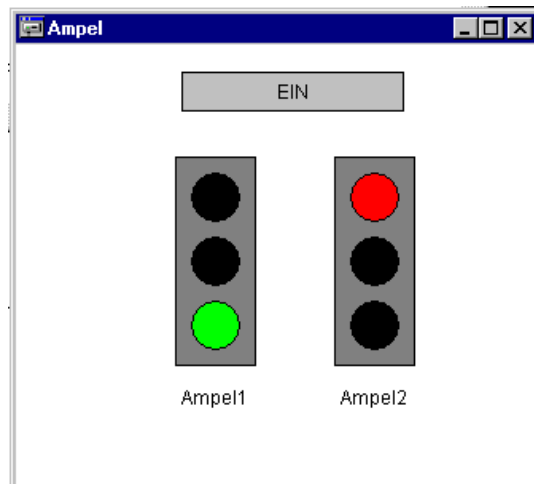
Konfiguracja wizualizacji, „Wpisywanie”



Czcionka w wizualizacji

W celu dopełnienia wizualizacji należy jeszcze wstawić dwa płaskie prostokąty, które powinny znaleźć się pod sygnalizatorami.

W oknie dialogowym wizualizacji należy każdorazowo w kategorii **Kolory** dla **Ramki** ustawić opcję „Bez koloru ramki” i w kategorii **Tekst** w polu **Zawartość** wpisać „Sygnalizator 1” wzgl. „Sygnalizator 2”. Wizualizacja będzie mieć wtedy następującą postać:



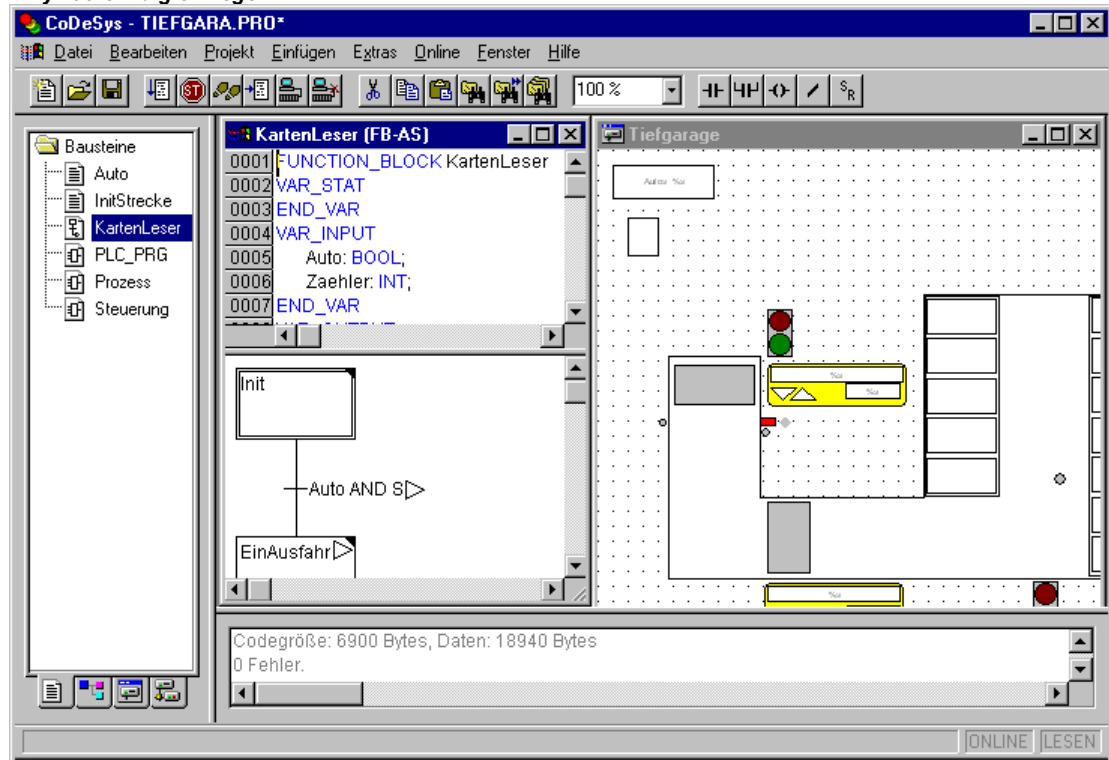
5

6 Szczegółowy opis składników

6.1 Okno główne...

Składniki okna głównego

Przykład okna głównego



Wymienione poniżej elementy znajdują się w oknie głównym programu CoDeSys (od góry do dołu):

- Pasek menu (wiele poleceń menu dostępnych jest także w menu kontekstowym, które można otworzyć prawym przyciskiem myszy).
- Pasek funkcji (opcjonalny); zawiera przyciski służące do szybkiego wykonywania poleceń menu.
- Organizator obiektów z kartami modułów, typów danych, wizualizacji i zasobów.
- Pionowy **separator ekranu** pomiędzy Organizерem obiektów a obszarem roboczym programu CoDeSys.
- Obszar roboczy, zawierający okna edytora (ich liczba nie jest ograniczona).
- Okno komunikatu (opcjonalne).
- Pasek stanu (opcjonalny); zawiera informacje na temat aktualnego statusu projektu.

Pasek menu

Pasek menu znajduje się przy górnej krawędzi okna głównego. Zawiera on wszystkie polecenia menu.

Datei Bearbeiten Projekt Einfügen Extras Online Fenster Hilfe

Pasek funkcji

Kliknięcie przyciskiem myszy symbolu z paska funkcji umożliwia szybsze wybranie polecenia menu. Wybór dostępnych symboli dostosowuje się automatycznie do aktywnego okna.





Po przytrzymaniu przez krótki czas kursora myszy nad symbolem na pasku funkcji wyświetlana jest nazwa symbolu w okienku etykiety.

W celu uzyskania opisu każdego symbolu na pasku funkcji należy wybrać w opcjach Pomocy edytor, którego mają dotyczyć informacje, a następnie kliknąć wybrany symbol na pasku funkcji.

Widok paska funkcji jest opcjonalny (patrz „Projekt” „Opcje” kategoria Obszar roboczy).



Organizer obiektów

Organizer obiektów znajduje się zawsze po lewej stronie programu CoDeSys. Poniżej widoczne są symbole czterech kart kategorii obiektów:  **Moduły**,  **Typy danych**,  **Wizualizacje** i  **Zasoby**. W celu przejścia pomiędzy poszczególnymi kategoriami obiektów należy kliknąć odpowiednią kartę lub też użyć lewego wzgl. prawego klawisza strzałki.

Dodatkowe symbole za lub przed wpisami obiektów oznaczają określone stany dotyczące zmian online oraz podłączenia ENI do bazy danych.

Sposób pracy z obiektami w Organizerze obiektów opisany został w rozdziale Zarządzanie obiektami.

Separator ekranu

Separator ekranu to granica pomiędzy dwoma nienakładającymi się oknami. W programie CoDeSys separator ekranu znajduje się pomiędzy Organizem obiektów a obszarem roboczym okna głównego, pomiędzy interfejsem (część deklaracyjna) a implementacją (część instrukcji) modułów oraz pomiędzy obszarem roboczym a oknem komunikatów.

Po umieszczeniu na separatorze ekranu kursora myszy, separator można przesuwac. Wystarczy przesunąć mysz, trzymając naciśnięty lewy przycisk.

Należy pamiętać o tym, że separator ekranu pozostaje stale na pozycji bezwzględnej, nawet w przypadku zmiany wielkości okna. Jeśli separatora ekranu nie widać i wydaje się, że go nie ma, wystarczy po prostu powiększyć okno.

Obszar roboczy

Obszar roboczy znajduje się po prawej stronie okna głównego programu CoDeSys. W obszarze tym otwierane są wszystkie edytory obiektów oraz zarządzania bibliotekami. Na pasku tytułu okna dialogowego wyświetlana jest zawsze nazwa obiektu, w przypadku modułów w nawiasach za nimi dodatkowo podawany jest skrót typu modułu oraz stosowany język programowania.

Pod punktem menu „Okno” znajdują się wszystkie polecenia, dotyczące zarządzania oknami.

Okno komunikatów

Okno komunikatów, oddzielone separatorem ekranu, znajduje się pod obszarem roboczym w oknie głównym.

Zawiera ono wszystkie komunikaty z ostatniego procesu kompilacji, weryfikacji i porównania. Można w nim wyświetlić również wyniki wyszukiwania i listę odsyłaczy.

Podwójne kliknięcie komunikatu w oknie komunikatów lub naciśnięcie klawisza <Enter> powoduje otwarcie edytora wraz z danym obiektem oraz zaznaczenie odpowiedniego wiersza obiektu. Za pomocą poleceń

„Edycja” „Następny błąd” i „Edycja” „Poprzedni błąd” można szybko przechodzić pomiędzy komunikatami o błędach.

Widok okna komunikatów jest opcjonalny (patrz „Okno” „Komunikaty”).

Pasek stanu

Na pasku stanu, znajdującym się przy dolnej ramce okna głównego programu CoDeSys, wyświetlane są informacje na temat aktualnego projektu oraz poleceń menu.

Jeśli informacja dotyczy danego elementu, pojęcie z prawej strony na pasku stanu wyświetlane jest przy użyciu czarnej czcionki, w przeciwnym razie czcionka jest szara.

Podczas pracy w trybie online słowo **Online** wyświetlane jest czarną czcionką, w przypadku pracy w trybie offline, czcionka zmieni się na szarą.

W trybie online na podstawie paska stanu można się zorientować, czy aktualnie aktywny jest tryb symulacji (**SIM**), czy jest przetwarzany program (**W TOKU**), czy został ustawiony punkt wstrzymania (**BP**) i czy zmienne są wymuszane (**WYMUSZAJ**).

W przypadku edytorów tekstowych wyświetlany jest numer wiersza i kolumny dla aktualnej pozycji kursora (np. **W.: 5, Kol.: 11**). Jeśli włączony jest tryb zastępowania, na pasku stanu wyświetlany jest na czarno wpis „**ZAS**”. Naciskając klawisz <Insert>, można przełączać pomiędzy trybem zastępowania i wprowadzania. Jeśli kursor myszy znajduje się w obrębie wizualizacji, aktualna **pozycja X** oraz **Y** kursora wyświetlana jest w pikselach w odniesieniu do górnego lewego rogu obrazu. Jeśli kursor myszy znajduje się nad **elementem** lub jeśli element jest edytowany, wyświetlany jest jego numer. W przypadku wybrania elementu do wstawienia, zostanie on również wyświetlony (np. **prostokąt**).

Jeśli wybrane zostało polecenie menu, które nie zostało jeszcze uruchomione, na pasku stanu ukazuje się krótki opis.

Widok paska stanu jest opcjonalny (patrz „Projekt” „Opcje” kategoria Obszar roboczy).

Menu podręczne

Kombinacja klawiszy: <Shift> + <F10>

Zamiast używania w celu wykonania polecenia paska menu, można użyć prawego przycisku myszy. Wyświetlone w ten sposób menu zawiera najczęściej stosowane polecenia, dotyczące zaznaczonego obiektu lub też aktywnego edytora. Wybór dostępnych poleceń dostosowuje się automatycznie do aktywnego okna.

6.2 Opcje projektu...

Ustawienia w kategorii „Projekt” „Opcje” służą do innego konfigurowania widoku okna głównego CoDeSys. O ile nie zaznaczono inaczej, są one zapisywane w pliku „CoDeSys.ini”, a więc są odtwarzane wraz z każdym kolejnym uruchomieniem programu CoDeSys.

Przegląd opcji ustawionych dla projektu tworzony jest w zasobach w węźle „Obszar roboczy”.

Polecenie to umożliwi otwarcie okna dialogowego **Opcje**. Możliwości ustawień podzielone są na różne kategorie. Kliknięciem myszy lub też za pomocą klawiszy strzałek należy wybrać po lewej stronie okna dialogowego żądaną kategorię i zmienić opcje po prawej stronie.

Kategorie:	zapisane w CoDeSys	zapisane w projekcie
Wczytywanie i zapisywanie	X	
Informacje o użytkowniku	X	
Edytor	X	
Obszar roboczy	X	
Kolory	X	

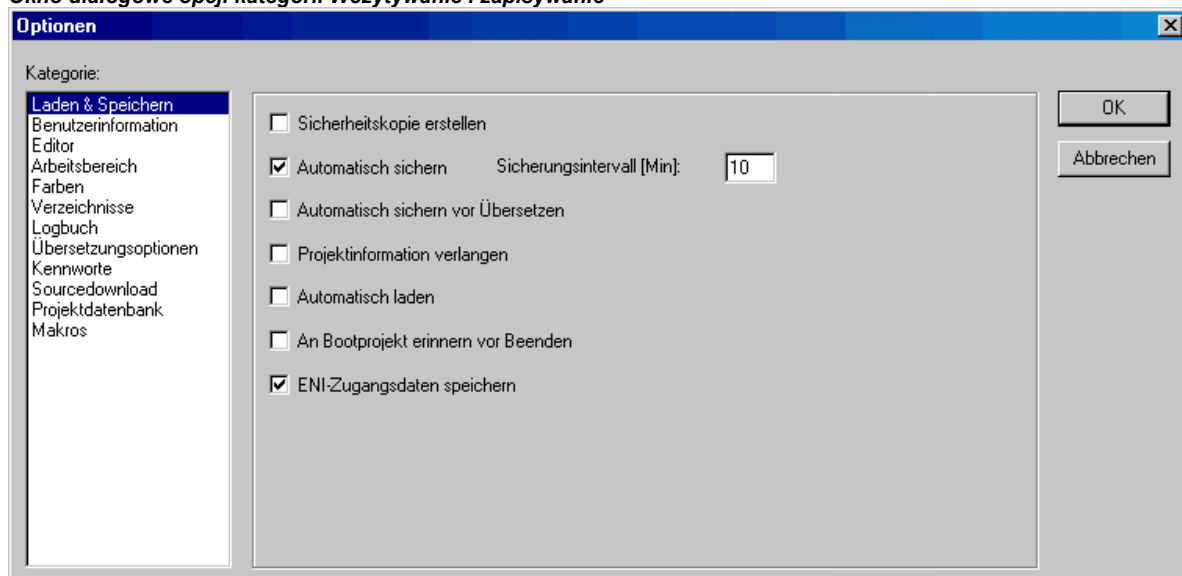
Opcje projektu...

Kategorie:	zapisane w CoDeSys	zapisane w projekcie
Katalogi	Kategoria Ogólne	Kategoria Projekt
Dziennik	X	
Opcje kompilacji		X
Hasła		X
Pobieranie źródła	X	
Konfiguracja symboli	X	
Baza danych projektu		X
Makra		X

Opcje wczytywania i zapisywania

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje” zostanie wyświetlone

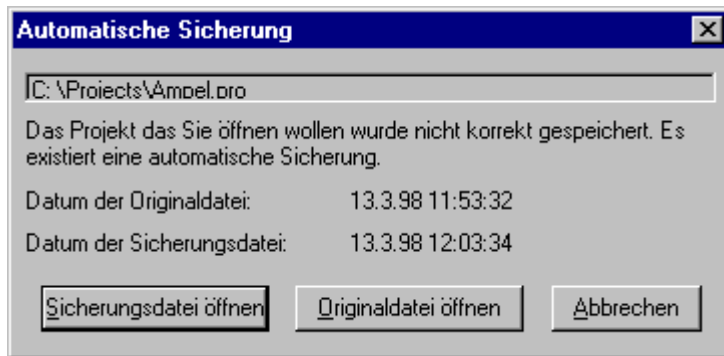
Okno dialogowe opcji kategorii Wczytywanie i zapisywanie



W przypadku włączenia opcji, przed opcją wyświetlany jest znacznik.

Tworzenie kopii zapasowej: Program CoDeSys zapisuje w przypadku każdego polecenia „Plik” „Zapisz” stary plik w pliku kopii zapasowej o rozszerzeniu „.bak”. Plik ten pozostaje w systemie również po zakończeniu projektu, w przeciwieństwie do pliku kopii zapasowej *.asd (patrz niżej, „Autom. kopia zapas.”). Można z niego również odtworzyć starszą wersję sprzed ostatniego zapisu.

Autom. kopia zapasowa: Otwarty projekt zapisywany jest w odstępach czasu ustawionych przez użytkownika (okres archiwizacji (min.)) w pliku tymczasowym z rozszerzeniem „.asd” w katalogu projektu. Plik ten jest usuwany podczas normalnego zamykania programu. Jeśli jednak program CoDeSys z jakiegóż powodu nie zostanie „normalnie” zamknięty (np. w przypadku awarii zasilania), plik nie zostanie usunięty. W przypadku kolejnego otwarcia takiego projektu zostanie wyświetlony następujący komunikat:



Można zdecydować, czy zostanie otwarty plik oryginalny, czy też kopia zapasowa.

Jeśli jako projekt otwarta jest biblioteka *.lib, zostanie utworzony odpowiedni plik zapasowy „*.asl”.

Autom. kopia zapasowa przed kompilacją: Projekt zapisywany jest przed każdą procedurą kompilacją. Tworzony jest przy tym plik z rozszerzeniem „.asd” wzgl. „.asl”, który ma takie same właściwości, jakie opisano w przypadku opcji „Autom. kopia zapas.”.

Żądaj informacji o projekcie: Podczas zapisywania nowego projektu lub też zapisywania istniejącego projektu pod nową nazwą, automatycznie wyświetlane jest okno dialogowe „Informacja o projekcie”. Informacje o projekcie można w każdej chwili wyświetlić i edytować za pomocą polecenia menu „Projekt” „Informacja o projekcie”.

Wczytaj automatycznie: Podczas następnego uruchamiania programu CoDeSys otwarty ostatnio projekt wczytywany jest automatycznie. Projekt można wczytać w trakcie uruchamiania programu CoDeSys, wpisując nazwę projektu w wierszu poleceń.

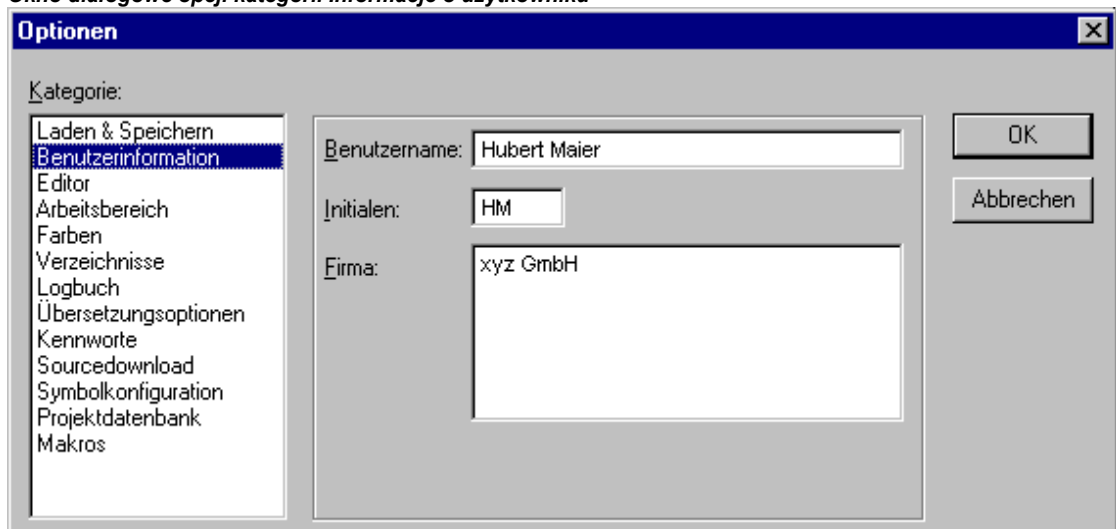
Przypominaj o proj. bootowalnym przed zakończeniem: Jeśli od czasu utworzenia projektu bootowalnego projekt został wczytany do sterownika w zmodyfikowanej postaci i nie utworzono nowego projektu bootowalnego, podczas wychodzenia z projektu wyświetlana jest informacja dla użytkownika: „Od operacji ostatniego pobierania nie utworzono żadnego projektu bootowalnego. Czy chcesz mimo to zakończyć?”

Zapisz dane dostępu do bazy projektu: Zapisywana jest nazwa użytkownika oraz hasło, wprowadzone ew. dla dostępu do bazy danych ENI. W przypadku danych dostępu, wprowadzonych w oknie „Otwórz projekt z bazy danych projektów” („Plik” „Otwórz”) nazwa użytkownika oraz hasło zapisywane są w pliku codesys.ini.

Opcje informacji o użytkowniku

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje”, zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe opcji kategorii Informacje o użytkowniku



Informacje o użytkowniku to: **Nazwa** użytkownika, jego **Inicjały** oraz **Firma**, w której pracuje. Każdy z wpisów można zmieniać. Dane te są przejmowane automatycznie w przypadku innych projektów, tworzonych w komputerze z zastosowaniem programu **CoDeSys**.

Opcje edytora

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje” zostanie wyświetlone przedstawione poniżej okno dialogowe:

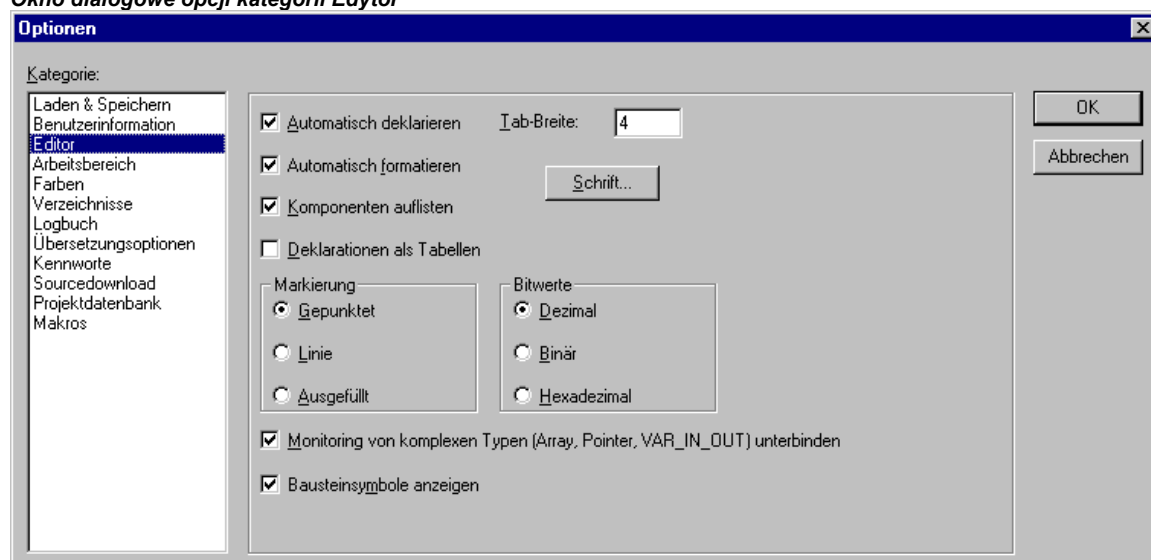
W przypadku włączenia opcji, przed opcją wyświetlany jest znacznik.

Możliwości ustawień edytorów są następujące:

Zadeklaruj automatycznie: We wszystkich edytorach po wprowadzeniu niezadeklarowanej jeszcze zmiennej wyświetlane jest okno dialogowe „Deklaracja zmiennych”, za pomocą którego można zadeklarować zmienną.

Formatuj automatycznie: Program CoDeSys wykonuje automatyczne formatowanie w edytorze list instrukcji oraz w edytorze deklaracji. Po wyjściu z wiersza przeprowadzane jest następujące formatowanie: 1. argumenty i słowa kluczowe zapisane małymi literami wyświetlane są wielkimi literami, 2. wstawiane są tabulatory, dzięki czemu powstaje jednolity podział kolumn.

Okno dialogowe opcji kategorii Edytor



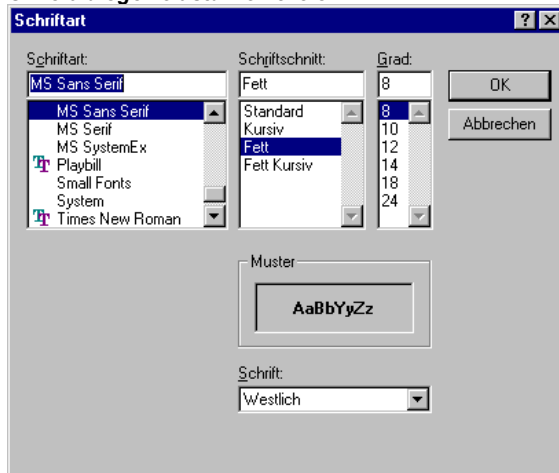
Wyświetl listę składników: Gdy opcja ta jest włączona, w programie CoDeSys dostępna jest „funkcja **Intellisense**”. W takim przypadku, jeżeli w miejscu, gdzie powinien zostać wprowadzony identyfikator, zostanie tylko wprowadzona kropka, zostanie wyświetlona lista wyboru wszystkich dostępnych w projekcie zmiennych globalnych. W przypadku wprowadzenia nazwy instancji bloku funkcji, po której następuje kropka, zostanie przedstawiona lista wyboru wejść i wyjść bloku funkcji danej instancji. Funkcja „Intellisense” dostępna jest w edytorach, w Administratorze list kontrolnych i instrukcji, w wizualizacji i w konfiguracji śledzenia.

Deklaracje jako tabela: Zamiast w zwykłym edytorze deklaracji zmienne można również edytować w postaci tabeli. Tabela ta ma właściwości segregatora, w którym znajdują się karty zmiennych wejściowych, wyjściowych lokalnych oraz zmiennych wejścia/wyjścia. W przypadku każdej zmiennej dostępne są pola **Nazwa**, **Adres**, **Typ**, **Inicjał** i **Komentarz**.

Szer. tabulatora: W tym miejscu można określić, jaką szerokość ma mieć tabulator, wyświetlany w edytorach. Domyślnie ustawiona jest szerokość czterech znaków, przy czym szerokość znaku zależy od ustawionego rodzaju czcionki.

Czcionka: Po naciśnięciu tego przycisku wyświetlane jest okno dialogowe „Rodzaj czcionki”. Tu należy wybrać cechy czcionki, jaka będzie używana we wszystkich edytorach CoDeSys. Rozmiar czcionki to podstawowa jednostka dla wszystkich operacji rysowania. Wybór większego rozmiaru czcionki powiększa również widok, a w rezultacie także wydruk w przypadku każdego edytora programu CoDeSys.

Okno dialogowe ustawień czcionki

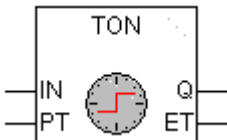


Zaznaczenie: Tu można wybrać, czy aktualne zaznaczenie w edytorach graficznych ma być wyświetlane w postaci kropkowanego prostokąta (**Kropkowanie**), za pomocą prostokąta o linii ciągłej (**Linia**) czy też prostokąta z wypełnieniem (**Wypełnienie**). Aktywny jest ten wybór, przed którym znajduje się kropka.

Wartości bitów: Tu można wybrać, czy binarne typy danych (BYTE, WORD, DWORD) mają być podczas monitorowania wyświetlane jako **Dziesiętne**, **Szesnastkowe** czy też **Binarne**. Aktywny jest ten wybór, przed którym znajduje się kropka.

Nie zezwalaj na monitoring typów złożonych (tablica, wskaźnik, VAR_IN_OUT): Jeśli opcja ta jest aktywna, złożone typy danych w rodzaju tablic, wskaźników i VAR_IN_OUT nie są wyświetlane w oknie monitora trybu online.

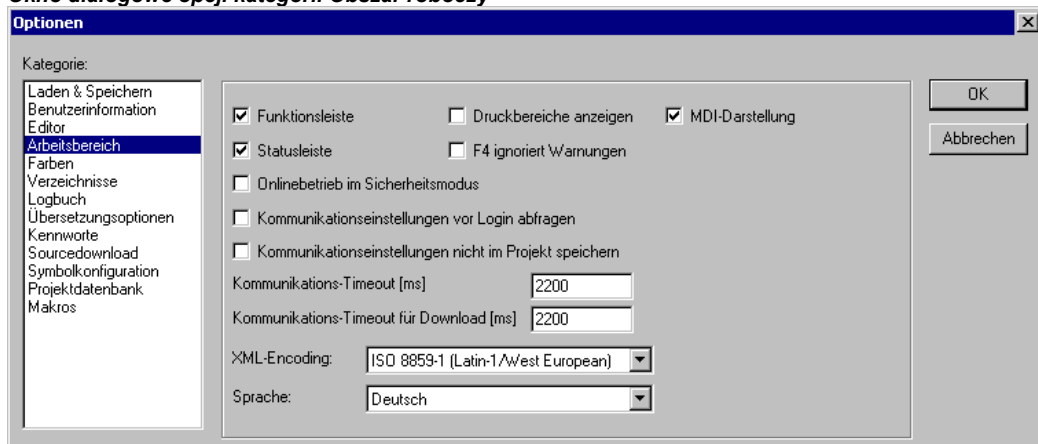
Pokazuj symbole modułów: Gdy opcja ta jest aktywna, w oknach modułów wyświetlane są symbole, o ile są one dostępne w katalogu biblioteki w postaci map bitowych. Nazwa pliku mapy bitowej musi składać się z nazwy modułu i rozszerzenia .bmp. Przykład: Dla modułu TON symbol znajduje się w pliku TON.bmp:



Opcje dot. obszaru roboczego

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje”, zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe opcji kategorii Obszar roboczy



Kliknięciem należy uaktywnić żądane opcje, w wyniku czego zostaną one zaznaczone za pomocą znacznika:

Opcje projektu...

Pasek funkcji: Pod paskiem menu wyświetlany jest pasek funkcji wraz z przyciskami, umożliwiającymi szybszy wybór poleceń menu.

Pasek stanu: Pasek stanu wyświetlany jest przy dolnej krawędzi okna głównego programu CoDeSys.

Tryb online z zabezpieczeniami: W trybie Online w przypadku poleceń „Start”, „Stop”, „Reset”, „2Punkt wstrzymania wł.”, „Cykl pojedynczy”, „Zapisz wartości”, „Wymuszanie wartości” i „Anuluj wymuszanie” wyświetlane jest okno dialogowe z zapytaniem o potwierdzenie, czy polecenie rzeczywiście ma zostać wykonane. Jeśli system czasu przebiegu oferuje taką możliwość, podczas wczytywania projektu do sterownika wyświetlane jest kolejne okno dialogowe: Wyświetlane są w nim dodatkowo informacje o projekcie już ew. zapisanym lub też wczytywanym na nowo do sterownika. Informacje o projekcie wyświetlane są również podczas tworzenia projektu bootowalnego, jeśli taki projekt znajduje się już w sterowniku. Opcja ta jest zapisywana wraz z projektem.

Sprawdzanie ustawień komunikacji przed zalogowaniem: Po wykonaniu poleceń „Online” „Logowanie” najpierw wyświetlane jest okno dialogowe parametrów komunikacji. Dopiero po jego zamknięciu za pomocą polecenia OK następuje przejście do trybu online.

Nie zapisuj danych komunikacji w projekcie: Ustawienia okna dialogowego parametrów komunikacji („Online” „Parametry komunikacji”) nie będą zapisywane w projekcie.

Pokaż obszary wydruku: W każdym oknie edytora zaznaczane są czerwoną przerywaną linią ograniczenia aktualnie ustawionego obszaru wydruku. Jego rozmiar zależy od właściwości drukarki (rozmiar papieru, orientacja) oraz od wielkości obszaru „zawartości” ustawionego szablonu drukowania.

Ignorowanie ostrzeżeń – F4: Po zakończeniu operacji kompilacji, po naciśnięciu klawisza F4 następuje przejście tylko do widoku wierszy z komunikatami o błędach, ostrzeżenia są ignorowane.

Wyświetlanie MDI: Domyślnie opcja ta (**M**ultiple-**D**ocument-**I**nterface) jest włączona, możliwe jest więc równoczesne otwieranie wielu obiektów (okien). Jeśli opcja ta zostanie wyłączona (tryb SDI, Single-Document-Interface), w obszarze roboczym można każdorazowo otworzyć tylko jedno okno, które jest wyświetlane w trybie pełnoekranowym. Wyjątek: Czynność programu może być wyświetlana w trybie MDI wraz z programem.

Limit czasu komunikacji [ms]: w przypadku standardowych usług komunikacyjnych: okres czasu w milisekundach, po upływie którego przerwana zostanie komunikacja z systemem docelowym (standardowe usługi komunikacyjne), jeśli nie zostanie stwierdzona dalsza aktywność. Możliwe wartości: 1-10000000. ms.

Limit czasu komunikacji dla pobierania [ms]: w przypadku dłuższych czynności komunikacyjnych (pobieranie programu, wysyłanie i pobieranie plików, tworzenie i sprawdzanie projektów bootowalnych): okres czasu w milisekundach, po upływie którego przerwana zostanie komunikacja z systemem docelowym, jeśli nie zostanie stwierdzona dalsza aktywność (czas oczekiwania na pobranie). Możliwe wartości: 1-10000000.

Kodowanie XML: Można wybrać format dla eksportu w postaci XML z programu CoDeSys. Ustawienie standardowe to ISO 8859-1. Dotyczy to komunikacji za pośrednictwem ENI, Message Interface i COM Automation Interface, jak również sterowanego przez użytkownika eksportu z programu CoDeSys. Wyjątek stanowi eksport XML Menedżera licencji.

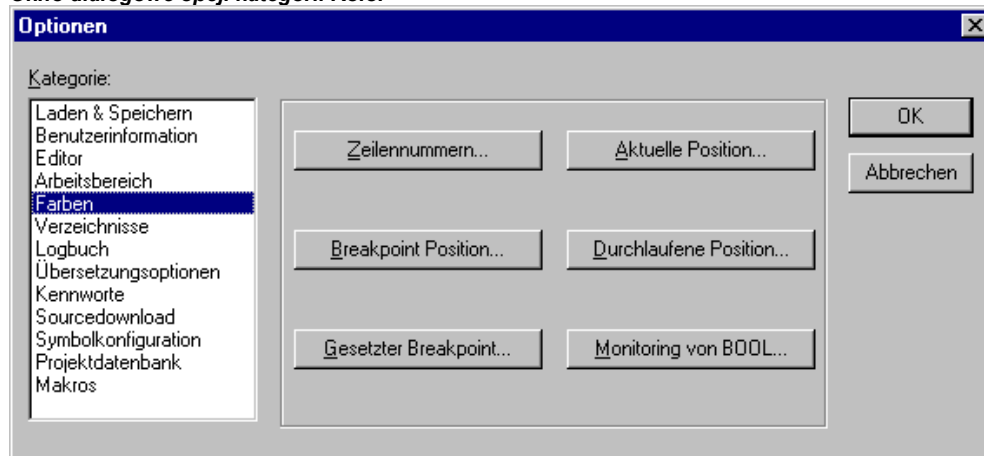
Język: Można wybrać, w jakim języku wyświetlane będą teksty menu, okien dialogowych oraz pomocy online.

Wskazówki: Należy pamiętać o tym, że wybór języka nie jest możliwy w systemie Windows 98!
Wprowadzone tu ustawienia zostaną zapisane dla obszaru roboczego w programie CoDeSys.

Opcje kolorów

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje”, zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe opcji kategorii Kolor



Tu można zmienić domyślne ustawienia kolorów programu CoDeSys dla opcji: **Numery wierszy** (domyślnie: jasnoszary), dla **pozycji punktów wstrzymania** (ciemnoszary), dla ustawionego **punktu wstrzymania** (jasnoniebieski), dla **aktualnej pozycji** (czerwony), dla **przechodnich pozycji** (zielony) lub dla **monitorowania wartości boolean** (niebieski).

Po wybraniu wymienionych przycisków zostanie wyświetlone okno dialogowe, umożliwiające wprowadzenie kolorów.

Okno dialogowe ustawień kolorów



Opcje katalogów

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje”, zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe opcji kategorii Katalogi



W obszarach **Projekt** oraz **Ogólne** można wpisywać katalogi, które program CoDeSys ma przeszukiwać w poszukiwaniu **bibliotek**, plików sterujących, **konfiguracyjnych** oraz **plików wizualizacji** (mapy bitowe, pliki XML dla tekstów dynamicznych itp.) wzgl. których ma używać do przechowywania plików **kompilacji** oraz źródłowych **plików do przesłania**. (Wskazówka: Pliki kompilacji to np. pliki .map oraz .list, ale nie pliki symboli! Te ostatnie zapisywane są w katalogu projektu.)

Naciśnięcie przycisku (...) za tym polem powoduje otwarcie okna dialogowego wyboru katalogu. Dla plików bibliotek oraz plików konfiguracyjnych można wprowadzać wiele ścieżek oddzielonych za pomocą średnika „;”.

Wskazówka: Ścieżki biblioteki można wprowadzać względnie wobec aktualnego katalogu projektu, poprzedzając je znakiem „.”. Wprowadzenie np. ciągu „.libs” powoduje wyszukiwanie bibliotek także w katalogu „C:\Programy\projects\libs”, o ile aktualny projekt znajduje się w katalogu „C:\Programy\projects”. Informacje na temat ścieżek dostępu do bibliotek zawiera rozdział 9.4, „Wstaw” „Następna biblioteka”.

Wskazówka: W nazwach ścieżek do katalogów nie należy używać spacji i znaków specjalnych z wyjątkiem „_”.

Dane w obszarze **Projekt** zapisywane są wraz z projektem, natomiast w obszarze **Ogólne** zapisywane są w pliku .ini systemu programowania i dotyczą wszystkich projektów.

W obszarze **System docelowy** wyświetlane są katalogi bibliotek i plików konfiguracyjnych, jakie są ustawione w systemie docelowym, np. poprzez informacje w pliku docelowym. Pola te są niedostępne do edycji, jednakże wpis można zaznaczyć i skopiować (menu podręczne prawego przycisku myszy).

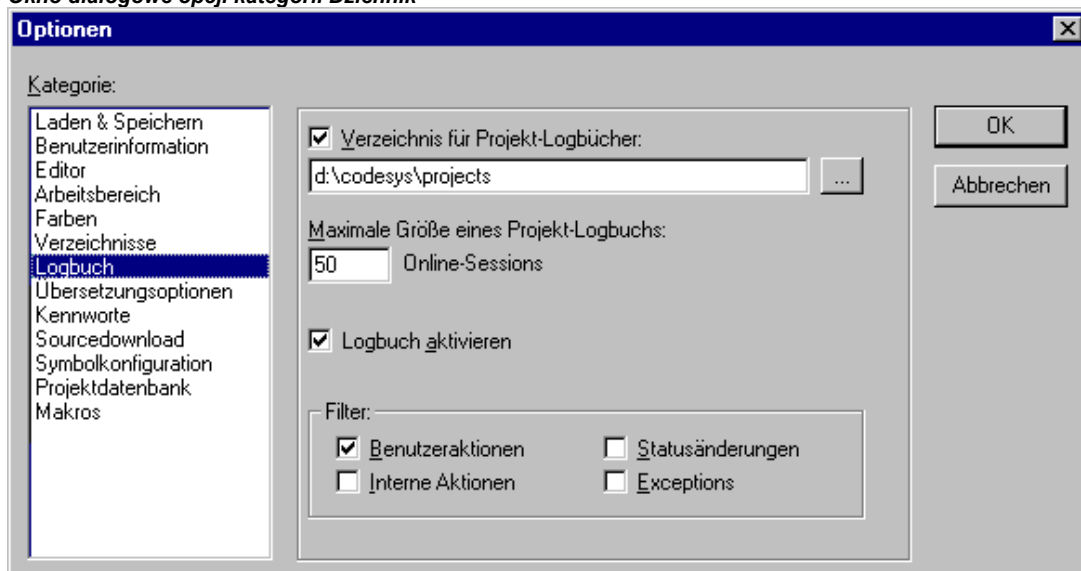
CoDeSys przeszukuje generalnie najpierw katalogi wpisane w obszarze „Projekt”, następnie w obszarze „System docelowy” (zdefiniowane w pliku docelowym), a na koniec w obszarze „Ogólne”. Jeśli znajdują się w nich pliki o tych samych nazwach, użyty zostanie plik z katalogu, który został przeszukany jako pierwszy.

Opcje dziennika

W przypadku wybrania w oknie dialogowym „Opcje” tej kategorii, zostanie wyświetlone przedstawione poniżej okno dialogowe: Można w nim skonfigurować plik, w którym jako w dzienniku projektu zapisywane będą chronologicznie podczas trybu online wszystkie czynności użytkownika i procesy wewnętrzne.

Jeśli zostanie otwarty istniejący już projekt, w przypadku którego nie utworzono jeszcze dziennika, zostanie otwarte okno dialogowe, przypominające o utworzeniu dziennika, w którym wpisy zapisywane będą począwszy od następnego logowania.

Okno dialogowe opcji kategorii Dziennik



Dziennik zapisywany jest automatycznie w postaci pliku binarnego w katalogu projektu podczas zapisywania projektu. Jeśli zajdzie potrzeba wybrania innego katalogu docelowego, można włączyć opcję **Katalog dziennika projektu:** i w polu edycji wpisać odpowiednią ścieżkę. Kliknięcie przycisku spowoduje wyświetlenie okna dialogowego „Wybierz katalog”.

Plikowi dziennika nadawana jest automatycznie nazwa projektu z rozszerzeniem .log. W polu **Maks. rozmiar dziennika projektu** można ustalić maks. ilość zapisywanych **sesji online**. Jeśli podczas zapisywania liczba ta zostanie przekroczona, w miejscu najnowszego wpisu usuwany jest każdorazowo wpis najstarszy.

Funkcję dziennika można włączyć lub wyłączyć za pomocą pola opcji **Włącz dziennik**.

W obszarze **Filtr** można wybrać rodzaj zapisywanych czynności. W oknie dziennika będą widoczne wzgl. w pliku dziennika będą zapisywane tylko czynności z kategorii oznaczonych w tym miejscu znacznikiem.

Okno dialogowe dziennika można otworzyć za pomocą polecenia „Okno” „Dziennik”.

Opcje kompilacji

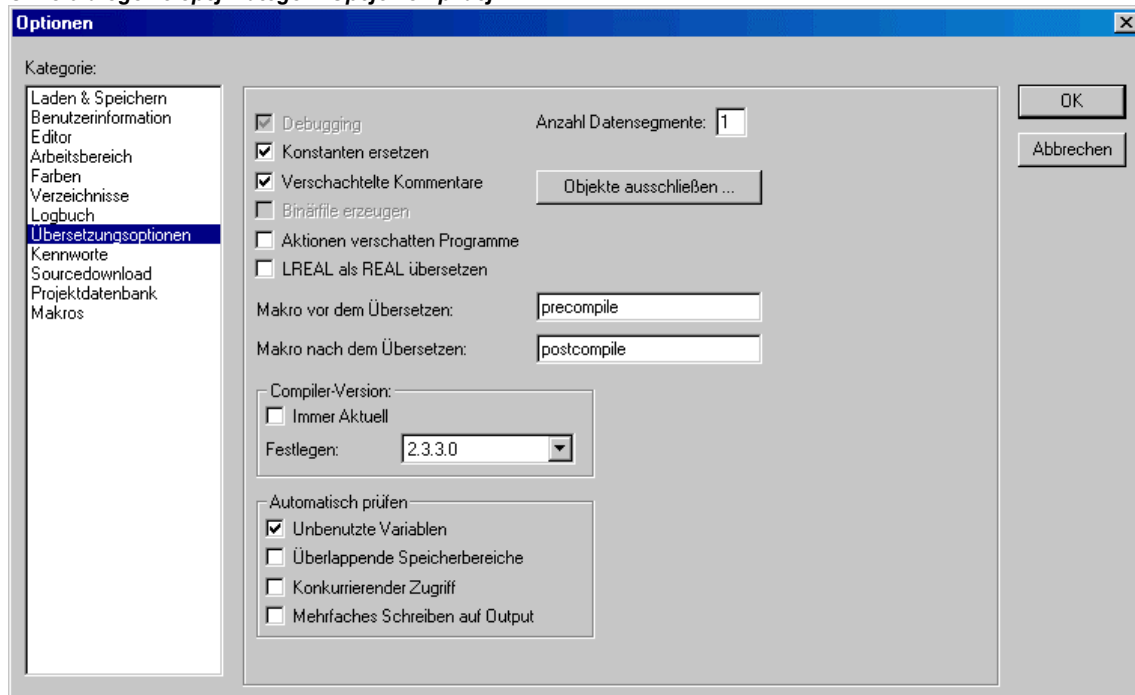
Ustawienia w tej kategorii okna dialogowego „Opcje” odnoszą się do kompilacji, a więc kompilacji projektu.

Debugowanie: Opcję tę można wybrać w zależności od systemu docelowego wzgl. jest ona ustawiona domyślnie. Jeśli jest ona włączona, dodatkowo generowany jest kod debugowania, tzn. kod może być znacznie bardziej obszerny. Kod debugowania jest potrzebny, aby możliwe było wykorzystanie oferowanych przez program CoDeSys funkcji debugowania (np. punktów wstrzymania). Po wyłączeniu tej opcji przetwarzanie projektu jest szybsze, zaś kod ma mniejszą objętość. Opcja ta jest zapisywana wraz z projektem.

Opcje projektu...

Zastap stałe: W przypadku każdej stałej typu skalarnego (a więc nie w przypadku ścieżek, tablic i struktur) ich wartość jest wczytywana bezpośrednio i w trybie online stałe wyświetlane są w kolorze zielonym. Tym samym niemożliwe jest wymuszanie, zapisywanie i monitoring stałej. Jeśli opcja ta jest wyłączona, wartość wczytywana jest poprzez dostęp zmiennej do miejsca w pamięci (umożliwia to wprowadzić zapis wartości zmiennej, oznacza jednakże dłuższy czas przetwarzania).

Okno dialogowe opcji kategorii Opcje kompilacji



Komentarze zagnieżdżone: Komentarze mogą być wprowadzane jako zagnieżdżone jedno w drugim. Przykład:

```
(*  
a:=inst.out; (* to be checked *)  
b:=b+1;  
*)
```

Komentarz rozpoczynający się od pierwszego nawiasu nie został w tym przypadku zamknięty po „checked”, lecz dopiero przez ostatni nawias.

Uwaga: Opcji tej należy zawsze używać ostrożnie: Jeśli ustawienie opcji w projekcie nie jest zgodne z używaną tam, utworzoną w programie CoDeSys biblioteką, wystąpią błędy kompilacji, które użytkownik może jedynie z trudem zinterpretować, a często także nie może ich usunąć!

Utwórz plik binarny: Podczas kompilacji w katalogu projektu tworzony jest binarny obraz wygenerowanego kodu (projekt bootowalny). Nazwa pliku: <projektname>.bin. Należy mieć również na uwadze możliwość zapisywania projektu bootowalnego oraz pliku wraz z odnośną sumą kontrolną online w sterowniku wzgl. offline w katalogu projektu za pomocą polecenia „Online” „Utwórz projekt bootowalny”.

Czynności przesłaniają programy: Opcja ta jest włączona domyślnie podczas tworzenia nowego projektu. Oznacza ona, że gdy lokalna czynność ma tę samą nazwę co zmienna lub program, podczas przetwarzania obowiązuje następująca hierarchia: zmienna lokalna przed czynnością lokalną przed zmienną globalną przed programem.

Uwaga: Jeśli zostanie otwarty projekt, utworzony przy użyciu wcześniejszej wersji programu CoDeSys, opcja ta jest domyślnie wyłączona. Zostanie więc zachowana hierarchia obowiązująca podczas tworzenia projektu (zmienna lokalna przez zmienną globalną przed programem przed czynnością lokalną).

Kompiluj LREAL jako REAL: Gdy opcja ta jest aktywna (dostępność uzależniona od systemu przebiegu; domyślnie: nieaktywna), podczas kompilowania projektu wartości LREAL traktowane są jak wartości REAL. Opcji tej można użyć do tworzenia projektów niezależnych od platformy systemowej.

Liczba segm. danych: Tutaj można ustalić, ile segmentów pamięci w sterowniku ma być zarezerwowanych dla danych projektu. Miejsce to jest niezbędne, aby możliwe było również przeprowadzenie zmian online, gdy dołączone zostaną nowe zmienne. Jeśli podczas kompilacji zostanie wyświetlony komunikat „Zajęta pamięć zmiennych globalnych.”, wprowadzoną tutaj liczbę należy powiększyć. Lokalne zmienne programowe są w tej relacji traktowane również jako zmienne globalne.

Wyklucz obiekty: Przycisk ten powoduje wyświetlenie okna dialogowego **Wykluczenie obiektów z kompilacji**. W przedstawionym tutaj drzewie należy wybrać te moduły projektu, które nie powinny być kompilowane podczas procesu kompilacji, a następnie uaktywnić opcję **bez kompilacji**. W strukturze drzewa wykluczone moduły zostaną wyświetlone w kolorze zielonym. Aby wykluczyć automatycznie wszystkie te moduły, które nie są w programie używane, należy nacisnąć przycisk **Wyklucz nieużywane**. Obiekt zaznaczony w Organizерze obiektów można ponadto również wyłączyć z kompilacji za pomocą polecenia „Wykluczenie z kompilacji”.

Wersja kompilatora: Tu można zdefiniować wersję kompilatora, jaka ma być użyta do procesu kompilacji. W programie CoDeSys od wersji V2.3.3 dostępne są zarówno aktualne, jak też dotychczasowe wersje kompilatora (dla każdej wersji/ każdego dodatku Service Pack / dla każdej łatki) wstecz do wersji V2.3.3. Jeśli zachodzi potrzeba, aby projekt był zawsze kompilowany przy użyciu najnowszej wersji kompilatora, należy uaktywnić opcję **Zawsze aktualna**. **W takim przypadku jednakże uwzględniane jest, czy projekt aktualnie otwarty jest w odpowiedniej wersji systemu programowania.** Jeśli jest inaczej, użyta zostanie wersja **kompilatora, odpowiadająca aktualnie używanemu systemowi programowania.** **W przypadku, gdy konieczne jest automatyczne kompilowanie z zastosowaniem określonej wersji,** można ją ustawić za pomocą pola wyboru **Ustal**.

Aby mieć wpływ na proces kompilacji, można wprowadzić dwa makra:

Makro w polu **Makro przed kompilacją** wykonywane jest przed procesem kompilacji, makro w polu **Makro po kompilacji** po nim. Nie można tu jednak wykonać następujących poleceń makr: file new, file open, file close, file saveas, file quit, online, project compile, project check, project build, project clean, project rebuild, debug, watchlist.

Sprawdzaj automatycznie:

W celu sprawdzenia poprawności semantycznej podczas każdego przebiegu kompilacji projektu można włączyć następujące opcje:

- Nieużywane zmienne
- Nakładające się obszary pamięci
- Konflikt dostępu
- Wielokrotny zapis na wyjściu

Wyniki przedstawiane są w oknie komunikatów. Procedury kontrolne można również uruchamiać wybiórczo za pomocą polecenia menu „Sprawdź” w menu „Projekt”.

Jeśli umożliwia to system docelowy, negatywne wyniki kontroli przedstawiane są jako błędy kompilacji.

Wskazówka: Wszystkie ustawienia, ustalone w oknie dialogowym „Opcje kompilacji”, zapisywane są razem z projektem.

Hasła

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje” zostanie wyświetlone przedstawione poniżej okno dialogowe:

Zabezpieczenie dostępu:

Plik projektu można zabezpieczyć przed niepowołanym dostępem poprzez zabezpieczenie jego otwierania i dokonywania zmian za pomocą hasła.

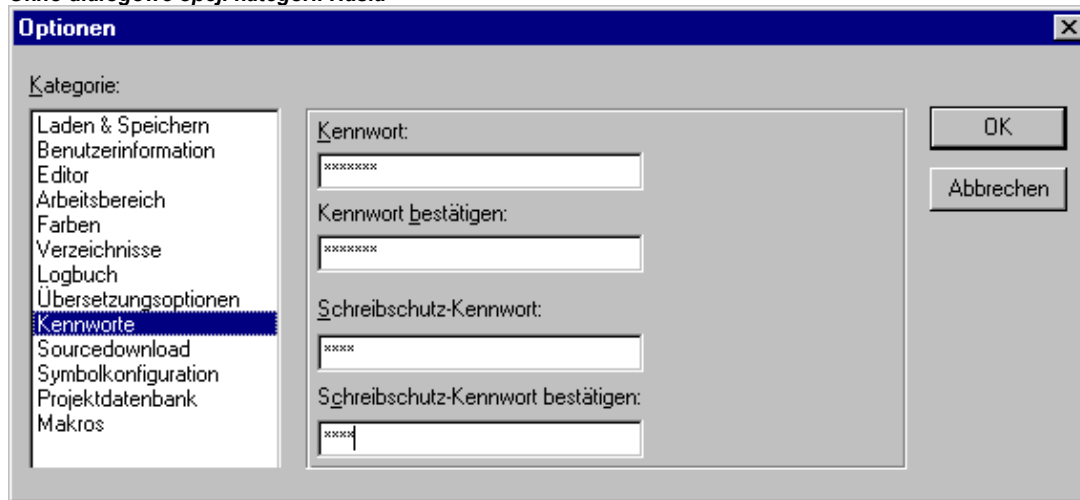
Żądane hasło należy wprowadzić w polu **Hasło**. Zamiast każdej wpisanej litery w polu wyświetlany jest symbol gwiazdki (*). To samo słowo należy wprowadzić w polu **Potwierdź hasło**. Okno należy zamknąć przyciskiem **OK**. Gdy zostanie wyświetlony komunikat:

Opcje projektu...

„Hasło oraz jego potwierdzenie nie są zgodne.”,

oznacza to, że w jednym z wpisów jest błąd. Dlatego też należy powtarzać oba wpisy tak długo, aż okno dialogowe zamknie się bez tego komunikatu.

Okno dialogowe opcji kategorii Hasła



Jeśli plik zostanie teraz zapisany i otwarty ponownie, zostanie wyświetlone okno dialogowe z żądaniem wprowadzenia hasła. Projekt zostanie otwarty tylko po wprowadzeniu prawidłowego hasła, w przeciwnym razie program CoDeSys wyświetli komunikat:

„Hasło jest nieprawidłowe.”

Zabezpieczenie przed zapisem:

Oprócz otwierania pliku hasłem można zabezpieczyć również zmiany pliku. W tym celu należy dokonać wpisu w polu **Hasło zabezpieczenia przed zapisem**, a następnie potwierdzić ten wpis w polu poniżej.

Projekt chroniony przed zapisem można otworzyć również bez hasła. W tym celu należy po wyświetleniu przez program CoDeSys żądania wprowadzenia hasła w trakcie otwierania pliku nacisnąć przycisk **Anuluj**. Można wtedy skompilować projekt, wczytać go do sterownika, przeprowadzić symulację itp., ale nie można go zmienić.

W przypadku zapomnienia hasła należy zwrócić się do producenta sterownika.

Hasła są zapisywane wraz z projektem.

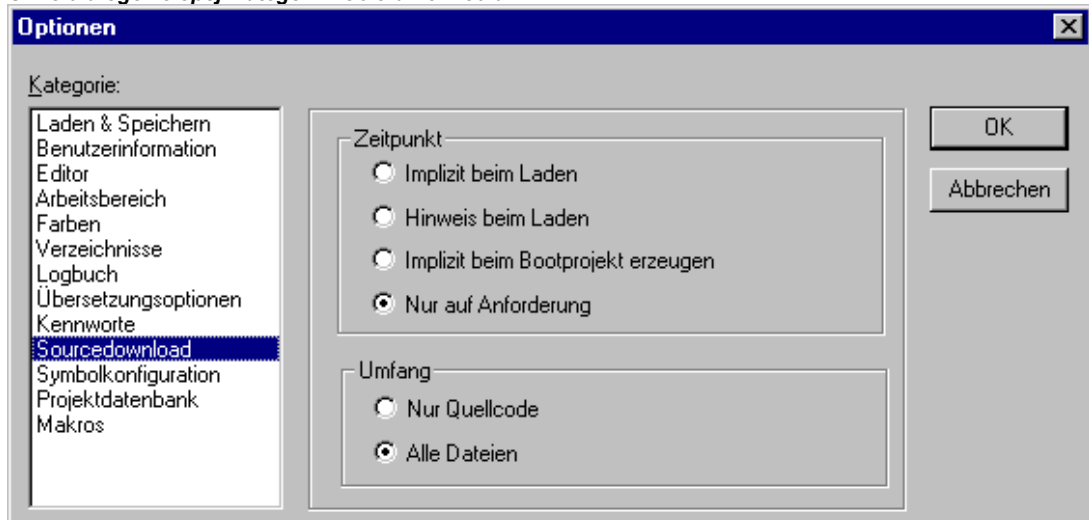
W celu utworzenia zróżnicowanych praw dostępu można ustalić grupy robocze („Projekt” „Obiekt Prawa dostępu” oraz „Hasła dla grupy roboczej”).

Należy ponadto pamiętać o rozszerzonej możliwości ochrony projektu poprzez **szyfrowanie** podczas zapisywania (patrz rozdz. 6.3, „Plik” „Zapisz jako”), które może służyć np do ochrony biblioteki przed użyciem bez wprowadzenia klucza.

Opcje pobierania źródła

W przypadku wybrania tej kategorii w oknie dialogowym „Opcje”, zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe opcji kategorii Pobieranie źródła



Można wybrać **czas** i **zakres** zapisywania kodu źródłowego projektu do sterownika. Dane zostaną w tym celu skompresowane.

Opcja **Tylko kod źródłowy** dotyczy wyłącznie pliku CoDeSys (rozszerzenie .pro).

Opcja **Wszystkie pliki** obejmuje dodatkowo pliki, takie jak np. odnośne biblioteki, mapy bitowe wizualizacji, pliki konfiguracyjne itp.

Opcja **Wewnętrznie podczas wczytywania** powoduje automatyczne wczytanie do sterownika wybranego zakresu danych po użyciu polecenia „Online” „Wczytaj”.

Opcja **Wskazówka podczas wczytywania** powoduje, że w przypadku polecenia „Online” „Wczytaj” wyświetlane jest okno dialogowe z pytaniem „Zapisać kod źródłowy w sterowniku?” Naciśnięcie przycisku **Tak**, spowoduje automatycznie wczytanie wybranego zakresu danych do sterownika, w przeciwnym razie należy wybrać **Nie**.

Opcja **Wewnętrznie przy projekcie bootowalnym** powoduje automatyczne wczytanie do sterownika wybranego zakresu danych przy użyciu polecenia „Online” „Utwórz projekt bootowalny”.

Opcja **Tylko na żądanie** sprawia, że wybrany zakres danych musi być jednoznacznie wczytany do sterownika za pośrednictwem polecenia „Online” „Wczytaj kod źródłowy”.

Projekt zapisany w sterowniku można wczytać ponownie za pomocą poleceń menu „Plik” „Otwórz” oraz „Wczytaj projekt ze sterownika”. Dane zostaną przy tym rozpakowane!

Opcje konfiguracji symboli

Udostępniane tu okno dialogowe służy do konfiguracji pliku symboli, który jest tworzony podczas kompilowania projektu. Plik symboli tworzony jest jako plik tekstowy <Nazwa projektu>.sym wzgl. binarny <Nazwa projektu>.sdb (format jest zależny od używanej wersji bramki) w katalogu projektu. Plik symboli jest niezbędny do wymiany danych ze sterownikiem za pomocą interfejsu symboli i używany jest do tego np. serwer OPC lub GatewayDDE.

Dostępne tu okno dialogowe (nie dostępne w trybie symulacji) służy do konfiguracji pliku symboli. Jest on tworzony jako plik tekstowy <Nazwa projektu>.sym wzgl. binarny <Nazwa projektu>.sdb (zależnie od używanej wersji bramki) w katalogu projektu. Plik symboli jest niezbędny do wymiany danych ze sterownikiem za pomocą interfejsu symboli i używany jest do tego np. serwer OPC lub GatewayDDE.

Jeśli wybrana została opcja **Utwórz wpisy symboli**, podczas każdej kompilacji projektu automatycznie tworzone są w pliku symboli wpisy symboli dla zmiennych projektu. W przeciwnym razie plik ten zawiera jedynie informacje o wersji pliku i projektu oraz sumę kontrolną.

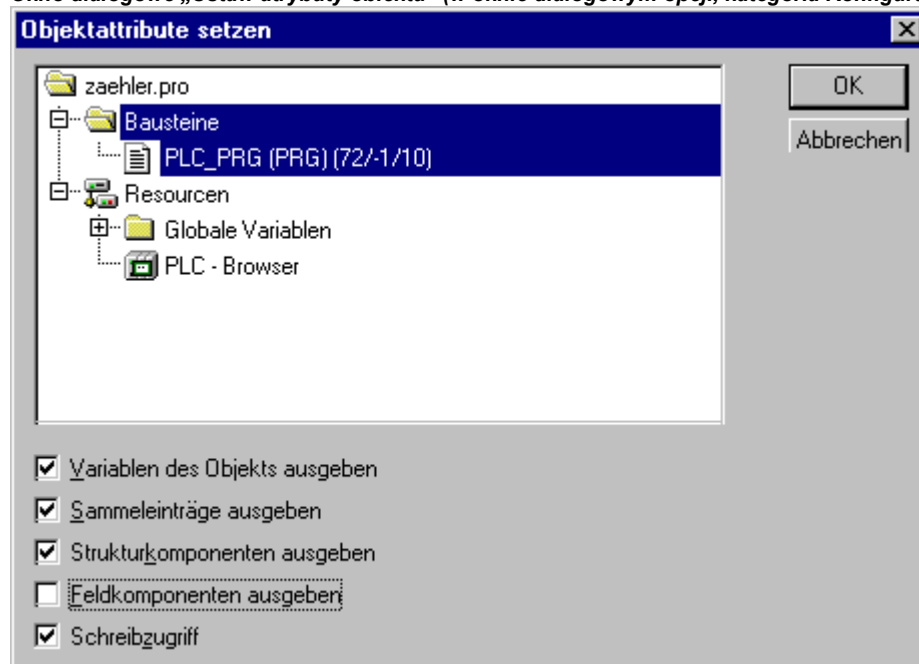
Jeśli włączona została ponadto opcja **Utwórz plik XML**, dodatkowo tworzona jest wersja XML pliku symboli. Jest ona generowana również w katalogu projektu i ma nazwę <Nazwa projektu>.SYM_XML.

W odniesieniu do konfiguracji wpisów w pliku symboli obowiązują poniższe zasady:

Opcje projektu...

- Jeśli opcja „Konfiguracja symboli z pliku INI” w ustawieniach systemu docelowego (plik odcelowy) jest aktywna, konfiguracja wpisów symboli wczytywana jest z pliku codesys.ini lub z innego wskazanego tam pliku .ini. (Okno dialogowe CoDeSys Atrybuty obiektu nie umożliwia w takim przypadku edycji.)
- Jeśli opcja „Konfiguracja symboli z pliku INI” nie jest aktywna, wpisy symboli tworzone są zgodnie z ustawieniami, jakie wprowadzono w oknie dialogowym „Ustaw atrybuty obiektu”. Można je utworzyć naciskając przycisk Konfiguruj plik symboli:

Okno dialogowe „Ustaw atrybuty obiektu” (w oknie dialogowym opcji, kategoria Konfiguracja symboli)



W strukturze drzewa katalogowego edytora wyboru należy wybrać zmienne, dla których utworzone zostaną wpisy symboli. Można w tym celu albo zaznaczyć moduły projektu, przez co automatycznie wybierane są odpowiednie zmienne, lub też można zaznaczyć poszczególne wpisy zmiennych. W odniesieniu do dokonanego wyboru można następnie ustawić żądane opcje poprzez kliknięcie przyciskiem myszy odpowiedniego pola w dolnej części okna dialogowego. Aktywne opcje oznaczone są znacznikiem. Możliwe jest ustawienie następujących opcji:

Utwórz zmienne obiektu: Zmienne wybranego obiektu zostaną umieszczone w pliku symboli.

Tylko wtedy, gdy aktywna jest opcja **Utwórz zmienne obiektu**, mogą być skuteczne następujące pozostałe opcje:

Utwórz wpisy zbiorowe: Dla struktur i tablic obiektu tworzone są wpisy, umożliwiające dostęp do wspólnych zmiennych.

Utwórz składniki struktury: Dla struktur obiektu tworzony jest odrębny wpis dla każdego składnika zmiennej.

Utwórz składniki pól: Dla tablic obiektu tworzony jest odrębny wpis dla każdego składnika zmiennej.

Dostęp do zapisu: Zmienne obiektu mogą być zmieniane przez serwer OPC.

Po wprowadzeniu ustawień opcji dla aktualnego wyboru zmiennych można – bez konieczności zamykania okna dialogowego za pomocą przycisku OK – wybrać inne moduły i również ustawić konfigurację opcji. Operację tę można wykonać kolejno w stosunku do dowolnej liczby wybranych modułów.

Jeśli okno dialogowe zostanie zamknięte za pomocą przycisku **OK**, zostaną zastosowane wszystkie zmiany konfiguracji, wprowadzone od momentu otwarcia okna dialogowego.

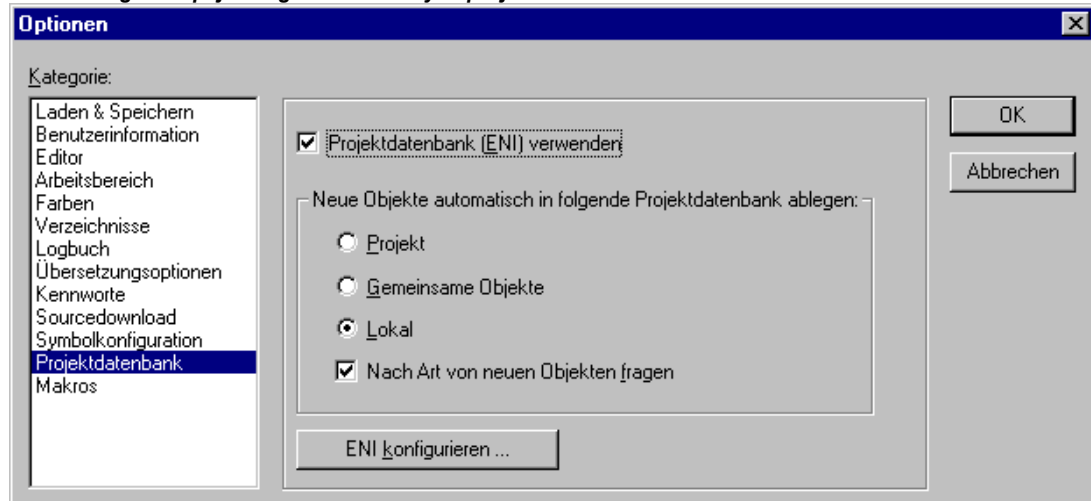
Wskazówka: Należy pamiętać o możliwości zastosowania za pomocą pragmy poszczególnych zmiennych bez prawa zapisu/odczytu lub też nawet o zaniechaniu ich przeniesienia do pliku symboli.

Opcje bazy danych projektu

W tym oknie dialogowym można ustalić, czy projekt ma być administrowany w bazie danych projektu oraz wprowadzić niezbędne w takim przypadku ustawienia interfejsu ENI.

Użyj bazy danych projektu (ENI): Opcję tę należy włączyć, jeśli dostęp do bazy danych projektu ma się odbywać za pośrednictwem serwera ENI, w celu zarządzania wszystkimi lub określonymi modułami projektu za pośrednictwem tej bazy danych. Warunkiem jest zainstalowanie serwera ENI oraz bazy danych projektu i zdefiniowanie uprawnionego użytkownika bazy danych. Należy w tym celu zapoznać się z dokumentacją serwera ENI wzgl. z informacjami z rozdziału 10, „Interfejs ENI programu CoDeSys”.

Okno dialogowe opcji kategorii Baza danych projektu



Jeśli opcja ta jest aktywna, w przypadku każdego obiektu w projekcie dostępne są funkcje (rejestrwanie, wywołanie itp.) bazy danych projektu. Z jednej strony pewne funkcje bazy danych są wtedy wykonywane automatycznie, o ile zostało to skonfigurowane w oknach dialogowych opcji, z drugiej zaś można również używać poleceń menu „Projekt” „Baza danych projektu” do wywoływania konkretnych funkcji. Ponadto w oknie dialogowym właściwości obiektu dostępna jest wtedy karta „Baza danych projektu”, za pomocą której można przypisać moduł do określonej kategorii bazy danych.

Zapisuj nowe obiekty automatycznie w następującej bazie danych projektu: Tu można wprowadzić ustawienia standardowe: W przypadku wstawienia nowego obiektu do projektu („Wklej obiekt”), jest on przypisywany automatycznie do jednej z ustawionych tu kategorii obiektów. Przyporządkowanie to jest widoczne we właściwościach obiektu („Projekt” „Obiekt” „Właściwości”) i można je tam zmieniać w odniesieniu do obiektu. Możliwe przyporządkowania:

Projekt: Moduł zapisywany jest w katalogu bazy danych, który został zdefiniowany w obszarze „Nazwa projektu” w oknie dialogowym ustawień ENI/obiektów projektu.

Obiekty wspólne: Administrowanie modulem odbywa się w katalogu bazy danych, który został zdefiniowany w obszarze „Nazwa projektu” w oknie dialogowym ustawień ENI/obiektów wspólnych.

Lokalnie: Administracja modulem nie odbywa się poprzez ENI w bazie danych projektu, lecz jest on wyłącznie zapisywany lokalnie w projekcie.

Oprócz kategorii „Obiekty projektu” oraz „Obiekty wspólne” dostępna jest trzecia kategoria bazy danych „Pliki kompilacji”, odnosząca się do obiektów, które powstają dopiero w trakcie kompilacji projektu, dlatego też ta kategoria nie jest tutaj istotna.

Pytaj o rodzaj nowego obiektu: Gdy opcja ta jest aktywna, podczas wstawiania nowego obiektu w projekcie otwiera się okno dialogowe „Obiekt” „Właściwości”, w którym można wybrać, do której spośród trzech wymienionych wyżej kategorii obiektów należeć ma moduł. W ten sposób można więc zastąpić ustawienie standardowe.

Konfiguruj ENI: Przycisk ten umożliwia dostęp do ustawień ENI, które wprowadzane są w trzech oknach dialogowych.

Opcje projektu...

Obiekty należące do projektu, które obsługiwane są w bazie danych, można przyporządkować do kategorii bazy danych „Obiekty projektu”, „Obiekty wspólne” lub też „Pliki kompilacji”. W przypadku każdej z tych kategorii, w kolejnych oknach dialogowych można ustalić w opcjach bazy danych projektu, w jakim katalogu bazy danych będą się znajdować i jakie ustawienia domyślne będą dotyczyć niektórych funkcji bazy danych:

- Okno dialogowe Konfiguracja ENI / Obiekty projektu
- Okno dialogowe Konfiguracja ENI / Obiekty wspólne
- Okno dialogowe Konfiguracja ENI / Pliki kompilacji

Należy pamiętać: Obiekty są w każdym przypadku zapisywane dodatkowo lokalnie, a więc razem z projektem.

Okna dialogowe wyświetlane są w przypadku pierwszej konfiguracji po kolei, przy czym użytkownikowi pomaga w konfiguracji **Kreator** (włączany przyciskiem **Dalej**). Ustawienia wprowadzone przy tym w pierwszym oknie dialogowym przejmowane są do dwóch pozostałych, dzięki czemu należy tylko wprowadzić żądane różnice.

Jeśli konfiguracja jest już dostępna, okna dialogowe zgrupowane są w postaci trzech kart w jednym oknie.

Jeśli przed wprowadzeniem konfiguracji nie została podjęta próba zalogowania do bazy danych (okno dialogowe logowania, wywoływane z poziomu menu „Projekt” „Baza danych projektu” „Logowanie”), okno dialogowe logowania zostanie otwarte w tym celu automatycznie.

Opcje obiektów projektu i obiektów wspólnych w odniesieniu do bazy danych projektu

Te okna dialogowe stanowią składnik ustawień opcji bazy danych projektu („Projekt” „Opcje” „Baza danych projektu”). Tu można zdefiniować, przy użyciu jakich parametrów dostęp odbywać się będzie w bazie danych zarządzanie obiektami kategorii „Projekt” oraz „Obiekty wspólne”. Obydwa okna dialogowe zawierają takie same punkty. (Trzecie okno dialogowe zawiera ustawienie dotyczące kategorii plików kompilacji.)

Połączenie ENI

Adres TCP/IP:	Adres komputera, na którym działa serwer ENI
Port:	Domyślnie: 80; musi zgadzać się z ustawieniem konfiguracji serwera ENI
Nazwa projektu:	Nazwa katalogu w bazie danych, w którym mają być przechowywane obiekty danej kategorii. Jeśli katalog istnieje już w bazie danych, można go wybrać w strukturze drzewa katalogowego projektów ENI, do którego dostęp umożliwia przycisk Jeśli jednak użytkownik nie został wcześniej zidentyfikowany jako użytkownik ENI za pośrednictwem okna dialogowego logowania, w przypadku naciśnięcia tego przycisku zostanie najpierw wyświetlone okno dialogowe logowania, w którym należy wprowadzić nazwę użytkownika oraz hasło dostępu ENI do trzech kategorii bazy danych.
Dostęp tylko do odczytu	Jeśli opcja ta jest aktywna, dostęp do zdefiniowanego tu katalogu bazy danych jest możliwy tylko w trybie do odczytu.

Okno dialogowe „Obiekty projektu” w opcjach kategorii Baza danych projektu

Przywołaj

Funkcja bazy danych Przywołaj (menu „Projekt” „Baza danych projektu”) oznacza, że aktualna wersja modułu z bazy danych zostanie skopiowana do otwartego lokalnie projektu, przy czym lokalna wersja zostanie zastąpiona. Procedura ta obejmie automatycznie wszystkie moduły zmienione w stosunku do lokalnej wersji projekt w każdym z punktów czasowych, który został uaktywniony (zaznaczony):

Przy otwieraniu projektu	Gdy projekt jest otwierany w programie CoDeSys
Natychmiast przy zmianach w ENI	Gdy w bazie danych zostanie zarejestrowana nowsza wersja modułu; moduł jest wtedy aktualizowany bezpośrednio w otwartym projekcie i wyświetlany jest odpowiedni komunikat
Przed każdą kompilacją	Przed każdym procesem kompilacji w programie CoDeSys

Wyrejestruj

Funkcja bazy danych Wyrejestruj oznacza, że moduł oznaczony zostanie jako będący „w edycji” i będzie zablokowany dla innych użytkowników, aż zostanie ponownie udostępniony poprzez ponowne zarejestrowanie lub też cofnięcie wyrejestrowania.

Jeśli jest aktywna opcja **Bezpośrednio przy rozpoczęciu zmiany**, wtedy wyrejestrowanie modułu następuje automatycznie, gdy tylko rozpocznie się jego edycja w projekcie. Jeśli obiekt został już wyrejestrowany przez innego użytkownika (co można rozpoznać po czerwonym krzyżyku przed nazwą obiektu w Organizercie obiektów), zostanie wyświetlony komunikat.

Opcje projektu...

Rejestrowanie

Funkcja bazy danych Rejestrowanie oznacza, że została utworzona nowa wersja obiektu w bazie danych. Starsze wersje zostaną zachowane. Możliwe punkty czasowe:

Przy zapisywaniu projektu	Jeśli opcja ta jest aktywna, każdy zmieniony moduł rejestrowany jest automatycznie podczas każdej operacji zapisywania projektu.
Po pomyślnej kompilacji	Jeśli opcja ta jest aktywna, po każdym bezbłędnym procesie kompilacji projektu rejestrowany jest każdy zmieniony obiekt.

W przypadku punktów Przywołaj, Wyrejestruj i Rejestrowanie można zawsze uaktywnić opcję zapytaj. W takim przypadku przed wykonaniem danej czynności wyświetlane jest okno dialogowe, w którym użytkownik musi ją jeszcze raz potwierdzić wzgl. anulować.

Punkty okna dialogowego „**Obiekty wspólne**” odpowiadają punktom opisanego wyżej okna „Obiekty projektu”. Ustawienia odnoszą się do wszystkich obiektów, które są przyporządkowane do kategorii bazy danych „Obiekty wspólne”.

Okna dialogowe wyświetlane są w przypadku pierwszej konfiguracji po kolei, przy czym użytkownikowi pomaga w konfiguracji **Kreator** (włączany przyciskiem **Dalej**). Ustawienia wprowadzone przy tym w pierwszym oknie dialogowym przejmowane są do dwóch pozostałych, dzięki czemu należy tylko wprowadzić żądane różnice.

Polecenie **Anuluj** powoduje zamknięcie okna dialogowego bez zapisywania dokonanych zmian. Jeśli zostanie zmieniona aktualnie dostępna konfiguracja opcji, nowe ustawienie (wszystkie trzy okna dialogowe) zapisywane jest za pomocą przycisku **OK** i następuje powrót do głównego okna dialogowego „Opcje” „Baza danych projektu”.

Opcje plików kompilacji w odniesieniu do bazy danych projektu

To okno dialogowe stanowi składnik ustawień opcji bazy danych projektu („Projekt” „Opcje” „Baza danych projektu”). Można w nim ustalić sposób zarządzania obiektami kategorii Pliki kompilacji w bazie danych. (Ponadto dostępne są dwa dalsze okna dialogowe, umożliwiające ustawienie opcji obiektów kategorii Projekt oraz kategorii Wspólne.)

Okno dialogowe Pliki kompilacji w kategorii Baza danych projektu

The screenshot shows a dialog box titled "Übersetzungsdateien" with a close button (X) in the top right corner. The dialog is divided into a main area with several input fields and a bottom area with three buttons. The input fields are: "ENI-Verbindung" (empty), "TCP/IP-Adresse:" (localhost), "Port:" (80), "Projektname:" (compile), "Benutzername:" (User1), and "Passwort:" (masked with asterisks). Below these fields are three checkboxes: "ASCII-Symbolinformation erzeugen (.sym)" (checked), "Binär-Symbolinformation erzeugen (.sdb)" (checked), and "Bootprojekt erzeugen" (unchecked). The bottom area contains three buttons: "< Zurück", "Fertig stellen", and "Abbrechen".

Informacje na temat pól wprowadzania **Adres TCP/IP**, **Port**, **Nazwa projektu**, patrz okno dialogowe Obiekty projektu / Obiekty wspólne.

Utwórz informacje o symbolu ASCII (.sym)	Jeśli opcja ta jest aktywna, plik symbolu *.sym (format tekstowy) wzgl. *.sdb (format binarny), zostanie zapisany w bazie danych zaraz po jego utworzeniu. Podczas generowania symboli obowiązują atrybuty obiektów ustawione w opcjach projektu w kategorii „Konfiguracja symboli”.
Utwórz binarne informacje o symbolu (.sdb)	
Utwórz projekt bootowalny	Jeśli opcja ta jest aktywna, projekt bootowalny zapisywany jest w bazie danych zaraz po jego utworzeniu.

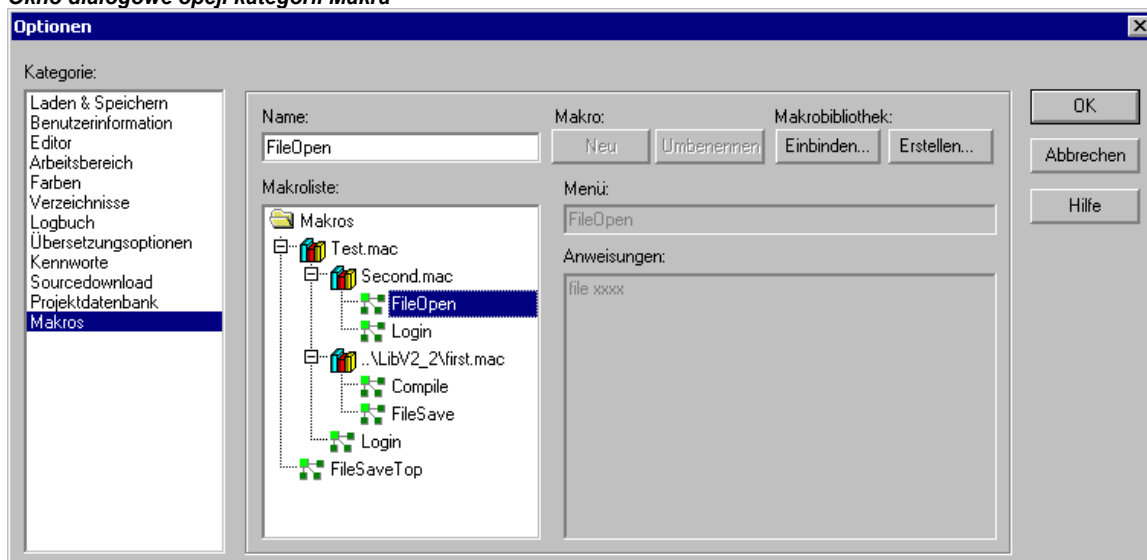
Okna dialogowe wyświetlane są w przypadku pierwszej konfiguracji po kolei, przy czym użytkownikowi pomaga w konfiguracji **Kreator** (włączany przyciskiem **Dalej**). Ustawienia wprowadzone przy tym w pierwszym oknie dialogowym przejmowane są do dwóch pozostałych, dzięki czemu należy tylko wprowadzić żądane różnice.

W przypadku naciśnięcia przycisku **Anuluj** następuje powrót do okna głównego, przy czym nie są zapisywane ustawienia karty „Pliki kompilacji”. (Pozostaną jednakże zachowane te ustawienia, które już zostały wprowadzone dla obiektów projektu i obiektów ogólnych.)

Opcje makr

W przypadku wybrania w oknie dialogowym „Opcje” tej kategorii, zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe opcji kategorii Makra



W tym oknie dialogowym można z poziomu poleceń pliku instrukcji mechanizmu przetwarzania wsadowego w programie CoDeSys zdefiniować makra, które będą następnie wywoływane w menu „Edycja” „Makra”.

W celu zdefiniowania nowych makr należy postępować w następujący sposób:

- W polu wprowadzania **Nazwa** wpisać nazwę tworzonego makra. Po kliknięciu przycisku **Nowy** nazwa ta zostanie umieszczona na **Liście makr** i tam zaznaczona. Lista makr ma strukturę drzewa. Utworzone lokalnie makra umieszczane są jedno pod drugim, ew. powiązane biblioteki makr (patrz niżej) wyświetlane są wraz z nazwą pliku biblioteki. Za pomocą znaku plus wzgl. minus przed nazwą biblioteki można otworzyć lub zamknąć listę elementów biblioteki.
- W polu **Menu** należy zdefiniować nazwę wpisu makra, pod którą makro zostanie umieszczone w menu „Edycja” „Makra”. W celu przypisania do danej litery skrótu, należy poprzedzić ją znakiem „&”. *Przykład:* nazwa „Ma&kro 1” tworzy wpis makra „Makro 1”.
- W polu edytora **Polecenia** należy wprowadzić następnie polecenia dla makra zaznaczonego na liście makr. Dozwolone są wszystkie polecenia mechanizmu przetwarzania wsadowego w programie CoDeSys oraz opisane w powiązaniu z nimi słowa kluczowe, ich listę można znaleźć naciskając przycisk **Pomoc**. Nowy wiersz polecenia wprowadza się naciskając klawisze

<Ctrl>Enter>. Prawym przyciskiem myszy można otworzyć menu podręczne z typowymi funkcjami edytora tekstowego. Powiązane ze sobą elementy polecenia można zgrupować za pomocą znaku cudzysłowu.

- Jeśli konieczne jest utworzenie dalszych makr, kroki od 1 do 3 należy wykonać ponownie przed zatwierdzeniem i zamknięciem okna dialogowego za pomocą przycisku OK.

Jeśli konieczne jest **usunięcie** makra, należy wybrać je na liście makr i nacisnąć klawisz .

W przypadku potrzeby zmiany nazwy makra, należy wybrać je na liście makr, w polu **Nazwa** wpisać nową nazwę i nacisnąć przycisk **Zmień nazwę**.

Aby **edytować** istniejące już makro, należy zaznaczyć je na liście makr i rozpocząć edycję w polach wprowadzania menu i/lub poleceń. Zmiany zostaną zastosowane po naciśnięciu przycisku OK.

Podczas opuszczania okna dialogowego za pomocą przycisku **OK** aktualny opis makra zapisywany jest w projekcie.

Wpisy menu makra wyświetlane są wtedy w kolejności ich definiowania w menu „Edycja” „Makra”. Sprawdzanie makra odbywa się dopiero podczas wykonywania polecenia menu.

Biblioteka makr:

Makra można zapisywać w zewnętrznych bibliotekach makr, które można potem podłączyć np. do innych projektów.

- Tworzenie biblioteki makr z makra aktualnego projektu:
Nacisnąć przycisk Utwórz. Zostanie wyświetlone okno dialogowe „Kopiowanie obiektów”, w którym znajduje się lista wszystkich dostępnych makr. Wybrane makra należy zaznaczyć i potwierdzić za pomocą przycisku OK. Okno wyboru zostanie zamknięte, a w jego miejsce zostanie otwarte okno dialogowe „Zapisz bibliotekę makrodefinicji”. Należy wpisać w nim nazwę i ścieżkę dla tworzonej biblioteki, a następnie nacisnąć przycisk Zapisz. Zostanie utworzona biblioteka o nazwie <nazwabiblioteki>.mac, a okno dialogowe zostanie zamknięte.
- Podłączanie biblioteki makr <nazwa biblioteki>.mac do aktualnego projektu.
Nacisnąć przycisk **Podłącz**. Zostanie wyświetlone okno dialogowe **Otwórz bibliotekę makrodefinicji**, w którym automatycznie wyświetlane są tylko pliki z rozszerzeniem *.mac. Należy wybrać żadaną bibliotekę i nacisnąć przycisk Otwórz. Okno dialogowe zostanie zamknięte, a biblioteka zostanie wyświetlona w strukturze drzewa listy makr.

Wskazówka: Makra z projektu można również eksportować („Projekt” „Eksportuj”).

6.3 Zarządzanie projektami...

Polecenia, które odnoszą się do całego projektu, dostępne są w punktach menu „Plik” oraz „Projekt”. Wskazówki na ten temat znajdują się w następnym rozdziale.

„Plik” „Nowy”

Symbol: 

Za pomocą tego polecenia można utworzyć pusty projekt o nazwie „bez_nazwy”. Nazwę tę należy zmienić podczas zapisywania.

„Plik” „Nowy z szablonu”

Za pomocą tego polecenia można otworzyć dowolny projekt, który zostanie użyty jako „szablon”, tzn. projekt nie musi być zapisany ze specjalnymi ustawieniami do tego celu. Zostanie wyświetlone okno dialogowe wyboru pliku projektu, który zostanie następnie otwarty z nazwą pliku „bez_nazwy”.

„Plik” „Otwórz”

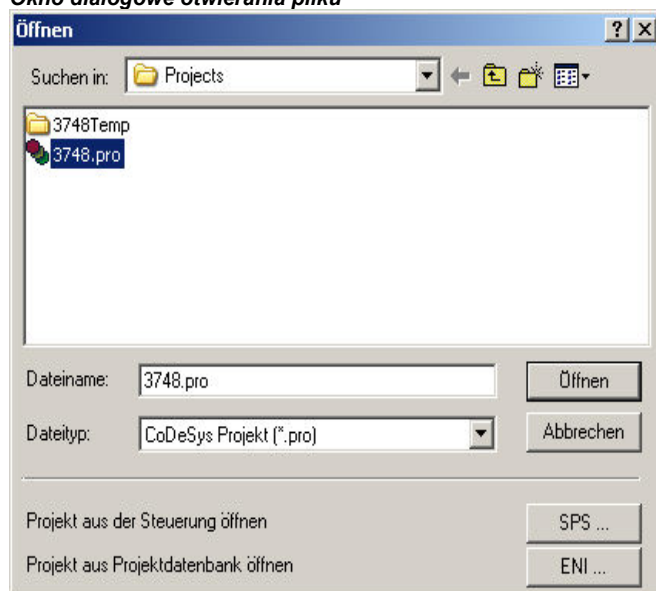
Symbol: 

Za pomocą tego polecenia można otworzyć istniejący już projekt. Jeśli projekt został już otwarty i zmieniony, program CoDeSys pyta, czy projekt ten ma być zapisany, czy też nie.

Zostanie wyświetlone okno dialogowe otwierania pliku i należy wybrać plik projektu z rozszerzeniem „*.pro” lub plik biblioteki z rozszerzeniem „*.lib”. Plik ten musi już istnieć; niemożliwe jest utworzenie projektu za pomocą polecenia „Otwórz”.

Aby wczytać plik projektu ze sterownika, należy nacisnąć przycisk **PLC....** Jeśli brak jeszcze połączenia ze sterownikiem, najpierw wyświetlane jest okno dialogowe parametrów komunikacji (patrz punkt menu „Online” „Parametry komunikacji”) służące do ustawiania parametrów transmisji. Jeśli połączenie online zostało już nawiązane, następuje sprawdzenie, czy w katalogu projektów w komputerze nie ma już plików projektów o tych samych nazwach. W tym przypadku wyświetlane jest okno dialogowe **Wczytaj projekt ze sterownika**, w którym można zdecydować, czy pliki lokalne mają być zastąpione przez pliki ze sterownika. (Procedura ta odpowiada w odwrotnej kolejności procedurze „Online” „Wczytaj kod źródłowy”, za pomocą której pliki źródłowe projektu zapisywane są w sterowniku. Nie należy mylić z poleceniem „Utwórz projekt bootowalny”!)

Okno dialogowe otwierania pliku



Otwórz projekt ze sterownika

Wskazówka: Należy pamiętać, że po wczytaniu projekt jeszcze nie ma nazwy. Należy zapisać go pod jakąś nazwą! Jeśli umożliwia to system docelowy, „oznaczenie” wprowadzone w informacji o projekcie zostanie automatycznie zastosowane jako nowa nazwa pliku. W takim przypadku podczas wczytywania projektu z PLC zostanie automatycznie otwarte okno dialogowe zapisywania pliku, w którym nazwa ta jest już wprowadzona i gdzie można ją potwierdzić lub zmodyfikować.

Jeśli do sterownika nie wczytano jeszcze projektu, zostanie wyświetlony odpowiedni komunikat o błędzie. Patrz także rozdz. 6.2, „Projekt” „Opcje”, kategoria „Pobieranie źródła”.

Otwórz projekt z bazy danych projektów

Opcja ta służy do otwierania projektu zarządzanego w bazie danych projektów ENI. Warunek stanowi dostęp do serwera ENI, obsługującego bazę danych. Należy nacisnąć przycisk **ENI...**, aby wyświetlić najpierw okno dialogowe „Obiekty projektu”, umożliwiające nawiązanie połączenia z serwerem.

Tu należy wprowadzić odpowiednie dane dostępu (adres TCP/IP, port, nazwę użytkownika, hasło, dostęp tylko do odczytu) oraz katalog bazy danych (nazwa projektu), z którego mają być wczytywane obiekty bazy danych, a następnie potwierdzić przyciskiem **Dalej**. Okno dialogowe zamknie się i otworzy się inne okno dla kategorii „Obiekty wspólne”. Tu należy wprowadzić dane dostępu. Klikając

przycisk **Zakończ**, można zamknąć okno dialogowe i automatycznie wczytać obiekty z ustawionych katalogów. Teraz można w opcjach projektu wprowadzić żądane ustawienia, które będą dotyczyły dalszej edycji projektu. Jeśli zarządzanie projektem ma w dalszym ciągu odbywać się poprzez bazę danych, należy ustalić odpowiednie parametry w oknach dialogowych kategorii Baza danych projektu.

Dane dostępu zapisywane są w pliku codesys.ini – nazwa użytkownika i hasło jednakże tylko wtedy, gdy aktywna jest opcja projektu „Zapisz dane dostępu do bazy projektu” (patrz rozdz. 6.2, kategoria Wczytywanie i zapisywanie).

Otwierane ostatnio projekty

W menu „Plik” w obszarze poniżej punktu menu „Zakończ” widoczna jest lista otwieranych ostatnio projektów. Wybierając wpis, można otworzyć projekt.

Jeśli dla projektu zdefiniowano hasła lub grupy roboczej, zostanie wyświetlone okno dialogowe w celu wprowadzenia hasła.

„Plik” „Zamknij”

Za pomocą tego polecenia można zamknąć otwarty aktualnie projekt. Jeśli projekt został zmieniony, program CoDeSys pyta, czy zmiany te należy zapisać czy też nie.

Jeśli zapisywany projekt nie ma jeszcze nazwy, należy mu ją nadać (patrz „Plik” „Zapisz jako”).

„Plik” „Zapisz”

Symbol:  **Kombinacja klawiszy: <Ctrl> + <S>**

Za pomocą tego polecenia można zapisać projekt, o ile został zmieniony.

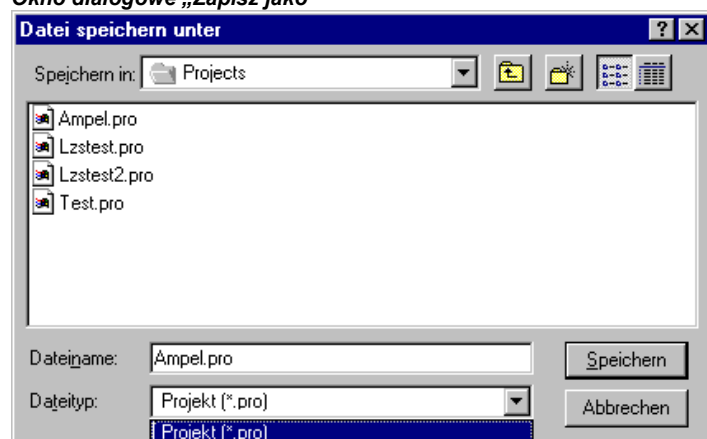
Jeśli zapisywany projekt nie ma jeszcze nazwy, należy mu ją nadać (patrz „Plik” „Zapisz jako”).

„Plik” „Zapisz jako”

Za pomocą tego polecenia można zapisać aktualny projekt w innym pliku lub jako bibliotekę. Pierwotny plik projektu pozostanie przy tym niezmieniony. Istnieje możliwość opatrzenia biblioteki zabezpieczeniem licencyjnym. W przypadku użycia należy wtedy wprowadzić nr identyfikacyjny licencji, który można uzyskać od producenta (patrz niżej).

Po wybraniu polecenia wyświetlane jest okno dialogowe zapisywania. Należy wybrać albo istniejącą już **Nazwę pliku**, albo też wprowadzić nową nazwę i wybrać żądany **typ pliku**.

Okno dialogowe „Zapisz jako”



Jeśli projekt ma zostać zapisany pod nową nazwą, należy wybrać typ pliku programu CoDeSys **Projekt (*.pro)**.

W przypadku wybrania typu pliku **Wersja projektu 1.5 (*.pro)** wzgl. **2.0 (*.pro)**, **2.1 (*.pro)** lub **2.2 (*.pro)**, aktualny projekt zapisywany jest w taki sposób, jakby został utworzony przy użyciu wersji 1.5 wzgl. 2.0, 2.1 lub 2.2. Mogą przy tym zostać utracone dane specyficzne dla wersji 2.3! Projekt można jednak edytować dalej przy użyciu wersji 1.5 wzgl. 2.0, 2.1 lub 2.2.

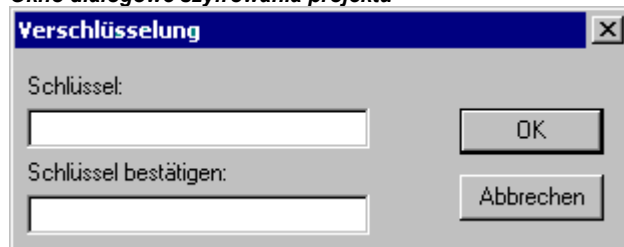
Aktualny projekt można również zapisać jako bibliotekę, aby umożliwić wykorzystanie utworzonych modułów w innych projektach. Należy w tym celu wybrać typ pliku **Wersja biblioteki wewnętrznej (*.lib)**

Jeśli moduły zostały zaimplementowane w innym języku programowania (np. C) i mają być podłączone, należy wybrać typ pliku **Biblioteka zewnętrzna (*.lib)**. W wyniku tego zostanie zapisany dodatkowy plik noszący również nazwę pliku biblioteki, jednakże o rozszerzeniu „.h”. Plik ten ma strukturę pliku C-Header i zawiera deklaracje wszystkich modułów, typów danych oraz zmiennych globalnych. W przypadku bibliotek zewnętrznych podczas symulacji przeprowadzana jest implementacja, zapisana w programie CoDeSys w odniesieniu do modułów. Przy użyciu prawdziwego sprzętu przetwarzana jest implementacja napisana w języku C.

Zapisywanie zaszyfrowanego pliku:

Aby zapisać projekt jako projekt zaszyfrowany lub jako bibliotekę, należy wybrać opcję **Zaszyfrowany projekt CoDeSys (*.pro)** wzgl. **Zaszyfrowana biblioteka wewnętrzna (*.lib)** lub **Zaszyfrowana biblioteka zewnętrzna (*.lib)**. W takim przypadku zostanie wyświetlone okno dialogowe „Szyfrowanie”, w którym można zdefiniować i potwierdzić klucz, bez którego otwarcie projektu nie będzie możliwe, wzgl. nie będzie możliwe podłączenie biblioteki:

Okno dialogowe szyfrowania projektu

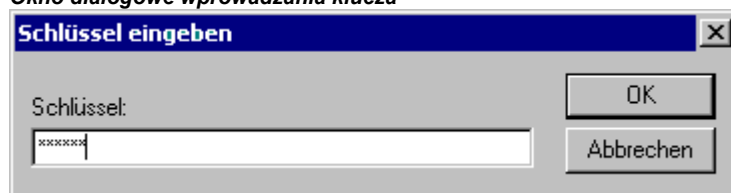


Szyfrowanie rozszerza ochronę projektu, który dotychczas obejmował jedynie nadawanie haseł dostępu i ochrony przed zapisem (patrz rozdz. 6.2, opcje haseł). Możliwości te są dostępne nadal, nie mogą one jednak zapobiec podłączeniu biblioteki do projektu bez wprowadzenia hasła.

Raz nadany klucz zapisywany jest podczas kolejnego zapisywania projektu. Klucz można zmienić tak samo jak w przypadku jego pierwszego nadania, za pośrednictwem okna dialogowego „Zapisz jako”.

Gdy zachodzi potrzeba ponownego otwarcia zaszyfrowanego projektu wzgl. włączenia zaszyfrowanej biblioteki do projektu, wyświetlane jest okno dialogowe wprowadzania klucza, w którym należy wpisać odpowiedni klucz, zdefiniowany podczas zapisywania.

Okno dialogowe wprowadzania klucza



Dodawanie zabezpieczenia licencyjnego do biblioteki:

Jeśli biblioteka ma być licencjonowana, tu można dodać niezbędne informacje licencyjne. Umożliwia to okno dialogowe „Edytuj informacje licencyjne”, otwierane za pomocą przycisku **Edytuj informację o licencji**. Odpowiednie informacje znajdują się w opisie zarządzania licencjami.

Po wprowadzeniu wszystkich danych należy nacisnąć przycisk **OK**. Aktualny projekt zapisywany jest we wskazanym pliku. Jeśli nowa nazwa pliku już istnieje, wyświetlane jest zapytanie, czy plik ma być zastąpiony.

W przypadku polecenia „Zapisz jako bibliotekę” kompilowany jest cały projekt. Jeśli wystąpi przy tym błąd kompilacji, projekt nie zostanie zapisany jako biblioteka i zostanie wyświetlona odpowiednia wskazówka.

„Plik” „Zapisz/wyślij archiwum...”

Za pomocą tego polecenia można tworzyć skompresowane pliki archiwum zip, zawierające wszystkie pliki, istotne dla projektu CoDeSys. Plik zip można zapisać w systemie plików lub też przesłać bezpośrednio pocztą elektroniczną.

Należy pamiętać: Funkcja archiwizacji nie nadaje się do odtwarzania środowisk projektu. Została ona przewidziana wyłącznie jako możliwość prostego zestawiania wszystkich plików należących do projektu. Podczas rozpakowywania archiwum należy dostosować ścieżki poszczególnych plików do danego środowiska programu CoDeSys.

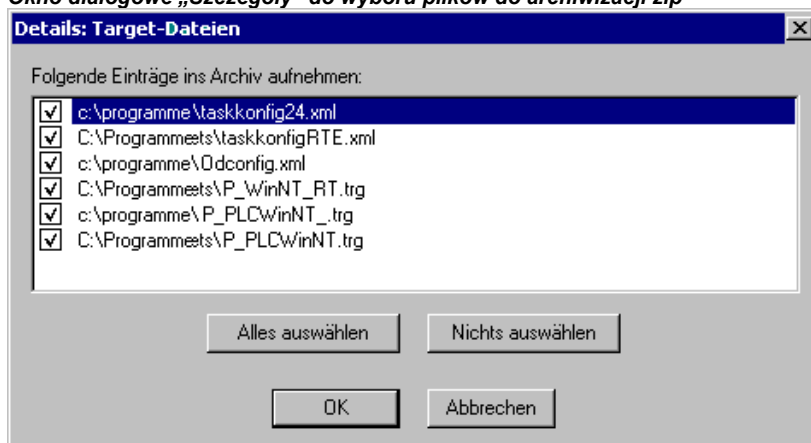
Po wykonaniu polecenia otwiera się okno dialogowe **Zapisz archiwum**. Można w nim zdefiniować, jakie kategorie plików mają być dołączane do archiwum projektu. Wybór kategorii wskazywany jest przez umieszczenie znacznika w polu wyboru bezpośrednio przed odpowiednią kategorią. Należy w tym celu kliknąć pole przyciskiem myszy lub też kliknąć dwukrotnie nazwę kategorii.

Okno dialogowe „Zapisz archiwum”



W przypadku każdej wybranej kategorii kopiowane są do pliku zip zasadniczo wszystkie istotne pliki. W odniesieniu do niektórych kategorii można jednak dokonać wyboru częściowego. Umożliwia to okno dialogowe „Szczegóły”, otwierane za pomocą przycisku **Szczegóły**.

Okno dialogowe „Szczegóły” do wyboru plików do archiwizacji zip



W oknie dialogowym wyświetlana jest lista wszystkich plików dostępnych w tej kategorii. Automatycznie wybrane są wszystkie pliki – wyjątek stanowi kategoria „Pliki docelowe”, w której wybrane są tylko pliki istotne dla ustawionego systemu docelowego.

W celu zmiany wyboru należy uaktywnić wzgl. dezaktywować żądane pliki. Za pomocą przycisków „Wybierz wszystko” wzgl. „Nie wybieraj niczego” można zaznaczyć wszystkie pliki na liście, pojedyncze kliknięcie w polu wyboru powoduje uaktywnienie wzgl. dezaktywację poszczególnych plików, podobnie jak dwukrotne kliknięcie wpisu. Ponadto wpisy można uaktywniać/dezaktywować, naciskając klawisz <Enter>, gdy dany wpis jest zaznaczony.

Jeśli okno dialogowe zostanie zamknięte za pomocą przycisku **Zapisz**, dokonany wybór zostanie zastosowany. Ustawienie zapisane jest do chwili ostatecznego utworzenia archiwum zip.

W głównym oknie dialogowym „Zapisz archiwum” po szarym tle pola wyboru można rozpoznać kategorie, dla których dokonano wyboru częściowego.

W poniższej tabeli przedstawiono zdefiniowane domyślnie kategorie danych oraz to, jakie pliki uwzględniane są każdorazowo automatycznie.

Kategoria	Przyporządkowane pliki
Plik projektu	<Nazwa projektu>.pro (plik projektu CoDeSys)
Biblioteki referencyjne	*.lib, *.obj, *.hex (biblioteki oraz ew. przyporządkowane pliki obj i hex.)
Informacje o kompilacji	*.ci (informacja ostatniego przebiegu kompilacji), *.ri (informacje o pobieraniu) <temp>.* (tymczasowe pliki kompilacji i pobierania) także dla symulacji
Plik INI	Codesys.ini
Dziennik	*.log (dziennik projektu)
Mapy bitowe	*.bmp (mapy bitowe stosowane w modułach projektu i w wizualizacjach)
Wpisy rejestracyjne	Registry.reg (wpisy Automation Alliance, Gateway i PLC; następujące gałęzie z rejestru: HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions „HKEY_LOCAL_MACHINE\SOFTWARE\AutomationAlliance”
Pliki ikon	*.sdb, *.sym (informacje o symbolach utworzone z projektu)
Pliki konfiguracyjne	Pliki konfiguracji sterownika (pliki konfiguracyjne, pliki źródłowe urządzeń, ikony itp.): np. *.cfg, *.con, *.eds, *.dib, *.ico ...
Pliki docelowe	*.trg (pliki docelowe w formacie binarnym dla wszystkich zainstalowanych celów) *.txt (pliki docelowe w formacie tekstowym dla wszystkich zainstalowanych celów, o ile są dostępne)

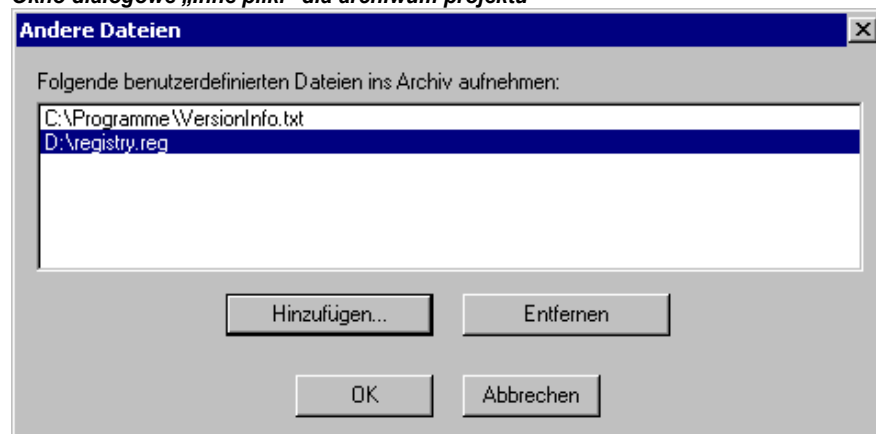
Zarządzanie projektami...

Kategoria	Przyporządkowane pliki
Brama lokalna	Pliki Gateway: Gateway.exe, GatewayDDE.exe, GClient.dll, GDrvBase.dll, GDrvStd.dll, Ghandle.dll, GSymbol.dll, GUtil.dll, ew. inne pliki DLL w katalogu Gateway
Pliki języka	Pliki języka (*.vis, *.xml) dla wizualizacji
Projekt bootowalny	Pliki projektu bootowalnego <nazwa projektu>.prg, <nazwa projektu>.chk wzgl. pliki projektu bootowalnego specyficzne dla celu.

Aby dodać do archiwum zip dowolne inne pliki, należy za pomocą przycisku **Inne pliki...** otworzyć okno dialogowe o takiej samej nazwie.

Można w nim utworzyć listę plików użytkownika. W tym celu za pomocą przycisku **Dodaj** otwierane jest standardowe okno dialogowe, umożliwiające otwarcie pliku. Należy wybrać w nim plik i potwierdzić za pomocą przycisku **Otwórz**. Plik zostanie wtedy dołączony do listy w oknie dialogowym „Inne pliki”. Za pomocą przycisku „Usuń” można usuwać wpisy z listy. Gdy lista jest gotowa, okno dialogowe należy zamknąć za pomocą przycisku „OK”, aby zapisać wpisy przeznaczone do utworzenia pliku zip.

Okno dialogowe „Inne pliki” dla archiwum projektu



Aby dodać do archiwum zip plik Readme, należy nacisnąć przycisk **Komentarz**. Zostanie otwarte okno dialogowe o tej samej nazwie, które zawiera pole edycji. Można wpisać w nim dowolny tekst. Jeśli okno dialogowe zostanie zamknięte przez naciśnięcie przycisku „OK”, podczas tworzenia archiwum zip zostanie utworzony plik o nazwie *Readme.txt*. Zawiera on tekst wprowadzony przez użytkownika. Do tekstu dołączana jest automatycznie data utworzenia (kompilacji) oraz numer wersji używanego aktualnie programu CoDeSys.

Tworzenie archiwum zip:

Po wprowadzeniu wszystkich żądanych ustawień w głównym oknie dialogowym można utworzyć archiwum zip. Dostępne są następujące przyciski:

- **Zapisz...** – umożliwia utworzenie i zapisanie pliku. Otwiera się standardowe okno dialogowe zapisywania pliku i można ustalić, gdzie plik będzie przechowywany. Domyślna nazwa pliku zip to <nazwa projektu>.zip. Potwierdzenie za pomocą przycisku Zapisz powoduje rozpoczęcie generowania archiwum. Przebieg przedstawiany jest w oknie dialogowym postępu oraz protokolowany w oknie komunikatów. Tam także wyświetlany jest komunikat, gdy wskazane pliki nie zostaną znalezione.
- **Wyślij...** – umożliwia utworzenie tymczasowego pliku zip i automatyczne wygenerowanie wiadomości e-mail, zawierającej w załączniku plik zip (<nazwa projektu>.zip). Warunkiem dla tej funkcji jest prawidłowe zainstalowanie MAPI (Messaging Application Programming Interface). Podczas generowania wiadomości e-mail wyświetlane jest okno dialogowe postępu i proces jest protokolowany w oknie komunikatów. Tymczasowy plik zip jest usuwany po dodaniu w formie załącznika do wiadomości e-mail.
- **Anuluj:** Okno dialogowe zostanie opuszczone bez tworzenia archiwum zip, a wprowadzone ustawienia nie zostaną zapisane.

Należy pamiętać: Po rozpakowaniu archiwum w innym systemie należy w razie potrzeby dostosować ścieżki dostępu do plików!

„Plik” „Drukuj”

Kombinacja klawiszy: <Ctrl> + <P>

Za pomocą tego polecenia można wydrukować zawartość aktywnego okna.

Po wybraniu polecenia wyświetlane jest okno dialogowe drukowania. Należy wybrać żadaną opcję lub skonfigurować drukarkę i kliknąć przycisk **OK**. Zawartość aktywnego okna zostanie wydrukowana. Możliwe są kolorowe wydruki ze wszystkich edytorów.

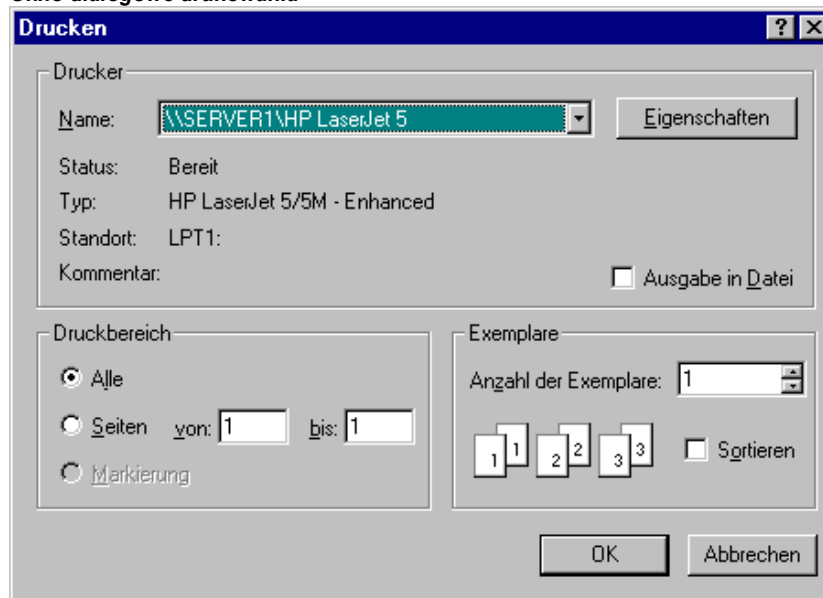
Można również podać **liczbę egzemplarzy** lub skierować wydruk do pliku.

Za pomocą przycisku **Właściwości** można otworzyć okno dialogowe konfiguracji drukarki.

Układ wydruku można ustalić za pomocą polecenia „Plik” „Ustawienia drukowania”.

Aby umożliwić uwzględnienie podziału na strony w oknach edytora już podczas pracy, można za pomocą opcji „Pokaż obszary wydruku” w oknie dialogowym „Projekt” „Opcje” „Obszar roboczy” włączyć wyświetlanie ustawionych aktualnie granic obszaru drukowania.

Okno dialogowe drukowania



Podczas drukowania w oknie dialogowym wyświetlana jest informacja o liczbie wydrukowanych aktualnie stron. W przypadku zamknięcia tego okna proces drukowania zostanie zatrzymany po następnym stronie.

W celu wykonania dokumentacji całego projektu należy użyć polecenia „Projekt” „Dokumentowanie”.

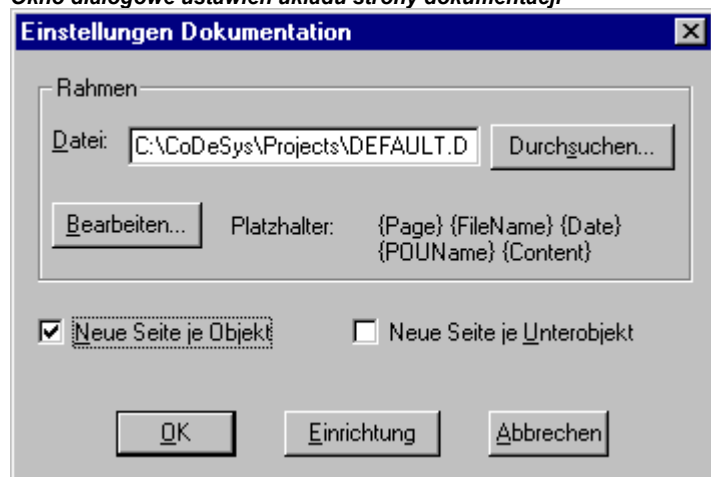
Aby utworzyć szablon dokumentacji do projektu, w którym można byłoby wprowadzać komentarze na temat wszystkich zmiennych używanych w projekcie, należy otworzyć listę zmiennych globalnych i użyć polecenia „Dodatki” „Utwórz szablon dokumentacji”.

Jeśli aktywne jest okno komunikatów, drukowana jest wierszami cała jego zawartość zgodnie z wyglądem w oknie. Możliwa zawartość: wynik kompilacji, lista odsyłaczy, wynik wyszukiwania, wynik porównania, protokołów przetwarzania wsadowego.

„Plik” „Ustawienia drukowania”

Za pomocą tego polecenia można ustawić układ drukowanych stron. Zostanie otwarte następujące okno dialogowe:

Okno dialogowe ustawień układu strony dokumentacji



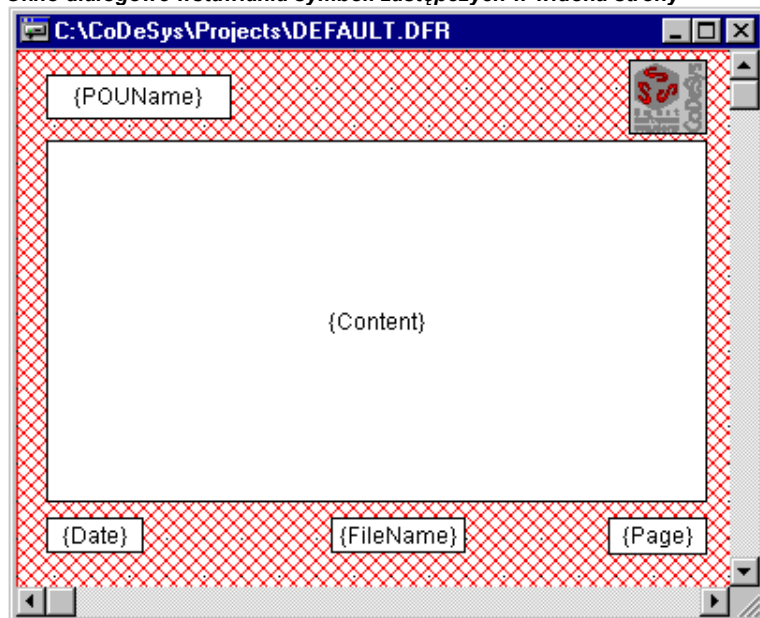
W polu **Plik** można wprowadzić nazwę oraz ścieżkę do pliku z rozszerzeniem „dfr”, w którym zostanie zapisany układ strony. Domyślnie szablon zapisywany jest w pliku DEFAULT.DFR.

Jeśli zachodzi potrzeba zmiany istniejącego układu, za pomocą przycisku **Znajdź** należy otworzyć okno dialogowe „Otwórz” i wybrać żądany plik.

Można również wybrać, czy **dla każdego obiektu** oraz **dla każdego podobiektu** ma być rozpoczynana nowa strona. Za pomocą przycisku **Konfiguracja** można otworzyć okno dialogowe konfiguracji drukarki.

Kliknięcie przycisku **Edycja** powoduje wyświetlenie szablonu umożliwiającego ustawienie układu strony. Można w nim ustawić liczbę stron, datę, nazwę pliku oraz modułu, jak również umieścić na stronie elementy graficzne oraz ustalić obszar tekstu, w jakim będzie drukowana dokumentacja. Powierzchnia strony, ustawiona domyślnie wg konfiguracji drukarki, jest zaznaczona czerwonymi kreskami.

Okno dialogowe wstawiania symboli zastępczych w widoku strony



Za pomocą punktu menu „**Wstaw**” „**Symbol zastępczy**”, a następnie poprzez wybór jednego z pięciu symboli zastępczych (strona, nazwa modułu, nazwa pliku, data, zawartość) można wstawić w widoku tzw. symbol zastępczy, rozciągając w tym celu prostokątne oznaczenie (poprzez przeciągnięcie kursorem myszy po przekątnej przy naciśniętym lewym przycisku myszy). Zostaną one zastąpione na wydruku w następujący sposób:

6 - Szczegółowy opis składników

Polecenie	Symbol zastępczy	Działanie
Strona	{Page}	W tym miejscu na wydruku widnieć będzie aktualna liczba stron.
Nazwa modułu	{POUName}	Tu pojawi się nazwa aktualnego modułu.
Nazwa pliku	{FileName}	Tu pojawi się nazwa projektu.
Data	{Date}	W tym miejscu znajdzie się aktualna data.
Zawartość	{Content}	W tym miejscu znajdzie się spis treści modułu.

Ponadto za pomocą polecenia „**Wstaw**” „**Mapa bitowa**” można wstawić na stronie element graficzny (np. logo firmy). Po wybraniu opcji wstawiania elementu graficznego w widoku strony należy również narysować prostokąt za pomocą myszy. Można również wstawiać inne elementy wizualizacji.

Jeśli szablon został zmieniony, program CoDeSys pyta podczas zamykania okna, czy zmiany te powinny zostać zapisane czy też nie.

Wskazówka: Aby możliwe było już podczas programowania uwzględnienie wielkości papieru, przewidzianej dla późniejszego wydruku projektu, należy najpierw ustawić w opisany tu sposób żądany format i włączyć w opcjach projektu, w kategorii „Obszar roboczy” opcję „Pokaż obszary wydruku”.

„Plik” „Zakończ”

Kombinacja klawiszy: <Alt> + <F4>

Za pomocą tego polecenia można zamknąć program CoDeSys.

Jeśli projekt jest otwarty, zamykanie nastąpi w sposób opisany w odniesieniu do polecenia „Plik” „Zapisz”.

„Projekt” „Kompiluj”

Kombinacja klawiszy: <F11>

Polecenie menu „Projekt” „Kompiluj” powoduje skompilowanie projektu. Proces kompilacji ma z zasady charakter przyrostowy, tzn. kompilacji na nowo podlegają tylko zmienione moduły. Niezbędna do tego celu informacja z ostatniego procesu kompilacji przechowywana jest podczas zapisywania projektu w pliku *.ci. Za pomocą tego polecenia można również uruchomić nieprzyrostowy proces kompilacji, gdy wcześniej użyte zostało polecenie „Projekt” „Wyczyść wszystko”.

W przypadku systemów docelowych, które zapewniają możliwość zmian online, po zakończeniu procedury kompilacji w Menedżerze obiektów wszystkie obiekty, które zostaną wczytane do sterownika podczas następnego pobierania, oznaczone są niebieską strzałką.

Procedura kompilacji, wykonywana za pomocą polecenia „Projekt” „Kompiluj”, przebiega automatycznie, jeśli zalogowanie do sterownika nastąpiło za pośrednictwem polecenia „Online” „Zaloguj”.

Informacje na temat powiązań pomiędzy poleceniami kompilacji projektu, pobierania, zmian online i logowania a systemem docelowym zawiera przegląd w rozdziale 6.6, „Online” „Zaloguj”.

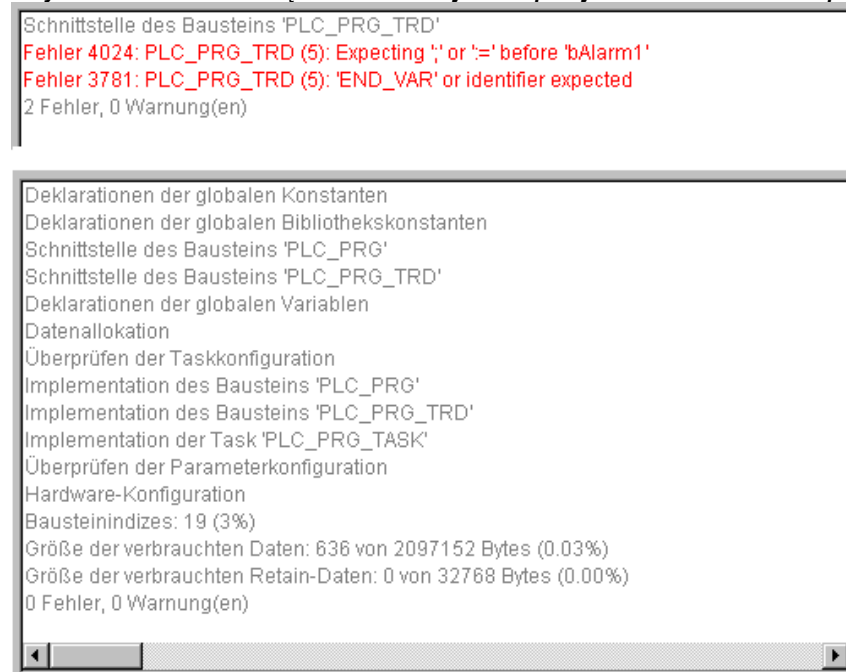
Podczas kompilacji otwierane jest okno komunikatów, w którym wskazywany jest postęp procesu kompilacji, ew. błędy i ostrzeżenia podczas procesu kompilacji, jak również informacje dotyczące wskaźników wzgl. użycia pamięci (zawsze w postaci liczby oraz wartości procentowej). Błędy i ostrzeżenia są numerowane. Za pomocą klawisza F1 można uzyskać dalsze informacje na temat zaznaczonego aktualnie błędu.

Jeśli w oknie dialogowym opcji w kategorii „Wczytywanie i zapisywanie” wybrano opcję **Kopia zapasowa przed kompilacją**, projekt zostanie zapisany przed skompilowaniem.

Pojedyncze obiekty wzgl. kilka obiektów można wykluczyć z kompilacji za pomocą polecenia menu podręcznego „Wykluczenie z kompilacji” wzgl. dzięki odpowiedniej konfiguracji („Wyklucz obiekty”) w opcjach kompilacji (patrz rozdział 6.2, opcje kompilacji).

Wskazówka: Odsyłacze powstają podczas kompilacji i są zapisywane w informacjach o kompilacji. Aby możliwe było zastosowanie poleceń „Wyświetl drzewo wywołań”, „Wyświetl listę odsyłaczy” oraz „Nieużywane zmienne”, „Konflikt dostępu” i „Wielokrotny zapis na wyjściu” menu „Projekt” „Sprawdź” wzgl. otrzymanie aktualnych rezultatów, po modyfikacji projekt należy ponownie skompilować.

Przykład komunikatów o błędach i informacji o kompilacji w oknie komunikatów projektu



„Projekt” „Kompiluj wszystko”

Za pomocą poleceń „Projekt” „Kompiluj wszystko” można, w przeciwieństwie do kompilowania przyrostowego („Projekt” „Kompiluj”), skompilować ponownie cały projekt. W takim przypadku nie zostanie jednak usunięta informacja o pobieraniu, jak ma to miejsce w przypadku polecenia „Wyczyść wszystko”. Należy także pamiętać o możliwości wykluczania obiektów z kompilacji (patrz rozdział 6.2, opcje kompilacji).

Informacje na temat powiązań pomiędzy poleceniami kompilacji projektu, pobierania, zmian online i logowania a systemem docelowym zawiera przegląd w rozdziale 6.6, „Online” „Zaloguj”.

„Projekt” „Wyczyść wszystko”

Za pomocą tego polecenia można usuwać informacje o ostatnim pobraniu i ostatnim procesie kompilacji.

Po wybraniu polecenia wyświetlane jest okno dialogowe, zawierające wskazówkę, że dokonanie zmian online jest już niemożliwe. W tym miejscu można anulować lub potwierdzić polecenie.

Wskazówka: Zmiana online po poleceniu „Wyczyść wszystko” jest możliwa tylko wtedy, jeśli uprzednio plik *.ri był zapisany wraz z informacjami o projekcie z ostatniego pobierania jako plik bez nazwy lub też poza katalogiem projektu (patrz „Wczytaj informacje o pobieraniu”) i teraz przed zalogowaniem ma zostać celowo ponownie wczytany.

Informacje na temat powiązań pomiędzy poleceniami kompilacji projektu, pobierania, zmian online i logowania a systemem docelowym zawiera przegląd w rozdziale 6.6, „Online” „Zaloguj”.

„Projekt” „Wczytaj informacje o pobieraniu”

Za pomocą tego polecenia można w sposób zamierzony ponownie wczytać informacje o pobieraniu związane z projektem. Po naciśnięciu przycisku polecenia otwiera się standardowe okno dialogowe „Plik” „Otwórz”.

Informacje o pobieraniu zapisywane są automatycznie podczas każdego pobierania oraz ew. – w zależności od systemu docelowego – przed każdym utworzeniem projektu bootowalnego w trybie offline w pliku o nazwie **<nazwa projektu><identyfikator celu>.ri**, zapisywanym w katalogu projektu. Jest on wczytywany wraz z każdym otwarciem projektu i służy do ustalenia, podczas ponownego logowania do sterownika, czy projekt w sterowniku odpowiada aktualnie otwartemu projektowi (ID-Check). Ponadto sprawdzane jest, w jakich modułach zmienił się wygenerowany kod. Tylko te moduły wczytywane są podczas pobierania w systemach, które zapewniają możliwość zmian online. Plik *.ri stanowi tym samym warunek zmian online.

Należy pamiętać: Za pomocą polecenia „Projekt” „Wyczyść wszystko” należący do projektu plik *.ri usuwany jest automatycznie z katalogu projektu, w wyniku czego początkowo brak jest możliwości zmiany online, chyba że plik *.ri został uprzednio zapisany w innym miejscu lub pod inną nazwą i można go ponownie wczytać.

Informacje na temat powiązań pomiędzy poleceniami kompilacji projektu, pobierania, zmian online i logowania a systemem docelowym zawiera przegląd w rozdziale 6.6, „Online” „Zaloguj”.

„Projekt” „Kompilacja na inny język”

Ten punkt menu umożliwia skompilowanie aktualnego pliku projektu na inny język wzgl. na wyświetlenie go w innym języku. Odbywa się to poprzez wczytanie pliku kompilacji, który został wygenerowany z projektu i uzupełniony zewnętrznie za pomocą edytora tekstowego o teksty przetłumaczone na żądany język danego kraju.

Dostępne są również następujące podpunkty menu:

- Utwórz plik kompilacji
- Kompiluj projekt
- Pokaż skompilowany projekt
- Przełącz kompilację

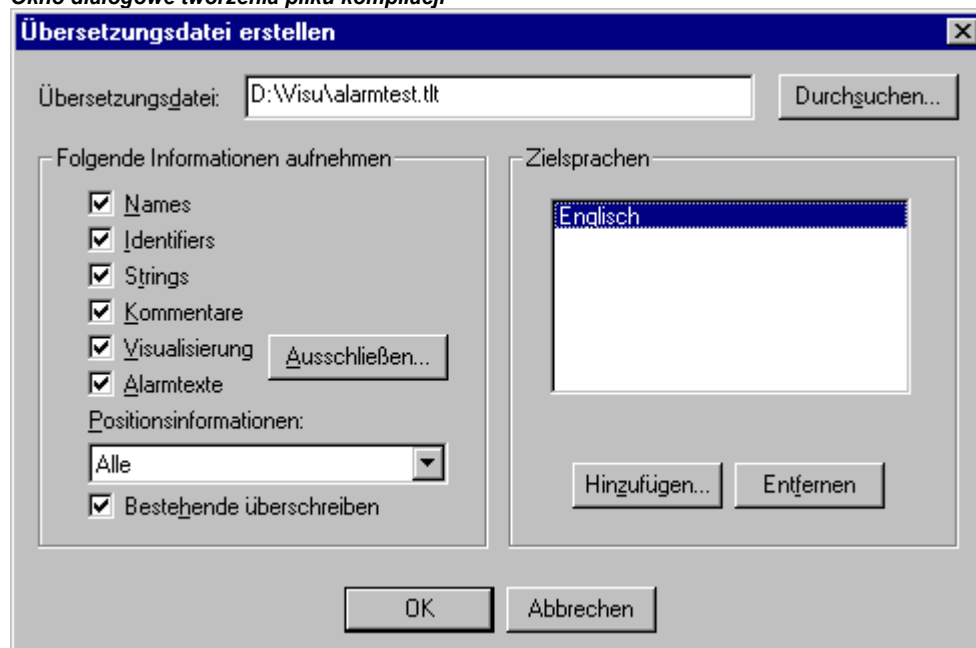
Utwórz plik kompilacji

To polecenie menu „Projekt” „Kompiluj na inny język” prowadzi do okna dialogowego „Utwórz plik kompilacji”.

W polu **Plik kompilacji** należy wprowadzić ścieżkę, wskazującą miejsce zapisania pliku. Standardowe rozszerzenie pliku to *.tlt – jest to plik tekstowy. Możliwe jest również zastosowanie rozszerzenia *.txt, co jest zalecane, jeśli plik np. ma być edytowany w programach, takich jak Excel czy Word, ponieważ w takim przypadku dane mają postać tabelaryczną.

Jeśli istnieje już plik kompilacji, który ma być poddany edycji, należy wpisać ścieżkę dostępu do tego pliku wzgl. użyć standardowego okna dialogowego systemu Windows, dostępnego po naciśnięciu przycisku **Znajdź** i umożliwiającego wybór pliku.

Okno dialogowe tworzenia pliku kompilacji



Do tworzonego wzgl. modyfikowanego pliku kompilacji można załączyć opcjonalnie następujące informacje z projektu, dzięki czemu będą one dostępne do kompilacji: **Nazwy** (nazwa, np. tytuł „Moduły” w Organizerze obiektów), **Identyfikatory** (oznaczenia), **Ścieżki**, **Komentarze**, **Wizualizacja**, **Teksty alarmowe**. Ponadto można zastosować w stosunku do tych elementów projektu **Informację o pozycji**.

Jeśli odpowiednie opcje są zaznaczone znacznikiem, informacje przejmowane są jako symbole języka z aktualnego projektu do tworzonego na nowo pliku kompilacji wzgl. uzupełniane w pliku już istniejącym. Jeśli odpowiednia opcja nie została wybrana, z pliku kompilacji zostaną usunięte wszelkie informacje danej kategorii, niezależnie od tego, z jakiego projektu pochodzą.

Teksty w wizualizacjach:

Jako teksty wizualizacji uznaje się tu elementy „Tekst” i „Tekst etykiety” elementów wizualizacji. W przypadku zastosowania pliku kompilacji do tekstów wizualizacji należy jednak zwrócić uwagę na następujące punkty:

- Plik *.tlt lub plik kompilacji *.txt może być używany tylko w programie CoDeSys i CoDeSys HMI, jednakże nie w wizualizacji elementu docelowego i w wizualizacji sieciowej. W razie potrzeby należy użyć specjalnego pliku języka wizualizacji *.vis.
- Przelączenie z jednego języka na inny jest możliwe zasadniczo tylko w trybie online, tzn. teksty wizualizacji nie są skompilowane za pomocą polecenia „Projekt” „Kompiluj na inny język”, lecz tylko wtedy, gdy w oknie dialogowym „Dodatki” „Ustawienia” ustawiono online odpowiedni język.
- Jeśli dla tekstów wizualizacji ma być użyty plik tlt lub txt („Tekst” i „Tekst etykiety” elementów wizualizacji), teksty te muszą zostać wprowadzone w oknie dialogowym konfiguracji elementu wizualizacji pomiędzy dwoma znakami „#” (np. #tekst#).
- Informacje na temat przelączenia języka w wizualizacji, patrz podręcznik programu CoDeSys_Wizualizacja_V23_PL.pdf.

Informacja o pozycji: Informacje te opisują za pomocą danych, takich jak ścieżka do pliku, moduł oraz wiersz, pozycję symbolu języka, jaki zostanie udostępniony do kompilacji. Do wyboru dostępne są trzy opcje:

- „Żadne”: Nie są generowane żadne informacje na temat pozycji.
- „Występuje pierwszy raz”: Do pliku kompilacji włączana jest ta pozycja, w której skompilowany element występuje po raz pierwszy.
- „Wszystkie”: Podawane są wszystkie pozycje, w których dany element występuje w projekcie.

6 - Szczegółowy opis składników

Jeśli edytowany jest utworzony wcześniej plik kompilacji, który zawiera już więcej informacji o pozycji, niż wybrano aktualnie, wtedy zostaną one odpowiednio skrócone lub całkowicie usunięte, niezależnie od tego, z jakiego projektu zostały wygenerowane.

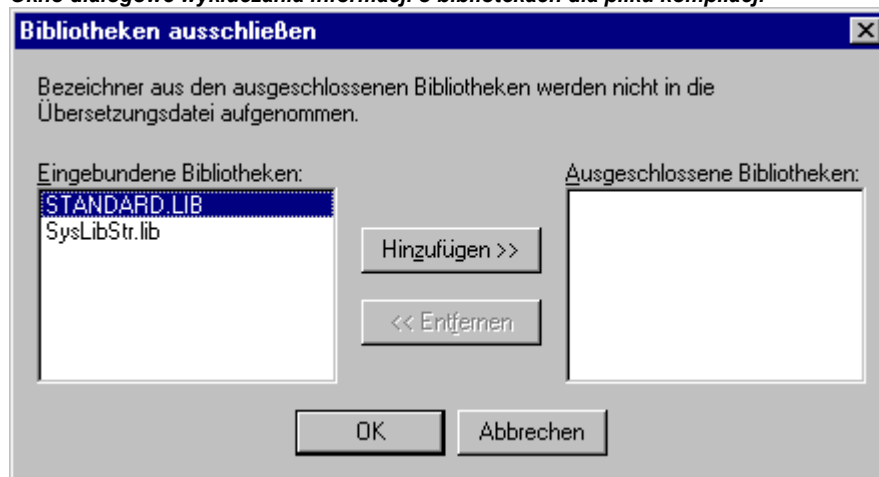
Wskazówka: Na każdy element (symbol języka) generowane są maks. 64 informacje o pozycji, nawet jeśli użytkownik w oknie dialogowym „Utwórz plik kompilacji” wybrał „Wszystkie” informacje o pozycji.

Zastąp bieżące: Zostaną zastąpione wszystkie istniejące już informacje o pozycji w pliku kompilacji, który jest aktualnie edytowany, niezależnie od tego, z jakiego projektu zostały wygenerowane.

Języki docelowe: Lista ta zawiera identyfikatory wszystkich języków, jakie zawiera plik kompilacji wzgl. jakie zostaną włączone po zamknięciu okna dialogowego „Utwórz plik kompilacji”.

Przycisk **Wyklucz** otwiera okno dialogowe „Wyklucz biblioteki”.

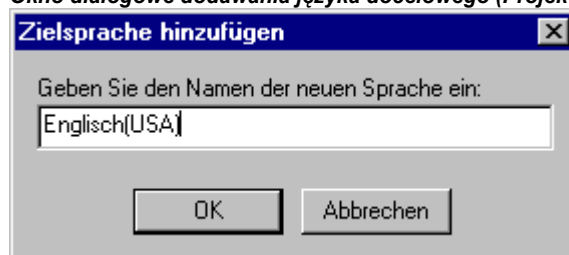
Okno dialogowe wykluczania informacji o bibliotekach dla pliku kompilacji



W tym oknie dialogowym spośród bibliotek załączonych do projektu można wybrać te, których informacje identyfikacyjne nie powinny zostać umieszczone w pliku kompilacji. W tym celu odpowiedni wpis wybierany jest za pomocą myszy z lewej tabeli **Powiązane biblioteki** i poprzez kliknięcie przycisku **Dodaj** umieszczany w prawej tabeli **Wykluczone biblioteki**. Za pomocą przycisku **Usuń** można również ponownie usunąć wybrany już wpis. Naciśnięcie przycisku „OK” powoduje potwierdzenie ustawienia i zamknięcie okna dialogowego.

Przycisk **Dodaj** otwiera okno dialogowe „Dodaj język docelowy”:

Okno dialogowe dodawania języka docelowego (Projekt, Kompilacja na inny język)



W polu edycji należy wprowadzić identyfikator języka, który ani na początku, ani na końcu nie może zawierać spacji ani znaków specjalnych (ę, ą, ź, ć itp.).

Za pomocą przycisku **OK** można zamknąć okno dialogowe „Dodaj język docelowy” i nowy język docelowy pojawi się na liście języków docelowych.

Przycisk **Usuń** powoduje usunięcie wybranego wpisu z listy.

W celu wygenerowania pliku kompilacji można również potwierdzić okno dialogowe „Utwórz plik kompilacji” za pomocą przycisku **OK**. Jeśli plik kompilacji o tej samej nazwie już istnieje, wyświetlane jest najpierw poniższe zapytanie, które należy potwierdzić za pomocą „Tak” lub „Nie”:

„Wskazany plik kompilacji już istnieje. Zostanie on teraz odpowiednio zmieniony, przy czym zostanie utworzona kopia zapasowa aktualnego pliku. Czy chcesz kontynuować?”

Odpowiedź **Nie** powoduje powrót do okna „Utwórz plik kompilacji” bez wykonywania żadnej czynności. Natomiast po wybraniu odpowiedzi **Tak** tworzona jest w tym samym katalogu kopia istniejącego już pliku kompilacji o nazwie pliku „Backup_of_<plik kompilacji>.xlt”, zaś dany plik kompilacji modyfikowany jest zgodnie z ustawionymi opcjami.

Podczas tworzenia pliku kompilacji wykonywane są następujące czynności:

- Dla każdego nowego języka docelowego generowany jest symbol zastępczy („##TODO”) dla wyświetlanego symbolu języka. (Patrz – „Edycja pliku kompilacji”).
- Jeśli zostanie zmieniony istniejący już plik kompilacji, nastąpi usunięcie wpisów plików języka, które znajdują się w pliku kompilacji, ale nie ma ich na liście języków docelowych, niezależnie od tego, z jakiego projektu zostały wygenerowane.

Edycja pliku kompilacji

Plik kompilacji należy otworzyć i zapisać jako plik tekstowy. Znaki ## oznaczają słowa kluczowe. Symbole zastępcze ##TODO w pliku można zastąpić aktualnymi tekstami kompilacji. Dla każdego symbolu tworzony jest akapit ograniczony identyfikatorem typu. Na przykład oznaczenia ##NAME_ITEM i ##END_NAME_ITEM wskazują początek i koniec akapitu nazwy obiektu w Organizercie obiektów. COMMENT_ITEM oznacza akapity dla komentarzy, IDENTIFIER_ITEM dla identyfikatorów, STRING_ITEM dla ścieżek, a VISUALTEXT_ITEM dla tekstów wizualizacji).

Poniżej przedstawiono przykładowy akapit w pliku kompilacji o formacie *.tlt dla nazwy (NAME_ITEM) używanego w projekcie modułu: ST_Visu. Przewidziane są języki docelowe angielski (USA) i francuski. W tym przykładzie zawarta jest również informacja o pozycji skompilowanego elementu projektu:

– przed kompilacją:

```
##NAME_ITEM
[D:\CoDeSys\projects\Bspdt_22.pro::ST_Visualisierung::0]
ST_Visualisierung
##English :: ##TODO
##French :: ##TODO
##END_NAME_ITEM
```

– po kompilacji:

W miejsce „##TODO” wstawiony został angielski wzgl. francuski odpowiednik słowa „Wizualizacja”:

```
##NAME_ITEM
[D:\CoDeSys\projects\Bspdt_22.pro::ST_Visualisierung::0]
ST_Visualisierung
##English :: ST_Visualization
##French :: ST_Visu
##END_NAME_ITEM
```

Należy pamiętać o tym, że przetłumaczone identyfikatory i nazwy obowiązują zgodnie z normą, i że ścieżki oraz komentarze umieszczone są w odpowiednich znakach nawiasów klamrowych. W przypadku komentarza (##COMMENT_ITEM), który znajduje się w pliku kompilacji z wpisem „(* comment 1)”, symbol zastępczy „##TODO” musi zostać zastąpiony przez „(* komentarz 1 *)”, zaś w przypadku ścieżki (##STRING_ITEM) „string1” przez „ciąg znaków1”.

Wskazówka: Poniższych fragmentów pliku kompilacji nie należy modyfikować bez dokładniejszej wiedzy: bloki języka, bloki znaczników, informacje o pozycji, teksty oryginalne.

Kompiluj projekt (na inny język)

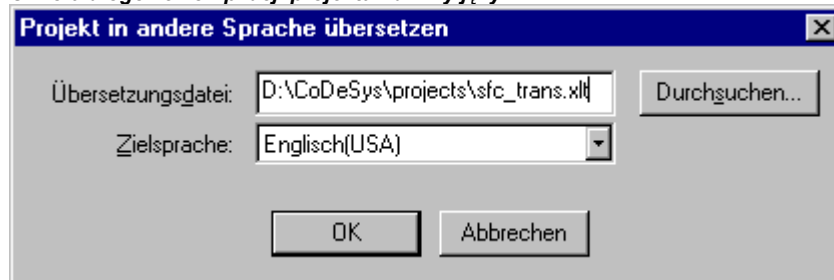
To polecenie menu „Projekt” „Kompiluj na inny język” prowadzi do okna dialogowego „Kompiluj projekt na inny język”.

Aktualny projekt może zostać skompilowany na inny język z użyciem aktualnego pliku kompilacji.

Wskazówka: Jeśli ma zostać zachowana wersja językowa projektu, w której został utworzony, kopię projektu należy zapisać przed kompilacją pod inną nazwą. **Procesu kompilacji nie można cofnąć.**

W związku z tym należy pamiętać, że projekt można w innym języku tylko wyświetlić, jednakże w widoku tym projektu nie można edytować.

Okno dialogowe kompilacji projektu na inny język



Aktualny projekt może zostać skompilowany na inny język z użyciem aktualnego pliku kompilacji.

Wskazówka: Jeśli ma zostać zachowana wersja językowa projektu, w której został utworzony, kopię projektu należy zapisać przed kompilacją pod inną nazwą. **Procesu kompilacji nie można cofnąć.**

W związku z tym należy pamiętać, że projekt można w innym języku tylko wyświetlić, jednakże w widoku tym projektu nie można edytować.

W polu **Plik kompilacji** należy wpisać ścieżkę do używanego pliku kompilacji. Za pomocą polecenia **Znajdź** można otworzyć standardowe okno dialogowe systemu Windows, umożliwiające wybór pliku.

W polu **Język docelowy** widoczna jest lista identyfikatorów języka, znajdujących się w pliku kompilacji i umożliwiających wybórżądanego języka docelowego.

Przycisk **OK** uruchamia kompilację aktualnego projektu na wybrany język docelowy za pomocą wskazanego pliku kompilacji. Podczas kompilacji wyświetlane jest okno dialogowe postępu oraz ew. komunikaty o błędach. Po kompilacji zamykane są okna dialogowe oraz wszystkie otwarte okna edycji projektu.

Polecenie **Anuluj** powoduje zamknięcie okna dialogowego bez modyfikacji aktualnego projektu.

Jeśli plik kompilacji zawiera błędne wpisy, po naciśnięciu przycisku OK wyświetlany jest komunikat błędu, zawierający ścieżkę dostępu do pliku oraz błędny wiersz, np.: „[C:\Programy\CoDeSys\projects\visu.tft (78)]; Oczekiwano tekstu kompilacji”.

Wskazówka: Należy zwracać uwagę na specyfikę tekstów w wizualizacji, patrz wyżej, „Utwórz plik kompilacji”.

Pokaż skompilowany projekt

Jeśli projekt zawiera już plik kompilacji, można wyświetlić jedną z przetłumaczonych wersji bez zastępowania projektu w oryginalnej wersji językowej.

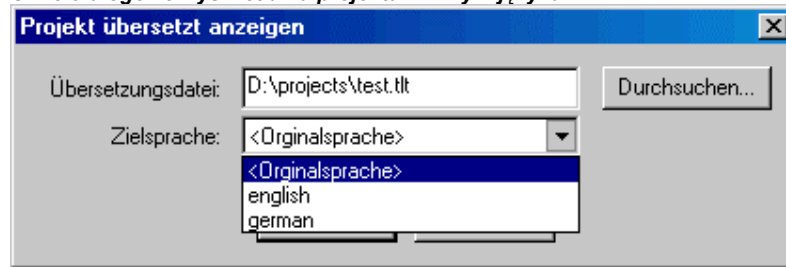
(O możliwości tej należy pamiętać w porównaniu z „rzeczywistym” tłumaczeniem projektu, które umożliwia polecenie „Kompiluj projekt na inny język” „Kompiluj projekt”).

To polecenie menu „Pokaż skompilowany projekt” z menu „Projekt” „Kompilacja na inny język” otwiera okno dialogowe „Pokaż skompilowany projekt”.

W polu **Plik kompilacji** należy wpisać ścieżkę do używanego pliku kompilacji. Za pomocą polecenia **Znajdź** można otworzyć standardowe okno dialogowe systemu Windows, umożliwiające wybór pliku.

W polu **Język docelowy** dostępna jest lista wyboru, która, oprócz wpisu „<Język oryginału>”, zawiera również identyfikatory języka, znajdujące się w pliku kompilacji. Język oryginalny to język zapisany aktualnie wraz z projektem. Zmienić go można tylko poprzez polecenie menu „Projekt” „Kompiluj”. Następnie należy wybrać jeden z dostępnych innych języków i zamknąć okno dialogowe za pomocą przycisku „OK”. Projekt zostanie wtedy wyświetlony w wybranym języku, **jednakże nie można go w tym widoku edytować!**

Okno dialogowe wyświetlania projektu w innym języku



Aby przełączyć się na język oryginału, można użyć polecenia „Przełącz kompilację”.

Wskazówka: Należy zwracać uwagę na specyfikę tekstów w wizualizacji, patrz wyżej, „Utwórz plik kompilacji”.

Przełącz kompilację

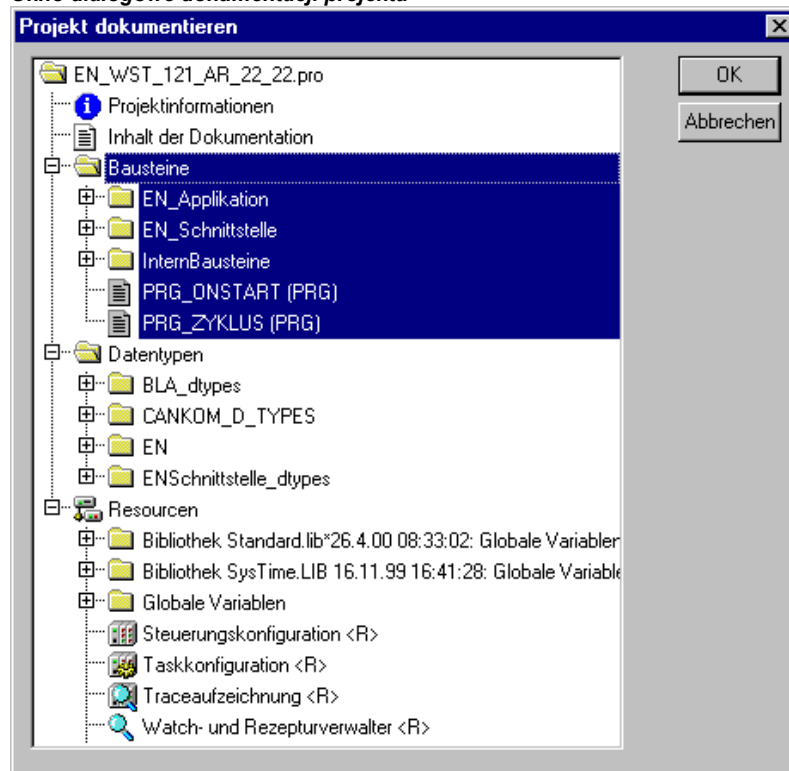
Jeśli za pomocą polecenia „Pokaż skompilowany projekt” (chroniony przed zapisem) widok projektu zostanie przetłumaczony na inny, udostępniony w pliku kompilacji język, polecenie „Przełącz kompilację” menu „Projekt” „Kompilacja na inny język” umożliwi przełączanie pomiędzy tą wersją językową oraz (edytowalną) wersją oryginalną.

Wskazówka: Należy zwracać uwagę na specyfikę tekstów w wizualizacji, patrz wyżej, „Utwórz plik kompilacji”.

„Projekt” „Dokumentowanie”

Polecenie to umożliwia wydrukowanie dokumentacji dla całego projektu.

Okno dialogowe dokumentacji projektu



Kompletna dokumentacja obejmuje:

- moduły,
- przegląd zawartości dokumentacji,
- typy danych,

6 - Szczegółowy opis składników

- wizualizacje,
- zasoby (zmienne globalne, konfigurację zmiennych, zapis śledzenia, konfigurację sterownika, konfigurację zadań, menedżer podglądu zmiennych i receptur),
- drzewa wywołania modułów i typów danych,
- listę odsyłaczy.

W przypadku ostatnich dwóch punktów projekt musi być skompilowany bezbłędnie.

Wydrukowane zostaną obszary zaznaczone w oknie dialogowym „Dokumentowanie projektu” (z niebieskim tłem).

Jeśli zaznaczony ma być cały projekt, należy zaznaczyć nazwę projektu w pierwszym wierszu.

Jeśli natomiast zachodzi potrzeba zaznaczenia pojedynczego obiektu, należy kliknąć dany obiekt wzgl. przemieścić kropkowany prostokąt na żądany obiekt za pomocą przycisków strzałek. Obiekty, które przed symbolem posiadają znak „plus”, to obiekty organizacyjne, zawierające dalsze obiekty. Za pomocą kliknięcia znaku plus obiekt taki można rozwinąć i zwinąć ponownym kliknięciem na wyświetlony teraz znak „minus”. W przypadku zaznaczenia obiektu organizacyjnego wszystkie zawarte w nim obiekty zostaną również zaznaczone. Przytrzymując naciśnięty klawisz <Shift>, można wybrać pewien zakres obiektów, zaś w przypadku klawisza <Ctrl> wiele obiektów pojedynczych.

Po dokonaniu wyboru należy nacisnąć przycisk **OK**. Pojawi się okno dialogowe wydruku. Układ drukowanej strony można ustalić za pomocą polecenia „Plik” „Ustawienia dokumentacji”.

„Projekt” „Eksportuj”

Program CoDeSys oferuje możliwość eksportowania wzgl. importowania modułów. Umożliwia to wymianę programów pomiędzy różnymi systemami programowania IEC.

Aktualnie stosowany jest standaryzowany format wymiany dla modułów w IL, ST i AS (format Common Elements wg IEC 61131-3). W przypadku modułów w LD i FBD oraz innych obiektów program CoDeSys posiada własny format zapisu, ponieważ brak jest dla nich formatu tekstowego w IEC 61131-3.

Wybrane obiekty zapisywane są w pliku ASCII.

Eksportować można moduły, typy danych, wizualizacje i zasoby. Ponadto można również eksportować wpisy w Administratorze biblioteki, tzn. informacje o powiązaniach z bibliotekami (nie same biblioteki!).

Uwaga: Ponowny import wyeksportowanego modułu FBD lub LD nie powiedzie się, jeśli w edytorze graficznym komentarz zawiera prosty apostrof ('), ponieważ jest on interpretowany jako początek ciągu znaków!

Po dokonaniu wyboru w oknie dialogowym (wybór przebiega w sposób opisany w odniesieniu do poleceń „Projekt” „Dokumentowanie”), można jeszcze zdecydować, czy wybór należy wyeksportować do pliku, czy też dla każdego obiektu zostanie wygenerowany odrębny plik eksportu. W tym celu należy włączyć lub wyłączyć opcję **Jeden plik na obiekt**, a następnie potwierdzić klikając przycisk <OK>. Zostanie wyświetlone okno dialogowe zapisywania plików. W nim należy wprowadzić nazwę pliku z rozszerzeniem „.exp” wzgl. katalog dla poszczególnych plików eksportu, które następnie zostaną utworzone pod nazwą „nazwa obiektu.exp”.

„Projekt” „Importuj”

W wyświetlonym oknie dialogowym, umożliwiającym otwarcie plików, należy wybrać żądany plik eksportu.

Dane zostaną zaimportowane do aktualnego projektu. Jeśli w projekcie istnieje już obiekt o tej samej nazwie, zostanie wyświetlone okno dialogowe z pytaniem „Czy chcesz go zastąpić?”: Jeśli zostanie wybrana odpowiedź **Tak**, obiekt w projekcie zostanie zastąpiony przez obiekt z pliku importu, w przypadku wybrania odpowiedzi **Nie**, do nazwy nowego obiektu zostanie jako uzupełnienie dodany zostanie dolny myślnik oraz kolejny numer („_0”, „_1” itd.). Opcja <Tak, wszystkie> wzgl. <Nie, wszystkie> ma zastosowanie do wszystkich obiektów.

Jeśli zostanie zaimportowana informacja o powiązaniu z biblioteką, biblioteka ta zostanie wczytana i dodana w Menedżerze biblioteki na końcu listy. Natomiast jeśli biblioteka została już wczytana do

projektu, nie zostanie wczytana ponownie. Jeżeli natomiast w pliku eksportu, który jest importowany, wskazany jest inny czas zapisu biblioteki, nazwa biblioteki zostanie w Menedżerze biblioteki oznaczona za pomocą znaku „*” (np. standard.lib*30.3.99 11:30:14), analogicznie jak w przypadku wczytywania projektu. Jeśli biblioteki nie można znaleźć, zostanie wyświetlone następujące okno dialogowe z informacją: „Nie można znaleźć biblioteki {<ścieżka>\}<nazwa> <data> <czas>”, analogicznie jak w przypadku wczytywania projektu.

Import jest protokolowany w oknie komunikatów.

„Projekt” „Import Siemens”

W podmenu „Import Siemens” można znaleźć polecenia importu modułów i zmiennych z plików Siemens STEP5.

Dostępne są następujące polecenia:

- „Importuj plik symboli SEQ”
- „Importuj plik S5”

Bliższe informacje można znaleźć w rozdziale **Błąd! Nie można odnaleźć źródła odсыlacza.** „Import Siemens”.

„Projekt” „Porównaj”

Polecenie to jest stosowane w celu porównania dwóch projektów lub aktualnej wersji otwartego projektu z wersją zapisaną ostatnio.

Przegląd:

Definicje:	aktualny projekt:	Projekt aktualnie edytowany
	projekt porównawczy:	Projekt otwierany w celu porównania
	tryb porównywania:	W tym trybie projekt jest wyświetlany po wybraniu polecenia.
	Jednostka:	Najmniejsza jednostka porównawcza, która może składać się z jednego wiersza (edytor deklaracji, ST, IL), sieci (FBD, LD) lub elementu/modułu (CFC, SFC).

W trybie porównania w podzielonym na dwie części oknie zestawiany jest projekt aktualny z projektem porównawczym, a różniące się moduły zaznaczane są kolorem.

W przypadku edytorów modułów możliwe jest też bezpośrednie zestawienie zawartości.

Przed dokonaniem porównania można włączyć filtry dot. uwzględniania spacji oraz komentarzy.

Ponadto można wybrać, czy w trybie porównania zmiany w obrębie istniejących jednostek mają być prezentowane jako takie, czy też wszystkie różniące się jednostki mają być zaznaczone jako „wstawione na nowo” wzgl. „już nieistniejące”.

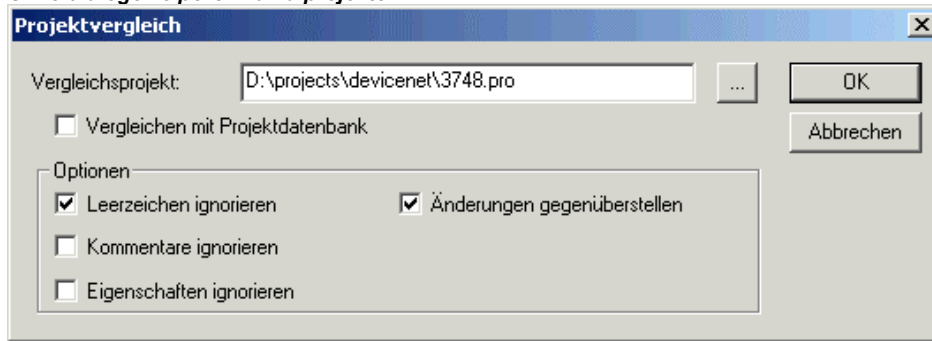
Wersja projektu porównawczego może zostać zastosowana w aktualnym projekcie w przypadku pojedynczych, różniących się jednostek lub też w stosunku do całego bloku zaznaczonych jednostek.


Należy pamiętać: Dopóki aktywny jest tryb porównania (patrz wiersz statusu: COMPARE), projektu nie można edytować!

Przeprowadzanie porównania projektów

Po wybraniu polecenia zostanie otwarte okno dialogowe „Porównanie projektów”:

Okno dialogowe porównania projektów



Należy wpisać ścieżkę do **Projektu porównawczego**. Za pomocą przycisku  można przejść do standardowego okna dialogowego otwierania pliku, które umożliwi wybranie projektu. Jeśli wpisana jest nazwa aktualnego projektu, używana aktualnie wersja projektu porównywana jest z wersją z ostatniego zapisu.

Jeśli projekt znajduje się w bazie danych ENI, porównać można wersję otwartą lokalnie z aktualną wersją z bazy danych. W tym celu należy włączyć opcję **Porównanie z bazą projektów**.

Można włączyć/wyłączyć następujące **Opcje** dotyczące porównania:

Ignoruj spacje: W przypadku różnic w zakresie ilości znaków spacji nie są zgłaszane komunikaty o różnicach.

Ignoruj komentarze: Nie są wyświetlane różnice, dotyczące komentarzy.

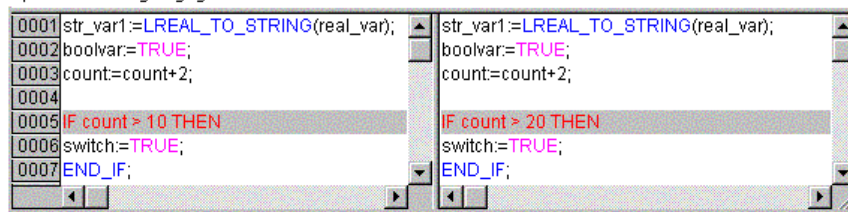
Ignoruj właściwości: Nie są wyświetlane różnice, dotyczące właściwości obiektów.

Porównaj zmiany: Jeśli opcja ta jest aktywna: W przypadku jednostki w obrębie modułu, która nie została usunięta ani dodana, a tylko zmieniona, w podzielonym oknie trybu porównania wersja projektu porównawczego zestawiona jest bezpośrednio z aktualnym projektem (zaznaczona na czerwono, patrz niżej). Jeśli opcja ta jest nieaktywna: Jednostka przedstawiana jest w projekcie porównawczym jako „już nieistniejąca”, a w aktualnym projekcie jako „wstawiona na nowo” (patrz niżej), a więc nie jest zestawiana bezpośrednio.

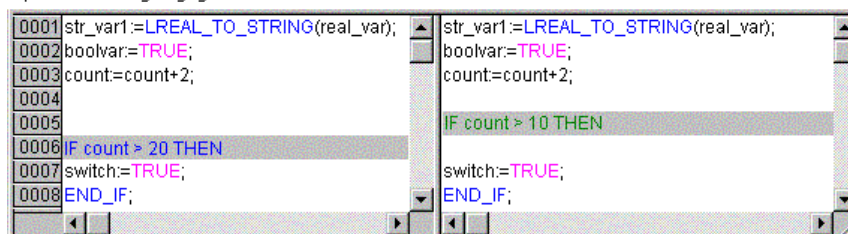
Przykład polecenia „Porównaj zmiany”

Wiersz 0005 został zmieniony w aktualnym projekcie (lewa połowa okna).

Option 'Änderungen gegenüberstellen' aktiviert:



Option 'Änderungen gegenüberstellen' nicht aktiviert:



Jeśli okno dialogowe „Porównanie projektu” zostanie zamknięte za pośrednictwem przycisku „OK”, porównanie zostanie przeprowadzone zgodnie z ustawieniami.

Wyświetlanie wyniku porównania

Wyniki prezentowane są początkowo w drzewie struktury projektu (przegląd projektu), z poziomu którego można następnie otwierać poszczególne moduły, aby zobaczyć zmiany ich zawartości.

1. Przegląd projektu w trybie porównania:

Po przeprowadzeniu porównania projektu otwiera się podzielone na dwie części okno dialogowe, przedstawiające drzewo struktury projektu w trybie porównania. Pasek tytułu zawiera wpis: „Porównanie projektów <ścieżka do aktualnego projektu> – <ścieżka do projektu porównawczego>”.

W lewej części okna wyświetlony jest aktualny projekt, w prawej części projekt porównawczy. Przegląd projektu przedstawia na najwyższej pozycji nazwę projektu i w pozostałej części odpowiada strukturze Organizera obiektów.

Moduły, wykazujące odstępstwa, wyróżniane są poprzez cieniowanie tła, kolor tekstu lub dopisek tekstowy:

Czerwony: Jednostka została zmodyfikowana wewnętrznie; jest wyświetlana w obu częściach okna na czerwono.

Niebieski: Jednostka znajduje się tylko w projekcie porównawczym; po przeciwnej stronie w drzewie struktury aktualnego projektu wstawiane jest puste miejsce.

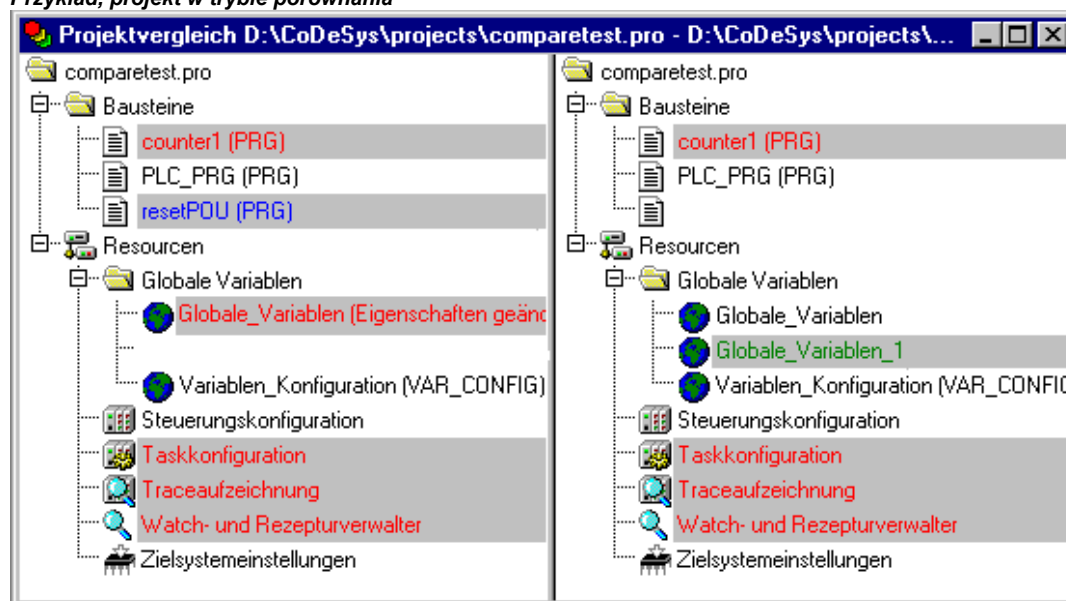
Zielony: Jednostka znajduje się tylko w aktualnym projekcie; po przeciwnej stronie w drzewie struktury projektu porównawczego wstawiane jest puste miejsce.

Czarny: Jednostka, w przypadku której nie stwierdzono różnic.

„**(Właściwości zostały zmienione)**”: Tekst ten wyświetlany jest za nazwą modułu w drzewie struktury aktualnego projektu, jeśli zostały znalezione różnice we właściwościach modułu.

„**(Prawa dostępu zostały zmienione)**”: Tekst ten wyświetlany jest za nazwą modułu w drzewie struktury aktualnego projektu, jeśli zostały znalezione różnice w prawach dostępu.

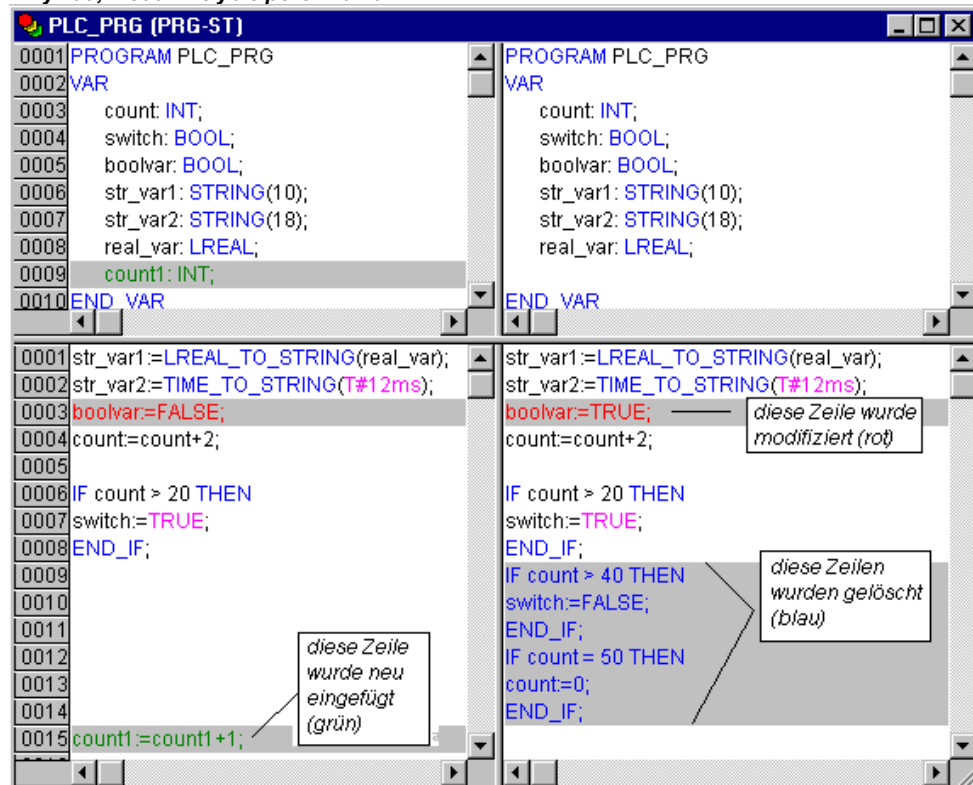
Przykład, projekt w trybie porównania



2. Zawartość modułu w trybie porównania:

Podwójne kliknięcie wiersza w przeglądzie projektu (patrz wyżej), powoduje **otwarcie modułu**.

Przykład, moduł w trybie porównania



Praca w trybie porównania (menu „Dodatki”, menu podręczne)

Jeśli w podzielonym na dwie części oknie kursor znajduje się w wierszu, w którym stwierdzono rozbieżność, menu „Dodatki” wzgl. menu podręczne (prawy przycisk myszy) oferuje, w zależności od tego, czy aktualny widok znajduje się w przeglądzie projektu, czy też w obrębie modułu, wybór następujących poleceń:

- „Następna różnica”
- „Poprzednia różnica”
- „Zastosuj zmianę”
- „Zastosuj pojedynczą zmianę”
- „Zastosuj właściwości”
- „Zastosuj prawa dostępu”

Wskazówka: Przejmowanie różnych fragmentów projektu (zmian) lub też praw dostępu jest możliwe tylko z projektu porównawczego do aktualnego projektu, nigdy odwrotnie.

„Dodatki” „Następna różnica”

Kombinacja klawiszy: <F7>

Polecenie to dostępne jest w trybie porównania (patrz wyżej, „Projekt” „Porównaj”).

Kursor przeskakuje do następnego miejsca (wiersz w przeglądzie projektu/wiersz wzgl. sieć w module), które wykazuje różnice.

„Dodatki” „Poprzednia różnica”

Kombinacja klawiszy: <Shift><F7>:

Polecenie to dostępne jest w trybie porównania (patrz wyżej, „Projekt” „Porównaj”).

Kursor przeskakuje do poprzedniego miejsca (wiersz w przeglądzie projektu/wiersz wzgl. sieć w module), które wykazuje różnice.

„Dodatki” „Zastosuj zmianę”

Kombinacja klawiszy: <Spacja>

Polecenie to dostępne jest w trybie porównania (patrz wyżej, „Projekt” „Porównaj”).

W przypadku wszystkich jednostek, powiązanych z jednostką, w której aktualnie umieszczony jest kursor, i które mają tak samo oznaczone zmiany (np. kolejne wiersze), wersja projektu porównawczego zostanie zastosowana w aktualnym projekcie (jest to możliwe tylko w tym kierunku!). Dane jednostki zostaną w wyniku tego wyświetlone w odpowiednim kolorze w lewej części okna.

Informacje na temat przejmowania zmian pojedynczych jednostek, patrz „Zastosuj pojedynczą zmianę”.

„Dodatki” „Zastosuj pojedynczą zmianę”

Kombinacja klawiszy: <Ctrl> <Spacja>

Polecenie to dostępne jest w trybie porównania (patrz wyżej, „Projekt” „Porównaj”).

Wersja projektu porównawczego przejmowana jest do aktualnego projektu (jest to możliwe tylko w tym kierunku!) jedynie w odniesieniu do jednostki porównawczej, w której aktualnie umieszczony jest kursor (np. wiersze w przeglądzie projektu lub wiersz wzgl. sieć w module). W wyniku tego odpowiednia jednostka zostanie wyświetlona w odpowiednim kolorze w lewej części okna.

Jeśli zmiana ta zostanie zastosowana w odniesieniu do modułu w drzewie struktury, który był zaznaczony na czerwono ze względu na stwierdzenie zmiany zawartości, wtedy moduł zostanie zaznaczony w aktualnym projekcie żółtą czcionką.

Moduły, które znalazły się w aktualnym projekcie tylko w wyniku polecenia „Zastosuj zmianę”, są również wyświetlane przy użyciu żółtej czcionki. Moduły, które zostały usunięte z aktualnego projektu tylko w wyniku polecenia „Zastosuj zmianę”, są wyświetlane przy użyciu żółtej czcionki w projekcie porównawczym.

„Dodatki” „Zastosuj właściwości”

Polecenie to dostępne jest w trybie porównania, a tam tylko w przeglądzie projektu (patrz wyżej, „Projekt” „Porównaj”).

W przypadku modułu, w którym umieszczony jest kursor, właściwości modułu zostaną przejęte z projektu porównawczego do aktualnego projektu (jest to możliwe tylko w tym kierunku!).

„Dodatki” „Zastosuj prawa dostępu”

Polecenie to dostępne jest w trybie porównania, a tam tylko w przeglądzie projektu (patrz wyżej, „Projekt” „Porównaj”).

W przypadku modułu, w którym umieszczony jest kursor, prawa dostępu zostaną przejęte z projektu porównawczego do aktualnego projektu (jest to możliwe tylko w tym kierunku!).

„Projekt” „Kopiuj”

Za pomocą tego polecenia obiekty (moduły, typy danych, wizualizacje i zasoby), jak również powiązania z bibliotekami z innych projektów można skopiować do projektu.

Polecenie to otwiera najpierw standardowe okno dialogowe otwierania pliku. Po wybraniu w nim pliku otwierane jest okno dialogowe, w którym można zaznaczyć żądane obiekty. Wybór odbywa się w taki sam sposób, jak w przypadku polecenia „Projekt” „Dokumentowanie”.

Jeśli w projekcie istnieje już obiekt o takiej samej nazwie, wtedy do nazwy nowego obiektu na końcu dodawany jest dolny myślnik oraz kolejny numer („_1”, „_2” itp.).

„Projekt” „Informacja o projekcie”

W tym punkcie menu można zapisać informacje o projekcie. Użycie polecenia powoduje otwarcie okna dialogowego „Informacja o projekcie”.

Jako informacje o projekcie wyświetlane są następujące dane:

- Nazwa pliku
- Ścieżka do katalogu
- Czas ostatniej zmiany (Zmieniono dnia)

Tych danych nie można zmienić.

Okno dialogowe wprowadzania informacji o projekcie

The screenshot shows a dialog box titled 'Projektinformation'. It has several input fields and buttons. The fields are: 'Dateiname' with value 'proj2.lib', 'Verzeichnis' with value 'C:\Projects', 'Geändert am' with value '7.8.02 13:36:39 / V(null)', 'Bezeichnung' with value 'Projekt 1', 'Autor' with value 'P.Huber', and 'Version' with value '1.1'. The 'Beschreibung' field is a text area containing 'Program controls a traffic light system'. On the right side, there are four buttons: 'OK', 'Abbrechen', 'Statistik...', and 'Lizenzinfo...'.

Ponadto można jeszcze dodać następujące informacje:

- **oznaczenie projektu:**

Należy pamiętać: O ile umożliwiał to system docelowy, wpisane tu oznaczenie proponowane jest automatycznie jako nazwa pliku, gdy projekt zostanie ponownie wczytany do programu CoDeSys za pomocą funkcji „Plik” „Otwórz” „Otwórz projekt ze sterownika” (otwiera się wtedy okno dialogowe zapisu pliku).

- nazwisko **autora**,
- **wersję** oraz
- **opis** projektu.

Dane te są opcjonalne.

Po naciśnięciu przycisku **Statystyka** wyświetlane są informacje statystyczne na temat projektu (patrz nast. rysunek). Zawierają one dane z informacji o projekcie, jak też liczbę **modułów**, **typów danych**, **zmiennych lokalnych** i **globalnych**, jakie zostały zapisane podczas ostatniej kompilacji.

Przycisk **O licencji...** można nacisnąć, jeśli chodzi o projekt CoDeSys, który został zapisany za pomocą polecenia „Plik” „Zapisz jako...” w postaci modułu licencjonowanego. W takim przypadku zostanie otwarte okno dialogowe „Edytuj informacje licencyjne”, w którym można zmodyfikować wzgl. usunąć informację o licencji. Patrz także rozdział 12, „Zarządzanie licencjami w programie CoDeSys”.

W przypadku wybrania opcji **Żądaj informacji o projekcie** w kategorii Wczytywanie i zapisywanie w oknie dialogowym opcji, podczas zapisywania nowego projektu lub podczas zapisywania projektu pod nową nazwą automatycznie zostanie otwarte okno dialogowe informacji o projekcie.


Przykład statystyki projektu



„Projekt” „Szukaj globalnie”

Za pomocą tego polecenia można szukać miejsc występowania danego tekstu w modułach, typach danych w obiektach zmiennych globalnych, w konfiguracji sterownika, w konfiguracji zadań oraz w częściach deklarycyjnych bibliotek. Jeśli zostało wprowadzone polecenie, otworzy się okno dialogowe, w którym można wybierać moduły i obiekty, które mają być przeszukiwane. Wybór odbywa się w taki sam sposób jak w przypadku polecenia „Projekt” „Dokumentowanie”.

Jeśli wybór zostanie potwierdzony przyciskiem **OK**, zostanie wyświetlone standardowe okno dialogowe funkcji szukania. Zostanie ono wyświetlone bezpośrednio, jeśli polecenie „Szukaj globalnie”

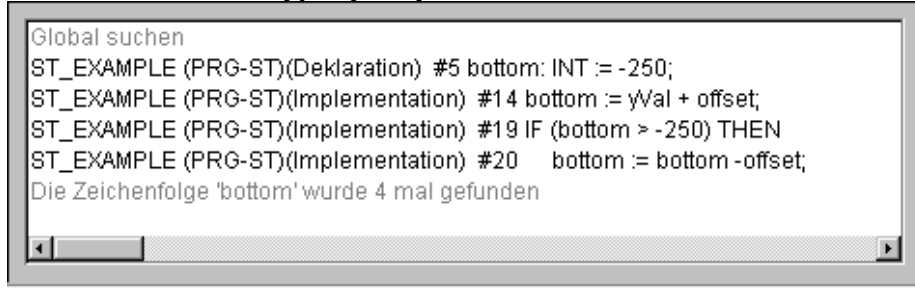
zostało wybrane za pomocą symbolu  z paska menu; wyszukiwanie obejmie wtedy automatycznie wszystkie możliwe do przeszukania elementy projektu. Wprowadzone ostatnio wyszukiwane ciągi znaków można wybrać za pośrednictwem pola listy rozwijanej

Szukaj. Gdy tekst zostanie znaleziony w obiekcie, obiekt wczytywany jest do odpowiedniego edytora wzgl. do Menedżera biblioteki i wyświetlana jest jego lokalizacja. Wyświetlanie znalezionej lokalizacji, jak również szukanie i dalsze przeszukiwanie odbywa się analogicznie do polecenia „Edycja” „Znajdź”.

W przypadku naciśnięcia przycisku **Do okna komunikatu**, wszystkie miejsca użycia szukanego ciągu znaków w wybranych obiektach przedstawiane są jako wiersze w formie tabelarycznej w oknie komunikatów. Na końcu podawana jest liczba znalezionych lokalizacji.

Jeśli okno komunikatów nie było otwarte, zostanie ono teraz wyświetlone. W odniesieniu do każdej znalezionej lokalizacji podawane są następujące informacje:

- Nazwa obiektu
- Lokalizacja w części deklarycyjnej (Decl) lub w części implementacyjnej (Impl) modułu
- Numery wierszy wzgl. sieci
- Kompletny wiersz w przypadku edytorów tekstowych
- Kompletna jednostka tekstowa w przypadku edytorów graficznych

Okno komunikatów zawierające wynik wyszukiwania

Podwójne kliknięcie wiersza w oknie komunikatów lub naciśnięcie klawisza <Enter> powoduje otwarcie edytora wraz z danym obiektem. Zostanie zaznaczony odpowiedni wiersz obiektu. Za pomocą kombinacji klawiszy <F4> i <Shift> + <F4> można szybko przechodzić pomiędzy poszczególnymi wierszami widoku.

„Projekt” „Zastąp globalnie”

Za pomocą tego polecenia można szukać miejsc występowania danego tekstu w modułach, typach danych lub w obiektach zmiennych globalnych w konfiguracji sterownika, w konfiguracji zadań i zastąpić ten tekst innym. Rozpoczęcie procedury i jej przebieg mają taki sam charakter jak w przypadku poleceń „Projekt” „Szukaj globalnie” wzgl. „Edycja” „Zastąp”. Jednakże nie są oferowane biblioteki do wyboru i nie jest możliwe wyświetlanie w oknie komunikatów.

„Projekt” „Sprawdź”

Za pomocą tego polecenia można otworzyć podmenu zawierające polecenia sprawdzania semantycznej poprawności projektu:

- Nieużywane zmienne
- Nakładające się obszary pamięci
- Konflikt dostępu
- Wielokrotny zapis na wyjściu

Wyniki przedstawiane są w oknie komunikatów.

Każda z tych funkcji sprawdza stan ostatniego procesu kompilacji. Jeśli projekt został od tego czasu zmieniony, w oknie komunikatów wyświetlane jest ostrzeżenie. Aby otrzymać aktualny wynik sprawdzenia, należy więc skompilować projekt ponownie.

Wskazówka: Procedury sprawdzające można również zdefiniować w opcjach projektu, w kategorii „Opcje kompilacji”, w taki sposób, aby były uruchamiane automatycznie podczas każdego procesu kompilacji.

Nieużywane zmienne

Ta funkcja menu „Projekt” „Sprawdź” (patrz wyżej) umożliwia wyszukiwanie zmiennych, które są zadeklarowane, jednak nie są używane przez program. Są one wskazywane wraz z nazwą modułu i wiersza, np. PLC_PRG (4) – var1. Nie są uwzględniane zmienne w bibliotekach.

Wyniki przedstawiane są w oknie komunikatów.

Nakładające się obszary pamięci

Ta funkcja menu „Projekt” „Sprawdź” (patrz wyżej) umożliwia sprawdzenie, czy w przypadku przypisania zmiennych za pomocą deklaracji „AT” nie powstaje nakładanie się określonych obszarów pamięci. Na przykład w wyniku przypisania zmiennej „var1 AT %QB21: INT” i „var2 AT %QD5: DWORD” powstaje nakładający się obszar, obejmujący łącznie 21 bajtów. Wyświetlany wynik ma taką postać:

```

Do %QB21 odnoszą się następujące zmienne:
PLC_PRG (3): var1 AT %QB21
PLC_PRG (7): var2 AT %QD5

```

Wyniki przedstawiane są w oknie komunikatów.

Konflikt dostępu

Ta funkcja menu „Projekt” „Sprawdź” (patrz wyżej) umożliwia wyszukiwanie obszarów pamięci adresów IEC, do których odwołuje się więcej niż jedno zadanie. Nie jest przy tym dokonywane rozróżnienie pomiędzy dostępem zapisu i odczytu. Wyświetlany wynik wygląda np. tak:

Do %MB28 odnoszą się następujące zadania:
 Task1 – PLC_PRG (6): %MB28 [Dostęp tylko do odczytu]
 Task2 – POU1.ACTION (1) %MB28 [Dostęp do zapisu]

Wyniki przedstawiane są w oknie komunikatów.

Wielokrotny zapis na wyjściu

Ta funkcja menu „Projekt” „Sprawdź” (patrz wyżej) umożliwia wyszukiwanie obszarów pamięci adresów IEC, do których w projekcie następuje dostęp zapisu z więcej niż jednego miejsca. Wyświetlany wynik ma taką postać:

Zapis %QB24 w następujących miejscach:
 PLC_PRG (3): %QB24
 PLC_PRG.POU1 (8): %QB24

Wyniki przedstawiane są w oknie komunikatów.

Grupy robocze

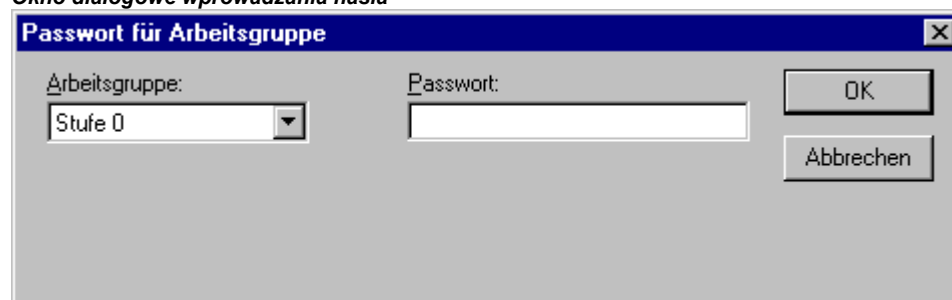
W programie CoDeSys można skonfigurować maks. osiem grup o różnych prawach dostępu do modułów, typów danych, wizualizacji i zasobów. Można ustalić prawa dostępu do pojedynczych lub do wszystkich obiektów. Każde otwarcie projektu związane jest z członkostwem w określonej grupie roboczej. Członek grupy musi dokonać autoryzacji za pomocą hasła.

Grupy robocze mają numerację od 0 do 7, przy czym grupa 0 ma prawa administratora, tzn. tylko członkowie grupy 0 mogą ustalać hasła i prawa dostępu do wszystkich grup wzgl. obiektów.

W przypadku utworzenia nowego projektu wszystkie hasła są początkowo puste. Dopóki dla grupy 0 nie zostanie ustalone hasło, każdy użytkownik ma dostęp do projektu automatycznie jako członek grupy 0.

Jeśli podczas wczytywania projektu zostanie ustalone hasło dla grupy roboczej 0, wtedy podczas otwierania projektu będzie wymagane wprowadzenie hasła dla wszystkich grup. Zostanie w tym celu wyświetlone następujące okno dialogowe:

Okno dialogowe wprowadzania hasła



W polu listy rozwijanej **Grupa robocza** po lewej stronie okna dialogowego należy ustawić grupę, do której należy użytkownik, a po prawej stronie wprowadzić odpowiednie

hasło. Nacisnąć przycisk **OK**. Jeśli hasło nie jest zgodne z zapisanym hasłem, zostanie wyświetlony komunikat:

„Hasło jest nieprawidłowe.”

Projekt zostanie otwarty dopiero po prawidłowym wprowadzeniu hasła.

Uwaga: Jeśli hasła nie zostaną nadane wszystkim grupom roboczym, wtedy projekt można otworzyć za pośrednictwem grupy, dla której nie ustalono hasła!

Za pomocą polecenia „Hasła dla grupy roboczej” można nadawać hasła, zaś polecenia „Obiekt” „Prawa dostępu” umożliwiają nadanie praw dla poszczególnych lub dla wszystkich obiektów.

„Projekt” „Hasła dla grupy roboczej”

Za pomocą tego polecenia można otworzyć okno dialogowe w celu wprowadzenia haseł dla grup roboczych. Polecenie to mogą wykonywać tylko członkowie grupy roboczej 0. Użycie polecenia powoduje otwarcie następującego okna dialogowego:

Okno dialogowe nadawania hasła grupom roboczym

W lewym polu listy rozwijanej Grupa robocza można wybrać grupę. Żądane hasło dla tej grupy należy wprowadzić w polu **Hasło**. Zamiast każdej wpisanej litery w polu wyświetlany jest symbol gwiazdki (*). To samo słowo należy wprowadzić w polu

Potwierdź hasło. Okno dialogowe należy po każdorazowym wprowadzeniu hasła zamknąć przyciskiem **OK**. W przypadku błędnego potwierdzenia zostanie wyświetlony komunikat:

„Hasło oraz jego potwierdzenie nie są zgodne.”,

i okno dialogowe pozostanie otwarte, aż do zamknięcia w wyniku wprowadzenia prawidłowego wpisu wzgl. użycia przycisku **Anuluj**.

Polecenie nadania hasła należy wywołać ponownie dla kolejnej grupy roboczej.

Uwaga: Jeśli hasła nie zostaną nadane wszystkim grupom roboczym, wtedy projekt można otworzyć za pośrednictwem grupy, dla której nie ustalono hasła!

Za pomocą polecenia „Obiekt” „Prawa dostępu” można nadawać prawa dla poszczególnych lub wszystkich obiektów.

W związku z ochroną projektu należy zwrócić ponadto uwagę na następujące rozdziały:

- Hasła ochrony dostępu i zapisu (rozd. 6.2, Opcje haseł)
- Szyfrowanie projektu podczas zapisu (rozd. 6.3, „Plik” „Zapisz jako”).

„Projekt” „Baza danych projektu”

Ten punkt menu jest dostępny, gdy w opcjach projektu, w kategorii „Baza danych projektu” została włączona opcja „Zastosuj bazę danych projektu (ENI)”. Umożliwia on dostęp do podmenu zawierającego polecenia dotyczące zarządzania obiektami wzgl. projektem w bazie danych podłączonej aktualnie poprzez interfejs ENI (patrz rozdział 10):

- Login (logowanie użytkownika do serwera ENI)

Jeśli obiekt jest zaznaczony w Organizercie obiektów i wybrano polecenie „Baza danych projektu” z **menu podręcznego** (prawy przycisk myszy), dla obiektu tego można za pomocą poniższych poleceń wywołać odpowiednie funkcje bazy danych. Jeśli uprzednio nie zostało wykonane logowanie użytkownika w ENI za pośrednictwem okna dialogowego **Logowanie do bazy danych**, najpierw automatycznie otwierane jest to okno, a polecenie jest wykonywane dopiero po pomyślnej weryfikacji uprawnień:

- Ustal
- Przywołaj
- Wyrejestruj

- Zarejestruj
- Cofnij wyrejestrowanie
- Pokaż różnice
- Pokaż historię wersji

Jeśli polecenie „Baza danych projektu” zostało wybrane w menu „Projekt”, zostaną wyświetlone dodatkowe punkty menu, odnoszące się do wszystkich obiektów projektu:

- Ustal wielokrotnie
- Przywołaj wszystko
- Wyrejestrowanie wielokrotne
- Rejestrowanie wielokrotne
- Cofnij wyrejestrowanie wielokrotne
- Historia wersji projektu
- Wersja etykiet
- Wstaw wspólne obiekty
- Odśwież status

Wyświetlanie statusu obiektów w odniesieniu do zarządzania w bazie danych projektu w Organizeryze obiektów:

	<p><u>Ikona z szarym cieniowaniem:</u> Zarządzanie obiektem poprzez bazę danych.</p> <p><u>Zielony znacznik</u> przed nazwą obiektu: Obiekt został wyrejestrowany z otwartego aktualnie projektu CoDeSys.</p> <p><u>Czerwony krzyżyk</u> przed nazwą obiektu: Obiekt jest aktualnie wyrejestrowany przez innego użytkownika.</p> <p><u><R></u> za nazwą obiektu Obiekt umożliwia dostęp jedynie do odczytu.</p>
	<p>Należy pamiętać:</p> <p>Niektóre obiekty (konfiguracja zadań, konfiguracja śledzenia, konfiguracja sterownika, ustawienia systemu docelowego, Menedżer podglądu zmiennych i receptur) są z zasady oznaczone symbolem <R>, dopóki nie są wyrejestrowane. W takim przypadku oznacza to, że nie zostanie wyświetlone automatyczne zapytanie „... Czy wyrejestrować obiekt?”, jeśli została rozpoczęta edycja obiektu; nie oznacza to jednakże automatycznie, że nie jest możliwy dostęp w celu zapisu. Ten ostatni przypadek można rozpoznać po tym, że nie można wybrać polecenia „Wyrejestruj”.</p>

Ustal

Polecenie: „Projekt” „Baza danych projektu” „Ustal”

Można ustalić, czy obiekt zaznaczony w Organizeryze obiektów będzie zarządzany przez bazę danych czy też tylko lokalnie (w projekcie). Wyświetlane jest w tym celu okno dialogowe, w którym można wybrać jedną z dwu kategorii bazy danych: „Projekt”, lub „Obiekty wspólne” lub kategorię „Lokalnie”. Patrz również rozdział 10.1.4, „Kategorie w obrębie bazy danych projektu”.

Symbole wszystkich obiektów, które są administrowane w bazie danych, wyświetlane są w Organizeryze obiektów z szarym cieniowaniem. Obiekty wspólne wyświetlane są przy użyciu turkusowej czcionki.

Przywołaj

Polecenie: „Projekt” „Baza danych projektu” „Przywołaj”

Aktualna wersja obiektu zaznaczonego w Organizeryze obiektów zostaje przywołana z bazy danych i zastępuje ona wersję lokalną. W przeciwieństwie do wyrejestrowania (patrz niżej) obiekt w bazie danych nie jest blokowany dla edycji przez innych użytkowników.

Wyrejestruj

Polecenie: „Projekt” „Baza danych projektu” „Wyrejestruj”

Obiekt zaznaczony w organizerze obiektów jest wyrejestrowywany z bazy danych i tym samym jego edycja przez innych użytkowników jest zablokowana.

W przypadku wywołania polecenia zostanie otwarte okno dialogowe „Wyrejestruj plik”: Można wprowadzić komentarz, który zostanie zapisany razem z historią wersji obiektu w bazie danych podczas procesu wyrejestrowywania. Podziały wierszy można wstawiać za pomocą kombinacji <Ctrl> + <Enter>.

Po potwierdzeniu okna dialogowego za pomocą przycisku „OK” wyrejestrowany obiekt oznaczany jest w Organizерze obiektów za pomocą zielonego znacznika, innym użytkownikom wyświetlany jest czerwony krzyżyk, oznaczający, że nie mogą go oni edytować.

Zarejestruj

Polecenie: „Projekt” „Baza danych projektu” „Zarejestruj”

Obiekt zaznaczony w organizerze obiektów zostanie zarejestrowany w bazie danych. Tym samym w bazie danych zostanie utworzona nowa wersja obiektu. Stare wersje zostaną zachowane.

W przypadku wywołania polecenia zostanie otwarte okno dialogowe „Zarejestruj plik”: Można wprowadzić komentarz, który zostanie zapisany razem z historią wersji obiektu w bazie danych podczas procesu wyrejestrowywania. Podziały wierszy można wstawiać za pomocą kombinacji <Ctrl> + <Enter>. Jeśli wersja obiektu w bazie danych różni się od wersji w projekcie lokalnym, wyświetlany jest odpowiedni komunikat i użytkownik może zdecydować, czy obiekt pomimo to powinien zostać zarejestrowany.

Po potwierdzeniu okna dialogowego za pomocą przycisku „OK” znika zielony znacznik przed nazwą modułu w Organizерze obiektów.

Cofnij wyrejestrowanie

Polecenie: „Projekt” „Baza danych projektu” „Cofnij wyrejestrowanie”

Wyrejestrowanie obiektu zaznaczonego w Organizерze obiektów oraz zmiany wprowadzone lokalnie w tym obiekcie zostaną cofnięte. Nie jest wyświetlane żadne okno dialogowe. Obiekt pozostaje w wersji niezmienionej i jest udostępniany w bazie danych do edycji przez innych użytkowników. Znika czerwony znacznik przed nazwą modułu w Organizерze obiektów.

Pokaż różnice

Polecenie: „Projekt” „Baza danych projektu” „Pokaż różnice”

Moduł, otwarty aktualnie do edycji w programie CoDeSys, jest wyświetlany w podzielonym na dwie części oknie, w którym edytowana wersja lokalna zestawiona jest z ostatnią aktualną wersją z bazy danych. Różnice wersji są wyróżniane optycznie w sposób podobny jak w przypadku porównania projektów (patrz „Projekt” „Porównaj”).

Pokaż historię wersji

Polecenie: „Projekt” „Baza danych projektu” „Pokaż historię wersji”

Zostanie otwarte okno dialogowe „Historia wersji <nazwa obiektu>”, zawierające tabelaryczną listę wszystkich wersji, które w przypadku aktualnie edytowanego obiektu zostały zarejestrowane w bazie danych wzgl. otrzymały etykietę. Podawane informacje to:

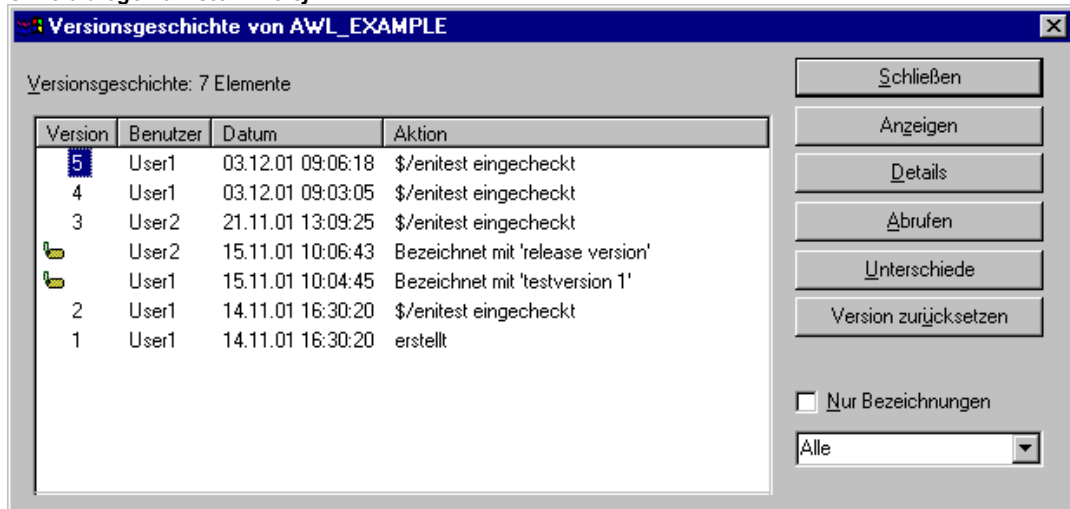
Wersja: Zależna od bazy danych numeracja wersji obiektu zarejestrowanych w kolejności czasowej. Wersje oznaczone etykietą nie otrzymują numeru wersji, lecz oznaczenie w postaci ikony etykiety.

Użytkownik: Nazwa użytkownika, który wykonał czynność dotyczącą obiektu.

Data: Data i czas czynności.

Czynność: Rodzaj czynności wykonanej w związku z obiektem. W zależności od bazy danych, np.: „utworzony” (obiekt został po raz pierwszy zarejestrowany w bazie danych), „zarejestrowany” lub „oznaczony <etykieta>” (ta wersja obiektu została opatrzona identyfikatorem).

Okno dialogowe historii wersji



Przyciski:

- **Zamknij:** Zamykanie okna dialogowego
- **Pokaż:** Wersja zaznaczona w tabeli otwierana jest w oknie programu CoDeSys. Pasek tytułu zawiera wpis „ENI: <nazwa projektu w bazie danych>/<nazwa obiektu>”
- **Szczegóły:** Otwiera się okno dialogowe „Szczegóły historii wersji”, zawierające następujące informacje: **Plik** (nazwa projektu i obiektu w bazie danych), **Wersja** (patrz wyżej), **Data** (patrz wyżej), **Użytkownik** (patrz wyżej), **Komentarz** (komentarz nadawany podczas rejestrowania wzgl. przypisywania etykiety). Za pomocą przycisków **Następna** wzgl. **Poprzednia** można przejść do szczegółów następnego lub poprzedniego wpisu w oknie dialogowym „Historia wersji ...”.
- **Przywołaj:** Wersja zaznaczona w tabeli wczytywana jest z bazy danych do programu CoDeSys i zastępuje wersję lokalną.
- **Różnice:** Jeśli w tabeli zostanie zaznaczona tylko jedna wersja obiektu, polecenie to sprawia, iż porównywana jest ona z aktualną wersją z bazy danych. W przypadku zaznaczenia dwóch wersji nastąpi ich porównanie. Różnice wyświetlane są w oknie podzielonym na dwie części podobnie jak w przypadku porównania projektów.
- **Poprzednia wersja:** Wersja zaznaczona w tabeli zostanie zastąpiona przez aktualną wersję z bazy danych. Wersje wstawione później zostaną usunięte! Polecenia tego można użyć w celu przywrócenia poprzedniego stanu i nadania mu statusu stanu aktualnego.
- **Tylko nazwy:** Jeśli opcja ta jest aktywna, do wyboru w tabeli będą wyświetlane tylko wersje opatrzone etykietą.
- Pole wyboru poniżej opcji „Tylko nazwy”: Tutaj wymienione są nazwy wszystkich użytkowników, którzy już wykonali w bazie danych czynności związane z obiektami projektu. Należy wybrać „Wszystkie” lub też jedną z nazw, aby otrzymać historię wersji dla wszystkich obiektów lub tylko dla obiektów edytowanych przez określonych użytkowników.

Ustal wielokrotnie

Polecenie „Projekt” „Baza danych projektu” „Ustal wielokrotnie”

Za pomocą tego polecenia można ustalić równocześnie dla wielu obiektów aktualnego projektu kategorię bazy danych, w jakiej będą one administrowane. Początkowo zostanie wyświetlone takie samo okno dialogowe „Właściwości obiektu” jak w przypadku polecenia „Ustal”. Tu można wybrać żadaną kategorię i zamknąć okno dialogowe za pomocą polecenia **OK**. Następnie zostanie wyświetlone okno dialogowe „**Wybór ENI**”, zawierające listę modułów projektu, dostępnych w przypadku ustawionej kategorii (np. w przypadku ustawionej kategorii „Zasoby” dostępne będą do wyboru tylko moduły zasobów projektu). Widok odpowiada strukturze drzewa używanej w Organizерze obiektów. Wybrane moduły należy zaznaczyć i potwierdzić za pomocą przycisku **OK**.

Przywołaj wszystko

Polecenie „Projekt” „Baza danych projektu” „Przywołaj wszystko”

W przypadku otwartego projektu aktualna wersja wszystkich obiektów kategorii Projekt wywoływana jest z bazy danych. Jeśli do bazy danych dodano obiekty, zostaną one dodane również lokalnie, jeśli natomiast w bazie danych usunięto obiekty, nie zostaną one usunięte lokalnie, lecz automatycznie przyporządkowane do kategorii „Lokalne”. W przypadku obiektów o kategorii „Zasoby” z bazy danych zostaną wywołane tylko te, które są już utworzone w lokalnym projekcie. Znaczenie polecenia opisano pod hasłem „Przywołaj”.

Wyrejestrowanie wielokrotne

Polecenie „Projekt” „Baza danych projektu” „Wyrejestrowanie wielokrotne”

Można wyrejestrować równocześnie wiele obiektów. W tym celu wyświetlane jest okno dialogowe „Wybór ENI”, zawierające strukturę drzewa obiektów projektu, podobnie jak w Organizерze obiektów. Moduły przeznaczone do wyrejestrowania należy zaznaczyć i potwierdzić za pomocą przycisku OK. Znaczenie polecenia opisano pod hasłem „Wyrejestruj”.

Rejestrowanie wielokrotne

Polecenie „Projekt” „Baza danych projektu” „Rejestrowanie wielokrotne”

Można wyrejestrować równocześnie wiele obiektów. Sposób postępowania dopowiada wielokrotnemu wyrejestrowaniu. Znaczenie polecenia opisano pod hasłem „Zarejestruj”.

Cofnij wyrejestrowanie wielokrotne

Polecenie „Projekt” „Baza danych projektu” „Cofnij wyrejestrowanie wielokrotne”

Wyrejestrowanie można cofnąć równocześnie w odniesieniu do wielu obiektów. Wybór odbywa się w taki sam sposób jak w przypadku polecenia „Wyrejestrowanie wielokrotne” „Rejestrowanie wielokrotne”.

Historia wersji projektu

Polecenie „Projekt” „Baza danych projektu” „Historia wersji projektu”

Polecenie to należy wybrać w celu uzyskania wglądu w historię wersji aktualnego projektu.

Zostanie wyświetlone okno dialogowe „Historia wersji <nazwa projektu w bazie danych>”, zawierające chronologiczną kolejność czynności (tworzenie, rejestrowanie, nadawanie etykiet) dotyczących obiektów należących do projektu. Liczba tych obiektów podawana jest za

historią wersji. Informacje związane z obsługą okna dialogowego opisano niżej pod hasłem „Historia wersji” w odniesieniu do pojedynczego obiektu, jednakże należy mieć na uwadze, że

- polecenie „Poprzednia wersja” jest dostępne tylko w przypadku pojedynczych obiektów.
- Polecenie „Przywołaj” oznacza, że wszystkie obiekty z wersji projektu zaznaczonej w tabeli zostaną wczytane do lokalnego projektu. Oznacza to, że obiekty lokalne zostaną zastąpione przez starszą wersję. Obiekty lokalne, których jeszcze nie zawiera starsza wersja projektu, nie zostaną jednak usunięte z wersji lokalnej! Jeśli wywołana zostanie wersja z etykietą, która zawiera również obiekty wspólne, użytkownik otrzymuje w oknie dialogowym możliwość wyboru, czy mają być one również wywołane, czy też nie.

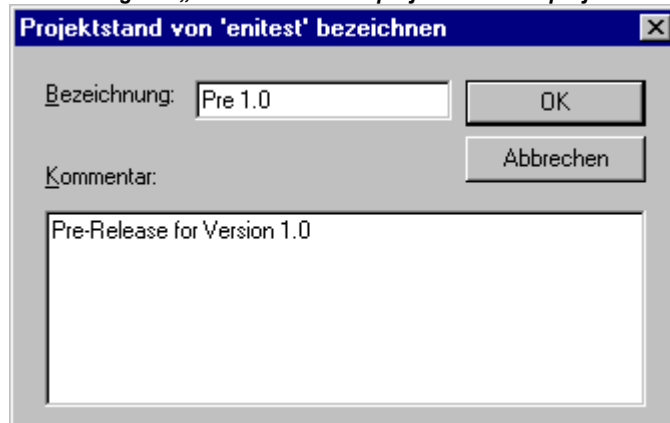
Wersja etykiet

Polecenie „Projekt” „Baza danych projektu” „Wersja etykiet projektu”

Polecenie to służy do grupowania pod jednym oznaczeniem aktualnego stanu obiektów, dzięki czemu możliwe jest później przywołanie tego właśnie stanu. Wyświetlane jest okno dialogowe „Stan projektu <nazwa projektu w bazie danych>”. Należy wprowadzić **oznaczenie** (etykietę) dla statusu projektu i opcjonalnie **komentarz**. Po potwierdzeniu za pomocą przycisku „OK” okno dialogowe zostanie zamknięte i oznaczenie oraz czynność etykietowania („oznaczone jako...”) zostaną wyświetlone w tabeli historii wersji zarówno pojedynczego obiektu, jak też projektu. Etykieta ta zostanie nadana również wspólnym obiektom projektu. Wersja z etykietą nie posiada numeru wersji, lecz można ją

rozpoznać po ikonie etykiety w kolumnie „Wersja”. Jeśli jest włączona opcja „Tylko nazwy”, wyświetlane są tylko wersje z etykietą.

Okno dialogowe „Oznaczanie stanu projektu <nazwa projektu>”



Wstaw wspólne obiekty

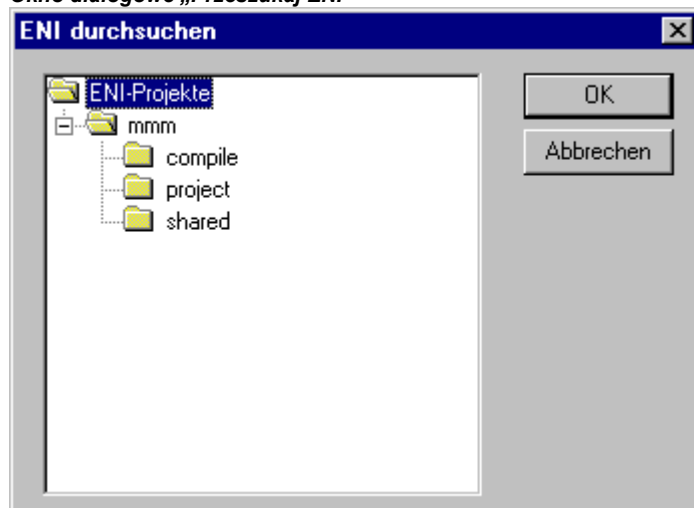
Polecenie „Projekt” „Baza danych projektu” „Wstaw wspólne obiekty”

Polecenie to służy do włączania do otwartego lokalnie projektu dodatkowych obiektów kategorii „Obiekty wspólne”, jakie dostępne są w bazie danych. W przypadku obiektów kategorii „Projekt” nie jest to konieczne, ponieważ w przypadku polecenia „Przywołaj wszystko” do projektu lokalnego wczytane zostaną wszystkie istniejące aktualnie obiekty bazy danych – również takie, które jeszcze nie zostały tam utworzone. W przypadku obiektów o kategorii „Wspólne” uwzględniane są po użyciu polecenia „Przywołaj wszystko” tylko obiekty włączone już do projektu.

Dodatkowy obiekt można wstawić w następujący sposób:

Polecenie otwiera okno dialogowe „Przeszukaj ENI”, zawierające listę wszystkich obiektów, które znajdują się we wskazanym z lewej strony katalogu projektu w bazie danych projektu. Należy wybrać żądane zasoby i nacisnąć przycisk **OK** lub kliknąć dwukrotnie. W ten sposób obiekt zostanie wstawiony do otwartego lokalnie projektu.

Okno dialogowe „Przeszukaj ENI”



Odśwież status

Polecenie „Projekt” „Baza danych projektu” „Odśwież status”

Polecenie to aktualizuje widok w organizerze obiektów, dzięki czemu wyświetlany jest aktualny status obiektów w odniesieniu do bazy danych.

Login

Polecenie to otwiera okno dialogowe logowania do bazy danych, w którym użytkownik musi zalogować się do serwera ENI w przypadku każdej kategorii bazy danych, aby uzyskać połączenie projektu z aktualną bazą danych. Dane dostępu muszą być więc dostępne zarówno na serwerze ENI (administracja ENI, zarządzanie użytkownikami), jak również w razie potrzeby w informacjach o zarządzaniu użytkownikami w bazie danych. Po wykonaniu polecenia otwiera się najpierw okno dialogowe logowania do kategorii „Obiekty projektu”.

Okno dialogowe „Login”

Wyświetlane są następujące elementy:

Baza danych: Obiekty projektu

Host: adres komputera serwera ENI (host), podany również w opcjach projektu / kategoria „Baza danych projektu” w polu „Adres TCP/IP”.

Projekt: nazwa projektu w bazie danych (patrz również w opcjach projektu, kategoria „Baza danych projektu”, obiekty projektu, pole „Nazwa projektu”)

Podaj **Nazwę użytkownika** i **Hasło** w obszarze **Legitymacja**. By zalogować się jako „Użytkownik anonimowy”, należy pozostawić pole „Nazwa użytkownika” puste.

W celu potwierdzenia wprowadzonych danych należy nacisnąć przycisk „OK”. Okno dialogowe obiektów projektu zostanie wtedy zamknięte i zostanie automatycznie wyświetlone okno dialogowe logowania dla kategorii „Obiekty wspólne”. Również tutaj należy wprowadzić odpowiednie dane dostępu, potwierdzić je za pomocą przycisku „OK” i postępować tak samo w przypadku trzeciego okna dialogowego logowania, dotyczącego kategorii „Pliki kompilacji”.

Okno dialogowe logowania otworzy się automatycznie w przypadku próby dostępu do bazy danych, zanim jeszcze użytkownik wykaże swe uprawnienia w opisany powyżej sposób.

Wskazówka: Jeśli wprowadzone tu dane dostępu do bazy danych mają być zapisane razem z projektem, w opcjach projektu, w kategorii „Wczytywanie i zapisywanie” należy włączyć opcję „Zapisywanie danych dostępu ENI”.

6.4 Zarządzanie obiektami...

Obiekt

Jako „obiekt” określane są moduły, typy danych, wizualizacje i zasoby (zmienne globalne, konfiguracja zmiennych, zapis śledzenia, konfiguracja sterownika, konfiguracja zadań, menedżer podglądu zmiennych i receptur). Foldery wstawione w celu stworzenia struktury projektu są częściowo również uwzględniane. Wszystkie obiekty projektu dostępne są w organizerze obiektów.





Po zatrzymaniu kursora myszy na krótki czas nad modulem w organizerze obiektów na etykiecie wyświetlany jest rodzaj modułu (program, funkcja lub blok funkcji); w przypadku zmiennych globalnych jest to słowo kluczowe (VAR_GLOBAL, VAR_CONFIG).

Dodatkowe symbole za lub przed wpisami obiektów oznaczają określone stany odnośnie zmian online oraz podłączenia ENI do bazy danych (patrz rozdz. 10, Powiązanie z bazą danych).

Za pomocą techniki „przeciągnij i upuść” można przenosić obiekty (a także foldery, patrz „Folder”) w obrębie rodzaju obiektu. Należy w tym celu wybrać obiekt i przenieść go w żądane miejsce, przytrzymując naciśnięty lewy przycisk myszy. Jeśli w przypadku przeniesienia wystąpi konflikt nazw, nowo wstawiony element oznaczany jest jednoznacznie przez dołączany kolejny numer (np. „Obiekt_1”).

Folder

Aby zachować przejrzystą strukturę większych projektów, należy grupować moduły, typy danych i wizualizacje oraz zmienne globalne w odpowiednich folderach.

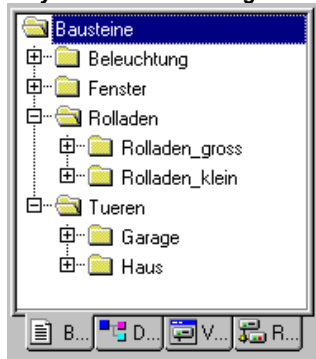
Możliwe jest tworzenie dowolnie głębokiej struktury poziomów folderów. Jeśli przed zamkniętym symbolem folderu znajduje się znak plus  , folder ten zawiera obiekty i/lub inne foldery. Za pomocą kliknięcia znaku plus można rozwinąć strukturę folderu i wyświetlić obiekty podporządkowane. Kliknięcie znajdującego się teraz przed nim znaku minus   powoduje ponowne zwinięcie widoku. W menu podręcznym dostępne są polecenia „Rozwiń węzły” oraz „Zwiń węzły”, które pełnią tę samą funkcję.

Foldery można przenosić techniką „przeciągnij i upuść”. Należy w tym celu wybrać folder i przenieść go w żądane miejsce przytrzymując naciśnięty lewy przycisk myszy. Jeśli w przypadku przeniesienia wystąpi konflikt nazw, nowo wstawiony element oznaczany jest jednoznacznie przez dołączany kolejny numer (np. „Nowy folder_1” lub „Obiekt_1”).

Kolejne foldery można wstawić za pomocą polecenia „**Nowy folder**”.

Wskazówka: Foldery nie mają żadnego wpływu na program, lecz służą jedynie do przejrzystego uporządkowania projektu.

Przykład folderów w Organizерze obiektów



„Nowy folder”

Za pomocą tego polecenia można wstawić nowy folder jako obiekt porządkowy. Jeśli został zaznaczony folder, nowy folder zostanie utworzony pod nim, a w przeciwnym razie na tym samym poziomie. Jeśli zaznaczono czynność, nowy folder zostanie wstawiony na poziomie modułu, do którego należy czynność.

Menu podręczne organizera obiektów, zawierające to polecenie, zostanie wyświetlone, gdy zaznaczony zostanie obiekt lub rodzaj obiektu i gdy zostanie naciśnięty prawy przycisk myszy lub kombinacja <Shift> + <F10>.

Nowy folder otrzymuje początkowo nazwę „Nowy folder”. Należy zwrócić uwagę na następującą konwencję nazw folderów:

- Foldery, które znajdują się na tym samym poziomie hierarchii, muszą mieć różne nazwy. Foldery na różnych poziomach mogą mieć takie same nazwy.
- Folder nie może mieć takiej samej nazwy jak obiekt, który znajduje się na tym samym poziomie.

Jeśli na tym samym poziomie znajduje się już folder o nazwie „Nowy folder”, każdy dodatkowy folder o tej nazwie otrzyma automatycznie kolejny numer (np. „Nowy folder 1”). Zmiana nazwy na już używaną nie jest możliwa.

„Rozwiń węzły” „Zwiń węzły”

Za pomocą polecenia „Rozwiń” można rozwinąć widok wszystkich obiektów, które znajdują się pod wybranym obiektem, natomiast polecenie „Zwiń” sprawia, że elementy podrzędne nie są widoczne.

Strukturę folderów można rozwijać i związać również poprzez dwukrotne kliknięcie lub naciśnięcie klawisza <Enter>.

Menu podręczne organizera obiektów, zawierające to polecenie, zostanie wyświetlone, gdy zaznaczony zostanie obiekt lub rodzaj obiektu i gdy zostanie naciśnięty prawy przycisk myszy lub kombinacja <Shift> + <F10>.

„Projekt” „Usuń obiekt”

Kombinacja klawiszy:

Za pomocą tego polecenia można usunąć z organizera obiektów, a tym samym również z projektu, zaznaczony obiekt (moduł, typ danych, wizualizację lub zmienne globalne) lub też folder ze znajdującymi się poniżej obiektami. Usunięcie można cofnąć za pomocą poleceń menu „Edycja” „Cofnij”.

Jeśli otwarte było okno edytora obiektu, zostanie ono automatycznie zamknięte.

W przypadku zastosowania do usunięcia polecenia „Edycja” „Wytnij”, obiekt dodatkowo przenoszony jest do schowka.

„Projekt” „Wstaw obiekt”

Kombinacja klawiszy: <Ins>

Za pomocą tego polecenia można utworzyć nowy obiekt. Rodzaj obiektu (moduł, typ danych, wizualizacja lub zmienne globalne) zależy od karty wybranej w organizerze obiektów. Należy przy tym pamiętać, że w razie potrzeby dla wybranego typu obiektu zostanie zastosowany zdefiniowany szablon. Jest to możliwe w przypadku obiektów typu „Zmienne globalne”, „Typ pliku”, „Funkcja”, „Moduł funkcji” lub „Program” – patrz niżej, rozdział „Zapisz jako szablon”.

W wyświetlonym oknie dialogowym należy wpisać **nazwę** nowego obiektu.

Należy przy tym przestrzegać następujących ograniczeń:

- Nazwa modułu nie może zawierać znaku spacji.
- Moduł nie może mieć takiej samej nazwy, jak inny moduł lub typ danych ani jak wizualizacja, ponieważ może to prowadzić do powstawania problemów w przypadku zmiany wizualizacji.
- Typ danych nie może zawierać takiej samej nazwy jak inny typ danych, wzgl. moduł.

- Lista zmiennych globalnych nie może mieć tej samej nazwy co inna lista zmiennych globalnych.
- Czynność nie może mieć tej samej nazwy co inna czynność tego samego modułu.
- Wizualizacja nie może mieć takiej samej nazwy jak inna wizualizacja ani jak moduł, ponieważ może to prowadzić do powstawania problemów w przypadku zmiany wizualizacji.

We wszystkich innych przypadkach takie same nazwy są dozwolone. Na przykład czynności różnych modułów mogą mieć te same nazwy, a wizualizacje mogą mieć takie same nazwy jak moduły.

W przypadku modułu należy dodatkowo wybrać typ modułu (program, funkcja lub blok funkcji) oraz język, w którym ma zostać zaprogramowany. Jako **typ modułu** domyślnie ustawiony jest „Program”, jako **język modułu** język ostatnio utworzonego modułu. Jeśli ma być utworzony moduł typu Funkcja, w polu tekstowym **Typ zwrotu** należy wprowadzić żądany typ danych. Dozwolone są przy tym wszystkie podstawowe i zdefiniowane typy danych (tablice, struktury, enumeracje, aliasy). Można stosować funkcję pomocy przy wpisywaniu (np. za pomocą klawisza <F2>).

Okno dialogowe tworzenia nowego modułu



Po potwierdzeniu wpisu za pomocą przycisku **OK**, co jest możliwe tylko wtedy, gdy została zachowana wskazana wyżej konwencja nazewnictwa, nowy obiekt zostanie utworzony w Organizерze obiektów i zostanie wyświetlone odpowiednie okno wprowadzania.

W przypadku zastosowania polecenia „Edycja” „Wklej”, znajdujący się w schowku obiekt zostanie wstawiony bez wyświetlania okna dialogowego. Jeśli nazwa wstawionego obiektu nie jest zgodna z konwencją nazewnictwa (patrz wyżej), do nazwy zostanie dodany po dolnym myślniku kolejny numer (np. „skrętwprawo_1”), nadając jej jednoznaczną formę.

Jeśli projekt jest połączony z bazą danych projektu za pośrednictwem interfejsu **ENI**, połączenie to można skonfigurować w taki sposób, aby podczas tworzenia nowego obiektu wyświetlane było pytanie, w jakiej kategorii bazy danych powinien on być administrowany. W takim przypadku zostanie wyświetlone okno dialogowe „Właściwości obiektu” w celu wybrania kategorii bazy danych. Patrz także rozdział 6.2, opis opcji projektu dla bazy danych projektu.

„Zapisz jako szablon”

Obiekty typu „Zmienne globalne”, „Typ pliku”, „Funkcja”, „Moduł funkcji” lub „Program” można zapisać jako szablon modułu. Należy w tym celu zaznaczyć obiekt w Organizерze obiektów i wybrać z menu podręcznego (prawy przycisk myszy) polecenie „Zapisz jako szablon”. W wyniku tego podczas wstawiania kolejnego obiektu tego samego typu stosowana jest część deklaracyjna szablonu initial. Stosowany jest zawsze ostatni utworzony szablon dla danego typu danych.

„Projekt” „Zmień nazwę obiektu”

Kombinacja klawiszy: <Spacja>

Za pomocą tego polecenia można nadać wybranemu aktualnie obiektowi lub folderowi nową nazwę. Należy przy tym przestrzegać zaleceń dotyczących jednoznaczności nazwy (patrz „Wstaw obiekt”). W przypadku ich naruszenia nie można zamknąć okna dialogowego za pomocą przycisku **OK**.

Jeśli otwarte było okno edycji obiektu, jego tytuł zmieni się automatycznie po zmianie nazwy obiektu.

Okno dialogowe zmiany nazwy modułu



„Projekt” „Konwertuj obiekt”

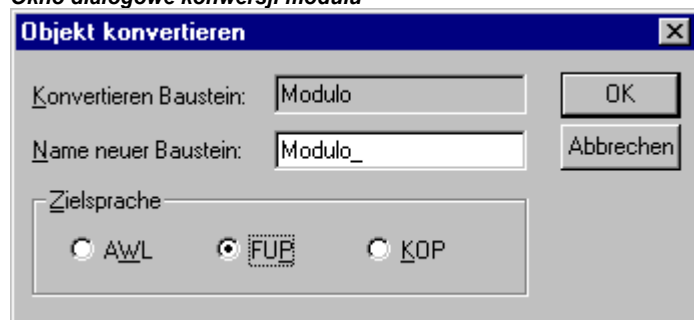
Polecenie to można stosować tylko w stosunku do modułów. Moduły w językach ST, FBD, LD i IL można skonwertować na jeden z trzech języków: IL, FBD i LD.

W tym celu projekt musi być skompilowany. Należy wybrać język, na który ma zostać przeprowadzona konwersja, i nadać nowemu modułowi nową nazwę. Należy pamiętać, że nowa nazwa modułu nie może być nazwą używaną już wcześniej. Następnie można nacisnąć przycisk **OK** i nowy moduł zostanie dodany do listy modułów.

Sposób przetwarzania podczas procesu konwersji odpowiada temu, jaki obowiązuje w przypadku kompilacji.

Uwaga: Czynności nie można konwertować.

Okno dialogowe konwersji modułu



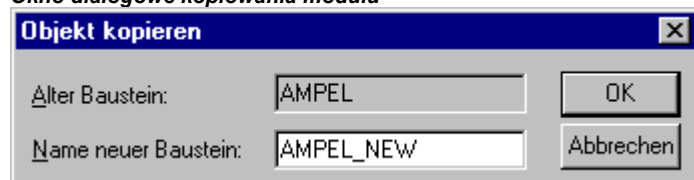
Należy mieć na uwadze następującą możliwość: Moduł, który został zaprogramowany w języku FBD, można prezentować i edytować za pomocą polecenia „Dodatki” „Widok” zarówno online, jak i offline także w edytorze LD bez wykonywania konwersji.

„Projekt” „Kopiuj obiekt”

Za pomocą tego polecenia można skopiować wybrany obiekt i zapisać go pod nową nazwą. W wyświetlonym oknie dialogowym należy wprowadzić nazwę nowego obiektu. Należy zwrócić uwagę na to, aby nazwa ta nie była używana już wcześniej, chyba że chodzi o czynność.

Natomiast w przypadku zastosowania polecenia „Edycja” „Kopiuj”, obiekt zostanie skopiowany do schowka i zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe kopiowania modułu



„Projekt” „Edytuj obiekt”

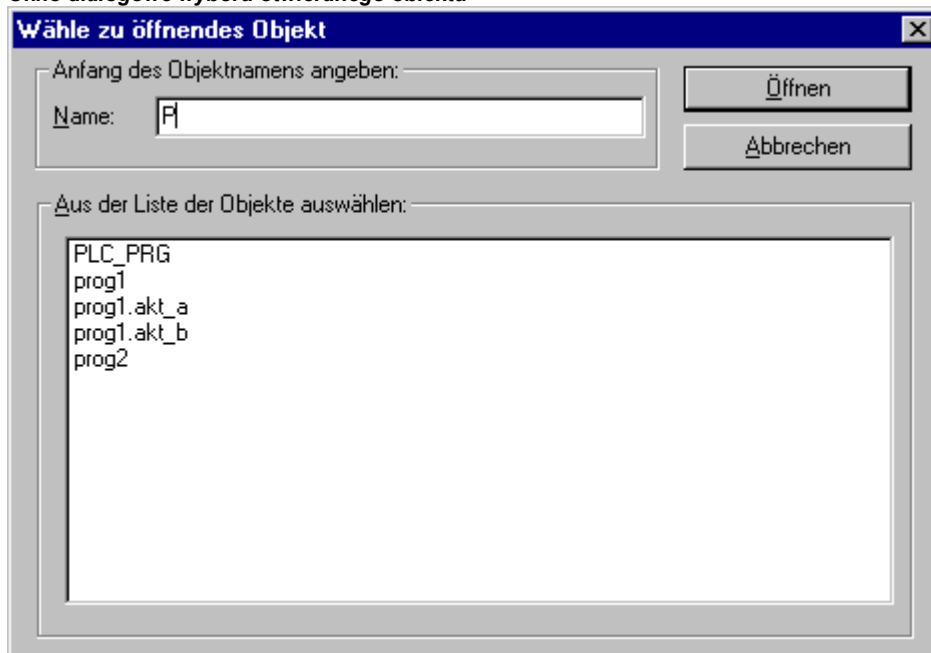
Kombinacja klawiszy: <Enter>

Za pomocą tego polecenia można wczytać do danego edytora obiekt zaznaczony w Organizery obiektów. Jeśli otwarte jest już okno z tym obiektem, staje się ono aktywne, tzn. wysuwa się na pierwszy plan i można je edytować.

Istnieją jeszcze dwie inne możliwości edytowania obiektu:

- Dwukrotne kliknięcie żądanego obiektu przyciskiem myszy
- W Organizерze obiektów należy wpisać pierwsze litery nazwy obiektu. Zostanie wyświetlone okno dialogowe wyboru obiektu, w którym pojawią się wszystkie obiekty ustawionego typu o wskazanych literach początkowych. Czynności wyświetlane są wg konwencji <nazwa modułu>.<nazwa czynności>. Ponieważ obiekty w oknie dialogowym wyboru obiektów są uporządkowane alfabetycznie, czynności modułu zawsze wymieniane są na liście za nim. Należy zaznaczyć żądany element na liście i nacisnąć przycisk Otwórz, aby wczytać obiekt do okna edycji. Obiekt ten jest przy tym zaznaczany również w Organizерze obiektów i rozwijane są wszystkie foldery i obiekty znajdujące się w hierarchii powyżej tego obiektu. Możliwość ta obsługiwana jest w przypadku typu zasobów wyłącznie dla zmiennych globalnych.

Okno dialogowe wyboru otwieranego obiektu



„Projekt” „Właściwości obiektu”

Polecenie to umożliwia otwarcie okna dialogowego „Właściwości” dla obiektu zaznaczonego w Organizерze obiektów.

Na karcie **Prawa dostępu** znajduje się to samo okno dialogowe, które również wyświetla się w przypadku użycia polecenia „Projekt” „Obiekt” „Prawa dostępu” i jest obsługiwane w opisany tam sposób.

To, czy dostępne są inne karty ustawiania właściwości obiektu i jakie są to karty, zależy od ustawień obiektu i folderów projektu:

Lista zmiennych globalnych:

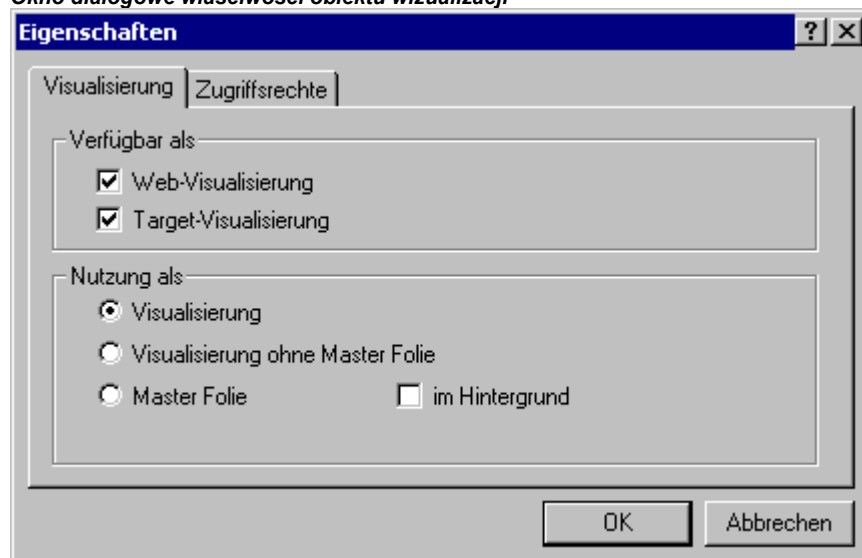
Na karcie „Lista zmiennych globalnych” wyświetlane są parametry skonfigurowane w celu aktualizacji listy oraz ew. w celu wymiany danych o globalnych zmiennych sieciowych. Można tutaj zmieniać wpisy. W przypadku utworzenia nowej listy zmiennych globalnych okno to otwierane jest za pośrednictwem polecenia „Wstaw obiekt”, gdy w Organizерze obiektów zaznaczony jest folder „Zmienne globalne” wzgl. jeden z wpisów znajdujących się poniżej (patrz rozdział 9.2, Zasoby, Zmienne globalne).

Obiekt wizualizacji:

Na karcie „Wizualizacja” można ustalić sposób zastosowania obiektu wizualizacji (patrz podręcznik wizualizacji w programie CoDeSys):

Dostępny jako: Jeśli w ustawieniach systemu docelowego włączona jest opcja „Wizualizacja sieciowa” wzgl. „Wizualizacja elementu docelowego”, odpowiednia opcja jest tutaj wyświetlana jako aktywna. Opcję tę należy wyłączyć, jeśli obiekt nie ma być dostępny w **wizualizacji sieciowej** i/lub w **wizualizacji elementu docelowego** (patrz podręcznik wizualizacji w programie CoDeSys).

Okno dialogowe właściwości obiektu wizualizacji



Wykorzystanie obiektu: Należy wybrać jedno z następujących ustawień dot. możliwości pracy ze „slajdami Master”:

Wizualizacja: Obiekt będzie używany jako zwykła wizualizacja.

Wizualizacja bez slajdu Master: Jeśli w projekcie zdefiniowany jest slajd Master, wizualizacja nie ma zastosowania do tego obiektu. Standardowo slajd Master umieszczany jest zawsze na pierwszym planie wizualizacji, chyba że została włączona opcja „**w tle**”.

Slajd Master: Obiekt będzie używany jako slajd Master.

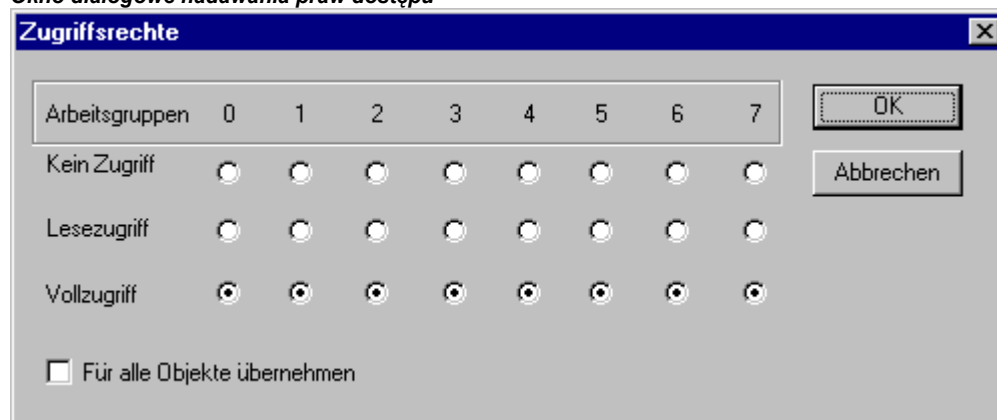
Baza danych projektu:

Jeśli projekt połączony jest z bazą danych ENI (patrz „Projekt” „Opcje” „Baza danych projektu”), dla każdego obiektu dostępna jest kolejna karta o tytule „Baza danych projektu”. Tu wyświetlane jest aktualne przyporządkowanie obiektu do jednej z kategorii baz danych wzgl. do kategorii „Lokalne” i tu można również to przyporządkowanie zmienić. Dalsze informacje na ten temat można znaleźć w rozdziale 10, „Interfejs ENI programu CoDeSys”.

„Projekt” „Prawa dostępu do obiektu”

Za pomocą tego polecenia można otworzyć okno dialogowe w celu nadania różnym grupom roboczym praw dostępu. Zostanie wyświetlone następujące okno dialogowe:

Okno dialogowe nadawania praw dostępu



Członkowie grupy roboczej 0 mogą nadawać teraz każdej grupie roboczej indywidualne prawa dostępu. Możliwe są przy tym trzy ustawienia:

- **Brak dostępu:** Członek danej grupy roboczej nie może otworzyć obiektu.

- **Dostęp do odczytu:** Członek danej grupy roboczej może otworzyć obiekt do odczytu, ale nie może go zmieniać.
- **Pełny dostęp:** Członek danej grupy roboczej może otworzyć i zmieniać obiekt.

Ustawienia odnoszą się do obiektu zaznaczonego aktualnie w Organizерze obiektów albo, jeśli wybrana została opcja **Zastosuj do wszystkich obiektów**, do wszystkich modułów, typów danych, wizualizacji i zasobów projektu.

Przyporządkowanie do grupy roboczej następuje podczas otwierania projektu poprzez zapytanie o hasło, jeśli nadane zostało hasło grupy roboczej 0.

Należy w związku z tym pamiętać również o dodatkowej możliwości nadawania praw dostępu dla grup roboczych w odniesieniu do obsługi elementów wizualizacji (patrz podręcznik wizualizacji w programie CoDeSys).

„Projekt” „Dodaj czynność”

Za pomocą tego polecenia można utworzyć w Organizерze obiektów czynność dla wybranego modułu. W wyświetlonym oknie dialogowym można wybrać nazwę czynności oraz język, w jakim czynność ma zostać zaimplementowana.

Nowa czynność zostanie umieszczona w Organizерze obiektów pod danym modułem. Przed modułem widnieje teraz znak plus. Proste kliknięcie znaku plus powoduje wyświetlenie obiektów czynności oraz umieszczenie znaku minus przed modułem. Ponowne kliknięcie znaku minus powoduje ukrycie czynności i ponowne wyświetlenie znaku plus. Ten sam efekt można również uzyskać za pomocą poleceń menu podręcznego „**Rozwiń węzły**” oraz „**Zwiń węzły**”.

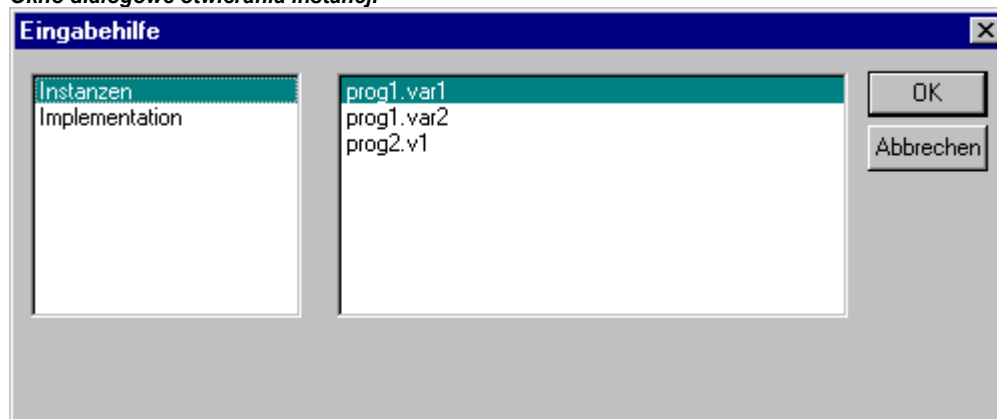
Dwukrotne kliknięcie czynności lub też naciśnięcie klawisza <Enter> powoduje wczytanie czynności do edytora w celu jej edycji.

„Projekt” „Otwórz instancję”

W przypadku tego polecenia można w trybie online otworzyć i wyświetlić instancję bloku funkcji wybranego w Organizерze obiektów. Tak samo za pomocą dwukrotnego kliknięcia bloku funkcji w Organizерze obiektów można przejść do okna dialogowego wyboru, w którym widoczna jest lista instancji bloku funkcji, jak również implementacje. Tu można wybrać żadaną instancję wzgl. implementację i potwierdzić za pomocą przycisku OK. W wyniku tego żadana instancja zostanie wyświetlona w oknie dialogowym.

Wskazówka: Instancje można otworzyć dopiero po zalogowaniu! (Projekt został poprawnie skompilowany i przesłany do sterownika za pomocą polecenia „Online” „Zaloguj”).

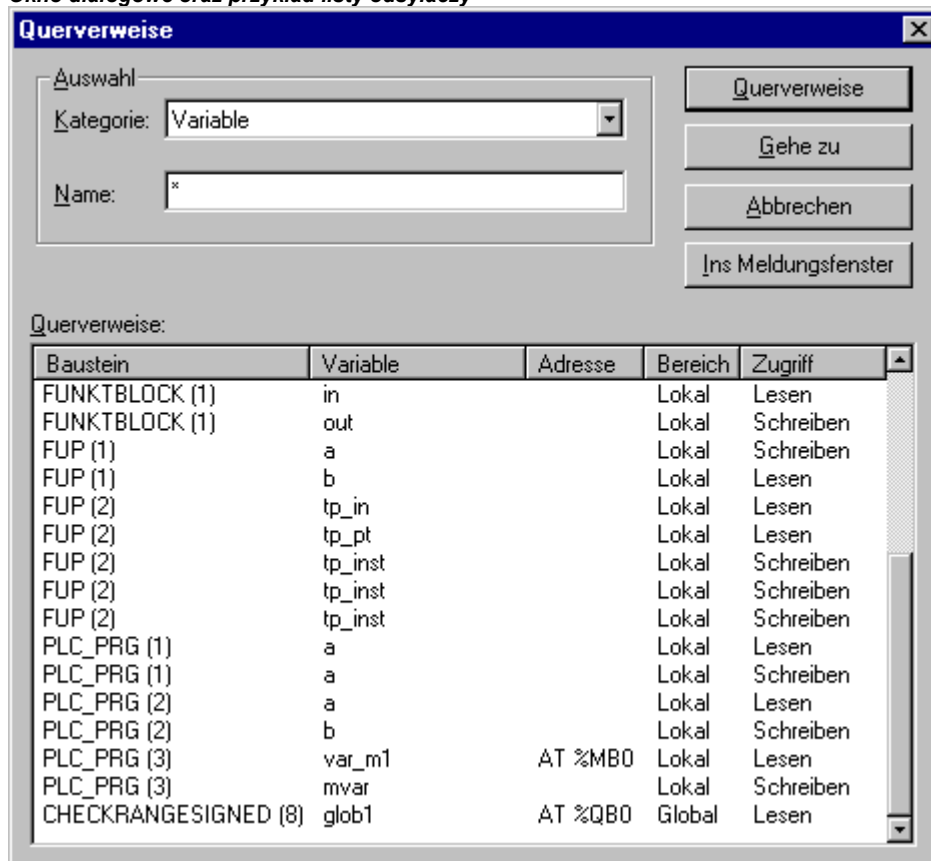
Okno dialogowe otwierania instancji



„Projekt” „Wyświetl listę odsyłaczy”

Za pomocą tego polecenia można otworzyć okno dialogowe umożliwiające wskazanie wszystkich miejsc użycia zmiennej, adresu lub modułu. Projekt musi być w tym celu skompilowany (patrz „Projekt” „Kompiluj”).

Okno dialogowe oraz przykład listy odsyłaczy



Należy najpierw wybrać **Kategorię** „Zmienna”, „Adres” lub „Moduł”, a następnie wprowadzić w polu **Nazwa** nazwężądanego elementu (można zastosować pomoc przy wprowadzaniu za pomocą klawisza <F2>). Aby otrzymać wszystkie elementy ustawionej kategorii, należy w polu nazwy wpisać „*”.

Jeśli od czasu ostatniej kompilacji projekt został zmieniony, na pasku tytułu okna dialogowego pojawi się wskazówka „(nieaktualne)”. Nowe odsyłacze nie są więc uwzględnione na tej liście!

Klikając przycisk **Odsyłacze**, można uzyskać listę wszystkich miejsc użycia. Oprócz modułu oraz numeru wiersza wzgl. sieci podawana jest również nazwa zmiennej i ew. powiązanie adresu. W kolumnie „Zakres” widnieje informacja, czy chodzi o zmienną lokalną czy też o globalną, zaś w kolumnie „Dostęp” – czy w danym miejscu dostęp do zmiennej to „Odczyt” czy też „Zapis”. Szerokość kolumny dostosowuje się automatycznie do najdłuższego wpisu.

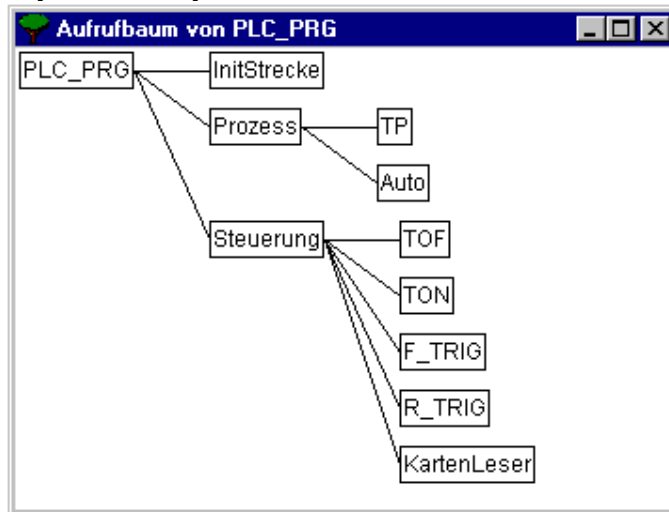
Po zaznaczeniu wiersza na liście odsyłaczy i kliknięciu przycisku **Idź do** lub też dwukrotnym kliknięciu danego wiersza, moduł wyświetlany jest w edytorze w odpowiednim miejscu. W ten sposób można przechodzić do wszystkich miejsc użycia bez pracochłonnego wyszukiwania.

Aby ułatwić sobie obsługę, można za pomocą przycisku **Do okna komunikatu** przejść aktualną listę odsyłaczy do okna komunikatów i stamtąd przechodzić do poszczególnych modułów.

„Projekt” „Wyświetl drzewo wywołań”

Za pomocą tego polecenia można otworzyć okno dialogowe, w którym wyświetlane jest drzewo wywołań obiektu wybranego w Organizery obiektów. Projekt musi być w tym celu bezbłędnie skompilowany (patrz „Projekt” „Kompiluj”). Drzewo wywołań pokazuje, jakie inne moduły są wywoływane w obiekcie.

Przykład drzewa wywołań:



6.5 Ogólne funkcje edycji...

Opisane poniżej polecenia dostępne są we wszystkich edytorach i częściowo w Organizery obiektów. Polecenia te znajdują się w punkcie menu „Edycja” oraz w menu podręcznym, otwieranym za pomocą prawego przycisku myszy.

Jeśli na komputerze zainstalowane jest oprogramowanie IntelliPoint, program CoDeSys obsługuje funkcję pokrętła i przycisk pokrętła Microsoft IntelliMouse. We wszystkich edytorach z funkcją powiększania: W celu powiększenia należy przytrzymać naciśnięty klawisz <Ctrl> podczas obracania pokrętłem do przodu. W celu pomniejszenia pokrętło należy obrócić do tyłu przy naciśniętym klawiszu <Ctrl>.

„Edycja” „Cofnij”

Kombinacja klawiszy: <Ctrl> + <Z>

Polecenie to umożliwia cofnięcie ostatniej wykonanej czynności w aktualnie otwartym oknie edytora wzgl. w Organizery obiektów lub też, w przypadku wielu wykonanych czynności, od momentu otwarcia tego okna. Dotyczy to wszystkich czynności w edytorach modułów, typów danych, wizualizacji i zmiennych globalnych oraz w Organizery obiektów.

Za pomocą polecenia menu „Edycja” „Przywróć” można ponownie wykonać cofniętą czynność.

Wskazówka: Polecenia „Cofnij” oraz „Przywróć” odnoszą się zawsze do aktualnego okna. Każde okno ma własną listę czynności. W przypadku potrzeby cofnięcia czynności w wielu oknach, należy za każdym razem uaktywnić odpowiednie okno. W przypadku cofania lub przywracania w Organizery obiektów, musi on być aktywny.

„Edycja” „Przywróć”

Kombinacja klawiszy: <Ctrl> + <Y>

Za pomocą tego polecenia można przywrócić cofniętą czynność („Edycja” „Cofnij”) w aktualnie otwartym oknie edytora wzgl. w Organizery obiektów.

Tyle razy, ile uprzednio zostało wykonane polecenie „Cofnij”, można również wykonać polecenie „Przywróć”.

Wskazówka: Polecenia „Cofnij” oraz „Przywróć” odnoszą się zawsze do aktualnego okna. Każde okno ma własną listę czynności. W przypadku potrzeby cofnięcia czynności w wielu oknach, należy za każdym razem uaktywnić odpowiednie okno. W przypadku cofania lub przywracania w Organizery obiektów, musi on być aktywny.

„Edycja” „Wytnij”

Symbol: 

**Kombinacja klawiszy: <Ctrl> + <X> lub <Shift> + **

Polecenie to usuwa zaznaczony element z edytora do schowka. Zaznaczenie zostanie usunięte z edytora.

W przypadku Organizera obiektów odbywa się to analogicznie w przypadku zaznaczonego obiektu, przy czym nie można usunąć wszystkich obiektów, np. konfiguracji sterownika.

Należy pamiętać o tym, że nie wszystkie edytory umożliwiają wycinanie, oraz że w niektórych edytorach mogą występować ograniczenia.

Forma wyboru zależy od danego edytora:

W edytorach tekstowych (IL, ST, deklaracje) zaznaczenie stanowi listę znaków.

W edytorze FBD i LD wybór stanowi pewną liczbę sieci, które każdorazowo zaznaczone są za pomocą kropkowanego prostokąta w polu liczb sieci, wzgl. okna dialogowego z poprzedzającymi liniami, polami i argumentami.

W edytorze AS wybór stanowi część ciągu znaków otoczonych kropkowanym prostokątem.

W celu wklejenia zawartości schowka, należy użyć polecenia „Edycja” „Wklej”. W edytorze AS można również użyć poleceń „Dodatki” „Dodaj gałąź równoległą (z prawej)” wzgl. „Wklej po”.

Aby umieścić wybrany element w schowku bez jego usuwania, należy użyć polecenia „Edycja” „Kopiuj”.

Aby usunąć wybrany obszar bez zmieniania zawartości schowka, należy użyć polecenia „Edycja” „Usuń”.

„Edycja” „Kopiuj”

Symbol:  **Kombinacja klawiszy: <Ctrl> + <C>**

Polecenie to kopiuje zaznaczony element z edytora do schowka. Zawartość okna edytora nie ulega przy tym zmianie.

W przypadku Organizera obiektów odbywa się to analogicznie jak w przypadku zaznaczonego obiektu, przy czym nie można skopiować wszystkich obiektów, np. konfiguracji sterownika.

Należy pamiętać o tym, że nie wszystkie edytory umożliwiają kopiowanie oraz że w niektórych edytorach mogą występować ograniczenia.

Forma wyboru zależy od danego edytora:

W edytorach tekstowych (IL, ST, deklaracje) zaznaczenie stanowi listę znaków.

W edytorze FBD i LD wybór stanowi pewną liczbę sieci, które każdorazowo zaznaczone są za pomocą kropkowanego prostokąta w polu liczb sieci, wzgl. okna dialogowego z poprzedzającymi liniami, polami i argumentami.

W edytorze AS wybór stanowi część ciągu znaków otoczonych kropkowanym prostokątem.

W celu wklejenia zawartości schowka, należy użyć polecenia „Edycja” „Wklej”. W edytorze AS można również użyć poleceń „Dodatki” „Dodaj gałąź równoległą (z prawej)” wzgl. „Dodatki” „Wklej po”.

Aby usunąć wybrany obszar i jednocześnie przenieść go do schowka, należy użyć polecenia „Edycja” „Wytnij”.

„Edycja” „Wklej”

Symbol:  **Kombinacja klawiszy: <Ctrl> + <V>**

Wstawia zawartość schowka w aktualnym miejscu kursora w oknie edytora. W edytorach graficznych polecenie to można wykonać tylko wtedy, gdy w wyniku wklejenia powstanie ponownie prawidłowa struktura.

W Organizерze obiektów obiekt wklejany jest ze schowka.

Należy pamiętać o tym, że nie wszystkie edytory umożliwiają wklejanie, oraz że w niektórych edytorach mogą występować ograniczenia.

Aktualna pozycja definiowana jest różnie w zależności od typu edytora:

W przypadku edytorów tekstowych (IL, ST, deklaracje) aktualna pozycja to pozycja migającego kursora (mała pionowa kreska, którą można pozycjonować za pomocą myszy).

W edytorze FBD i LD aktualna pozycja to pierwsza sieć w obrębie kropkowanego prostokąta w obszarze numerów sieci. Zawartość schowka wstawiana jest przed tą siecią. Jeśli została skopiowana struktura częściowa, zostanie ona wstawiona przed zaznaczonym elementem.

W edytorze AS aktualna pozycja określana jest na podstawie zaznaczenia, oznaczonego kropkowanym prostokątem. Zawartość schowka wklejana jest, w zależności od zaznaczenia oraz od zawartości schowka, przed tym zaznaczeniem lub też w nowej gałęzi (równoległej lub alternatywnej) na lewo od zaznaczenia.

W AS w celu wklejenia zawartości schowka można również zastosować polecenia „**Dodatki**” „**Dodaj gałąź równoległą (z prawej)**” wzgl. „**Dodatki**” „**Wklej po**”.

Aby umieścić wybrany element w schowku bez jego usuwania, należy użyć polecenia „Edycja” „Kopiuj”.

Aby usunąć wybrany obszar bez zmieniania zawartości schowka, należy użyć polecenia „Edycja” „Usuń”.

„Edycja” „Usuń”

Kombinacja klawiszy:

Usuwa zaznaczony obszar z okna edytora. Zawartość schowka nie ulega przy tym zmianie.

W przypadku Organizera obiektów odbywa się to analogicznie jak w przypadku zaznaczonego obiektu, przy czym nie można wyciąć wszystkich obiektów, np. konfiguracji sterownika.

Forma wyboru zależy od danego edytora:

W edytorach tekstowych (IL, ST, deklaracje) zaznaczenie stanowi listę znaków.

W edytorze FBD i LD wybór stanowi pewną liczbę sieci, które każdorazowo zaznaczone są za pomocą kropkowanego prostokąta w polu liczb sieci.

W edytorze AS wybór stanowi część ciągu znaków otoczonych kropkowanym prostokątem.

W Menedżerze biblioteki wybrany element to aktualnie wybrana nazwa biblioteki.

Aby usunąć wybrany obszar i jednocześnie przenieść go do schowka, należy użyć polecenia „Edycja” „Wytnij”.

„Edycja” „Znajdź”

Symbol: 

Za pomocą tego polecenia można w aktualnym oknie edytora szukać określonego miejsca w tekście. Zostanie otwarte standardowe okno dialogowe szukania. Pozostanie ono otwarte aż do momentu naciśnięcia przycisku „Anuluj”.

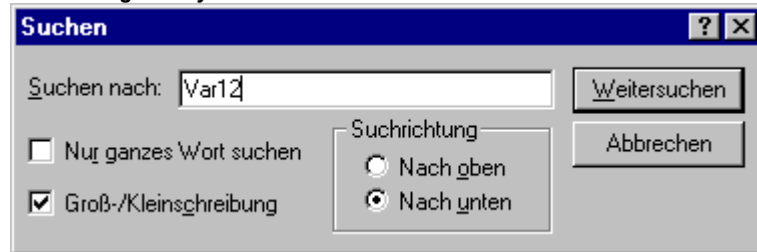
W polu **Szukaj** można wpisać wyszukiwany ciąg znaków wzgl. zostanie w nim automatycznie wyświetlony ciąg zaznaczony w oknie edytora. Wprowadzone ostatnio wyszukiwane ciągi znaków można wybrać za pośrednictwem pola listy rozwijanej „Szukaj”.

Ponadto można wybrać, czy wyszukiwany tekst ma być wyszukiwany z opcją **Tylko** jako **całe słowo**, czy też jako część słowa, czy podczas szukania ma być uwzględniona **Pisownia wielkimi/małymi literami** i czy wyszukiwanie ma rozpoczynać się od aktualnej pozycji kursora **W górę** czy też **W dół**. W edytorze graficznym CFC uwzględniane jest przy tym geometryczne rozmieszczenie elementów od strony lewej u góry do prawej na dole. Należy pamiętać, że w modułach FBD przetwarzanie odbywa się od prawej do lewej!

Przycisk **Znajdź następny** uruchamia wyszukiwanie. Rozpoczyna się ono od wybranej pozycji i postępuje w wybranym kierunku. W przypadku znalezienia tekstu następuje jego zaznaczenie. Jeśli tekst nie zostanie znaleziony, wyświetlany jest komunikat. Wyszukiwanie można przeprowadzić wiele razy po kolei, aż do osiągnięcia początku wzgl. końca zawartości okna edytora.

Należy pamiętać, że znaleziony tekst może być zakryty przez okno dialogowe wyszukiwania.

Okno dialogowe wyszukiwania



„Edycja” „Znajdź następny”

Symbol:  **Kombinacja klawiszy: <F3>**

Za pomocą tego polecenia można przeprowadzić wyszukiwanie w oparciu o te same parametry, jak w przypadku ostatniego zastosowania polecenia „Edycja” „Znajdź”. Należy pamiętać, że w modułach FBD przetwarzanie odbywa się od prawej do lewej!

„Edycja” „Zamień”

Za pomocą tego polecenia można wyszukać określony tekst, dokładnie w taki sam sposób, jak w przypadku polecenia „Edycja” „Znajdź”, i zastąpić go innym. Po wybraniu polecenia zostanie otwarte okno dialogowe „Zamień”. Pozostanie ono otwarte aż do momentu naciśnięcia przycisku **Anuluj** wzgl. **Zamknij**.

W polu **Szukaj** zostanie automatycznie wyświetlony ciąg znaków zaznaczony w oknie edytora, można jednak również ponownie wpisać wyszukiwany ciąg znaków. Przycisk **Zamień** powoduje zastąpienie pierwszego edytowalnego ciągu znaków tekstem wprowadzonym w polu **Zamień na**.

Za pomocą przycisku **Znajdź następny** można przejść do kolejnego miejsca, w którym został znaleziony ten ciąg znaków. Należy pamiętać, że w modułach FBD przetwarzanie odbywa się od prawej do lewej!

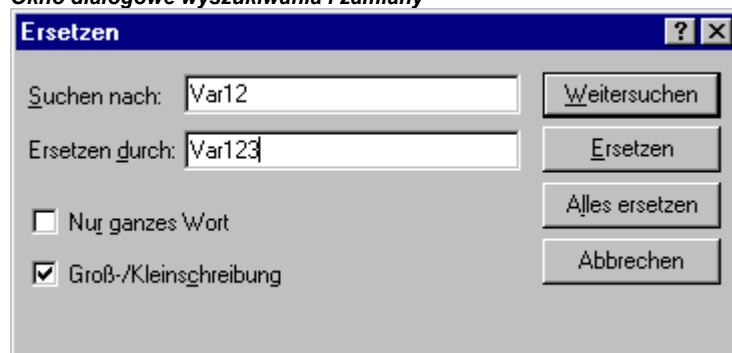
Za pomocą przycisku **Zamień wszystko** można zastąpić wyszukiwany ciąg znaków w całym projekcie, o ile są to miejsca edytowalne.

Należy pamiętać, że tekstu nie można zastąpić w miejscach chronionych przed zapisem (elementy konfiguracji zadań i sterownika, biblioteki). Ciągi znaków w edytowalnych częściach konfiguracji (nazwy zadań, programów, identyfikatory wejść/wyjść) można zastępować.

Wyszukiwane i zastępowane ciągi znaków, które wprowadzono ostatnio, można wybrać za pośrednictwem odpowiedniego pola listy rozwijanej.

Po zakończeniu procesu zamiany wyświetlany jest komunikat o liczbie zmian tekstu.

Okno dialogowe wyszukiwania i zamiany



„Edycja” „Asystent deklaracji”**Kombinacja klawiszy: <F2>**

Polecenie to powoduje wyświetlenie okna dialogowego z wyborem wpisów możliwych do wprowadzenia w aktualnej pozycji kursora w oknie edytora. W lewej kolumnie należy wybrać żadaną kategorię wpisu, zaznaczyć w prawej kolumnie żądany wpis i potwierdzić wybór za pomocą przycisku **OK**. Wybrany wpis zostanie wstawiony w tym miejscu.

Oferowane za każdym razem kategorie zależą od aktualnej pozycji kursora w oknie edytora, tzn. od tego, co można wstawić w danym miejscu (np. zmienne, operatory, moduły, konwersje itp.).

Jeśli zaznaczona jest opcja **Z argumentami**, podczas wstawiania wybranego elementu uwzględniane są również przekazywane argumenty. Przykłady: Wybór bloku funkcji fu1, który posiada zdefiniowaną zmienną wejściową var_in: fu1(var_in:=);

Wstawienie funkcji func1, która jako parametr przekazujący var1 i var2 wymaga: func1(var1,var2);

Zasadniczo możliwe jest przełączanie pomiędzy widokiem ze strukturą i bez struktury dostępnych elementów. Umożliwia to włączenie bądź wyłączenie opcji **Widok struktury**.

Wskazówka: Podczas wprowadzania zmiennych istnieje ponadto możliwość zastosowania funkcji „Intellisense” (patrz rozdział 6.2, Opcje edytora).

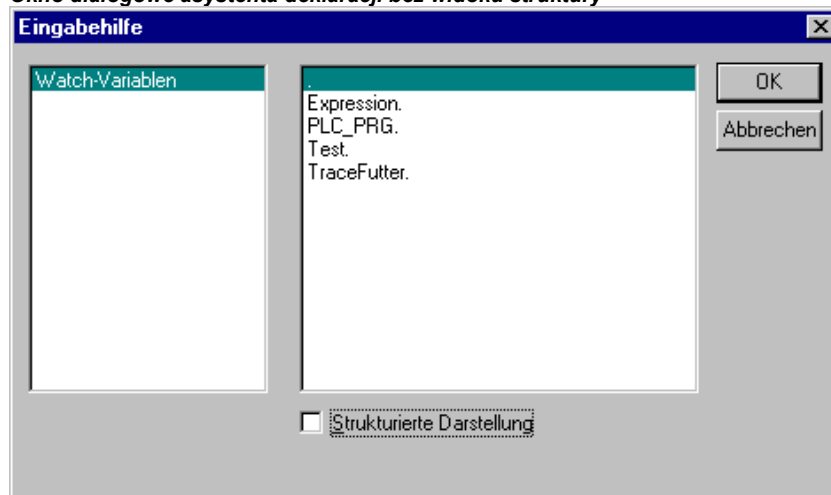
Widok bez struktury

Moduły, zmienne oraz typy danych w każdej kategorii sortowane są liniowo w kolejności alfabetycznej.

W niektórych miejscach (np. na liście kontrolnej) niezbędne są wielostopniowe nazwy zmiennych. W takim wypadku okno dialogowe asystenta deklaracji zawiera listę wszystkich modułów, jak też pojedynczą kropkę zmiennych globalnych. Po każdej nazwie modułu znajduje się kropka. Jeśli moduł zostanie zaznaczony przez podwójne kliknięcie wzgl. naciśnięcie klawisza <Enter>, otworzy się lista przypisanych do niego zmiennych. Jeśli dostępne są instancje i typy danych, można je rozwinąć dalej. Kliknięcie przycisku **OK** powoduje zastosowanie wybranej ostatnio zmiennej.

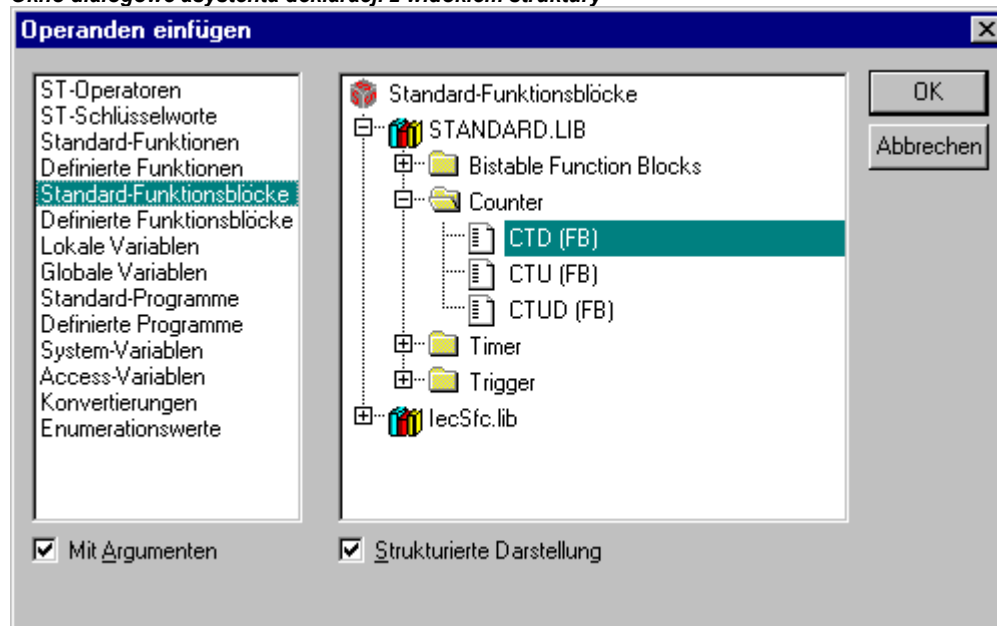
Do **Widoku struktury** można przełączyć się włączając tę opcję.

Okno dialogowe asystenta deklaracji bez widoku struktury

**Widok struktury**

Jeśli wybrany został **Widok struktury** moduły, zmienne oraz typy danych zostaną uporządkowane wg hierarchii. Jest to możliwe w przypadku standardowych programów, funkcji i bloków funkcji, zdefiniowanych programów, funkcji i bloków funkcji, zmiennych globalnych i lokalnych, zdefiniowanych typów, zmiennych monitoringu. Uporządkowany wizualnie widok odpowiada widokowi w Organizery obiektów; jeśli obejmuje on elementy w bibliotekach, są one umieszczone w kolejności alfabetycznej na najwyższym miejscu, zaś układ jest taki jak w Menedżerze biblioteki.

Okno dialogowe asystenta deklaracji z widokiem struktury



Zmienne wejściowe i wyjściowe bloków funkcji, zadeklarowane jako zmienne lokalne lub globalne, zamieszczone są w kategorii „Zmienne lokalne” wzgl. „Zmienne globalne” poniżej nazwy instancji (np. Inst_TP.ET, Inst_TP.IN, ...). Można do nich przejść, wybierając nazwę instancji (np. Inst_TP) i klikając przycisk **OK**.

Jeśli została tu zaznaczona instancja bloku funkcji, można wybrać opcję **Z argumentami**. Wtedy w językach tekstowych ST i IL oraz w konfiguracji zadań zostanie wstawiona nazwa instancji oraz parametr wejściowy bloku funkcji.

Np. w przypadku wybrania Inst (DeklarationInst : TON;) wstawiony zostanie wpis:

```
Inst(IN:= ,PT:=)
```

Jeśli nie wybrano tej opcji, wstawiana jest tylko nazwa instancji. W przypadku języków graficznych oraz w oknie kontrolnym wstawiana jest z zasady jedynie nazwa instancji.

Składniki struktur wyświetlane są w sposób analogiczny do instancji bloków funkcji.

W przypadku enumeracji pod typem enumeracji umieszczana jest lista poszczególnych wartości enumeracji. Kolejność: enumeracje bibliotek, enumeracje typów danych, enumeracje lokalne modułów.

Obowiązuje generalna zasada, że wierszy, które zawierają obiekty podporządkowane, nie można wybierać (poza instancjami, patrz wyżej), lecz tylko rozwijać i zwijać – analogicznie do wielostopniowych nazw zmiennych.

Jeśli asystent deklaracji zostanie wywołany w Menedżerze podglądu zmiennych i receptur wzgl. podczas wybierania zmiennych śledzenia w oknie dialogowym konfiguracji śledzenia, wtedy możliwe jest dokonanie również **wyboru wielokrotnego**. Przytrzymując naciśnięty klawisz <Shift> można wybrać pewien zakres zmiennych, zaś w przypadku klawisza <Ctrl> wiele pojedynczych zmiennych. Wybrane zmienne zostaną zaznaczone. Jeśli podczas zaznaczania obszaru zaznaczone zostały wiersze, które nie zawierają prawidłowych zmiennych (np. nazwy modułów), wiersze te nie zostaną uwzględnione w wyborze. W przypadku wyboru pojedynczego takich wierszy nie można zaznaczyć.

W oknie kontrolnym oraz w oknie konfiguracji śledzenia można przejmować struktury, tablice lub instancje z asystentu deklaracji. Ponieważ do dwukrotnego kliknięcia przypisane jest rozwijanie i zwijanie elementu, wybór można w takich przypadkach potwierdzić jedynie poprzez naciśnięcie przycisku **OK**. W wyniku tego wybrane zmienne zostaną wpisane wierszami do okna kontrolnego, tzn. każda wybrana zmienna zostanie zapisana w jednym wierszu. W przypadku zmiennych śledzenia każda zmienna jest wpisywana w jednym wierszu listy zmiennych śledzenia.

Jeśli podczas wpisywania wybranych zmiennych zostanie przekroczona maks. liczba 20 zmiennych śledzenia, zostanie wyświetlony komunikat o błędzie „Dozwolonych jest maks. 20 zmiennych”. Kolejne wybrane zmienne nie zostaną więc wtedy umieszczone na liście.

Wskazówka: Niektóre wpisy (np. zmienne globalne) aktualizowane są przez asystenta deklaracji dopiero po procesie kompilacji.

Do widoku bez struktury można przełączyć się, wyłączając opcję **Widok struktury**.

„Edycja” „Deklaracja zmiennych”

Kombinacja klawiszy: <Shift> + <F2>

Za pomocą tego polecenia można otworzyć okno dialogowe deklaracji zmiennych, które otwierane jest także w przypadku ustawionej opcji projektu „Zadeklaruj automatycznie”, gdy tylko w oknie deklaracji wprowadzona zostanie nowa zmienna.

„Edycja” „Następny błąd”

Kombinacja klawiszy: <F4>

Po błędnym skompilowaniu projektu można za pomocą tego polecenia wyświetlić następny błąd wzgl. następne ostrzeżenie. Zawsze uaktywniane jest odpowiednie okno edytora i zaznaczane jest błędne miejsce, a równocześnie w oknie komunikatów wyświetlany jest odpowiedni komunikat o błędzie wzgl. ostrzeżenie. Jeśli przy stopniowym przechodzeniu przy użyciu klawisza F4 mają być ignorowane komunikaty, musi być aktywna opcja „Ignorowanie ostrzeżeń F4” w menu „Projekt” „Opcje” „Obszar roboczy”.

„Edycja” „Poprzedni błąd”

Kombinacja klawiszy: <Shift> + <F4>

Po błędnym skompilowaniu projektu można za pomocą tego polecenia wyświetlić poprzedni błąd wzgl. poprzednie ostrzeżenie. Zawsze uaktywniane jest odpowiednie okno edytora i zaznaczane jest błędne miejsce, a równocześnie w oknie komunikatów wyświetlany jest odpowiedni komunikat o błędzie wzgl. ostrzeżenie. Jeśli przy stopniowym przechodzeniu przy użyciu klawisza F4 mają być ignorowane komunikaty, musi być aktywna opcja „Ignorowanie ostrzeżeń F4” w menu „Projekt” „Opcje” „Obszar roboczy”.

„Edycja” „Makra”

W tym punkcie menu zgrupowane są wszystkie makra, zdefiniowane dla aktualnego projektu (patrz rozdział 6.2, Opcje makr). Jeśli wybrane zostanie żądane makro i jest ono możliwe do wykonania, otworzy się okno dialogowe „Wykonaj makro”. W nim wyświetlana jest nazwa makra oraz aktualny wiersz polecenia. Za pomocą przycisku **Anuluj** można zatrzymać przetwarzanie makra, przy czym aktualny wiersz polecenia zostanie przetworzony do końca. Zostanie przy tym wyświetlony odpowiedni komunikat w oknie komunikatów, a w trybie online dokonany zostanie wpis w dzienniku „:<Makro>: Wykonanie przerwane przez użytkownika”.

Makra można wykonywać zarówno w trybie offline, jak i online. Jednak zawsze wykonywane są tylko polecenia dostępne w danym trybie.

6.6 Ogólne funkcje online...

Polecenia dostępne w trybie online zebrane są w punkcie menu „**Online**”. Wykonanie niektórych poleceń jest uzależnione od aktywnego edytora.

Polecenia online dostępne są dopiero po zalogowaniu.

Dzięki funkcjonalności „**Zmiana online**” istnieje możliwość dokonywania zmian programu podczas działania sterownika. Patrz także: „Online” „Zaloguj”.

Informacje na temat powiązań pomiędzy poleceniami kompilacji projektu, pobierania, zmian online i logowania a systemem docelowym opisano poniżej, patrz rozdział „Zależności: logowanie...”.

W poniższych akapitach opisano poszczególne polecenia w trybie online:

„Online” „Zaloguj”, Zmiana online

Symbol: 

Kombinacja klawiszy: <Alt> + <F8>

Polecenie to powoduje połączenie systemu programowania ze sterownikiem (lub też uruchamia program symulacji) i przełącza na tryb online.

Jeśli aktualny projekt nie został skompilowany od momentu jego otwarcia wzgl. od ostatniej zmiany, zostanie skompilowany teraz (jak w przypadku polecenia „Projekt” „Kompiluj”). W razie wystąpienia błędów podczas kompilacji, program CoDeSys nie zostanie przełączony do trybu online.

Jeśli aktualny projekt uległ zmianie od czasu ostatniego pobrania do sterownika, ale nie został zamknięty, i jeśli informacje o ostatnim pobieraniu nie zostały usunięte za pomocą polecenia „Projekt” „Wyczyść wszystko”, po poleceniu „Zaloguj” zostanie wyświetlone okno dialogowe z zapytaniem:

"Program został zmieniony. Czy chcesz wczytać te zmiany? (zmiana online)".

Wybór odpowiedzi <Tak> spowoduje potwierdzenie wczytania do sterownika zmienionych części projektu podczas logowania. Także i w tym przypadku w poniższych rozdziałach dostępne są wskazówki na temat zmian online oraz wykres przedstawiający zależności: logowanie – kompilacja – pobieranie – zmiana online.

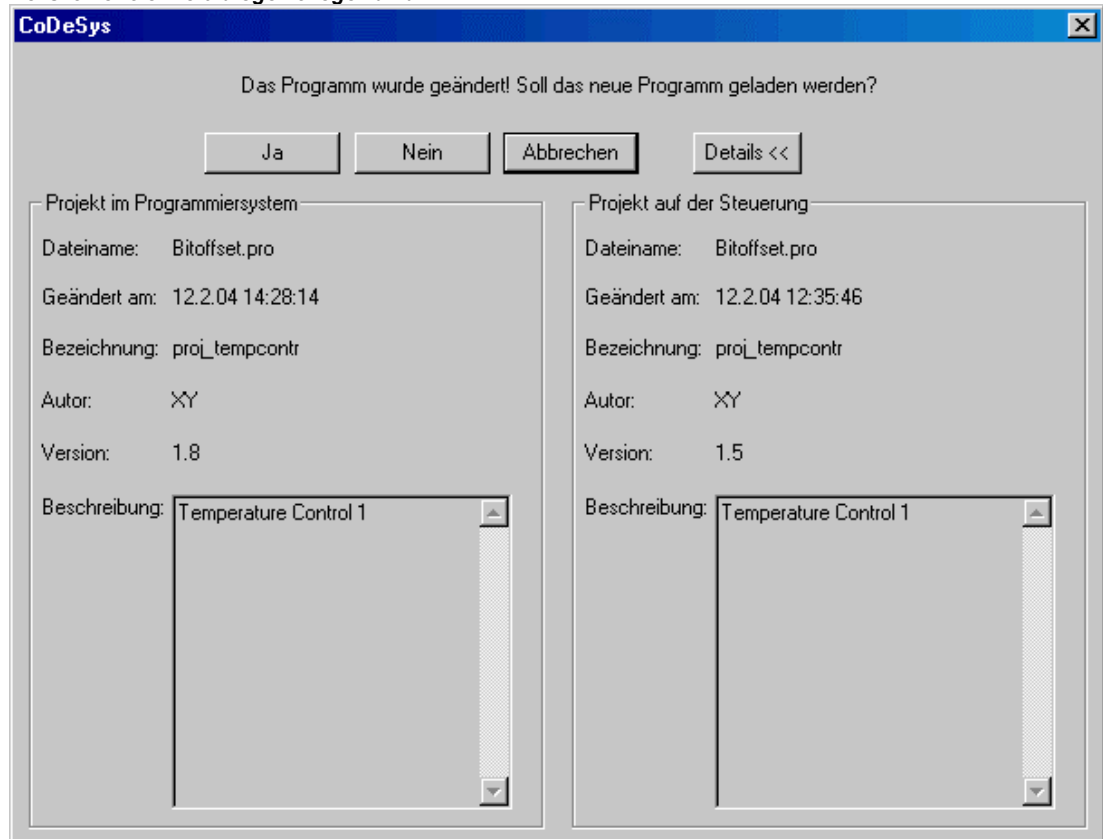
W przypadku wybrania opcji <Nie> nastąpi zalogowanie bez wczytywania do sterownika zmian dokonanych od czasu ostatniego pobrania. Odpowiedź <Anuluj> spowoduje przerwanie wykonywania polecenia. Opcja <Wczytaj wszystko> umożliwi ponowne wczytanie całego projektu do sterownika.

Jeśli w opcjach projektu, w kategorii „Obszar roboczy” włączona jest opcja „Praca online w trybie bezpieczeństwa” i system docelowy obsługuje tę funkcjonalność, w oknie dialogowym logowania automatycznie prezentowane są dodatkowo informacje o projekcie aktualnie wczytanym do systemu oraz o już istniejącym w sterowniku. Można je zamknąć za pomocą przycisku „Szczegóły <<”. Jeśli opcja obszaru roboczego nie jest aktywna, informacje o projekcie można otworzyć bezpośrednio za pomocą przycisku „Szczegóły >>”.

Należy pamiętać, że przycisk domyślny, czyli uaktywniany automatycznie, zależy do ustawień w systemie docelowym.

Wskazówka: Zmiana online nie jest możliwa po wprowadzeniu zmian w konfiguracji zadań, konfiguracji sterownika, po wstawieniu biblioteki wzgl. po wykonaniu polecenia „Projekt” „Wyczyść wszystko” (patrz niżej). W przypadku zmiany online nie następuje ponowna inicjalizacja, tzn. zmiany wartości inicjalizacji nie są uwzględniane! Zmienne zapisu (retain) w przypadku zmiany online zachowują swe wartości – inaczej niż w przypadku ponownego pobrania projektu (patrz niżej, „Online” „Wczytaj”).

Rozszerzone okno dialogowe logowania



Po pomyślnym zalogowaniu dostępne są wszystkie funkcje online (o ile zostały wprowadzone odpowiednie ustawienia w obszarze „**Opcje**”, kategoria „Opcje kompilacji”).

Aby z trybu online powrócić ponownie do trybu offline, należy użyć polecenia „Online” „Wyloguj”.

Inne możliwe komunikaty systemowe podczas logowania:

Błąd:

„Wybrany profil sterownika nie odpowiada systemowi docelowemu...”

Należy sprawdzić, czy system docelowy ustawiony w ustawieniach systemu docelowego (zasoby) jest zgodny z parametrami ustawionymi w obszarze „**Online**” „**Parametry komunikacji**”.

Błąd:

„Błąd komunikacji. Nastąpi wylogowanie.”

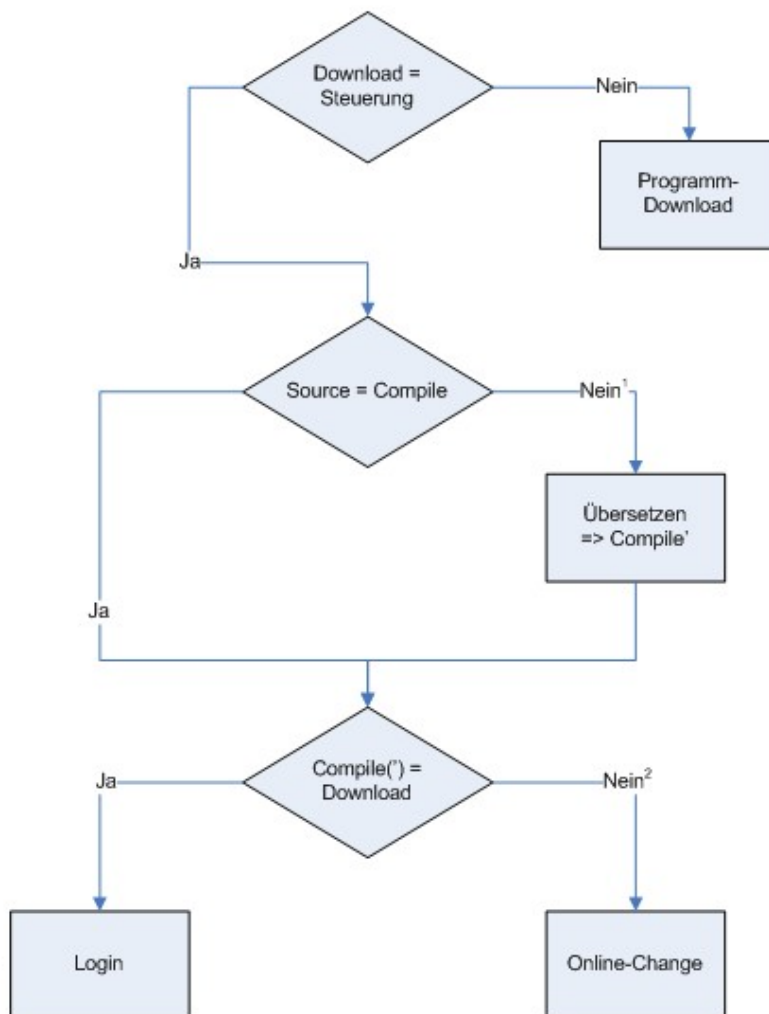
Należy sprawdzić, czy sterownik działa. Należy sprawdzić, czy parametry ustawione w obszarze „Online” „Parametry komunikacji” są zgodne z parametrami w sterowniku. Zwłaszcza należy zwrócić uwagę, czy ustawiony jest prawidłowy port i czy zgodne są prędkości transmisji w sterowniku i w systemie programowania. Jeśli używany jest serwer Gateway, należy sprawdzić, czy ustawienie kanału jest prawidłowe.

Zależności: logowanie – kompilacja – pobieranie – zmiana online:

Poniżej przedstawiono wykres obrazujący zależności pomiędzy logowaniem, kompilacją, wczytywaniem i zmianami online:

Stosowane są następujące pojęcia:

- Źródło (source) aktualny projekt CoDeSys (plik *.pro, na komputerze lokalnym)
- Kompiluj (compile) informacja o kompilacji z ostatniego procesu kompilacji, jest niezbędna do kompilacji przyrostowej (plik *.ci, na komputerze lokalnym)
- Pobierz (download) informacja o tym, co ostatnio zostało wczytane do sterownika (plik *.ri, w komputerze lokalnym)
- Sterownik aktualny projekt w sterowniku (plik *.prg)



¹: z.B. geänderte Bibliotheken

²: z.B. geänderte Compiler Version oder geänderter Programmcode

Wskazówki na temat zmiany online

- Zmiana online nie jest możliwa po wprowadzeniu zmian w konfiguracji zadań, konfiguracji sterownika, po wstawieniu biblioteki wzgl. po wykonaniu polecenia „Projekt” „Wyczyść wszystko” (patrz niżej).
- Jeśli usunięto (np. za pomocą polecenia „Projekt” „Wyczyść wszystko”) informację o pobieraniu (Plik <nazwa projektu><identyfikator docelowy>.ri), utworzoną podczas ostatniego wczytywania projektu (może być to również zmiana online), zmiana online nie jest już możliwa, chyba że plik ri

6 - Szczegółowy opis składników

został zapisany w innym miejscu lub pod inną nazwą i można go ponownie wczytać za pomocą polecenia „Projekt” „Wczytaj informacje o pobieraniu”. Patrz także niżej, „Zmiana online projektu...”.

- W przypadku zmiany online nie następuje ponowna inicjalizacja, tzn. zmiany wartości inicjalizacji nie są uwzględniane!
- Zmienne zapisu (retain) w przypadku zmiany online zachowują swe wartości – inaczej niż w przypadku ponownego pobrania projektu (patrz niżej, „Online” „Wczytaj”).

Zmiana online projektu działającego w wielu sterownikach:

Aby projekt *proj.pro* mógł być używany w dwóch takich samych sterownikach PLC1 i PLC2 (taki sam system docelowy) i aby upewnić się, że po dokonaniu zmian projektu za pomocą funkcji zmiany online można będzie zaktualizować projekt w obydwu sterownikach, należy postępować w następujący sposób:

(1) Wczytać projekt do sterownika PLC1, zapisać informacje o pobieraniu dla PLC1:

1. Ustanowić połączenie ze sterownikiem PLC1 (Online/Parametry komunikacji) i wczytać *proj.pro* do PLC1 (Online/Zaloguj, Wczytaj). Utworzony zostanie przy tym równolegle do *proj.pro* plik *proj00000001.ri*, zawierający informacje o pobieraniu.
2. Zmienić nazwę pliku *proj00000001.ri*, np. *naproj00000001_PLC1.ri*. Taki zapis pod inną nazwą jest konieczny, ponieważ w przypadku ponownego pobrania projektu *proj.pro* plik *proj00000001.ri* zostałby nadpisany nowymi informacjami o pobieraniu, a tym samym zostałyby utracone informacje o pobieraniu do PLC1.
Uruchomić projekt i wylogować się ponownie.

(2) Wczytać projekt do sterownika PLC1, zapisać informacje o pobieraniu dla PLC2:

1. Ustanowić teraz połączenie ze sterownikiem PLC2 (Online/Parametry komunikacji) i wczytać *proj.pro* do PLC2 (Online/Zaloguj, Wczytaj). W wyniku tego zostanie utworzony równolegle do *proj.pro* ponownie plik *proj00000001.ri*, zawierający teraz informacje na temat aktualnego pobierania.
2. Należy zapisać nowy plik *proj00000001.ri* poprzez zmianę nazwy np. na *proj00000001_PLC2.ri*.
3. Uruchomić projekt na PLC2 i wylogować się ponownie.

(3) Zmiana projektu:

W programie CoDeSys należy teraz wprowadzić zmiany do pliku *proj.pro*, który ma zostać wczytany za pomocą zmiany online do projektu działającego w sterowniku.

(4) Zmiana online w PLC1, ponowne zapisanie informacji o pobieraniu:

1. Aby możliwe było dokonanie zmiany online pliku *proj.pro* w PLC1, należy najpierw ponownie udostępnić informacje o pobieraniu zapisane przy pobieraniu pliku *proj.pro* do PLC1. Program CoDeSys poszukuje w tym celu po zalogowaniu pliku *proj00000001.ri*. Plik *ri* aktualny dla PLC1 został zapisany jako *proj00000001_PLC1.ri*.
Istnieją teraz dwie możliwości:
(a) można ponownie zmienić nazwę pliku *proj00000001_PLC1.ri* na *proj00000001.ri*. Dzięki temu podczas logowania do PLC1 automatycznie zostanie uwzględniona prawidłowa informacja o pobieraniu i zostanie zaproponowana dla zmiany online.
(b) zamiast tego można przed zalogowaniem wczytać za pomocą polecenia „Projekt” „Wczytaj informacje o pobieraniu” tylko plik *proj00000001_PLC1.ri*. Tym samym można uniknąć zmieniania nazwy i zapewnić sobie również możliwość zmiany online.
2. Podczas zmiany online w PLC1 został utworzony ponownie aktualny plik *proj00000001.ri* z nowymi informacjami o pobieraniu. Należy go zapisać w sposób przedstawiony w punkcie 4, aby zapewnić gotowość kolejnej zmiany online w PLC1.

(5) Zmiana online w PLC2, ponowne zapisanie informacji o pobieraniu:

Aby przeprowadzić teraz zmianę online pliku *proj.pro* również w sterowniku PLC2, należy postępować w sposób opisany w punkcie (4) odpowiednio z plikiem *proj00000001_PLC2.ri*.

(6) Każda następną zmianą online po zmianie projektu: wg punktów od (3) do (5)

Komunikaty systemowe podczas logowania

Błąd:

„Wybrany profil sterownika nie odpowiada systemowi docelowemu...”

Należy sprawdzić, czy system docelowy ustawiony w ustawieniach systemu docelowego (zasoby) jest zgodny z parametrami ustawionymi w obszarze „Online” „Parametry komunikacji”.

Błąd:

„Błąd komunikacji. Nastąpi wylogowanie.”

Należy sprawdzić, czy sterownik działa. Należy sprawdzić, czy parametry ustawione w obszarze „Online” „Parametry komunikacji” są zgodne z parametrami w sterowniku. Zwłaszcza należy zwrócić uwagę, czy ustawiony jest prawidłowy port i czy zgodne są prędkości transmisji w sterowniku i w systemie programowania. Jeśli używany jest serwer Gateway, należy sprawdzić, czy ustawienie kanału jest prawidłowe.

Błąd:

„Program został zmieniony! Czy chcesz wczytać nowy program?”

Aktualny projekt w edytorze nie pasuje do projektu wczytanego obecnie do sterownika. Dlatego niemożliwe jest monitorowanie i debugowanie. Można teraz wybrać odpowiedź **Nie**, wylogować się ponownie i otworzyć prawidłowy projekt lub też za pomocą odpowiedzi **Tak** wczytać do sterownika aktualny projekt.

Komunikat:

„Program został zmieniony! Czy chcesz wczytać te zmiany? (ZMIANA ONLINE)”

Projekt działa w sterowniku. System docelowy obsługuje funkcję „Zmiana online”, a projekt został zmieniony od czasu ostatniego pobrania wzgl. ostatniej zmiany online w sterowniku. Teraz można zdecydować, czy te zmiany mają być wczytane podczas działania programu sterownika, czy też wykonanie polecenia zostanie przerwane. Można także wczytać cały skompilowany kod, naciskając w tym celu przycisk **Wczytaj wszystko**.

„Online” „Wyloguj”

Symbol:  **Kombinacja klawiszy <Ctrl> + <F8>**

Połączenie ze sterownikiem zostanie przerwane wzgl. program symulacji zostanie zamknięty i nastąpi przejście do trybu offline.

Aby przejść do trybu online należy użyć polecenia „Online” „Zaloguj”.

„Online” „Wczytaj”

Polecenie to powoduje wczytanie skompilowanego projektu do sterownika (**Pobierz**, nie mylić z „Online” „Wczytaj kod źródłowy!”).

Jeśli stosowana jest opcja generowania kodu C, przed wczytaniem uruchamiany jest kompilator C, który generuje plik do pobrania. W przeciwnym razie plik do pobrania generowany jest już podczas kompilacji.

Informacje o pobieraniu zapisywane są w pliku <nazwa projektu>000000ar.ri, który jest używany podczas zmiany online w celu porównania aktualnego programu z programem wczytanym ostatnio do sterownika, dzięki czemu mogą być wczytane ponownie tylko zmienione części programu. Plik ten zostanie usunięty po użyciu polecenia „Projekt” „Wyczyść wszystko!” Należy pamiętać, że także podczas zmiany online tworzony jest nowy plik *.ri. Patrz informacje na temat „Zmiana online projektu działającego w wielu sterownikach” powyżej w rozdziale „Online” „Zaloguj”.

W zależności od systemu docelowego podczas każdego tworzenia projektu bootowalnego w trybie offline może być tworzony automatycznie nowy plik *.ri.

Tylko trwałe zmienne (patrz rozdział 8.2.1) zachowują swą wartość także po pobraniu.

„Online” „Start”

Symbol:  **Kombinacja klawiszy: <F5>**

Polecenie to uruchamia przetwarzanie programu użytkownika w sterowniku wzgl. w symulacji.

Polecenie może zostać wykonane bezpośrednio po poleceniu „Online” „Wczytaj”, lub po zatrzymaniu programu użytkownika w sterowniku za pomocą polecenia „Online” „Stop”, lub też kiedy program użytkownika zatrzyma się w punkcie wstrzymania lub jeśli zostało wykonane polecenie „Online” „Cykl pojedynczy”.

„Online” „Stop”

Symbol:  **Kombinacja klawiszy <Shift> + <F8>**

Polecenie to zatrzymuje przetwarzanie programu użytkownika w sterowniku wzgl. w symulacji pomiędzy dwoma cyklami.

W celu kontynuowania przetwarzania programu należy użyć polecenia „Online” „Start”.

„Online” „Reset”

Polecenie to przywraca wszystkie zmienne, z wyjątkiem zmiennych retain (VAR RETAIN), do wartości, przy jakich zostały zainicjowane (a więc także zmienne zadeklarowane przy użyciu VAR PERSISTENT!). Zmienne, którym nie przypisano bezpośrednio wartości inicjującej, otrzymują standardowe wartości inicjujące (liczby całkowite np. 0). Przed zastąpieniem wszystkich zmiennych program CoDeSys wyświetla zapytanie o potwierdzenie. Sytuacja ta jest podobna do sytuacji awarii prądu lub włączenia/wyłączenia sterownika (gorący start) podczas działania programu.

W celu ponownego uruchomienia sterownika i przetwarzania programu należy użyć polecenia „Online” „Start”.

Patrz także: „Online” „Reset źródła”, „Online” „Zimny reset” oraz przegląd procedur ponownej inicjalizacji w rozdziale 8.2.1, „Zmienne remanentne”.

„Online” „Reset (zimny)”

Polecenie to odpowiada poleceniu „Reset” z tą różnicą, że wszystkie zmienne, a więc również zmienne retain, resetowane są do wartości inicjujących. Sytuacja jest podobna jak w przypadku startu programu, który został wczytany do sterownika (zimny start). Patrz także: „Online” „Reset”, „Online” „Reset (źródło)” oraz przegląd procedur ponownej inicjalizacji w rozdziale 8.2.1, „Zmienne remanentne”.

„Online” „Reset (źródło)”

Polecenie to powoduje przywrócenie wszystkich zmiennych, również remanentnych (VAR RETAIN i VAR PERSISTENT) do wartości inicjujących i usunięcie programu użytkownika ze sterownika. Sterownik jest resetowany do stanu pierwotnego. Patrz także: „Online” „Reset”, „Online” „Reset zimny” oraz przegląd procedur ponownej inicjalizacji w rozdziale 8.2.1, „Zmienne remanentne”.

„Online” „Włącz/wyłącz punkt wstrzymania”

Symbol:  **Kombinacja klawiszy: <F9>**

Polecenie to powoduje wstawienie punktu wstrzymania w aktualnej pozycji w aktywnym oknie. Jeśli w aktualnej pozycji znajduje się już punkt wstrzymania, zostanie on usunięty.

Pozycja, w której można umieścić punkt wstrzymania, zależy od języka, w którym napisany został moduł znajdujący się w aktywnym oknie.

W edytorach tekstowych (IL, ST) punkt wstrzymania wstawiany jest w wierszu, w którym znajduje się kursor, jeśli wiersz ten stanowi pozycję punktu wstrzymania. Pozycję punktu wstrzymania można rozpoznać po ciemnoszarym kolorze pola numeru wiersza (w przypadku ustawienia standardowego). W celu umieszczenia bądź usunięcia punktu wstrzymania w edytorach tekstowych można również kliknąć pole numeru wiersza.

W FBD i LD punkt wstrzymania ustawiany jest na aktualnie zaznaczonej sieci. W celu umieszczenia bądź usunięcia punktu wstrzymania w edytorze FBD lub LD można również kliknąć pole numeru sieci.

W AS punkt wstrzymania umieszczany jest w aktualnie zaznaczonym kroku. W celu umieszczenia wzgl. usunięcia punktu wstrzymania można w AS nacisnąć również klawisz <Shift>, a następnie dwukrotnie kliknąć klawiszem myszy.

Jeśli punkt wstrzymania jest umieszczony, wtedy pole numeru wiersza lub pole numeru sieci wzgl. krok wyświetlany jest z tłem w jasnoniebieskim kolorze (ustawienie standardowe).

Jeśli podczas przetwarzania programu osiągnięty zostanie punkt wstrzymania, wtedy program zatrzymuje się i odpowiednie pole wyświetlane jest z tłem w kolorze czerwonym (ustawienie standardowe). Aby kontynuować przetwarzanie programu, należy użyć poleceń „Online” „Start”, „Online” „Krok pojedynczy w” lub „Online” „Krok pojedynczy przez”.

W celu umieszczenia wzgl. usunięcia punktów wstrzymania można również skorzystać z okna dialogowego punktu wstrzymania.

„Online” „Okno dialogowe punktu wstrzymania”

Polecenie to otwiera okno dialogowe edycji punktów wstrzymania w całym projekcie. Okno dialogowe wskazuje ponadto aktualnie umieszczone punkty wstrzymania.

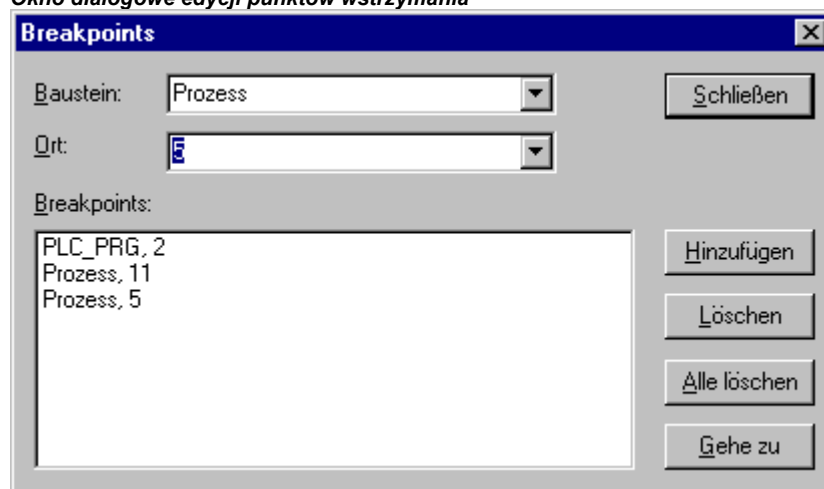
W celu umieszczenia punktu wstrzymania na liście rozwijanej **Moduł** należy wybrać moduł, zaś na liście **Miejsce** wiersz wzgl. sieć, gdzie ma zostać umieszczony punkt wstrzymania i nacisnąć przycisk **Dodaj**. Punkt wstrzymania zostanie dodany do listy.

W celu usunięcia punktu wstrzymania należy go zaznaczyć i nacisnąć przycisk **Usuń**.

Za pomocą przycisku **Usuń wszystkie** można usunąć wszystkie punkty wstrzymania.

Aby przejść w edytorze do miejsca, w którym umieszczony został określony punkt wstrzymania, należy zaznaczyć odpowiedni punkt i nacisnąć przycisk **Idź do**.

Okno dialogowe edycji punktów wstrzymania



W celu umieszczenia wzgl. usunięcia punktów wstrzymania można również skorzystać z polecenia „Online” „Włącz/wyłącz punkt wstrzymania”.

„Online” „Krok pojedynczy przez”

Symbol:  **Kombinacja klawiszy: <F10>**

Za pomocą tego polecenia wykonywany jest pojedynczy krok, przy czym podczas wywoływania modułów zatrzymanie nastąpi dopiero po jego przetworzeniu. W AS przetwarzana jest kompletna czynność.

Jeśli aktualna instrukcja stanowi wywołanie funkcji lub bloku funkcji, wtedy wykonywana jest kompletna funkcja lub też blok funkcji. Aby przejść do pierwszej instrukcji wywołanej funkcji wzgl. wywołanego bloku funkcji, należy użyć polecenia „Online” „Krok pojedynczy w”.

Gdy zostanie osiągnięta ostatnia instrukcja, wtedy program przechodzi do następnej instrukcji wywołanego modułu.

„Online” „Krok pojedynczy w”

Kombinacja klawiszy: <F8>

Przetwarzany jest pojedynczy krok, przy czym zatrzymanie następuje podczas wywoływania modułów przed wykonaniem pierwszej instrukcji modułu.

W razie potrzeby następuje przejście do wywołanego modułu.

Jeśli aktualna pozycja stanowi wywołanie funkcji lub bloku funkcji, wtedy polecenie przechodzi do pierwszej instrukcji wywołanego modułu.

We wszystkich innych sytuacjach polecenie zachowuje się dokładnie tak jak polecenie „Online” „Krok pojedynczy przez”.

„Online” „Cykl pojedynczy”

Kombinacja klawiszy: <Ctrl> + <F5>

Polecenie to wykonuje pojedynczy cykl sterujący i zatrzymuje się po tym cyklu.

Polecenie to można powtarzać w sposób ciągły, aby przechodzić za każdym razem o jeden cykl.

Cykl pojedynczy kończy się, gdy zostanie wykonane polecenie „Online” „Start”.

„Online” „Zapisz wartości”

Kombinacja klawiszy: <Ctrl> + <F7>

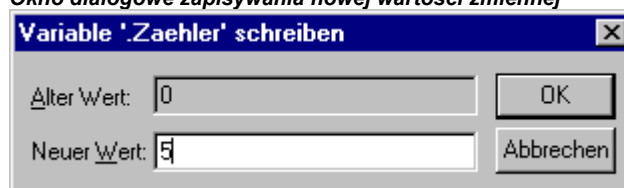
Za pomocą tego polecenia na początku cyklu ustawiana jest (jednorazowo!) zdefiniowana przez użytkownika wartość jednej lub wielu zmiennych.

Można zmienić wartości wszystkich zmiennych jednoelementowych, które są również widoczne podczas monitorowania.

Zanim będzie można wykonać polecenie „Zapisz wartości”, należy przygotować wartość zmiennej do zapisania:

- W przypadku zmiennych innych niż boolean należy dwukrotnie kliknąć w wierszu, w którym zadeklarowana jest zmienna lub też zaznaczyć zmienną i nacisnąć klawisz <Enter>. Zostanie wtedy wyświetlone okno dialogowe „Zapisywanie zmiennej <x>”, w którym można wprowadzić wartość, jaka ma zostać zapisana w zmiennej.

Okno dialogowe zapisywania nowej wartości zmiennej



- W przypadku zmiennych boolean w wyniku dwukrotnego kliknięcia przyciskiem myszy wiersza, w którym zadeklarowana jest zmienna, następuje przełączenie wartości (zmiana pomiędzy TRUE i FALSE bez żadnej innej wartości) bez wyświetlania okna dialogowego.

Przewidziana do zapisania nowa wartość wyświetlana jest w kolorze turkusowym w nawiasach ostrokatnych za dotychczasową wartością deklaracji, np.:

```
bvar = TRUE < := FALSE >
```

```
ivar = 509 < := 65 >
```

Wskazówka: Wyjątek w zakresie wyświetlania zapisywanych wartości: W edytorze FBD i LD wartość wyświetlana jest w kolorze turkusowym bez nawiasów ostrokatnych obok nazwy zmiennej.

Ustawianie wartości można przeprowadzić dla dowolnej ilości zmiennych.

Wartości, wpisane w celu zapisania dla zmiennych, można również w ten sam sposób skorygować lub ponownie usunąć. Tak samo możliwe jest to za pomocą poleceń „Online” „Okno dialog. Zapisywanie/Wymuszanie”.

Wartości zaznaczone w celu zapisania zapisywane są na **liście zapisu (lista kontrolna)**, gdzie pozostają aż do czasu rzeczywistego zapisania, usunięcia lub przeniesienia poleceniem „Wymuszanie wartości” na listę wymuszania. Wgląd w listę kontrolną oraz listę wymuszania zapewnia okno dialogowe Zapisywanie/wymuszanie.

Polecenie zapisania wartości ustawionych na liście zapisu można znaleźć w dwóch miejscach:

- Polecenie „Zapisz wartości” w menu „Online”.
- Przycisk „Zapisz wartości” w oknie dialogowym „Edycja listy zapisu i wymuszania”.

Jeśli zostanie wykonane polecenie „Zapisz wartości”, wszystkie wartości, zawarte na liście zapisu zostaną zapisane jednorazowo na początku cyklu w odpowiednich zmiennych w sterowniku, a tym samym usunięte z listy zapisu. (Jeśli zostanie wykonane polecenie „Wymuszanie wartości”, wtedy odpowiednie zmienne zostaną również usunięte z listy zapisu i umieszczone na liście wymuszania).

Wskazówka: W języku przebiegu za pomocą polecenia „Zapisz wartości” nie można zmieniać poszczególnych wartości, z których składa się wyrażenie przejścia. Polega to na tym, że podczas monitorowania „łącznej wartości” wyrażenia nie są wyświetlane wartości poszczególnych zmiennych (np. „a AND b” wyświetlane jest jako TRUE tylko wtedy, jeśli rzeczywiście obie zmienne mają wartość TRUE).

Natomiast w FBD w wyrażeniu, które np. używane jest jako wejście bloku funkcji, monitorowana jest tylko pierwsza zmienna. Dlatego też polecenie „Zapisz wartości” możliwe jest tylko w przypadku tej zmiennej.

„Online” „Wymuszanie wartości”

Kombinacja klawiszy: <F7>

Za pomocą tego polecenia ustawiana jest wartość jednej lub wielu zmiennych, trwale zdefiniowana przez użytkownika. Ustawianie odbywa się przy tym w systemie czasu przebiegu zawsze na początku i na końcu cyklu.

Przebieg czasowy w jednym cyklu: 1. Odczytywanie wejść, 2. Wymuszanie wartości, 3. Przetwarzanie kodu, 4. Wymuszanie wartości, 5. Zapis wyjść.

Funkcja jest aktywna tak długo, aż zostanie anulowana bezpośrednio przez użytkownika (polecenie „Online” „Anuluj wymuszanie”) lub też aż nastąpi wylogowanie z systemu programowania.

W celu ustawienia nowych wartości należy najpierw utworzyć listę zapisu. Zmienne zawarte na liście zapisu są odpowiednio oznaczone w widoku monitorowania. **Lista zapisu** (lista kontrolna) przekształcana jest na **listę wymuszania**, gdy tylko zostanie wykonane polecenie „Online” „Wymuszanie wartości”. Wgląd w listę kontrolną oraz listę wymuszania zapewnia okno dialogowe Zapisywanie/wymuszanie. Możliwe, że istnieje już aktywna lista wymuszania, która zostanie odpowiednio zaktualizowana. Lista zapisu zostanie opróżniona i nowe wartości zostaną przedstawione jako „wymuszone”; np.:

```
bvar = FALSE  
ivar = 44
```

Modyfikacje listy wymuszania przekazywane są zawsze do programu podczas kolejnej procedury „Wymuszanie wartości”.

Uwaga: Lista wymuszania powstaje podczas pierwszego wymuszenia zmiennych znajdujących się na liście zapisu, podczas gdy lista zapisu istnieje już przed pierwszym zapisaniem zawartych w niej zmiennych.

Wskazówka: O ile umożliwia to system docelowy, lista wymuszania pozostaje zachowana w sterowniku, nawet po przerwaniu połączenia np. w wyniku wylogowania.

Polecenie wymuszenia zmiennej (a tym samym umieszczenia na liście wymuszania) znajduje się w następujących miejscach:

- Polecenie „Wymuszanie wartości” w menu „Online”.

- Przycisk „Wymuszanie wartości” w oknie dialogowym „Edycja listy zapisu i wymuszania”.

Wskazówka: W języku przebiegu za pomocą polecenia „Wymuszanie wartości” nie można zmieniać poszczególnych wartości, z których składa się wyrażenie przejścia. Polega to na tym, że podczas monitorowania „łącznej wartości” wyrażenia nie są wyświetlane wartości poszczególnych zmiennych (np. „a AND b” wyświetlane jest jako TRUE tylko wtedy, jeśli rzeczywiście obie zmienne mają wartość TRUE). Natomiast w FBD w wyrażeniu, które np. używane jest jako wejście bloku funkcji, monitorowana jest tylko pierwsza zmienna. Dlatego też polecenie „Wymuszanie wartości” możliwe jest tylko w przypadku tej zmiennej.

„Online” „Anuluj wymuszanie”

Kombinacja klawiszy: <Shift> + <F7>

Za pomocą tego polecenia można zakończyć wymuszanie wartości zmiennych w sterowniku. Zmienne ponownie zmieniają swe wartości na normalne.

Zmienne wymuszone można podczas monitorowania rozpoznać po wyświetlanych na czerwono wartościach. Istnieje możliwość generalnego usunięcia całej listy wymuszania lub zaznaczenia tylko poszczególnych zmiennych przed wykonaniem polecenia anulowania wymuszania.

W celu kompletnego usunięcia listy wymuszania, czyli anulowania wymuszania **dla wszystkich zmiennych**, należy wybrać jedną z następujących możliwości:

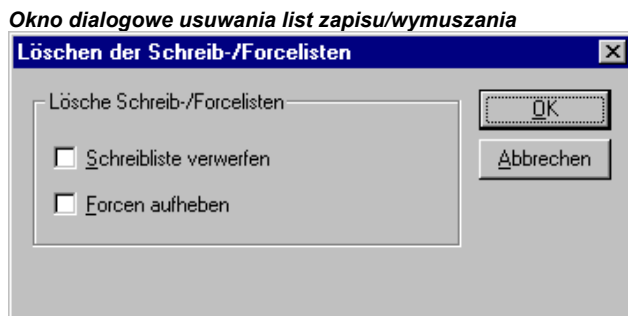
- Polecenie „Anuluj wymuszanie” w menu „Online”.
- Przycisk „Anuluj wymuszanie” w oknie dialogowym „Edycja listy zapisu i wymuszania”.
- Usunięcie kompletnej listy wymuszania za pomocą okna dialogowego „Usuwanie list zapisu/wymuszania” (patrz niżej). Okno to wyświetlane jest po wybraniu polecenia „Online” „Anuluj wymuszanie”.

Aby anulować wymuszanie tylko **dla pojedynczych zmiennych** z listy wymuszania, zmienne te należy najpierw w tym celu zaznaczyć. Można wybrać w tym celu jedną z następujących możliwości. Zmienne zaznaczone do wymuszenia można rozpoznać po dodatkowym tekście <Anuluj wymuszanie> w kolorze turkusowym.

- Dwukrotne kliknięcie wiersza, w którym zadeklarowana jest nieboolean wymuszona zmienna, powoduje otwarcie okna dialogowego „Zapisywanie zmiennej <x>”. Należy nacisnąć przycisk <Anuluj wymuszanie dla tej zmiennej>.
- Ponowne dwukrotne kliknięcie przyciskiem myszy wiersza, w którym zadeklarowana jest wymuszona zmienna boolean, powoduje przełączenie wskaźnika za zmienną na <Anuluj wymuszanie>.
- Należy teraz usunąć wartość w polu edycji kolumny „Wartość wymuszona” w oknie dialogowym zapisywania/wymuszania, które można otworzyć za pomocą polecenia menu „Online”.

Jeśli w przypadku wszystkich żądanych zmiennych za wartością w oknie deklaracji widoczne jest ustawienie „<Anuluj wymuszanie>”, należy wykonać polecenie „Wymuszanie wartości”, które spowoduje przekazanie do programu nowej zawartości listy wymuszania.

Jeśli podczas wykonywania polecenia „Anuluj wymuszanie” aktualna lista zapisu (patrz „Online” „Zapisz wartości”) nie jest pusta, zostanie wyświetlone okno dialogowe „Usuwanie list zapisu/wymuszania”, w którym użytkownik musi zdecydować, czy chce wybrać tylko warunek **Anuluj wymuszanie** czy także odrzucić listę zapisu, czy też oba te warunki.

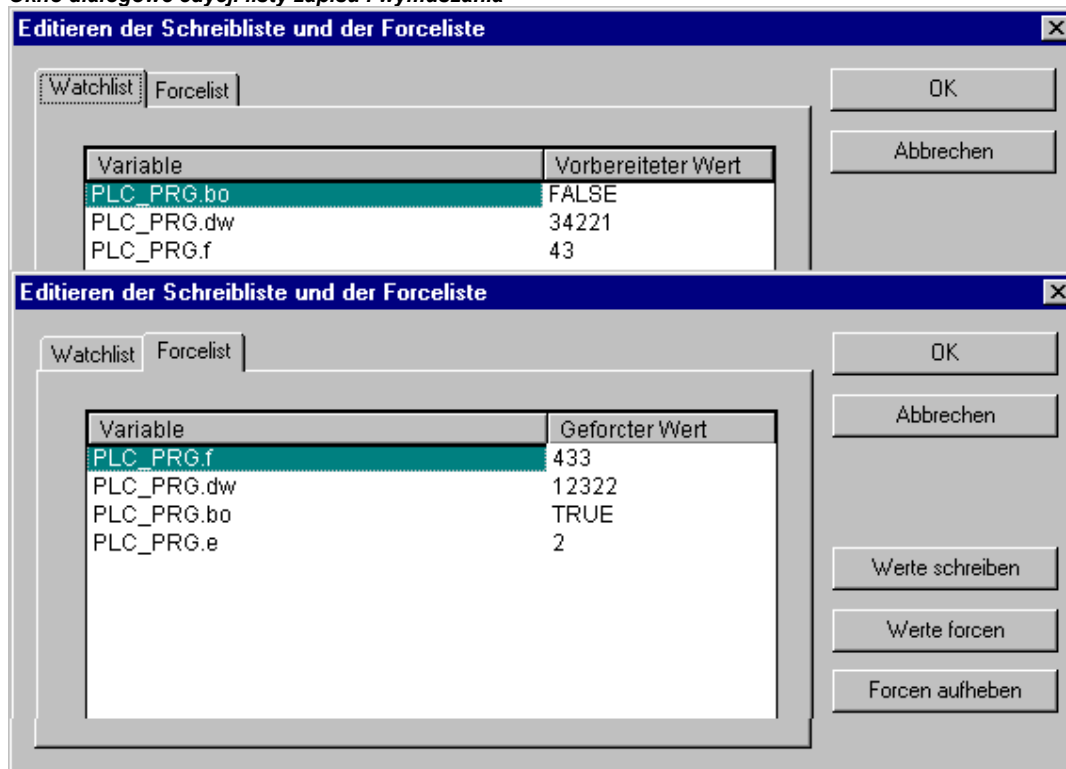


„Online” „Okno dialog. Zapisywanie/wymuszanie”

Polecenie to otwiera okno dialogowe przedstawiające aktualną listę zapisu na dwóch kartach (**Lista kontrolna**) oraz listę wymuszania (**Lista wymuszania**). W tabeli wyświetlana jest każdorazowo nazwa zmiennej oraz jej przygotowana do zapisania wzgl. wymuszona wartość.

Zmienne trafiają w wyniku użycia poleceń „Online” „Zapisz wartości” na listę kontrolną i są przenoszone na listę wymuszania za pomocą polecenia „Online” „Wymuszanie wartości”. Wartości można tu edytować w kolumnach „Przygotowana wartość” lub „Wartość wymuszona”, otwierając wpis w polu edycji poprzez kliknięcie go przyciskiem myszy. W przypadku wpisu niespójnego z typem zostanie wyświetlony komunikat o błędzie. Jeśli wartość zostanie usunięta, oznacza to, że wpis z listy zapisu zostanie usunięty wzgl. że zmienna zostanie zaznaczona w celu anulowania wymuszania, gdy tylko nastąpi wyjście z okna dialogowego za pomocą polecenia innego niż **Anuluj**.

Okno dialogowe edycji listy zapisu i wymuszania



Poniższe polecenia, które odpowiadają poleceniom w menu Online, dostępne są za pośrednictwem przycisków:

Wymuszanie wartości: Wszystkie wpisy aktualnej listy zapisu zostaną przeniesione na listę wymuszania, tzn. wartości zmiennych w sterowniku będą „wymuszane”. Wszystkie zmienne, zaznaczone za pomocą polecenia „Anuluj wymuszanie”, nie będą już wymuszane. Okno dialogowe zostanie następnie zamknięte.

Zapisz wartości: Wszystkie wpisy aktualnej listy zapisu zostaną jednorazowo zapisane w odpowiednich zmiennych w sterowniku. Okno dialogowe zostanie następnie zamknięte.

Anuluj wymuszanie: Wszystkie wpisy z listy wymuszania zostaną usunięte wzgl. jeśli dostępna jest lista zapisu, otwierane jest okno dialogowe „Usuwanie list zapisu/wymuszania”, w którym użytkownik musi zdecydować, czy chce wybrać tylko warunek **Anuluj wymuszanie**, lub **Odrzuć listę zapisu** lub oba te warunki. Okno dialogowe zostanie zamknięte potem wzgl. po zamknięciu okna wyboru.

„Online” „Hierarchia wywołań”

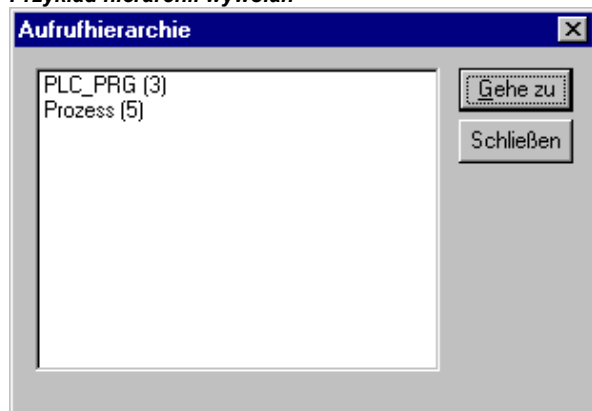
Polecenia tego można użyć, gdy symulacja została zatrzymana w punkcie wstrzymania. Wyświetlone zostanie okno dialogowe z listą modułów, które aktualnie znajdują się w stosie wywołań.

Pierwszym modułem jest zawsze PLC_PRG, ponieważ od niego rozpoczyna się przetwarzanie.

Ostatnim modułem jest zawsze ten moduł, w którym aktualnie zatrzymane zostało przetwarzanie.

Po wybraniu jednego z modułów i naciśnięciu przycisku **Idź do**, wybrany moduł zostanie wczytany do okna i wyświetlony zostanie wiersz lub sieć, w której odbywa się przetwarzanie.

Przykład hierarchii wywołań



„Online” „Kontrola przebiegu”

Od ustawień aktualnego systemu docelowego zależy, czy kontrola przebiegu zostanie włączona przez użytkownika za pośrednictwem tego punktu menu, czy też nie. Jeśli jest włączona, przed punktem menu pojawi się znacznik. Następnie zaznaczony zostanie każdy wiersz lub każda sieć, które zostały przetworzone podczas ostatniego cyklu sterownika.

Pole numeru wiersza wzgl. pole numeru sieci przetworzonych wierszy lub sieci zostanie wyświetlone w kolorze zielonym (ustawienie standardowe). W edytorze IL przy lewej krawędzi każdego wiersza wstawiane jest dodatkowe pole, w którym wyświetlana jest aktualna zawartość akumulatora. W edytorach graficznych schematu funkcji i schematu styków we wszystkich liniach połączeń, które nie przekazują wartości boolean, wstawiane jest dodatkowe pole. Jeśli te wyjścia i wejścia zostaną przyporządkowane, wartość, przekazywana za pomocą linii połączenia, zostanie wyświetlona w tym polu. Linie połączeń, które przekazują wyłącznie wartości boolean, wyświetlane są w kolorze niebieskim (ustawienie standardowe), gdy przekazują wartość TRUE, dzięki czemu można stale śledzić przepływ informacji.

Należy pamiętać:

1. Czas przebiegu programu wydłuża się wskutek kontroli przebiegu. W przypadku programów cyklicznych o wysokim obciążeniu może to prowadzić do przekroczenia cyklu.
2. W aktywnych pozycjach punktów wstrzymania nie jest wyświetlana kontrola przebiegu.
3. Jeśli dla danego zadania zdefiniowana została funkcja Watchdog, zostanie ona wyłączona, dopóki aktywna jest kontrola przebiegu.

„Online” „Symulacja”

Jeśli **Symulacja** jest włączona, przed punktem menu pojawi się znacznik.

W trybie symulacji program użytkownika działa na tym samym komputerze w środowisku Windows. Tryb ten jest używany w celu przetestowania projektu. Komunikacja pomiędzy komputerem a symulacją wykorzystuje mechanizm Windows Message.

Jeśli program nie znajduje się w trybie symulacji, wtedy działa w sterowniku. Komunikacja pomiędzy komputerem a sterownikiem działa zazwyczaj za pośrednictwem złącza szeregowego lub bramy (Gateway).

Status tego znacznika zapisywany jest wraz z projektem.

Wskazówka: Moduły z bibliotek zewnętrznych nie działają w symulacji.

„Online” „Parametry komunikacji”

Polecenie to otwiera okno dialogowe, umożliwiające ustawienie parametrów komunikacji, jakie obowiązują w odniesieniu do komunikacji pomiędzy komputerem lokalnym i systemem czasu przebiegu za pośrednictwem serwera Gateway. (W przypadku zastosowania serwera OPC lub DDE należy w jego konfiguracji ustawić te same parametry komunikacji).

Patrz poniższe punkty:


- Zasada działania systemu Gateway
- Wyświetlanie w oknie dialogowym „Parametry komunikacji”
- Ustawianie żądanego serwera Gateway oraz kanału
- Konfigurowanie nowego kanału dla lokalnego serwera Gateway
- Porady dotyczące edycji parametrów w oknie dialogowym parametrów komunikacji

Zasada działania systemu Gateway

Za pomocą serwera Gateway można uzyskać połączenie lokalnego komputera z jednym lub z wieloma systemami czasu przebiegu. W przypadku każdego serwera Gateway należy specjalnie skonfigurować typ wywoływanych systemów czasu przebiegu. Połączenie z żądanym serwerem Gateway ustawiane jest w komputerze lokalnym. Możliwe jest przy tym, że zarówno ten serwer, jak również systemy czasu przebiegu będą działać w komputerze lokalnym.

Jeśli brama (Gateway) została zainstalowana lokalnie, wtedy wymiana pomiędzy systemem programowania a bramą odbywa się za pomocą pamięci współdzielonej lub za pośrednictwem protokołu TCP/IP. Jeśli serwer Gateway działa na zewnętrznym komputerze, konieczne jest upewnienie się, że został on w nim uruchomiony. Połączenie z nim jest możliwe tylko za pomocą protokołu TCP/IP.

Serwer Gateway uruchamiany jest automatycznie, gdy tylko na komputerze, w którym jest zainstalowany, w programie CoDeSys zostanie otwarte okno dialogowe parametrów komunikacji, lub gdy nastąpi zalogowanie do docelowego systemu czasu przebiegu. Jeśli w komputerze zainstalowana jest wersja serwera Gateway niekompatybilna z systemem programowania, zostanie wyświetlony odpowiedni komunikat. Zalogowanie nie jest wtedy możliwe.

Gotowość systemu Gateway można rozpoznać po pojawieniu się symbolu  z prawej strony na pasku zadań. Gdy nastąpi połączenie z systemem czasu przebiegu za pośrednictwem serwera Gateway, symbol ten zaczyna dodatkowo świecić.

Menu Gateway:

Kliknięcie prawym przyciskiem myszy symbolu Gateway powoduje otwarcie punktów menu **Help**, **About**, **Change Password**, **Inspection**, **Exit**.

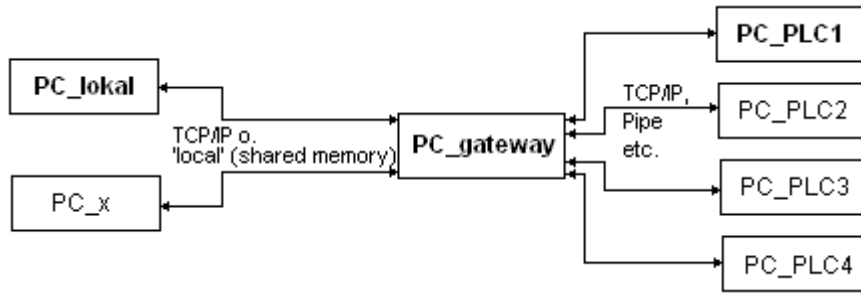
Polecenie **About** umożliwia wyświetlenie informacji o wersji serwera Gateway.

Za pomocą polecenia **Change Password** można otworzyć okno dialogowe, w którym można nadać lub zmienić hasło do lokalnego serwera Gateway. Jeśli jest skonfigurowane takie zabezpieczenie, bezpośrednio po wybraniu serwera Gateway w oknie dialogowym parametrów komunikacji wzgl. po pierwszym zalogowaniu do serwera Gateway następuje żądanie wprowadzenia hasła.

Polecenie **Inspection** umożliwia otwarcie okna dialogowego Gateway Inspector, umożliwiającego monitorowanie kanałów (sprawdzanie, które kanały są dostępne, jakie usługi są aktywne itp.). Za pomocą polecenia menu **Help** można otworzyć pomoc online dot. interfejsu użytkownika Gateway po to, by uzyskać informacje na temat obsługi funkcji kontrolnej Inspector.

Polecenie **Exit** umożliwia wyłączenie serwera Gateway.

Prezentacja systemu Gateway:



PC_lokal to komputer lokalny, **PC_x** to inny komputer również korzystający z serwera Gateway. **PC_gateway** to komputer, w którym zainstalowany jest serwer Gateway, **PC_PLC1** do **PC_PLC4** to komputery, w których działają systemy czasu przebiegu. Rysunek przedstawia moduły osobno, ale jest również możliwe, że serwer Gateway i/lub system czasu przebiegu zainstalowany jest również w komputerze lokalnym.

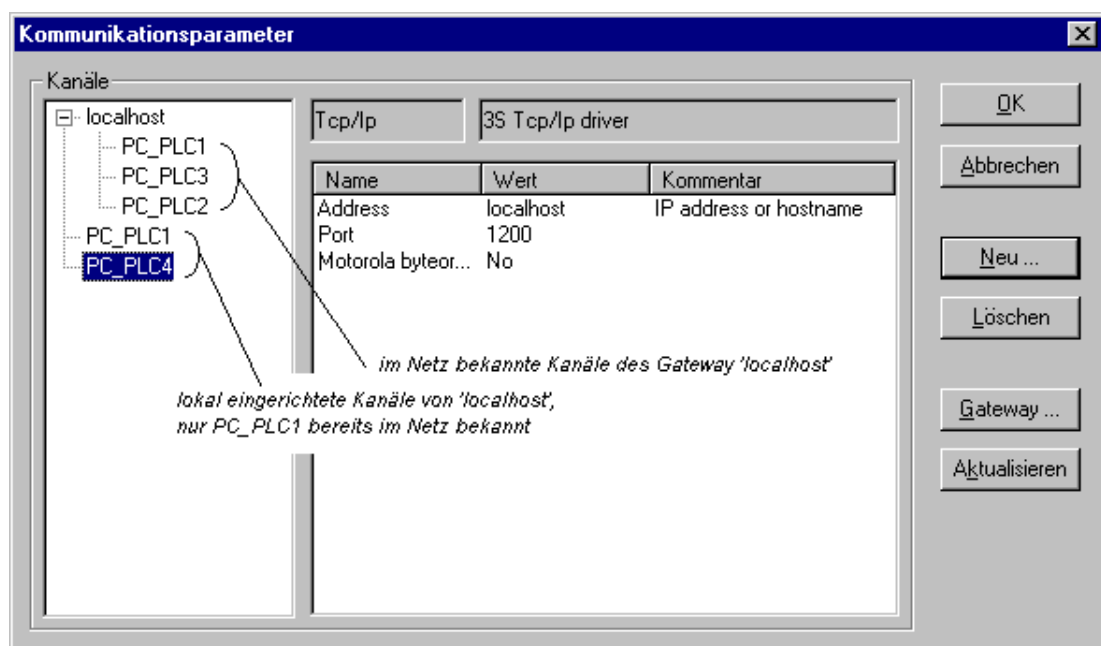
Uwaga: Należy pamiętać, że połączenie lokalnego komputera z serwerem Gateway – jeśli jest on zainstalowany na innym komputerze – jest możliwe tylko za pośrednictwem protokołu TCP/IP, a więc komputer musi być odpowiednio do tego wyposażony! Jeśli natomiast serwer Gateway znajduje się w komputerze lokalnym, możliwe jest również połączenie za pośrednictwem pamięci współdzielonej (szeregowo).
Połączenia serwera Gateway z różnymi komputerami, na których działają systemy czasu przebiegu, odbywają się za pomocą różnych protokołów (TCP/IP, Pipe itp.).

Wyświetlanie w oknie dialogowym „Parametry komunikacji”

To okno dialogowe służy do wybierania serwera Gateway, za pomocą którego ma być nawiązane połączenie np. ze sterownikiem. Ponadto dla serwera Gateway, zainstalowanego na lokalnym komputerze, można tworzyć nowe kanały i definiować ich parametry połączenia, dzięki czemu będą one później dostępne również dla innych komputerów w sieci.

Aktualnie obowiązujące ustawienia można przywołać w każdej chwili za pomocą przycisku **Aktualizuj**.

Jeśli parametry komunikacji zostały skonfigurowane zgodnie z przykładowym schematem, pokazanym w punkcie „Zasada działania serwera Gateway”, okno dialogowe będzie wyglądać w następujący sposób:



Rubryka **Kanały** zawiera dwie kategorie połączeń:

- Z jednej strony wyświetlane są wszystkie kanały, jakie oferuje w sieci aktualnie podłączony serwer Gateway o nazwie „localhost”, np. dla połączenia z komputerem sterującym. (Serwer Gateway został wybrany za pośrednictwem okna dialogowego, wybieranego przy użyciu przycisku „Gateway”). Na najwyższej pozycji, za znakiem minus znajduje się adres wzgl. nazwa serwera Gateway. W tym przykładzie serwer działa w komputerze lokalnym. Oferowany domyślnie adres „localhost” odpowiada zazwyczaj adresowi IP 127.0.0.1 komputera lokalnego (PC_lokal). Poniżej (z wcięciem do prawej) widnieją trzy adresy komputerów z systemami czasu przebiegu, dla których na serwerze Gateway skonfigurowane są kanały (PC_PLC1 do 3). Kanały te można skonfigurować zarówno z komputera lokalnego, jak też z innego komputera (PC_x), jakie były/są połączone z serwerem Gateway.
- Druga kategoria wyświetlanych kanałów obejmuje wszystkie połączenia z serwerem Gateway, konfigurowane z lokalnego komputera (tu „localhost”) – np. za pomocą okna dialogowego konfiguracji. Tworzą one „gałąź” prowadzącą od znaku minus bezpośrednio w dół do PC_PLC1 i PC_PLC4. Adresów kanałów nie trzeba koniecznie podawać na serwerze Gateway. W przedstawionym powyżej przykładzie parametry konfiguracji dla PC_PLC4 są wprawdzie zapisane lokalnie w projekcie, jednakże do serwera Gateway zostałyby przekazane dopiero podczas następnego logowania do systemu czasu przebiegu. Miało to już miejsce w przypadku PC_PLC1, który właśnie dlatego w drzewie kanałów wyświetlany jest dodatkowo jako „podgałąź” „lokalnego hosta”.

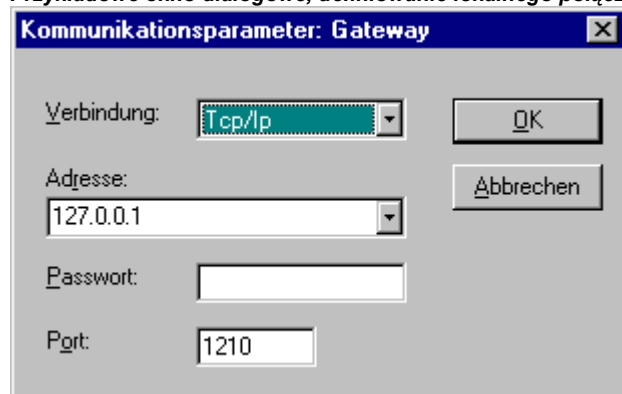
W środkowej części okna dialogowego można zawsze znaleźć oznaczenie kanału wybranego z lewej strony, a w pozycjach **Nazwa**, **Wartość** i **Komentarz** odpowiednie parametry.

Ustawianie żądanego serwera Gateway oraz kanału

1. Wybieranie serwera Gateway w oknie dialogowym parametrów komunikacji:

Aby zdefiniować połączenie z żądanym serwerem Gateway, za pomocą przycisku **Gateway** należy otworzyć okno dialogowe „Parametry komunikacji Gateway”.

Przykładowe okno dialogowe, definiowanie lokalnego połączenia z serwerem Gateway



Można w nim wprowadzić wzgl. edytować poniższe dane:

- typ połączenia komputera użytkownika z komputerem, w którym działa serwer Gateway, wybrany w celu użycia. Jeśli serwer Gateway działa na komputerze lokalnym, połączenie jest możliwe za pośrednictwem pamięci współdzielonej („lokalnie”) lub też za pośrednictwem protokołu „TCP/IP”, jeśli konieczne jest połączenie z innym komputerem, można to zrobić tylko za pośrednictwem TCP/IP.
- adres komputera, w którym działa serwer Gateway, wybrany w celu użycia: adres IP wzgl. odpowiednia nazwa symboliczna, np. localhost. Uwaga: Niedozwolone jest poprzedzanie zakresów adresu cyfrą „0” (np. niedopuszczalne: „010.107.084.050”, wpis musi mieć postać: „10.107.84.50”). Podczas pierwszej konfiguracji jako nazwa komputera (adres) oferowany jest standardowo wpis 'localhost', co oznacza, że nastąpi wywołanie serwera Gateway zainstalowanego lokalnie. Nazwa „localhost” jest w większości przypadków ustawiana automatycznie w sposób identyczny z lokalnym adresem IP 127.0.0.1, ew. może jednak być konieczne wpisanie jej bezpośrednio w polu „Adres”. Jeśli zachodzi potrzeba połączenia

z serwerem Gateway na innym komputerze, nazwę „localhost” należy zastąpić jego nazwą lub adresem IP.

- hasło wybranego serwera Gateway, jeśli znajduje się on na komputerze zdalnym. Jeśli zostanie ono wprowadzone błędnie lub nie zostanie wprowadzone, wyświetlany jest komunikat o błędzie. Należy pamiętać: Zainstalowanemu lokalnie serwerowi Gateway można nadać hasło w następujący sposób: Kliknąć prawym przyciskiem myszy symbol Gateway na dole, z prawej strony paska symboli i wybrać wpis „Change password”. Zostanie wyświetlone okno dialogowe zmiany lub wprowadzania hasła. W przypadku dostępu do lokalnego serwera Gateway, ewentualnie wprowadzone hasło nie będzie sprawdzane.
- **port** komputera, w którym działa wybrany serwer Gateway; z reguły wartość odpowiednia dla wybranego serwera Gateway jest już wypełniona domyślnie

Jeśli okno dialogowe zostanie zamknięte za pomocą przycisku **OK**, w rubryce **Kanały** okna dialogowego „Parametry komunikacji” na najwyższej pozycji zostanie wyświetlony odpowiedni wpis (adres komputera), zaś poniżej dostępne kanały serwera Gateway.

2. Ustawianie żadanego kanału na wybranym serwerze Gateway:

Klikając myszą na odpowiedni wpis, należy wybrać jeden z kanałów. Następnie odpowiednie parametry zostaną wyświetlone w tabeli. Jeśli nawiązanie połączenia z wybranym serwerem Gateway jest niemożliwe – ponieważ być może nie został uruchomiony lub niezgodny jest adres – w nawiasie za adresem pojawi się wpis „niepołączony” i zostanie wyświetlony komunikat „Nie znaleziono serwera Gateway o takich ustawieniach”. W takim wypadku należy przeprowadzić skróconą kontrolę.

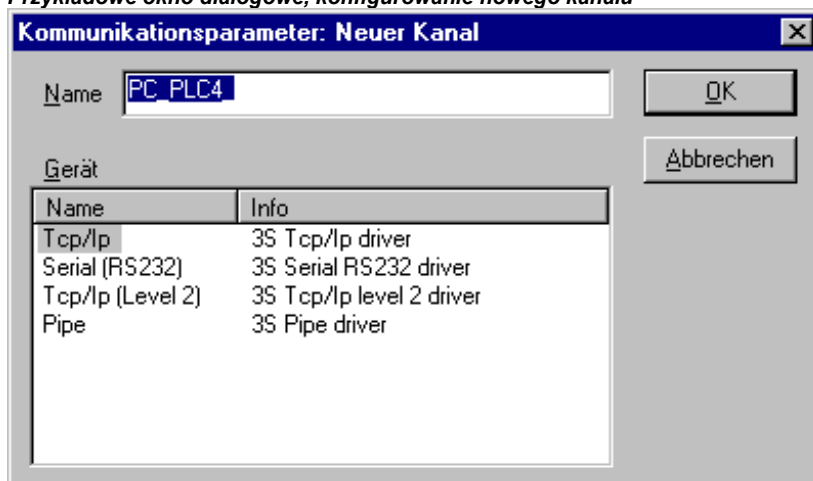
Jeśli wymagany kanał został ustawiony, okno dialogowe można zamknąć za pomocą polecenia **OK**. Ustawienia są zapisywane wraz z projektem.

Konfigurowanie nowego kanału dla lokalnego serwera Gateway

W oknie dialogowym parametrów komunikacji można skonfigurować nowe kanały dla podłączonego aktualnie serwera Gateway. Kanały te będą dostępne dla kolejnych połączeń serwera, np. dla połączenia ze sterownikiem. Dostępne możliwości zależą przy tym od sterowników urządzeń, zainstalowanych indywidualnie na komputerze.

Należy nacisnąć przycisk **Nowy**. Zostanie wyświetlone okno dialogowe „Parametry komunikacji: Nowy kanał”:

Przykładowe okno dialogowe, konfigurowanie nowego kanału



- W polu wprowadzania Nazwa automatycznie oferowana jest nazwa użyta dla wprowadzonego ostatnio kanału. Jeśli dotychczas nie został jeszcze zdefiniowany kanał, oferowana jest aktualna nazwa Gateway z dolnym myślnikiem, np. „localhost_”. Tu można edytować nazwę kanału. Nazwy kanałów mają charakter czysto informacyjny, jednoznaczność nie jest konieczna, ale zalecana.

- W tabeli w kolumnie Urządzenie znajduje się lista urządzeń dostępnych w komputerze Gateway. Z kolumny Nazwa, klikając przyciskiem myszy, należy wybrać oferowany sterownik – w kolumnie Info znajduje się ew. komentarz.

W przypadku zamknięcia okna dialogowego „...Nowy kanał” za pomocą przycisku **OK**, nowo zdefiniowany kanał zostanie wyświetlony w oknie dialogowym „Parametry komunikacji” jako kolejny wpis w kategorii **Kanały** (na najniższej pozycji pod znakiem minus). Jest on początkowo zapisany jedynie lokalnie w projekcie (patrz wyżej)! W tym stadium można edytować kolumnę **Wartość**. Ustawione parametry należy potwierdzić za pomocą przycisku **OK** i opuścić w ten sposób okno dialogowe „Parametry komunikacji”.

Aby nowo ustawiony kanał Gateway był dostępny wraz ze swymi parametrami także na serwerze Gateway xy, a tym samym również dla innych komputerów korzystających z tego serwera Gateway, konieczne jest **zalogowanie do systemu czasu przebiegu**. Jeśli potem zostanie ponownie otwarte okno dialogowe „Online” „Parametry komunikacji”, nowy kanał zostanie wyświetlony w widoku drzewa kanałów pod adresem wzgl. nazwą komputera Gateway xy z dodatkowym wcięciem względem swej dotychczasowej pozycji. Wskazuje to na fakt, że jest on dostępny w sieci. Można teraz otworzyć okno dialogowe parametrów komunikacji na komputerze innym niż lokalny, wybrać Gateway xy i używać jego nowego kanału.

Jeśli podczas logowania zostanie wyświetlony błąd komunikacji, może to być ew. zamknięte złącze (np. COM1 w przypadku połączenia szeregowego), ponieważ może ono być zajęte przez inne urządzenie. Być może również nie jest włączony sterownik.

Parametrów kanału określonego raz na serwerze Gateway nie można edytować już w oknie dialogowym konfiguracji. Pola parametrów mają kolor szary. Połączenie można usunąć, dopóki nie jest ono aktywne.

Uwaga: Należy pamiętać, że procedura usuwania kanału jest nieodwracalna. Usunięcie następuje w chwili naciśnięcia przycisku Usuń!

Porady dotyczące edycji parametrów w oknie dialogowym parametrów komunikacji:

W oknie dialogowym parametrów komunikacji można edytować tylko pola tekstowe z kolumny **Wartość**.

Po wybraniu za pomocą myszy pola tekstowego można przejść do trybu edycji, klikając je dwukrotnie lub naciskając klawisz spacji. Za pomocą klawisza <Enter> można zakończyć wprowadzanie tekstu.

Klawisze <Tab> wzgl. <Shift> + <Tab> umożliwiają przejście do następnej wzgl. poprzedniej możliwości edycji.

Podczas edytowania wartości numerycznych, za pomocą klawiszy strzałek wzgl. klawiszy przewijania można zmniejszyć lub zwiększyć wartość o jedną lub dziesięć jednostek. Dwukrotne kliknięcie przyciskiem myszy zwiększa wartość każdorazowo o jedną jednostkę. W przypadku wartości numerycznych skonfigurowana jest kontrola typu: kombinacje klawiszy <Ctrl> + <Home> wzgl. <Ctrl> + <End> umożliwiają każdorazowo ustawienie najniższej wzgl. najwyższej wartości spośród wartości możliwych do wprowadzenia dla danego typu parametru.

Skrócona kontrola w przypadku nieudanego połączenia z Gateway

Następujące punkty należy sprawdzić w przypadku nieudanego połączenia z wybranym komputerem Gateway (w oknie dialogowym parametrów komunikacji, za adresem serwera Gateway, w polu „Kanały” zostanie wyświetlony komunikat „niepołączony”):

- Czy serwer Gateway jest uruchomiony (czy wyświetlony jest trójkolorowy symbol u dołu z prawej strony na pasku symboli)?
- Czy adres IP, wprowadzony w oknie dialogowym „Gateway: Parametry komunikacji” jest rzeczywiście adresem serwera Gateway? (sprawdzić za pomocą procedury „ping”)
- Czy połączenie TCP/IP działa lokalnie? Być może błąd leży w TCP/IP.

„Online” „Wczytaj kod źródłowy”

Za pomocą tego polecenia kod źródłowy projektu wczytywany jest do sterownika. Nie należy go mylić z kodem powstającym podczas kompilacji projektu! Opcje obowiązujące podczas pobierania (czas, zakres) można ustawić w oknie dialogowym „Projekt” „Opcje” „Pobieranie źródła”.

„Online” „Utwórz projekt bootowalny”

Jeśli to polecenie zostanie wykonane w trybie **online**, skompilowany projekt zostanie zapisany w sterowniku w taki sposób, aby podczas ponownego uruchomienia mógł być automatycznie wczytany przez sterownik. W zależności od systemu docelowego zapisywanie projektu bootowalnego odbywa się w różny sposób. Np. w systemach 386 tworzone są trzy pliki: **default.prg** – zawierający kod projektu, **default.chk** – zawierający sumę kontrolną kodu, **default.sts** – zawierający status sterownika po ponownym uruchomieniu (Start/Stop).

Polecenie „Online” „Utwórz projekt bootowalny” jest dostępne także w trybie **offline**, jeśli projekt został bezbłędnie skompilowany. W takim przypadku w katalogu projektu tworzony jest dla projektu bootowalnego plik **<nazwa projektu>.prg** oraz dla sumy kontrolnej kodu plik **<nazwa projektu>.chk**. Po zmianie nazwy można je skopiować do sterownika.

Jeśli w sterowniku znajduje się już projekt bootowalny, a ponadto w opcjach projektu, w kategorii „Obszar roboczy” została włączona opcja „Praca online w trybie bezpieczeństwa”, wtedy podczas tworzenia nowego projektu bootowalnego zostanie wyświetlone okno dialogowe, w którym prezentowane są **informacje o projekcie** – zarówno projektu bootowalnego wczytanego do systemu programowania, jak i projektu znajdującego się w sterowniku. Funkcję tę musi jednak obsługiwać system docelowy!

Również w zależności od ustawień w systemie docelowym podczas tworzenia projektu bootowalnego w trybie offline tworzony jest równocześnie nowy plik *.ri (informacje na temat pobierania i kompilacji). W razie potrzeby (zależnie od systemu docelowego), jeśli plik taki już istnieje, wyświetlane jest okno dialogowe z zapytaniem.

Wskazówka: Jeśli aktywna jest opcja projektu „Wewnętrznie przy proj. bootowalnym” (kategoria Pobieranie źródła), w przypadku użycia polecenia „Online” „Utwórz projekt bootowalny” wybrany zakres danych źródła zostanie automatycznie wczytany do sterownika.

„Online” „Zapisz plik w sterowniku”

Za pomocą tego polecenia można wczytać dowolny plik do sterownika. Otwiera ono okno dialogowe „Zapisz plik w sterowniku”, w którym można zaznaczyć wybrany plik. Po potwierdzeniu wyboru za pomocą przycisku **Otwórz** okno dialogowe zostanie zamknięte, a plik zostanie wczytany do sterownika i tam zapisany pod tą samą nazwą. Wczytywanie wskazywane jest za pomocą wskaźnika postępu.

Za pomocą polecenia „Online” „Wczytaj plik ze sterownika” można ponownie wczytać plik zapisany w sterowniku.

„Online” „Wczytaj plik ze sterownika”

Przy użyciu tego polecenia można ponownie wczytać plik zapisany w sterowniku za pomocą polecenia „Online” „Zapisz plik w sterowniku”. Zostanie wyświetlone okno dialogowe **Wczytaj plik ze sterownika**. W polu **Nazwa pliku** należy wpisać nazwę żadanego pliku, a za pomocą okna wyboru ustawić katalog w komputerze, do którego powinien on zostać wczytany po zamknięciu okna dialogowego za pomocą przycisku **Zapisz**.

6.7 Okno...

W punkcie menu „**Okno**” znajdują się wszystkie polecenia, dotyczące zarządzania oknami. Są to zarówno polecenia automatycznego rozmieszczania oknem, jak również dotyczące otwierania Menedżer biblioteki, dziennika oraz do przełączania pomiędzy otwartymi oknami.

Na końcu menu znajduje się lista wszystkich otwartych okien w takiej kolejności, w jakiej zostały otwarte. Kliknięcie na danym wpisie powoduje przejście do żądanego okna. Przed aktywnym oknem zostanie wyświetlony znacznik.

W poniższych akapitach opisano poszczególne polecenia menu „Okno”:

„Okno” „Sąsiadująco”

Za pomocą tego polecenia można rozmieścić wszystkie okna w obszarze roboczym obok siebie, dzięki czemu nie nakładają się one i wypełniają cały obszar roboczy.

„Okno” „Jedno pod drugim”

Za pomocą tego polecenia można ustawić wszystkie okna w obszarze roboczym jedno pod drugim. Dzięki temu nie nakładają się one i wypełniają cały obszar roboczy.

„Okno” „Kaskada”

Za pomocą tego polecenia można ustawić wszystkie okna w obszarze roboczym w układzie kaskady jedno za drugim.

„Okno” „Rozmieść symbole”

Za pomocą tego polecenia można rozmieścić wszystkie zminimalizowane okna w obszarze roboczym w rzędzie przy dolnej krawędzi obszaru roboczego.

„Okno” „Zamknij wszystkie”

Za pomocą tego polecenia można zamknąć wszystkie okna otwarte w obszarze roboczym.

„Okno” „Komunikaty”

Kombinacja klawiszy: <Shift> + <Esc>

Za pomocą tego polecenia można otworzyć wzgl. zamknąć okno komunikatów wraz z komunikatami z ostatniego procesu kompilacji, kontroli lub porównania.

Jeśli okno komunikatów jest otwarte, w menu przed poleceniem pojawi się znacznik.

„Okno” „Zarządzanie biblioteką”

Polecenie to umożliwia otwarcie Menedżera biblioteki.

„Okno” „Dziennik”

Za pomocą tego polecenia można otworzyć okno dziennika, w którym wyświetlane są protokoły z ostatniej sesji online (patrz rozdział 9.5, Zasoby, Dziennik).

6.8 Pomoc...

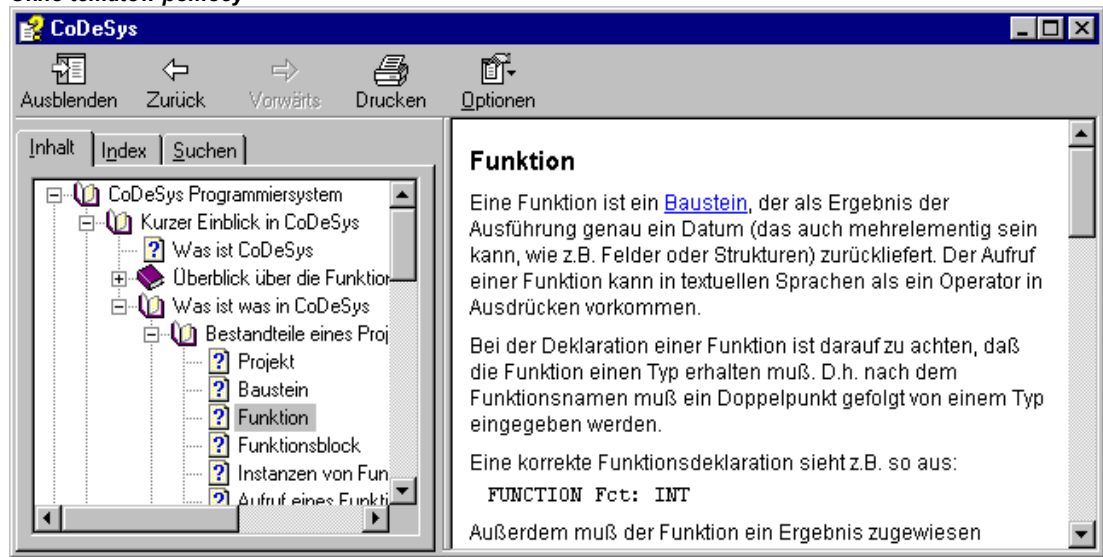
„Pomoc” „Zawartość” i „Szukaj”

Za pomocą polecenia „Zawartość” wzgl. „Szukaj” z menu „Pomoc” można otworzyć okno tematów pomocy, wyświetlane za pomocą przeglądarki pomocy w formacie HTML (począwszy od wersji Internet Explorera v4.1).

Karta **Zawartość** zawiera spis treści. Klikając myszą na jeden z symboli książki wzgl. znaki plus/minus przed nimi, można otwierać i zamykać tematy pomocy. Zawartość strony zaznaczonej w spisie treści wyświetlana jest w prawej części okna Pomocy. Powiązania w tekście Pomocy z innymi stronami Pomocy wzgl. rozwijane fragmenty tekstu lub rysunki można rozpoznać po innym kolorze oraz podkreśleniu. Kliknięcie przyciskiem myszy otwiera daną stronę docelową wzgl. rozszerzony tekst lub rysunek.

Na karcie **Indeks** można również szukać określonego terminu, natomiast karta **Szukaj** umożliwia wyszukiwanie w całym tekście wszystkich stron pomocy. Należy postępować wg instrukcji opisanych na poszczególnych kartach.

Okno tematów pomocy



Pomoc kontekstowa

Kombinacja klawiszy: <F1>

W celu otwarcia okna Pomocy online można nacisnąć klawisz <F1> w aktywnym oknie, oknie dialogowym lub przy wybranym poleceniu menu. Jeśli wybrane jest polecenie menu, zostanie wyświetlona strona pomocy bezpośrednio dla tego polecenia. Jeśli zaznaczony zostanie tekst (np. słowo kluczowe, funkcja standardowa lub komunikat o błędzie w oknie komunikatów), za pomocą klawisza <F1> można wyświetlić odpowiednie wskazówki pomocy.

8 Edytory

8.1 Poniższe uwagi dotyczą wszystkich edytorów

Struktura edytora

Edytory wszystkich języków programowania w CoDeSys składają się z części deklaracyjnej oraz ze szkieletu. Szkielet może składać się z edytora tekstowego lub graficznego, część deklaracyjną stanowi zawsze edytor tekstowy. Szkielet oraz część deklaracyjna oddzielone są od siebie poziomym separatorem ekranu, który można w razie potrzeby przesunąć, klikając go przyciskiem myszy i przesuwając w górę lub w dół przy naciśniętym przycisku.

Granice wydruku

Pionowe i poziome ograniczenia stron, odnoszące się do druku zawartości edytora, wyświetlane są w postaci czerwonych, przerywanych linii, jeśli została wybrana uprzednio opcja „**Pokaż obszary wydruku**” w opcjach projektu w oknie dialogowym „**Obszar roboczy**”. Obowiązują przy tym wartości domyślne ustawionej drukarki, jak również rozmiar i szablon wydruku wybrany w menu „**Plik**” „**Ustawienia drukowania**”. Jeśli nie ustawiono drukarki wzgl. szablonu wydruku, przyjmowane jest przyporządkowanie domyślne (Default.DFR i drukarka standardowa). Poziome granice wydruku oznaczane są w taki sposób, jakby w oknie „Ustawienia drukowania” wybrana została opcja „Nowa strona na obiekt” lub „Nowa str. na obiekt podrz.”. Najniższa granica nie jest wyświetlana.

Uwaga: Dokładne wyświetlanie granic wydruku zapewnione jest tylko w przypadku ustawienia współczynnika powiększenia na 100%.

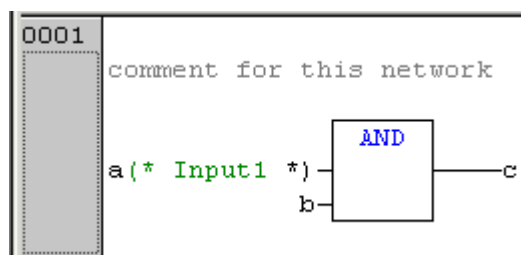
Komentarz

Komentarze użytkownika muszą zawierać się w specjalnych ciągach znaków „(*) i „*”)”. *Przykład:* (*To jest komentarz.*)

Komentarze są dozwolone we wszystkich edytorach tekstowych w dowolnym miejscu, tzn. we wszystkich deklaracjach w językach IL i ST oraz w samodzielnie zdefiniowanych typach danych. Jeśli projekt zostanie wydrukowany przy użyciu **szablonu wydruku**, w opartych na tekście częściach programu komentarz wprowadzony przy deklaracji zmiennej widoczny będzie zawsze za zmienną.

W edytorach graficznych FBD i LD do każdej sieci można dodać komentarze. Należy w tym celu wyszukać sieć, do której dołączony zostanie komentarz i użyć polecenia „**Wstaw**” „**Komentarz**”. Ponadto komentarze można wstawiać tam, gdzie wprowadzane są nazwy zmiennych.

Przykład komentarza sieci w FBD oraz komentarza za zmienną wejściową:



W LD można do każdego styku wzgl. do każdej cewki dołączyć komentarz, jeśli możliwość taka jest ustawiona w opcjach wyświetlania w menu „Dodatki” „Opcje”.

W CFC istnieją specjalne moduły komentarzy, które można umieszczać w dowolny sposób.

W AS można wprowadzić komentarz do kroku w oknie dialogowym edycji atrybutów kroku.

Dozwolone są również **komentarze zagnieżdżone**, jeśli w oknie dialogowym „Projekt” „Opcje” „Opcje kompilacji” została włączona odpowiednia opcja.

Przytrzymując krótko w trybie online kursor myszy nad zmienną, można wyświetlić w razie potrzeby adres oraz komentarz do zmiennej w oknie **etykiety**.

Powiększanie wywołanego modułu

Kombinacja klawiszy: <Alt> + <Enter>

Polecenie to jest dostępne w menu podręcznym (<F2>) lub w menu „Dodatki”, jeśli kursor w edytorze tekstowym znajduje się na nazwie wywołanego modułu wzgl. gdy w edytorze graficznym zaznaczone jest pole danego modułu. Powiększenie powoduje otwarcie danego modułu w oknie edytora.

Jeśli jest to moduł z biblioteki, wywoływany jest Menedżer bibliotek i wyświetlany jest odpowiedni moduł.

„Dodatki” „Otwórz instancję”

Polecenie to odpowiada poleceniu „Projekt” „Otwórz instancję”. Jest ono dostępne w menu podręcznym lub też w menu „Dodatki”, jeśli kursor w edytorze tekstowym znajduje się na nazwie bloku funkcji wzgl. gdy w edytorze graficznym zaznaczone jest pole bloku funkcji.

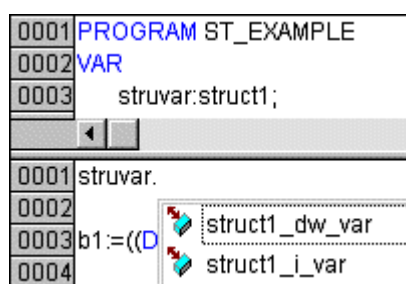
Funkcja „Wyświetl listę składników”

Jeśli w opcjach projektu, w kategorii „Edytor” zaznaczona jest opcja „Wyświetl listę składników”, we wszystkich edytorach, w Menedżerze podglądu zmiennych i receptur, w wizualizacji oraz w konfiguracji śledzenia dostępna jest „funkcja Intellisense”:

- Jeśli zamiast identyfikatora wprowadzona zostanie kropka „.”, otworzy się lista wyboru wszystkich zmiennych lokalnych i globalnych. Z tej listy można wybrać element i naciskając klawisz „Enter” wprowadzić go po kropce. Funkcja wstawiania działa również po dwukrotnym kliknięciu elementu na liście.
- Jeśli jako identyfikator wstawiona zostanie instancja bloku funkcji lub zmienna zdefiniowana jako struktura z następującą po niej kropką, po wprowadzeniu kropki otworzy się lista wyboru zmiennych wejściowych i wyjściowych bloku funkcji wzgl. składników struktury.

Przykład:

Po wprowadzeniu „struvar.” -> oferowane będą składniki struktury struct1:



- Jeśli jako identyfikator zostanie wprowadzony dowolny ciąg znaków, a następnie naciśnięta kombinacja klawiszy <Ctrl> + <Spacja>, zostanie wyświetlona lista wyboru wszystkich dostępnych w projekcie modułów i zmiennych globalnych, przy czym pierwszy, który zaczyna się od tego ciągu jest zaznaczony i można go zastosować w programie naciskając klawisz „Enter”.

Etykieta offline dla zmiennych

W trybie offline w przypadku wszystkich edytorów obowiązuje następująca zasada: Jeśli kursor myszy znajduje się na edytowalnym identyfikatorze, w etykietyce wyświetlana jest nazwa identyfikatora, klasa zmiennej (np. VAR_GLOBAL), typ danych, atrybuty zmiennej (np. RETAIN), adres oraz komentarz.

8.2 Edytor deklaracji

8.2.1 Praca z użyciem edytora deklaracji

Edytor deklaracji jest używany w przypadku deklaracji zmiennych modułów oraz zmiennych globalnych do deklarowania typów danych oraz w Menedżerze podglądu zmiennych i receptur. Posiada on funkcje typowe dla systemu WIndows oraz możliwość wykorzystania funkcji IntelliMouse, jeśli jest zainstalowany odpowiedni sterownik.

W trybie zastępowania na pasku stanu wyświetlany jest na czarno wpis „ZAS”, zaś pomiędzy trybem zastępowania i wpisywania można przełączać się za pomocą klawisza <Ins>.

Deklaracja zmiennych obsługuje funkcję **Syntaxcoloring**.

Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

Wskazówka: Należy pamiętać o możliwości wpływania na właściwości zmiennej za pomocą pragmy podczas procesu kompilacji wzgl. prekompilacji (patrz rozdział 8.2.3).

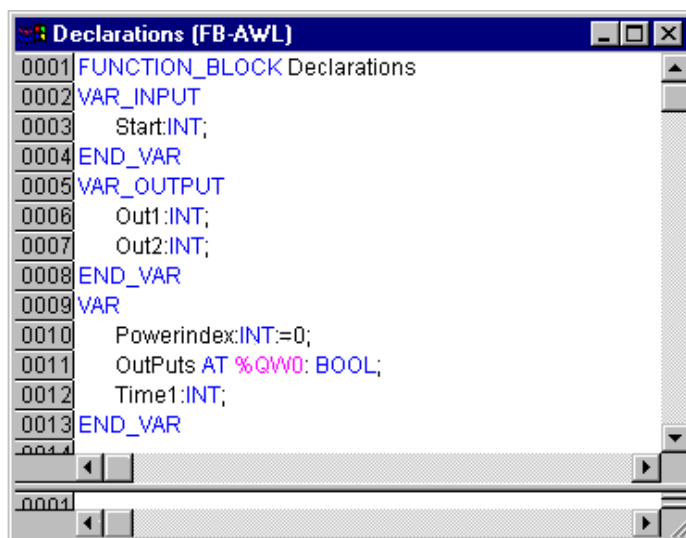
Część deklaracyjna

W części deklaracyjnej modułu deklarowane są wszystkie zmienne, które są używane tylko w tym module. Mogą to być zmienne wejściowe, wyjściowe, wejściowe/wyjściowe, zmienne lokalne, zmienne remanentne i stałe. Składnia deklaracji opiera się na standardzie IEC 61131-3.

Nadawanie identyfikatorów opisano również poniżej w rozdziałach „Deklaracja zmiennej”, jak również Załącznik L, „Zalecenia odnośnie nadawania identyfikatorów”.

Należy pamiętać o możliwości zastosowania **Szablону obiektu** do początkowego wypełnienia części deklaracyjnej podczas tworzenia nowego obiektu typu „Zmienne globalne”, „Typ danych”, „Funkcja”, „Moduł funkcji” lub „Program” (patrz rozdział 4.3, „Plik” „Otwórz z szablonu”). Ponadto istnieje możliwość wstawienia instrukcji pragma (patrz rozdział 8.2.3).

Przykład prawidłowej deklaracji zmiennych w edytorze CoDeSys:



```

0001 FUNCTION_BLOCK Declarations
0002 VAR_INPUT
0003     Start:INT;
0004 END_VAR
0005 VAR_OUTPUT
0006     Out1:INT;
0007     Out2:INT;
0008 END_VAR
0009 VAR
0010     Powerindex:INT:=0;
0011     OutPuts AT %QW0: BOOL;
0012     Time1:INT;
0013 END_VAR
0014
0001
  
```

Zmienne wejściowe

Pomiędzy słowami kluczowymi

VAR_INPUT i **END_VAR** deklarowane są wszystkie zmienne, służące jako zmienne wejściowe danego modułu, co oznacza, że w miejscu wywołania może być podana wartość zmiennej podczas wywołania.

Przykład:

```
VAR_INPUT
  iIn1:INT; (* 1. zmienna wejściowa*)
END_VAR
```

Przykład dostępu do zmiennej wejściowej bloku funkcji

Blok funkcji FBD posiada zmienną wejściową in1 typu int.

Deklaracja:

```
PROGRAM prog
VAR
  Inst:FUB;
END_VAR
```

Część programowa w IL:

```
LD 17
ST inst.iIn1
CAL inst
```

Część programowa w ST:

```
inst(iIn1:=17);
```

Zmienne wyjściowe

Pomiędzy słowami kluczowymi

VAR_OUTPUT i **END_VAR** deklarowane są wszystkie zmienne, służące jako zmienne wejściowe danego modułu, co oznacza, że wartości te są zwracane do modułu wywołującego, tam można je sprawdzić i stosować dalej.

Przykład:

```
VAR_OUTPUT
  iOut1:INT; (* 1. zmienna wyjściowa*)
END_VAR
```

Zmienne wejściowe/wyjściowe

Pomiędzy słowami kluczowymi

VAR_IN_OUT i **END_VAR** można zadeklarować wszystkie zmienne, które służą jako zmienne wejściowe i wyjściowe danego modułu.

Uwaga: W przypadku tych zmiennych zmieniana jest bezpośrednio wartość przekazanych zmiennych („Przełącz jako wskaźnik”, Call-by-Reference). Dlatego też wartość wejściowa takich zmiennych nie może być stała. Dlatego również zmienne VAR_IN_OUT bloku funkcji nie mogą być wczytywane i zapisywane z zewnątrz bezpośrednio za pośrednictwem polecenia <instancja bloku funkcji>.<zmienna wejściowa/wyjściowa>! *Przykład:*

Przykład:

```
VAR_IN_OUT
  iInOut1:INT; (* 1. zmienna wejściowa/wyjściowa *)
END_VAR
```

Zmienne lokalne

Pomiędzy słowami kluczowymi

VAR i **END_VAR** deklarowane są wszystkie zmienne lokalne modułu. Nie mają one połączenia na zewnątrz, tzn. z zewnątrz nie można dokonać w nich zapisu.

Przykład:

```
VAR
  iLoc1:INT; (* 1. zmienna lokalna *)
END_VAR
```

Zmienne remanentne

Zmienne remanentne mogą zachować swą wartość przez typowy czas przebiegu programu. Należą do nich zmienne retain i zmienne trwałe.

Przykład:

```
VAR RETAIN
  iRem1:INT; (* 1. zmienna remanentna *)
END_VAR
```

- Zmienne retain oznaczone są słowem kluczowym **RETAIN**. Zmienne te zachowują swą wartość po niekontrolowanym zakończeniu, jak również po zwykłym wyłączeniu i włączeniu sterownika (np. w przypadku polecenia „Online” „Reset”). Podczas ponownego uruchamiania programu następuje dalsze przetwarzanie w oparciu o zapisane wartości. Przykładem zastosowania mógłby być licznik sztuk w instalacji produkcyjnej, który powinien kontynuować liczenie po awarii zasilania.

Wszystkie inne zmienne inicjowane są ponownie, albo w oparciu o inicjowane wartości, albo też na podstawie inicjalizacji standardowych.

W przeciwieństwie do zmiennych trwałych zmienne retain są jednak inicjowane od nowa po ponownym pobraniu programu.

- Zmienne trwałe oznaczone są słowem kluczowym **PERSISTENT**. W przeciwieństwie do zmiennych retain zachowują one swą wartość tylko po ponownym pobraniu („Online” „Wczytaj”), lecz nie po poleceniach „Online” „Reset”, „Online” „Reset źródła” lub „Online” „Zimny reset” (patrz rozdział 6.6), ponieważ nie są one zapisywane w „obszarze retain”. Jeśli również zmienne trwałe powinny zachować poprzednie wartości po niekontrolowanej awarii sterownika, należy zadeklarować je dodatkowo jako VAR RETAIN. Przykładem zastosowania „**trwałych zmiennych retain**” może być licznik godzin roboczych, który po awarii zasilania powinien zliczać dalej.

Uwaga:

- Jeśli zmienna lokalna jest zadeklarowana w **programie** jako RETAIN, wtedy dokładnie ta zmienna zapisywana jest w obszarze retain (jak zmienna globalna retain).
- Jeśli zmienna lokalna zadeklarowana jest w **bloku funkcji** jako RETAIN, kompletna instancja tego bloku funkcji zapisana jest w obszarze retain (wszystkie dane modułu), przy czym jako zmienna retain traktowana będzie tylko zadeklarowana zmienna retain.
- Jeśli zmienna lokalna w **funkcji** zadeklarowana jest jako RETAIN, nie wywołuje to żadnego skutku.
Zmienna nie zostanie zapisana w obszarze retain! Jeśli zmienna lokalna w funkcji zostanie zadeklarowana jako PERSISTENT, nie wywoła to również żadnego skutku.
- Zmiennym remanentnym nie należy przypisywać **żadnych adresów znaczników** (%M), ponieważ prowadzi to do konfliktu zakresów pamięci!

x = wartość pozostaje zachowana – = wartość jest inicjowana ponownie

po poleceniu online	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
Reset	-	x	-	x
Reset (zimny)	-	-	-	-
Reset (źródło)	-	-	-	-
Wczytaj (= Download)	-	-	x	x
Zmiana online	x	x	x	x

Stałe, Typed Literals

Stałe oznaczane są słowem kluczowym

CONSTANT. Można je deklarować lokalnie lub globalnie.

Składnia:

VAR CONSTANT wzgl. **VAR_GLOBAL CONSTANT**

```
<identyfikator>:<Typ>:= <inicjalizacja>;
```

END_VAR**Przykład:**

```
VAR CONSTANT
  c_iCon1:INT:=12; (* 1. stała *)
END_VAR
```

Listę możliwych stałych oraz informację o zastosowaniu typowanych stałych (Typed Literals) można znaleźć w Załącznik D.

Zmienne zewnętrzne

Zmienne globalne, które należy zaimportować do modułu, oznaczane są słowem kluczowym EXTERNAL. Pojawiają się one online również w oknie kontrolnym edytora deklaracji.

Jeśli deklaracja VAR_EXTERNAL nie jest zgodna z deklaracją globalną, podczas kompilacji pojawia się następujący komunikat o błędzie: „Deklaracja '<nazwa>' jest niezgodna z deklaracją globalną!”.

Jeśli brak jest zmiennej globalnej, wyświetlany jest następujący komunikat: „Nieznana zmienna globalna '<zmienna>'!”.

Przykład:

```
VAR_EXTERNAL
  iVarExt1:INT:=12; (* 1-sza zmienna zewn. *)
END_VAR
```

Słowa kluczowe

Słowa kluczowe należy we wszystkich edytorach pisać wielkimi literami. Słów kluczowych nie można używać jako nazw zmiennych.

Deklaracja zmiennych

Deklaracja zmiennej ma następującą składnię:

```
<identyfikator> {AT <adres>} :<typ> {:= <inicjalizacja>};
```

Elementy w nawiasach klamrowych {} są opcjonalne.

W przypadku **Identyfikatora**, a więc nazwy zmiennej, należy zwrócić uwagę, aby nie zawierała ona spacji i znaków specjalnych, nie może być również deklarowana podwójnie ani identyczna ze słowami kluczowymi. Nie jest uwzględniana pisownia wielkimi/małymi literami, nazwa VAR1, Var1 i var1 oznacza tę samą zmienną. Dolne myślniki mają znaczenie dla identyfikatorów, np. „A_BCD” oraz „AB_CD” to różne identyfikatory. Niedozwolone jest stosowanie kilku podkreśleń kolejno na początku lub wewnątrz identyfikatora. Długość identyfikatora, jak również obszar znaczący są nieograniczone.

Patrz Załącznik L, Zalecenia odnośnie nadawania nazw w przypadku identyfikatorów.

Wszystkie deklaracje zmiennych oraz elementy typów danych mogą zawierać **inicjalizacje** (przyporządkowanie wartości początkowej). Są one wprowadzane przez operator przyporządkowujący „:=”. W przypadku zmiennych typu elementarnego inicjalizacje to wartości stałe. Inicjalizacja domyślna dla wszystkich deklaracji to 0.

Przykład:

```
iVar1:INT:=12; (* zmienna całkowita o wartości inicj. 12 *)
```

Jeśli zmienna ma być powiązana bezpośrednio z konkretnym adresem, wtedy zmienną należy zadeklarować przy użyciu słowa kluczowego **AT**.

W celu szybkiego wprowadzenia deklaracji należy użyć **trybu skróconego wprowadzania**.

W blokach funkcji zmienne można wskazywać również z niepełnymi danymi adresu. Aby wykorzystać takie zmienne w lokalnej instancji, należy w tym celu dokonać wpisu w konfiguracji zmiennej (zasoby).

Należy pamiętać o możliwości **automatycznego deklarowania (patrz niżej) oraz użycia dyrektywy pragma** do modyfikacji właściwości zmiennych podczas procesu kompilacji (patrz rozdział 8.2.3).

Deklaracja AT

Jeśli zmienna ma być powiązana bezpośrednio z konkretnym adresem, wtedy zmienną należy zadeklarować przy użyciu słowa kluczowego **AT**. Zaletą takiego sposobu postępowania jest to, że adresowi można nadać charakterystyczną nazwę i że ew. zmianę sygnału wejściowego lub wyjściowego (a mianowicie w deklaracji) wystarczy wprowadzić w jednym miejscu.

Należy pamiętać, że zmienne ustawione dla wejścia nie są dostępne do zapisu.

Przykłady:

```
bSchalterHeizung7 AT %QX0.0: BOOL;
wLichtschrankenimpuls AT %IW2: WORD;
xAblage AT %MX2.2: BOOL;
```

Wskazówka: Jeśli zmienne boolean osadzone są w adresie bajtu, Word lub DWORD, do 1 bajtu należy przypisać wartość TRUE lub FALSE, nie tylko do pierwszego bitu po przesunięciu (offset)!

„Wstaw” „Słowa kluczowe deklaracji”

Za pomocą tego polecenia można otworzyć listę wszystkich słów kluczowych, jakich można użyć w części deklaracyjnej modułu. Po wybraniu słowa kluczowego i potwierdzeniu wyboru, słowo zostanie wstawione w aktualnej pozycji kursora.

Listę można wyświetlić również, wywołując funkcję pomocy przy wpisywaniu (<F2>) i wybierając kategorię **Deklaracje**.

„Wstaw” „Typy”

Za pomocą tego polecenia można uzyskać wybór możliwych typów dla deklaracji zmiennej. Listę można również wyświetlić, wywołując funkcję pomocy przy wpisywaniu (<F2>).

Typy są podzielone na kategorie:

- typy standardowe – BOOL, BYTE itp.,
- typy zdefiniowane – struktury, typy wyliczania itp.,
- standardowe bloki funkcji dla deklaracji instancji,
- zdefiniowane bloki funkcji dla deklaracji instancji.

Program CoDeSys obsługuje wszystkie typy standardowe normy IEC 61131-3.

Przykłady zastosowania poszczególnych typów zawiera Załącznik D.

Syntaxcoloring

We wszystkich edytorach podczas implementacji i deklaracji zmiennych stosowane są ułatwienia w postaci wyróżnień. Dzięki kolorom tekstu łatwiej jest uniknąć błędów wzgl. szybciej można je wykryć.

Dzięki temu można natychmiast zauważyć niezamknięty komentarz, przechodzący w ten sposób w instrukcję, jak też uniknąć błędnego wpisywania słów kluczowych itp.

Stosowane są następujące wyróżnienia kolorystyczne:

Niebieski	Słowa kluczowe
Zielony	Komentarze w edytorach tekstowych
Różowy	Stałe specjalne (np. TRUE/FALSE, T#3s, %IX0.0)
Czerwony	Błędny wpis (np. nieprawidłowa stała czasu, pisownia słowa kluczowego małą literą, ...)
Czarny	Zmienne, stałe, operatory przyporządkowujące, ...

Tryb skróconego wprowadzania

Edytor deklaracji programu CoDeSys oferuje możliwość stosowania trybu skróconego wprowadzania. Jest on włączany, gdy wiersz zostanie zakończony kombinacją klawiszy <Ctrl> + <Enter>.

Stosowane są następujące formy skróconego wprowadzania:

- Wszystkie identyfikatory, z wyjątkiem ostatniego identyfikatora w wierszu, przekształcane są na identyfikatory zmiennych deklaracji.
- Typ deklaracji ustalany jest przez ostatni identyfikator w wierszu, przy czym obowiązuje zasada:
 - B lub BOOL daje w wyniku BOOL
 - I lub INT daje w wyniku INT
 - R lub REAL daje w wyniku REAL
 - S lub STRING daje w wyniku STRING
- Jeśli ustalenie typu nie jest możliwe w oparciu o te reguły, przyjmowany jest typ BOOL, a ostatni identyfikator nie jest stosowany jako typ (przykład 1).
- Każda stała, w zależności od typu deklaracji, przekształcana jest w inicjalizację lub długość sekwencji (przykłady 2 i 3).
- Adres rozszerzany jest (jak w przypadku %MD12) o atrybut AT ... (przykład 4).
- Tekst po średniku (;) przekształcany jest w komentarz (przykład 4).
- Wszystkie inne znaki w wierszu są ignorowane (jak np. znak wykrzyknika w ostatnim przykładzie).

Przykłady:

Forma skrócona	Deklaracja
A	A: BOOL;
A B I 2	A, B: INT := 2;
ST S 2; sekwencja	ST: STRING(2); (* sekwencja *)
X %MD12 R 5; liczba rzeczywista	X AT %MD12: REAL := 5.0; (* liczba rzeczywista *)
B !	B: BOOL;

Automatyczne deklarowanie

Jeśli w oknie dialogowym „Projekt” „Opcje” w kategorii **Edytor** wybrana została opcja **Zadeklaruj automatycznie**, wtedy we wszystkich edytorach po wprowadzeniu niezadeklarowanej jeszcze zmiennej zostanie wyświetlone okno dialogowe, za pomocą którego można zadeklarować tę zmienną.

Okno dialogowe deklaracji zmiennych

Za pomocą pola listy rozwijanej **Klasa** należy wybrać, czy chodzi o zmienną lokalną (**VAR**), wejściową (**VAR_INPUT**), wyjściową (**VAR_OUTPUT**), wejściowo-wyjściową (**VAR_IN_OUT**) czy też o zmienną globalną (**VAR_GLOBAL**).

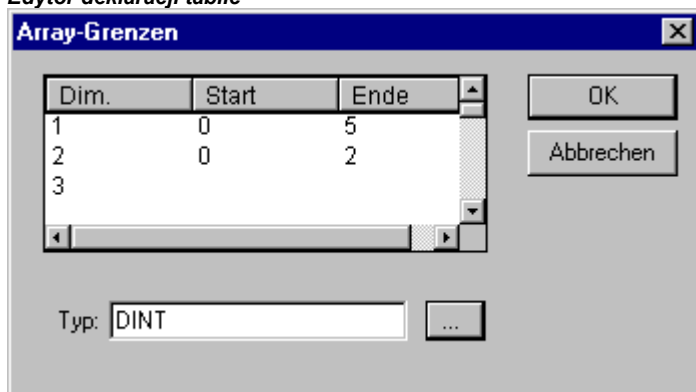
Za pośrednictwem opcji **CONSTANT**, **RETAIN**, **PERSISTENT** można zdefiniować, czy jest to zmienna stała czy też remanentna.

W polu **Nazwa** umieszczana jest domyślnie nazwa zmiennej, wprowadzonej w edytorze, zaś w polu **Typ** wartość BOOL. Za pomocą przycisku  można wyświetlić okno dialogowe asystenta deklaracji, umożliwiające wybór spośród możliwych typów danych.

Deklaracja **tablic** (pól):


Jeśli jako typ zmiennej zostanie wybrany ARRAY (pole), zostanie otwarte okno dialogowe wprowadzania granic tablicy.

Edytor deklaracji tablic




Dim.	Start	Ende
1	0	5
2	0	2
3		

Typ: DINT

W przypadku każdego z trzech możliwych wymiarów (wym.) można w polach Start i Koniec wprowadzić granice tablicy (zakres wartości DINT!), klikając po prostu przyciskiem myszy odpowiednie pole, w wyniku czego otworzy się ramka edycji. W polu Typ podawany jest typ danych tablicy. Za pomocą przycisku  można otworzyć odnośne okno dialogowe pomocy przy wpisywaniu.

Podczas opuszczania okna dialogowego granic tablicy za pomocą przycisku OK, na podstawie danych wypełniane jest w formacie IEC pole „Typ” w oknie dialogowym deklaracji zmiennej. Przykład: ARRAY [1..5, 1..3] OF INT

W polu Wartość początkowa można wpisać wartość początkową deklarowanej zmiennej. Jeśli stanowi ona tablicę lub inną prawidłową strukturę, za pomocą przycisku  można otworzyć specjalne okno dialogowe inicjalizacji wzgl. okno pomocy przy wpisywaniu w przypadku innych typów.

W oknie dialogowym inicjalizacji w odniesieniu do tablicy wyświetlana jest lista elementów tablicy i za pomocą kliknięcia przyciskiem myszy miejsca za znakiem „:=” można otworzyć pole edycji, umożliwiające wpisanie wartości początkowej elementu.

W oknie dialogowym inicjalizacji w odniesieniu do Struktury pojedyncze składniki wyświetlane są w postaci drzewa struktury. W nawiasach za nazwą zmiennej znajduje się typ oraz domyślna wartość początkowa składnika, a za nią następuje „:=”. Po kliknięciu pola za „:=” otwiera się pole edycji, w którym można wprowadzić żadaną wartość początkową. Jeśli składnikiem jest tablica, klikając znak plus przed nazwą tablicy w oknie dialogowym inicjalizacji można rozwijać poszczególne pola tablicy i edytować wartości początkowe.

Po opuszczeniu okna dialogowego inicjalizacji za pomocą przycisku **OK** w polu Wartość początkowa okna dialogowego deklaracji wyświetlana jest inicjalizacja tablicy wzgl. struktury w formacie IEC.

Przykład: x:=5,feld:=2,3,struct2:=(a:=2,b:=3)

W polu **Adres** można powiązać deklarowaną zmienną z adresem IEC (deklaracja AT).

W razie potrzeby można wprowadzić **Komentarz**. Za pomocą kombinacji klawiszy <Ctrl> + <Enter> w komentarzu można umieścić podziały wierszy.

Naciśnięcie przycisku **OK** powoduje zamknięcie okna dialogowego deklaracji i zmienna wpisywana jest, zgodnie ze składnią IEC, do odpowiedniego edytora deklaracji.

Wskazówka: Okno dialogowe deklaracji zmiennej można otworzyć również za pośrednictwem polecenia „Edycja” „Zmienne” „Deklaracja” (patrz Ogólne funkcje edycji). Jeśli kursor znajduje się na zmiennej, w trybie offline za pomocą kombinacji klawiszy <Shift> <F2> można otworzyć okno autodeklaracji, zawierające aktualne ustawienia zmiennych.

Numery wierszy w edytorze deklaracji

W trybie offline pojedynczym kliknięciem wybranego numeru wiersza można zaznaczyć cały wiersz tekstowy.

W trybie online pojedyncze kliknięcie wybranego wiersza umożliwia rozwijanie lub zwijanie zmiennej, jeśli jest to zmienna posiadająca rozbudowaną strukturę.

Deklaracje jako tabela

Jeśli otwarty jest edytor deklaracji i włączona została opcja **Deklaracje jako tabele** w oknie dialogowym opcji, w kategorii **Edytor** lub za pośrednictwem menu podręcznego, edytor deklaracji wyświetlany jest w postaci tabelarycznej. Podobnie jak w segregatorze można wtedy wybierać poszczególne karty rodzajów zmiennych i edytować zmienne.

W przypadku każdej zmiennej dostępne są w celu wypełnienia następujące pola:

- Nazwa: Tu należy wpisać identyfikator zmiennej.
- Adres: W razie potrzeby należy wpisać adres zmiennej (deklaracja AT)
- Typ: Tu należy wpisać typ zmiennej. (W przypadku tworzenia instancji bloku funkcji wpisać blok funkcji.)
- Inicjalizacja: Tu należy wpisać ew. wartość inicjalizacji zmiennej. (odpowiednio do operatora przyporządkowującego „ := ”.)
- Komentarz: W razie potrzeby tu można wprowadzić Komentarz.

Można bez problemu przełączać pomiędzy obydwooma typami widoku edytora deklaracji. W trybie online widok jest różny.

Wstawianie nowej deklaracji:

W celu wprowadzenia nowej deklaracji należy wybrać polecenie „**Wstaw**” „**Nowa deklaracja**” (patrz niżej). Pod wierszem, w którym aktualnie znajduje się kursor myszy, zostanie wtedy wstawiony nowy wiersz, który można edytować w odpowiedni sposób.

Sortowanie deklaracji:

Aby posortować wpisy w tabeli, kursor myszy należy umieścić na pasku numerów wierszy przy lewej krawędzi i w menu podręcznym wybrać jedno z następujących poleceń:

- **Sortuj wg nazwy:** Wszystkie wiersze będą sortowane alfabetycznie według nazw znajdujących się w kolumnie Nazwa.
- **Sortuj wg adresów:** Wszystkie wiersze będą sortowane alfabetycznie według adresów znajdujących się w kolumnie Adres.
- **Sortuj wg typu:** Wszystkie wiersze będą sortowane alfabetycznie według typów znajdujących się w kolumnie Typ.
- **Pierwotna kolejność:** Wiersze będą ponownie wyświetlane w tej kolejności, w jakiej zostały pierwotnie wprowadzone.

Edytor deklaracji jako tabela

	VAR	VAR_INPUT	VAR_OUTPUT	VAR_IN_OUT	CONST
	Name	Adresse	Typ	Initial	Kommentar
0001	AMPEL1		AMPEL		
0002	AMPEL2		AMPEL		
0003	VERZ		WARTEN		
0004	ZAEHLER		INT		

„Wstaw” „Nowa deklaracja”

Za pomocą tego polecenia można wprowadzić nową zmienną do tabeli deklaracji w edytorze deklaracji. Jeśli kursor znajduje się aktualnie w komórce tabeli, nowa zmienna zostanie wprowadzona przed tym wierszem, w przeciwnym razie zostanie dołączona na końcu tabeli. Ponadto można na końcu tabeli dołączyć nową deklarację, naciskając klawisz tabulatora lub przycisk strzałki w prawo w ostatniej komórce tabeli.

W efekcie zostanie wstawiona zmienna, która w polu **Nazwa** będzie mieć domyślną nazwę „Nazwa”, a w polu **Typ** – „Bool”. Wartości należy zmienić na żądane wartości. Nazwa oraz typ wystarczają dla kompletnej deklaracji zmiennej.

8.2.2 Edytory deklaracji w trybie online

W trybie online edytor deklaracji ma postać okna monitora. W każdym wierszu znajduje się zmienna, po niej następuje znak równości (=) oraz wartość zmiennej. Jeśli zmienna jest do tego momentu niezdefiniowana, wyświetlane są trzy znaki zapytania (???). W przypadku bloków funkcji wartości wyświetlane są tylko w odniesieniu do otwartych instancji (polecenie „Projekt” „Otwórz instancję”).

Przed każdą zmienną wieloelementową znajduje się znak plus. Naciskając klawisz <Enter> lub klikając dwukrotnie taką zmienną, można ją otworzyć; na przykładzie rozwinięta jest struktura Sygnalizator1:

```

+---AMPEL1
  |---.STATUS = 3
  |---.GRLEN = FALSE
  |---.GEL3 = FALSE
  |---.ROT = TRUE
  |---.AUS = FALSE

```

W przypadku rozwiniętych zmiennych wszystkie ich składniki wyświetlane są w kolejności. Przed zmiennymi widnieje znak minus. Za pomocą ponownego podwójnego kliknięcia wzgl. naciśnięcia klawisza <Enter> można zwinąć zmienną i ponownie ukazać się znak plus.

Naciśnięcie klawisza <Enter> lub dwukrotne kliknięcie zmiennej jednoelementowej powoduje otwarcie okna dialogowego zapisu zmiennej (patrz „Ogólne funkcje online”). Można w nim zmienić wartość aktualnej zmiennej. W przypadku zmiennych boolean okno dialogowe nie jest wyświetlane, lecz następuje przełączenie wartości.

Nowa wartość wyświetlana jest za zmienną w kolorze turkusowym, w nawiasach ostrokątnych i pozostaje niezmieniona.

```
bvar = TRUE < := FALSE >
ivar = 509 < := 65 >
```

Jeśli zostanie użyte polecenie „Online” „Zapisz wartości”, wszystkim zmiennym zostaną nadane wybrane wartości i będą one ponownie wyświetlane w kolorze czarnym.

```
bvar = TRUE
ivar = 65
```

Jeśli zostanie użyte polecenie „Online” „Wymuszanie wartości”, wszystkim zmiennym zostaną nadane wybrane wartości, dopóki nie zostanie użyte polecenie „Anuluj wymuszanie”. W takim wypadku wartość wymuszona zmieni kolor na czerwony.

```
bvar = FALSE
ivar = 44
```

8.2.3 Dyrektywy pragma w edytorze deklaracji

Dyrektywa pragma służy do sterowania procesem kompilacji. Znajduje się ona wraz z dodatkowym tekstem w wierszu programu lub w odrębnym wierszu edytora deklaracji.

Dyrektywa pragma umieszczana jest w nawiasach klamrowych (bez uwzględniania pisowni wielką lub małą literą):

```
{ <tekst instrukcji> }
```

Jeśli kompilator nie jest w stanie zinterpretować tekstu instrukcji w prawidłowy sposób, cała dyrektywa pragma traktowana jest jak komentarz i pomijana. Wyświetlane jest jednak ostrzeżenie: „Zignorowane polecenie kompilatora «<tekst instrukcji>!»”.

W zależności od typu i zawartości dyrektywy pragma odnosi się ona do wiersza, w którym się znajduje, wzgl. do wszystkich następujących wierszy, aż zostanie anulowana ponownie przez odpowiednią dyrektywę pragma, lub aż ta sama dyrektywa zostanie wykonana w oparciu o inne parametry lub gdy zostanie osiągnięty koniec pliku. Jako plik rozumie się przy tym: część deklaracyjna, część implementacyjna, lista zmiennych globalnych, deklaracja typu.

Nawias otwierający może znaleźć się przy tym bezpośrednio za nazwą zmiennej. Nawias otwierający i zamykający musi znajdować się w tym samym wierszu.

Program CoDeSys umożliwia aktualnie stosowanie następujących dyrektyw pragma:

- Pragma {flag} inicjalizacji, monitorowania, tworzenia symboli
- Pragma {bitaccess...} dostępu bitowego
- Pragma {parameter...}, {template...}, {instance...} do tworzenia wpisów w Menedżerze parametrów
- Pragma {library ...} – umożliwia wyświetlanie/ukrywanie części deklaracyjnych biblioteki w Menedżerze bibliotek

Dyrektywy pragma służące do inicjalizacji, monitorowania, tworzenia symboli, dostępu bitowego

Pragma {flag} inicjalizacji, monitorowania, tworzenia symboli:

Składnia: {flag [<flags>] [off|on]}

Za pomocą tej dyrektywy pragma można wpływać na właściwości deklaracji zmiennych.

<flags> może stanowić kombinację następujących znaczników:

noinit:	Zmienna nie będzie inicjowana.
nowatch:	Zmienna nie będzie monitorowana.
noread:	Zmienna zostanie wyeksportowana do pliku symboli bez prawa odczytu.
nowrite:	Zmienna zostanie wyeksportowana do pliku symboli bez prawa zapisu.
noread, nowrite:	Zmienna nie zostanie wyeksportowana do pliku symboli.

Za pomocą modyfikacji „on” dyrektywa pragma wywiera wpływ na wszystkie kolejne deklaracje zmiennych, aż zostanie anulowana przez pragnę {flag off}, wzgl. do czasu zastąpienia przez inną pragnę {flag <flags> on}.

Bez modyfikacji „on” lub „off” pragma oddziałuje jedynie na aktualną deklarację zmiennej (jest to ta deklaracja, która kończy się następnym średnikiem).

Przykłady zastosowania pragmy {flag}:

Inicjalizacja i monitorowanie zmiennych:

Zmienna a nie będzie inicjowana i nie będzie monitorowana. Zmienna b nie będzie inicjowana:

```
VAR
  a : INT {flag noinit, nowatch};
  b : INT {flag noinit };
END_VAR
```

```

VAR
  {flag noinit, nowatch on}
  a : INT;
  {flag noinit on}
  b : INT;
  {flag off}
END_VAR

```

Obie zmienne nie będą inicjowane.

```

  {flag noinit on}

VAR
  a : INT;
  b : INT;
END_VAR

  {flag off}

VAR
  {flag noinit on}
  a : INT;
  b : INT;
  {flag off}
END_VAR

```

Eksport zmiennych do pliku symboli:

Znaczniki „noread” oraz „nowrite” służą do tego, aby w module, który ma prawo odczytu i/lub zapisu, nadać pojedynczym zmiennym ograniczone prawo dostępu. Wartość domyślna zmiennej to ustawienie, jakie ma moduł, w którym jest zadeklarowana zmienna. Jeśli zmienna nie ma ani prawa odczytu, ani zapisu, nie zostanie wyeksportowana do pliku symboli.

Przykłady:

Moduł otrzyma prawo odczytu i zapisu, a wtedy za pomocą następującej dyrektywy pragma możliwe będzie wyeksportowanie zmiennej „a” tylko z prawem zapisu, a zmiennej „b” nie będzie można wyeksportować wcale:

```

VAR
  a : INT {flag noread};
  b : INT {flag noread, nowrite};
END_VAR

VAR
  { flag noread on}
  a : INT;
  { flag noread, nowrite on}
  b : INT;
  {flag off}
END_VAR

```

Obie zmienne: a i b nie zostaną wyeksportowane do pliku symboli:

```

  { flag noread, nowrite on }
VAR
  a : INT;
  b : INT;
END_VAR
  {flag off}

VAR
  { flag noread, nowrite on }
  a : INT;
  b : INT;
  {flag off}
END_VAR

```

Pragma sumuje wszystkie podporządkowane deklaracje zmiennych.

Przykład: (wszystkie używane moduły eksportowane są z prawem odczytu i zapisu)

```
a : afb;

...

FUNCTION_BLOCK afb
VAR
  b : bfb {flag nowrite};
  c : INT;
END_VAR

...

FUNCTION_BLOCK bfb
VAR
  d : INT {flag noread};
  e : INT {flag nowrite};
END_VAR
```

„a.b.d”: nie zostanie wyeksportowane.

„a.b.e”: zostanie wyeksportowane tylko z prawem odczytu.

„a.c”: zostanie wyeksportowane z prawem odczytu i zapisu.

Pragma {bitaccess...} dostępu bitowego:

Za pomocą tej pragmy można zdefiniować prawidłowe symboliczne dostępy bitowe do **Struktur**, które opierają się na stałych globalnych. Symbole te oferowane są wtedy zarówno przez pomoc przy wpisywaniu oraz funkcję „Intellisense”, jak też używane w celu wyświetlania dostępu bitowych podczas monitorowania w edytorze deklaracji. Wyświetlane są tam również używane stałe globalne.

Należy pamiętać: Musi być włączona opcja projektu „Zastąp stałe” (kategoria „Opcje kompilacji”, patrz rozdział 6.2)!

Pragma musi być umieszczona w definicji struktury w odrębnym wierszu. Wiersz ten nie jest zakończony średnikiem.

Składnia: {bitaccess <zmienna globalna> <numer bitu> '<komentarz>'}

<Stała globalna>: Nazwa stałej globalnej, która musi być zdefiniowana na liście zmiennych globalnych.

<numer bitu> : Wartość stałej globalnej, zdefiniowana na liście zmiennych globalnych.

Patrz przykład w Załącznik B, Argumenty w programie CoDeSys, Adresowanie bitów w zmiennych.

Pragmy do tworzenia wpisów w Menedżerze parametrów

Za pomocą pragm w obrębie deklaracji zmiennych można automatycznie tworzyć wpisy na listach parametrów, które są administrowane przez Menedżera parametrów. Menedżer parametrów jest dostępny w systemie programowania w zależności od systemu docelowego, tzn. musi być uaktywniony w ustawieniach systemu docelowego (funkcje sieciowe).

Ogólne informacje na temat składni:

- Pragma umieszczana jest w nawiasach klamrowych. Pisownia wielką/malą literą nie ma znaczenia. Jeśli zostanie dołączona do „normalnych” deklaracji zmiennych, musi znaleźć się przed średnikiem zamykającym deklarację zmiennych, na jakie ma oddziaływać.
- Pragmy używane w oknach VAR_CONFIG znajdują się każda w oddzielnym wierszu i nie są zamykane średnikiem!
- <name>: Nazwa listy parametrów w Menedżerze parametrów. Jeśli lista zmiennych jeszcze nie istnieje, zostanie utworzona automatycznie.
- <key>: Nazwa atrybutu – tzn. tytuł kolumny na liście parametrów; np. „Nazwa”, „Wartość”, „Poziom dostępu” itp. – oraz to, jakie klucze można podać, zależy od definicji typu listy parametrów w systemie docelowym. Wszystkie definicje kluczy (key) znajdują się w nawiasach kwadratowych,

oddzielone w pragmie od siebie za pomocą spacji. Należy przestrzegać składni wpisów na listach instancji dla elementów tablic, struktur, wzgl. bloków funkcji (patrz 3).

- <value>: Wartość, która zostanie wprowadzona na listę dla atrybutu zdefiniowanego za pomocą <key>. Należy przy tym pamiętać, że wartości, które zawierają znaki spacji, muszą być ujęte w podwójne znaki apostrofu. Przykład: ...accesslevel="read only"....

Wskazówka: Dyrektywy pragma skutkują już w przypadku zmiany aktywnego elementu (prekompilowanie), a więc w przypadku opuszczenia edytora deklaracji. Błędne wpisy pragm zgłaszane są dopiero podczas kompilacji projektu.

Możliwe jest generowanie następujących wpisów:

1. Wpisy na listach parametrów typu „Lista zmiennych”

(a) z części deklaracyjnej programów i list zmiennych globalnych:

Dla zmiennych w obrębie deklaracji PROGRAM lub VAR_GLOBAL można utworzyć wpis na liście parametrów typu „Zmienne”, jeśli zostanie ona zadeklarowana w następujący sposób: (Jeśli lista parametrów jeszcze nie istnieje, zostanie utworzona.)

Składnia: {parameter list=<name> [<key>=<value> <key>=<value> ...dalsze keys] }

Przykład: W programie została zadeklarowana zmienna bvar, która ma zostać wpisana na listę parametrów parlist1 typu listy zmiennych pod nazwą bvar1, z wartością 102, indeksem 16#1200 i subindeksem 16#2.

```
VAR
  bvar:INT{parameter list=parlist1 [name=bvar1 value=102 index=16#1200
  subindex=16#1 ] };
END_VAR
```

(b) za pomocą deklaracji w interfejsie VAR_CONFIG:

Dla zmiennych można utworzyć wpis na liście parametrów typu „Zmienne”, jeśli zostały one zadeklarowane w oknie VAR_CONFIG w następujący sposób: (Jeśli lista parametrów jeszcze nie istnieje, zostanie utworzona.)

Składnia: {parameter list=<name> path=<path> [<key>=<value> <key>=<value> ...dalsze keys] }

<path> ścieżka zmiennej, dla której ma zostać utworzony wpis, np. „PLC_PRG.act1.var_x”

Przykład: w przypadku zmiennej var_x zostanie utworzony wpis na liście parametrów varlist1, jako nazwa symboliczna zostanie wpisana nazwa xvar.

```
VAR_CONFIG
  {parameter list=varlist1 path=PLC_PRG.act1.var_x [ name=xvar ] }
END_VAR
```

2. Wpisy na listach parametrów typu „Szablon” z bloków funkcji oraz struktur

W przypadku deklaracji zmiennych w blokach funkcji i strukturach można tworzyć wpisy na listach parametrów typu „Szablon”. (Jeśli szablon jeszcze nie istnieje, zostanie utworzony.)

Składnia: {template list=<name> [<key>=<value> <key>=<value> ...dalsze keys] }

Przykład: Zmienna strvar, stanowiąca element struktury stru1, powinna zostać wpisana pod nazwą (member) „struvar1” i z poziomem dostępu Accesslevel=low w szablonie „vor1” w Menedżerze parametrów:

```
TYPE stru :
  STRUCT
    ivar:INT;
    strvar:STRING{template list=vor1 [member=struvar1 accesslevel=low] };
  END_STRUCT
END_TYPE
```

3. Wpisy na listach parametrów typu „Instancja” (dla zmiennych tablic, bloków funkcji lub struktur)

(a) z programów i list zmiennych globalnych:

W przypadku deklaracji zmiennych tablic, bloków funkcji lub struktur w obrębie programu lub listy zmiennych globalnych można utworzyć bezpośrednio listę instancji w Menedżerze parametrów.

Składnia: {instance list=<name> template=<template> baseindex=<index> basesubindex=<subindex> [<key>=<value dla pierwszego elementu > <key>=<value dla pierwszego elementu > ...dalsze keys dla pierwszego elementu] | [<key>=<value dla drugiego elementu > <key>=<value dla drugiego elementu > ...dalsze keys dla drugiego elementu] | [keys dla dalszych elementów]}

Dla tablic klucz „template” definiowany jest na podstawie dostępnego zawsze pośrednio szablonu „ARRAY”, w przypadku struktur i bloków funkcji odpowiedni szablon musi istnieć w Menedżerze parametrów i należy go wprowadzić w tym miejscu.

W przypadku każdego pojedynczego elementu tablicy wzgl. struktury lub bloku funkcji można zdefiniować wstępnie indywidualny wpis na liście parametrów: Można na przykład dla każdego elementu wskazać odrębną definicję [name=<elementname>]. Kolejne definicje kluczy (key) poszczególnych elementów (dla każdego elementu w obrębie tego samego nawiasu prostokątnego!) oddzielone są za pomocą spacji i odnoszą się automatycznie do elementów w rosnącej kolejności indeksu (member). Jeśli definicji kluczy nie jest tyle samo co tablic, wzgl. struktura lub blok funkcji zawiera elementy lub zmienne, wszystkie pozostałe elementy otrzymają takie same wartości jak element zdefiniowany indywidualnie jako ostatni (wyjątek stanowi key „name” w przypadku tablic, patrz niżej)! (Patrz niżej, przykład 1b).

Automatyczne mechanizmy wpisywania tablic na listach parametrów w odniesieniu do klucza „name”:

- Jeśli w pragmie nie zostanie wstępnie zdefiniowana nazwa dla elementu tablicy, ten element oraz wszystkie pozostałe na liście parametrów otrzymają automatycznie nazwę <nazwa modułu>_<nazwa zmiennej tablicy>_<odpowiedni indeks tablicy>. Przykład: Zmienna tablicy ARRVAR [1..8] of INT w module PLC_PRG ma zostać wpisana za pośrednictwem pragmy na listę parametrów; jeśli nie została podana definicja klucza „name”, poszczególne elementy tablicy na liście parametrów otrzymają automatycznie nazwy „PLC_PRG_arrvar_1” do „PLC_PRG_arrvar_8”.
- Jeśli dla pierwszego elementu tablicy w pragmie zostanie zdefiniowana wstępnie dowolna nazwa „<name>_<pierwszy indeks obszaru tablicy>”, dalsze elementy tablicy na liście parametrów otrzymają automatycznie nazwę „<name>_<odpowiedni indeks>”. Przykład: dla zmiennej tablicy ARRVAR [1..8] zdefiniowany został w pragmie pierwszy element „[name=xyz_1]” -> na liście parametrów pojawią się nazwy od xyz_1 do xyz_8.

Uwaga: W przypadku zmiennych tablic nie należy podawać wartości klucza „member”, jest on bowiem automatycznie tworzony dla każdego elementu tablicy na podstawie indeksu tablicy.

Przykłady:

Przykład 1a: Zmienna tablicy arr_1 zadeklarowana została w taki sposób, aby w Menedżerze parametrów została utworzona lista instancji „arrinst” (o ile jeszcze nie istnieje!), na której wpisane zostaną elementy tablicy, przy czym każdy element zostanie najpierw wpisany pod symboliczną nazwą xname_<indeks> (będzie w dalszym ciągu edytowalny na liście), natomiast subindeks będzie powiększany dla każdego wpisu o 1, poczynając od wartości 0 (basesubindex). Ustawienie Accesslevel=low zostanie zastosowane do wszystkich elementów.

```
arr_1: ARRAY [1..8] OF INT{instance list=arrinst template=ARRAY baseindex=16#0
basesubindex=16#0 [name=xname_1]};
```

Przykład 1b: Dla zmiennej tablicy arr_1 tylko dla elementów od 1 do 4 na liście parametrów zdefiniowano już różne nazwy, elementy od 5 do 8 otrzymają z tego względu nazwę elementu 4, z dołączonym podkreśleniem i odpowiednim indeksem, a więc xname_5 do xname_8. Należy pamiętać, że dalsze definicje kluczy dla określonego elementu muszą być wprowadzone w obrębie tego samego nawiasu kwadratowego, tak jak tu w przypadku pierwszego i czwartego elementu dla poziomu dostępu:

```
arr_1: ARRAY [1..8] OF INT{instance list=arrinst template=ARRAY baseindex=16#0
basesubindex=16#0[name=aname accesslevel=high] [name=bname] [name=cname]
[name=xname accesslevel=medium]};
```

Edytor Menedżera parametrów dla przykładu 1a i 1b, tablica wpisana na liście instancji:

Przykład 1a:

Name	Member	Value	Index	SubIn...	Accesslevel	Accessright	Min	Max
xname_1	[1]		16#0	16#0	low			
xname_2	[2]		16#0	16#1	low			
xname_3	[3]		16#0	16#2	low			
xname_4	[4]		16#0	16#3	low			
xname_5	[5]		16#0	16#4	low			
xname_6	[6]		16#0	16#5	low			
xname_7	[7]		16#0	16#6	low			
xname_8	[8]		16#0	16#7	low			

Synchrone Aktionen Vorlage Basisindex 16#0

Basisvariable PLC_PRG.arr_1 Basis-Subindex 16#0

Przykład 1b:

Name	Memb...	Value	Index	SubIn...	Accesslevel	Acces...	Min	Ma...
aname	[1]		16#0	16#0	high			
bname	[2]		16#0	16#1	medium			
cname	[3]		16#0	16#2	medium			
xname	[4]		16#0	16#3	medium			
xname_5	[5]		16#0	16#4	medium			
xname_6	[6]		16#0	16#5	medium			
xname_7	[7]		16#0	16#6	medium			
xname_8	[8]		16#0	16#7	medium			

Synchrone Aktionen Vorlage ARRAY Basisindex 16#0

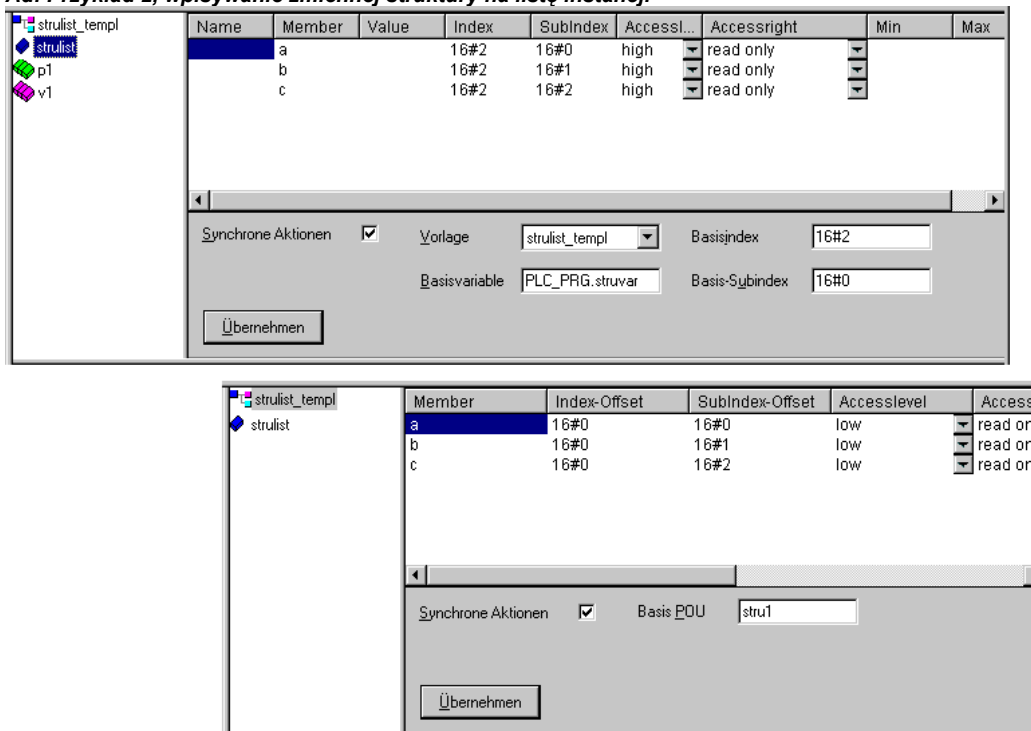
Basisvariable PLC_PRG.arr_1 Basis-Subindex 16#0

Przykład 2: Zmienna strukturalna struvar typu stru1 została zadeklarowana w taki sposób, aby w Menedżerze parametrów została utworzona lista instancji „strulist” (o ile jeszcze nie istnieje), bazująca na szablonie strulist_temp i w której jako wpisy pojawiają się zmienne a,b,c istniejącej już struktury stru1. Każda zmienna podczas wpisywania nie otrzyma jeszcze nazwy symbolicznej, poziom dostępu ustawiony jest jako wysoki (high), a do indeksu zdefiniowanego w szablonie dodawana jest zawsze wartość 2. Należy zwrócić uwagę na to, aby wskazany szablon instancji był dostępny w Menedżerze parametrów:

```
struvar:stru1{instance list=strulist template=strulist_temp baseindex=16#2
basesubindex=16#0 [accesslevel=high] };
```

Edytor deklaracji

Ad: Przykład 2, wpisywanie zmiennej struktury na listę instancji



(b) za pomocą deklaracji w interfejsie VAR_CONFIG:

W przypadku zmiennych mogących tworzyć instancje, wpisy na liście instancji w Menedżerze parametrów można definiować bezpośrednio za pomocą deklaracji w oknie VAR_CONFIG. Deklaracja ta jest niezależna od ewentualnych konfiguracji zmiennych! (Jeśli lista instancji jeszcze nie istnieje, zostanie utworzona.)

Należy zwrócić uwagę na to, aby wskazany szablon (<template>) był dostępny w Menedżerze parametrów.

Składnia: {instance list=<name> path=<path> template=<template> baseindex=<index> basesubindex=<subindex> [<key>=<value> <key>=<value> ...dalsze keys] }

<path>: Ścieżka instancji zmiennej; np. „PLC_PRG.fb1inst”, przy czym fb1inst jest instancją bloku funkcji fb1.

Przykład: Za pomocą następującego wpisu w oknie VAR_CONFIG (niezależnie od ewentualnych konfiguracji zmiennych!), na liście instancji „varinst1” tworzone są wpisy wszystkich zmiennych bloku funkcji fb1 w oparciu o szablon „fb1_templ” (który już musi być dostępny). W celu przesunięcia indeksu, które jest zdefiniowane w szablonie, do każdego wpisu dodawana jest wartość 2 (baseindex), zaś do przesunięcia subindeksu nie dodaje się niczego (basesubindex). Każdy wpis otrzymuje nazwę symboliczną „fb1var”, którą należy jeszcze wyedytować na liście.

```
VAR_CONFIG
{instance list=varinst1 path=PLC_PRG.fb1 template=fb1_templ baseindex=16#2
basesubindex=16#0 [ name=fb1var ]}
END_VAR
```

Pragma umożliwiająca wyświetlanie/ukrywanie części deklaracyjnych biblioteki w Menedżerze bibliotek

Za pomocą dyrektywy pragma {library public} oraz {library private} można w bibliotece utworzonej w programie CoDeSys zdefiniować, jakie wiersze/fragmenty wierszy części deklaracyjnej mają być wyświetlane względnie ukrywane w Menedżerze bibliotek podczas późniejszego zastosowania biblioteki w projekcie. Nie ma to wpływu na wyświetlanie części implementacyjnej.

W ten sposób można np. ukryć przed użytkownikiem komentarze lub też określone części deklaracji zmiennych w bibliotece. Dyrektywy pragma odnoszą się zawsze do pozostałej części tego samego wzgl. do kolejnych wierszy, aż zostaną one anulowane przez inną dyrektywę pragma.

Składnia: `{library public}` Poniższy tekst będzie wyświetlany w Menedżerze bibliotek.
`{library private}` Poniższy tekst nie będzie wyświetlany.

Przykład: Poniżej przedstawiono część deklaracyjną biblioteki utworzonej w programie CoDeSys. Komentarz „(* this is for all *)” powinien być widoczny po włączeniu biblioteki do Menedżera bibliotek, zaś komentarz „(* but this is not for all *)” nie. Zmienne local oraz in2 również nie powinny być widoczne:

```
{library public}(*this is for all*){library private}(*this is not for all*)
{library public}
FUNCTION afun : BOOL
VAR_INPUT
    in: BOOL;
END_VAR
{library private}
VAR
    local: BOOL;
END_VAR
{library public}
VAR_INPUT
    in2: BOOL;
{library private}
    in3: BOOL;
{library public}
END_VAR
```

Dyrektywy pragma do deklaracji nietrwałych typów danych

Zazwyczaj obowiązuje zasada, że nawet jeśli tylko jedna zmienna w bloku funkcji lub w strukturze została zadeklarowana jako trwała (persistent), wtedy w przypadku użycia instancji automatycznie wszystkie składniki w informacji „persistent” (persistent.dat) zostaną zapisane w systemie czasu przebiegu. Aby zaoszczędzić miejsce w pamięci, dyrektywę pragma można

Składnia: {nonpersistent}

zastosować w deklaracji bloku funkcji. Sprawia to, że tylko zmienne bloku funkcji/struktury, zadeklarowane jako trwałe, zostaną wpisane do informacji „persistent”.

Przykład: Jeśli instancja następującego bloku funkcji zostanie zadeklarowana jako trwała, w obszarze informacyjnym Persistent zostaną zapisane tylko zmienne local oraz fblevel3. Bez dyrektywy pragma {nonpersistent} zostałyby tam zapisane wszystkie zmienne bloku funkcji:

```
FUNCTION_BLOCK FB_Level_2
{nonpersistent}
VAR_INPUT
    bvar_in : BOOL;
END_VAR
VAR_OUTPUT
    bvar_out : BOOL;
END_VAR
VAR
    ivar2 : INT;
END_VAR
VAR PERSISTENT
    local : INT := 33;
    fblevel3 : FB_Level_3;
END_VAR
```

8.3 Edytory tekstowych języków programowania

8.3.1 Praca w edytorach tekstowych

Edytory używane do części implementacyjnej (edytor list instrukcji oraz edytor ST) programu CoDeSys dysponują typowymi funkcjami edytorów systemu Windows.

Implementacja w edytorach tekstowych obsługuje kolorowanie składni (Syntaxcoloring).

Jeśli włączony jest tryb zastępowania, na pasku stanu wyświetlany jest na czarno wpis „**ZAS**”. Naciskając klawisz <Insert> można przełączać pomiędzy trybem zastępowania i wprowadzania.

Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

„Wstaw” „Operator” w edytorach tekstowych

Za pomocą tego polecenia w oknie dialogowym wyświetlane są wszystkie operatory, jakie są dostępne w aktualnym języku.

Jeśli wybrany zostanie jeden z operatorów i lista zostanie zamknięta za pomocą przycisku **OK**, zaznaczony operator zostanie wstawiony w miejscu aktualnej pozycji kursora. (Obsługa w taki sam sposób jak w przypadku pomocy przy wpisywaniu).

„Wstaw” „Argument” w edytorach tekstowych

Za pomocą tego polecenia w oknie dialogowym wyświetlane są wszystkie zmienne, jakie są dostępne w postaci argumentów. Można wybrać, czy wyświetlana będzie lista zmiennych globalnych, lokalnych czy też zmiennych systemowych.

Jeśli wybrany zostanie jeden z argumentów i okno dialogowe zostanie zamknięte za pomocą przycisku **OK**, zaznaczony argument zostanie wstawiony w miejscu aktualnej pozycji kursora. (Obsługa w taki sam sposób jak w przypadku pomocy przy wpisywaniu).

„Wstaw” „Funkcja” w edytorach tekstowych

Za pomocą tego polecenia można wyświetlić w oknie dialogowym wszystkie funkcje. Można wybrać, czy wyświetlana będzie lista funkcji zdefiniowanych przez użytkownika czy też standardowych.

Jeśli wybrana zostanie jedna z funkcji okno dialogowe zostanie zamknięte za pomocą przycisku **OK**, zaznaczona funkcja zostanie wstawiona w miejscu aktualnej pozycji kursora. (Obsługa w taki sam sposób jak w przypadku pomocy przy wpisywaniu).

Jeśli w oknie dialogowym zostanie wybrana opcja **Z argumentami**, wtedy niezbędne zmienne wejściowe zostaną wprowadzone wraz z funkcją.

„Wstaw” „Blok funkcji” w edytorach tekstowych

Za pomocą tego polecenia można wyświetlić w oknie dialogowym wszystkie bloki funkcji. Można wybrać, czy wyświetlana będzie lista bloków funkcji zdefiniowanych przez użytkownika czy też standardowych.

Jeśli wybrany zostanie jeden z bloków funkcji i okno dialogowe zostanie zamknięte za pomocą przycisku **OK**, zaznaczony blok funkcji zostanie wstawiony w miejscu aktualnej pozycji kursora. (Obsługa w taki sam sposób jak w przypadku pomocy przy wpisywaniu).

Jeśli w oknie dialogowym zostanie wybrana opcja **Z argumentami**, wtedy niezbędne zmienne wejściowe zostaną wprowadzone wraz z blokiem funkcji. Nie muszą one jednak być koniecznie przyporządkowane.

Wywołanie modułu z parametrami wyjściowymi w edytorach tekstowych

Parametr wyjściowy wywołanego modułu w językach tekstowych IL i ST można przypisać bezpośrednio w wywołaniu.

Przykład: parametrowi wyjściowemu out1 modułu afbinst przypisano zmienną a.

```
IL: CAL afbinst(in1:=1, out1=>a)
```

```
ST: afbinst(in1:=1, out1=>a);
```

Jeśli moduł wstawiany jest w oknie implementacji modułu ST lub IL przy użyciu funkcji pomocy przy wprowadzaniu (<F2>) z opcją **Z argumentami**, wtedy jest on automatycznie wyświetlany w tej składni wraz z parametrami. Parametry nie muszą być jednak koniecznie przyporządkowane.

Edytory tekstowe w trybie online

Funkcje online w edytorach to wstawianie punktu wstrzymania oraz przetwarzanie pojedynczych kroków (krokowe). Wraz funkcją monitorowania użytkownik ma więc do dyspozycji funkcję debugowania w debugerze zaawansowanych języków w systemie Windows.

W trybie online okno edytora tekstowego podzielone jest w pionie na dwie części. Po lewej stronie okna znajduje się normalny tekst programu, po prawej natomiast wyświetlane są zmienne, których wartości zmieniają się w danym wierszu. Szerokość podzielonego okna można zmieniać przeciągając pionowy pasek za pomocą myszy lub też w oknie dialogowym „Ustawienia monitoringu”.

Wyświetlanie odbywa się w taki sam sposób jak w części deklaracyjnej. Tzn. gdy sterownik działa, wyświetlane są aktualne wartości danych zmiennych.

Podczas monitorowania wyrażeń oraz zmiennych adresowanych bitowo należy mieć na uwadze następujące kwestie: W przypadku wyrażeń wyświetlana jest zawsze wartość całego wyrażenia. Przykład: a AND b wyświetlane jest na niebiesko lub z „:=TRUE” (jeśli a i b mają wartość TRUE). W przypadku zmiennych adresowanych bitowo monitorowana jest zawsze wywoływana wartość bitu (np. a.3 niebieski wzgl. z :=TRUE, jeśli a ma wartość 4). Przytrzymując kursor myszy przez krótki czas nad zmienną, można wyświetlić typ, adres oraz komentarz do zmiennej w oknie etykiety.

„Dodatki” „Ustawienia monitoringu”

Za pomocą tego polecenia można skonfigurować okno monitoringu. W edytorach tekstowych podczas monitorowania okno dzielone jest na lewą połowę, w której znajduje się program, oraz prawą, w której monitorowane są wszystkie zmienne, znajdujące się w odpowiednich wierszach programu.

Ustawić można **Szerokość** obszaru monitorowania w oknie tekstowym, oraz **Odstęp**, jaki mają mieć dwie monitorowane zmienne w wierszu. Wskazanie odstępu 1 odpowiada przy tym wysokości wiersza w odniesieniu do wybranego rodzaju czcionki. Należy pamiętać, że szerokość podzielonego okna można także zmieniać poprzez przeciągnięcie

paska rozdzielającego.

Okno dialogowe ustawień monitoringu



Pozycje punktów wstrzymania w edytorze tekstowym

Ponieważ wewnątrz w programie CoDeSys wiersze IL grupowane są w jeden wiersz kodu C, nie w każdym wierszu można umieścić punkty wstrzymania. Pozycje punktów wstrzymania to wszystkie miejsca w programie, w których mogą zmieniać się wartości zmiennych lub też w których następuje rozgałęzienie przebiegu programu (wyjątek: wywołania funkcji; w takim przypadku w razie potrzeby punkt wstrzymania należy umieścić w funkcji). Umieszczanie punktu wstrzymania w pozycjach znajdujących się pomiędzy tymi punktami nie ma uzasadnienia, ponieważ od poprzedniej pozycji punktu wstrzymania dane nie mogły się zmienić.

Tym samym możliwe są w IL następujące pozycje punktów wstrzymania:

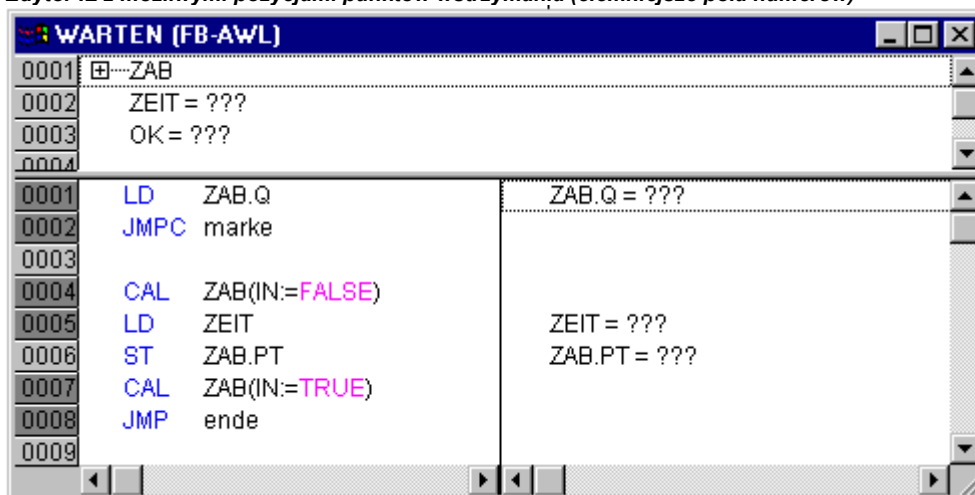
- Na początku modułu
- W każdym LD, LDN (lub też w znaczniku, jeśli LD znajduje się bezpośrednio po znaczniku)
- Przy każdym JMP, JMPC, JMPCN
- Przy każdym znaczniku
- Przy każdym CAL, CALC, CALCN
- Przy każdym RET, RETC, RETCN
- Na końcu modułu

W przypadku tekstu strukturalnego (ST) możliwe są następujące pozycje punktów wstrzymania:

- Przy każdym przyporządkowaniu
- Przy każdej instrukcji RETURN i EXIT
- w wierszach, w których są przetwarzane warunki (WHILE, IF, REPEAT)
- Na końcu modułu

Pozycje punktów wstrzymania są oznaczone w ten sposób, że pole numeru wiersza wyświetlane jest w kolorze szarym.

Edytor IL z możliwymi pozycjami punktów wstrzymania (ciemniejsze pola numerów)



W celu umieszczenia punktu wstrzymania użytkownik klika przy użyciu myszy pole numeru wiersza, w którym chce umieścić punkt wstrzymania. Jeśli wybrane pole to pozycja punktu wstrzymania, wtedy kolor pola numeru wiersza zmienia się z ciemnoszarego na jasnoszary i punkt wstrzymania uaktywniany jest w sterowniku.

Podobnie w celu usunięcia punktu wstrzymania należy kliknąć pole numeru wiersza, zawierające punkt wstrzymania przeznaczony do usunięcia.

Umieszczanie i usuwanie punktów wstrzymania umożliwia również menu („**Online**” „**Włącz/wyłącz punkt wstrzymania**”), klawisz funkcyjny <F9> lub wybranie symbolu na pasku funkcji.

Co się dzieje w punkcie wstrzymania?

Gdy w sterowniku zostanie osiągnięty punkt wstrzymania, na ekranie wyświetlany jest wycinek z odpowiednim wierszem. Pole numeru wiersza, w którym zatrzymał się sterownik, jest czerwone. W sterowniku następuje zatrzymanie przetwarzania programu użytkownika.

Jeśli program zatrzymał się w punkcie wstrzymania, przetwarzanie można kontynuować, wybierając polecenie „**Online**” „**Start**”.

Ponadto za pomocą polecenia „**Online**” „**Krok pojedynczy przez**” wzgl. „**Krok pojedynczy do**” można przejść tylko do następnej pozycji punktu wstrzymania. Jeśli instrukcja, w której aktualnie

nastąpiło zatrzymanie, jest poleceniem CAL wzgl. jeśli w wierszach do następnej pozycji punktu wstrzymania znajduje się wywołanie funkcji, za pomocą polecenia „**Krok pojedynczy przez**” wywołanie zostanie pominięte, zaś polecenie „**Krok pojedynczy do**” umożliwi przejście do wywołanego modułu.

Numery wierszy edytora tekstowego

Numery wierszy edytora tekstowego wskazują numer każdego wiersza tekstu implementacji danego modułu.

W trybie offline pojedynczym kliknięciem wybranego numeru wiersza można zaznaczyć cały wiersz tekstowy.

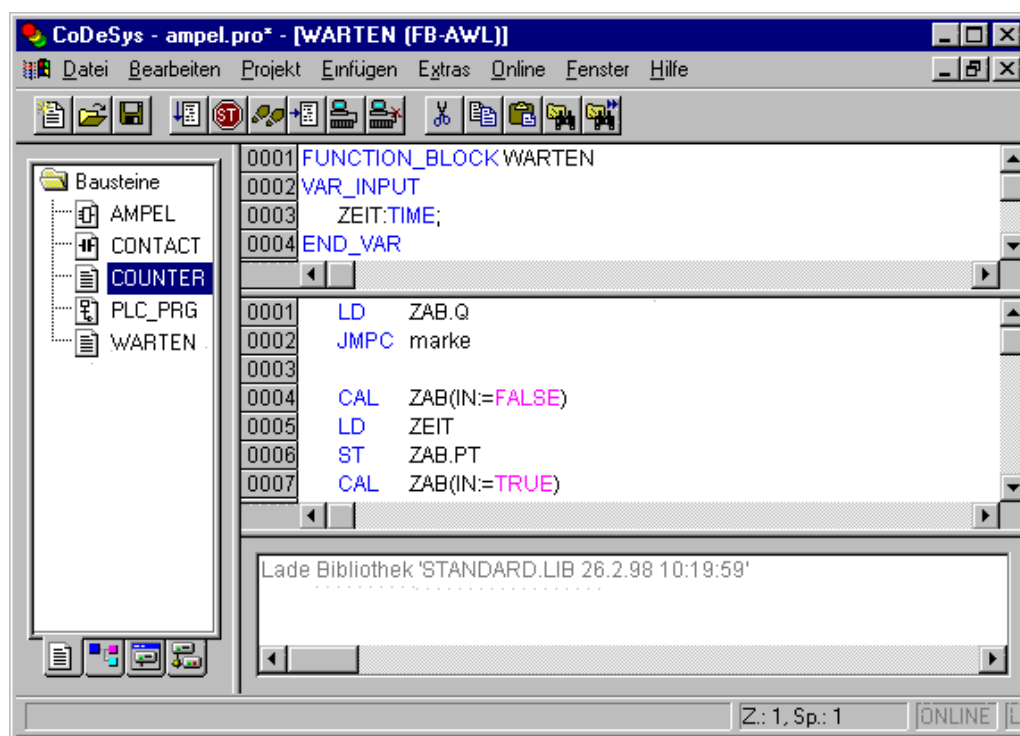
W trybie online kolor tła numeru wiersza wskazuje stan punktu wstrzymania w każdym wierszu:

- ciemnoszary: wiersz ten to możliwa pozycja punktu wstrzymania;
- jasnoniebieski: w tym wierszu umieszczono punkt wstrzymania;
- czerwony: przetwarzanie programu znajduje się w tym punkcie.

W trybie online kliknięcie przyciskiem myszy zmienia stan punktu wstrzymania w danym wierszu.

8.3.2 Edytor list instrukcji

Tak wygląda moduł napisany w IL w odpowiednim edytorze CoDeSys:



Wszystkie edytory modułów składają się z części deklaracyjnej oraz ze szkieletu. Są one od siebie oddzielone separatorem ekranu.

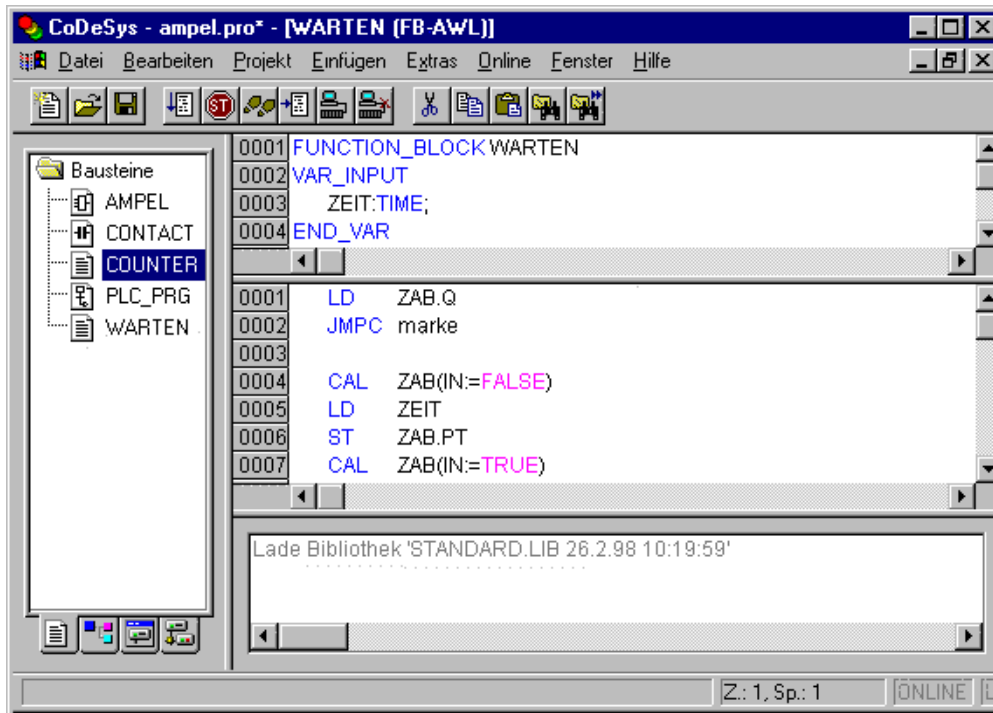
Edytor list instrukcji to edytor tekstowy, posiadający typowe funkcje edytorów tekstowych systemu Windows. Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

Możliwe jest wielokrotne wywołanie modułu. Przykład:

```
CAL CTU inst(
CU:=%IXI0,
PV:=(
LD A
ADD 5
)
)
```

Informacje na temat języka, patrz lista instrukcji (IL).

Tak wygląda moduł napisany w IL w odpowiednim edytorze CoDeSys:



Wszystkie edytory modułów składają się z części deklaracyjnej oraz ze szkieletu. Są one od siebie oddzielone separatorem ekranu.

Edytor list instrukcji to edytor tekstowy, posiadający typowe funkcje edytorów tekstowych systemu Windows. Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

Możliwe jest wielokrotne wywołanie modułu. Przykład:

```

CAL CTU inst(
CU:=%IXI0,
PV:=(
LD A
ADD 5
)
)
    
```

Informacje na temat języka, patrz lista instrukcji (IL).

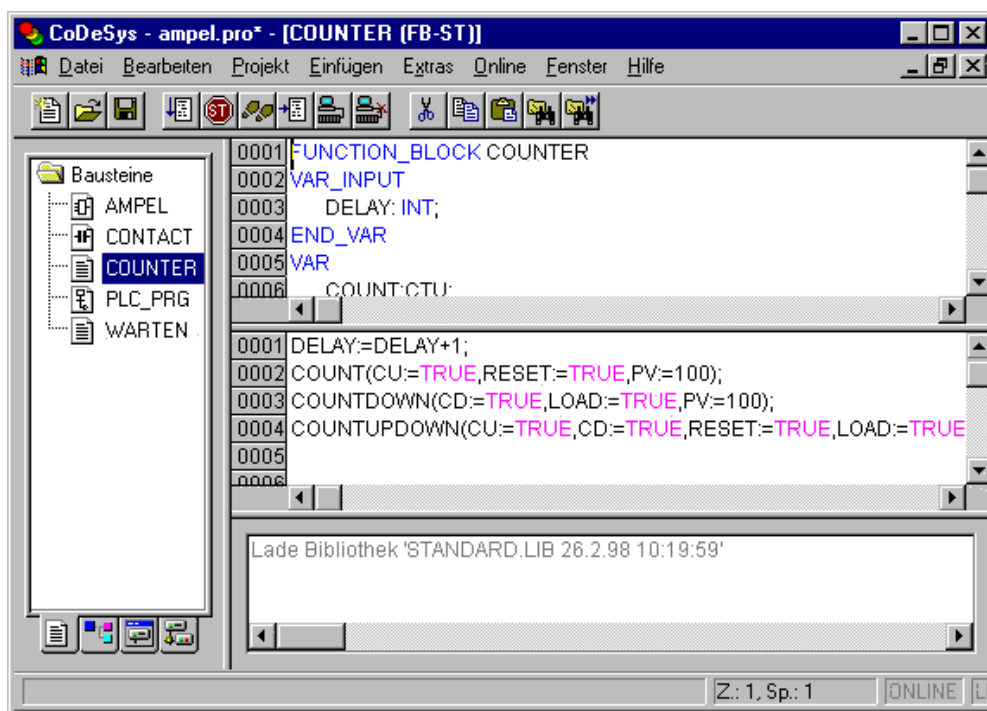
IL w trybie online

Za pomocą polecenia „Online” „Kontrola przebiegu” w edytorze IL po lewej stronie wiersza wstawiane jest dodatkowe pole, w którym prezentowana jest zawartość akumulatora.

Dalsze informacje na temat edytora IL w trybie online, patrz rozdział „Edytory tekstowe w trybie online”.

8.3.3 Edytor tekstu strukturalnego...

Tak wygląda moduł napisany w ST w odpowiednim edytorze CoDeSys:



Wszystkie edytory modułów składają się z części deklaracyjnej oraz ze szkieletu. Są one od siebie oddzielone separatorem ekranu.

Edytor tekstu strukturalnego to edytor tekstowy, posiadający typowe funkcje edytorów tekstowych systemu Windows. Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

Dalsze informacje na temat edytora ST w trybie online, patrz rozdział „Edytory tekstowe w trybie online”.

Informacje na temat języka, patrz „Tekst strukturalny (ST)”.

8.4 Edytory graficznych języków programowania...

8.4.1 Praca w edytorach graficznych

Edytory graficznych języków, języka przebiegu, schematu styków LD, schematu funkcji FBD oraz graficznego schematu funkcji CFC mają wiele cech wspólnych. Punkty te zostały omówione zbiorczo w ustępach (patrz niżej): Powiększenie, Sieć, Znaczniki skoku, Komentarze sieciowe, Wstaw sieć, Edytory tekstowe w trybie online. Ponadto dostępne są również odrębne opisy LD, FBD i CFC oraz języka przebiegu AS.

Implementacja w edytorach graficznych obsługuje kolorowanie składni (Syntaxcoloring).

Powiększenie

Obiekty w rodzaju modułów, czynności, przejść itp. w językach AS, LD, FBD, CFC i w wizualizacjach można powiększać i pomniejszać za pomocą funkcji powiększenia. Dotyczy tym wszystkich elementów znajdujących się w oknie części implementacyjnej – część deklaracyjna pozostaje niezmienniona.

Standardowo każdy obiekt wyświetlany jest przy zastosowaniu współczynnika powiększenia 100%. Ustawiony współczynnik powiększenia zapisywany jest w projekcie jako właściwość obiektu.

Drukowanie dokumentacji projektu odbywa się zawsze przy zastosowaniu widoku 100%.

Współczynnik powiększenia można ustawić za pośrednictwem listy wyboru na pasku narzędzi. Można wybrać wartości w zakresie od 25% do 400%, indywidualne wartości powiększenia w zakresie pomiędzy 10% a 500% można wprowadzić ręcznie.

Wybór współczynnika powiększenia jest dostępny tylko wtedy, gdy kursor znajduje się w obrębie obiektu utworzonego przy użyciu języka graficznego lub też w obrębie obiektu wizualizacji.

Pozycje kursora w edytorach można wybierać także po powiększeniu obiektu i przechodzić do nich za pomocą przycisków strzałek. Rozmiar tekstu dopasowuje się do współczynnika powiększenia i do ustawionego rozmiaru czcionki.

Wykonanie wszystkich poleceń menu, służących do obsługi edytora (np. wstawienie pola) zgodnie z pozycją kursora, jest możliwe w przypadku każdego współczynnika powiększenia i nie powoduje jego zmiany.

W trybie online każdy obiekt wyświetlany jest zgodnie z ustawionym współczynnikiem powiększenia, zaś funkcje online są dostępne bez ograniczeń.

W przypadku zastosowania funkcji IntelliMouse obiekt można powiększać/pomniejszać, przytrzymując naciśnięty klawisz <CTRL> i obracając jednocześnie pokrętkę myszy do przodu lub wstecz.

Sieć

W edytorach LD i FBD program umieszczany jest na liście sieci. Każda sieć oznaczona jest po lewej stronie kolejnym numerem sieci i zawiera strukturę przedstawiającą każdorazowo wyrażenie logiczne lub arytmetyczne, wywołanie programu, funkcji lub bloku funkcji, skok lub instrukcję Return.

Znaczniki skoku

Do każdej sieci przyporządkowany jest znacznik skoku, który może być również pusty. Znacznik można edytować, klikając pierwszy wiersz sieci, bezpośrednio obok numeru sieci. Teraz można wprowadzić znacznik, umieszczając za nim dwukropek.

Komentarze, podziały wierszy, „Dodatki” „Opcje”

Do każdej sieci można dodać kilkunastuwierszowy komentarz. W oknie dialogowym „Opcje schematu funkcji oraz schematu styków”, otwieranym za pomocą polecenia „Dodatki” „Opcje” można wprowadzić ustawienia dotyczące komentarzy:

Okno dialogowe opcji schematu funkcji oraz schematu styków

The screenshot shows a dialog box titled "Funktionen- und Kontaktplan Optionen". It has several input fields and checkboxes. At the top, there are two rows: "Minimale Kommentargröße:" with a value of 0 and "Zeilen" next to it, and "Maximale Kommentargröße:" with a value of 4 and "Zeilen" next to it. Below these are two buttons: "OK" and "Abbrechen". A section titled "Alternatives Look & Feel für Kontaktplan" contains several checkboxes: "Kommentare pro Kontakt" (checked), "Zeilen für Variablenkommentar:" with a value of 2 and "Zeilen" next to it, "Zeilen für Variablentext:" with a value of 1 and "Zeilen" next to it, "Netzwerke mit Umbrüchen" (checked), "Mit Symbol ersetzen, nach Eingabe der Adresse" (unchecked), "Kontaktkommentar mit Symbolkommentar vorbesetzen" (unchecked), "Adresse des Symbols anzeigen" (checked), and "Variablenkommentare pro Netzwerk im Ausdruck anzeigen" (checked). At the bottom right of this section is a button labeled "Optionen anwenden".

Maksymalny rozmiar komentarza: Liczba wierszy, dostępnych maksymalnie dla komentarza sieci (wartość ustawiona domyślnie to 4).

Minimalny rozmiar komentarza: Liczba wierszy dostępnych lub wyświetlanych generalnie dla komentarzy. Jeśli w tym miejscu zostanie wprowadzona np. cyfra 2, wtedy na każdym początku sieci po wierszu etykiety dostępne będą dwa puste wiersze komentarza. Wartość domyślna to w tym przypadku 0, co ma tę zaletę, że w obrębie ekranu mieści się więcej sieci.

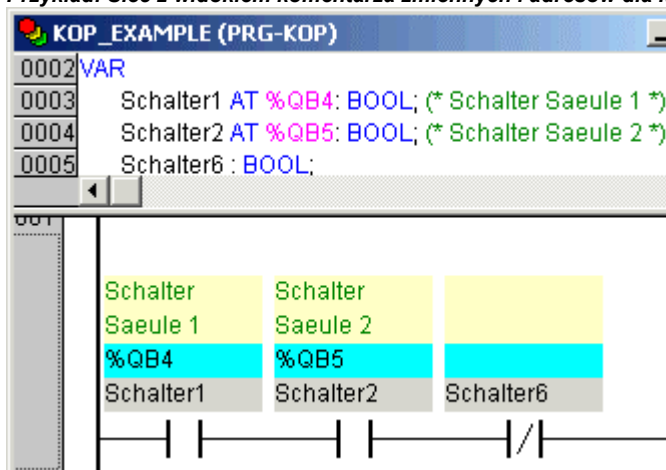
Jeśli minimalny rozmiar komentarza sieci jest większy niż 0, wtedy w celu wprowadzenia komentarza należy kliknąć wyświetlony wiersz komentarza i wpisać komentarz. W przeciwnym razie należy najpierw wybrać sieć, do której ma zostać dołączony komentarz, a następnie za pomocą polecenia „Wstaw” „Komentarz” wstawić wiersz komentarza. Komentarze wyświetlane są w kolorze szarym – w odróżnieniu od tekstu programu.

Alternatywny wygląd: Poniższe opcje umożliwiają alternatywny sposób wyświetlania sieci.

Komentarze/styk (tylko w przypadku schematu styków): Jeśli włączono tę opcję, można dodawać komentarze do poszczególnych styków i cewek. W polu **Wiersze do komentarzy zmiennych** należy wprowadzić żadaną liczbę wierszy, które są do tego przewidziane i mają być wyświetlane. Zostanie wtedy wyświetlone nad stykiem wzgl. cewką pole komentarza, w którym można wpisać tekst.

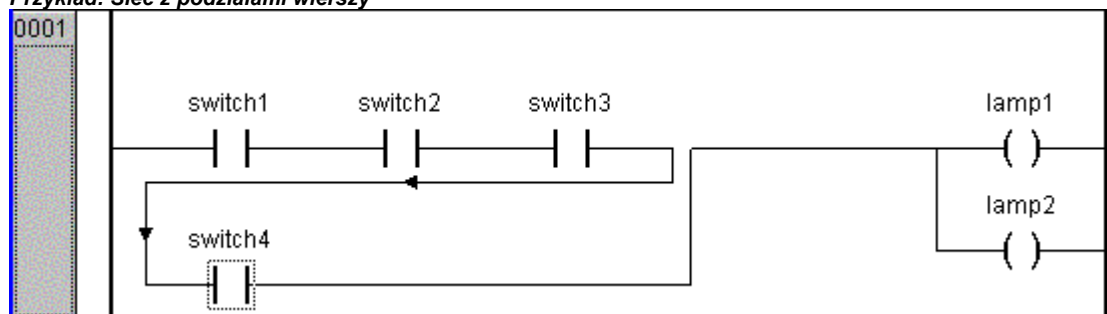
Jeśli włączona jest opcja „Komentarze/styk” można ponadto zdefiniować liczbę wierszy (**Wiersze dla tekstu zmiennych**), jaka będzie używana dla nazwy zmiennych styku wzgl. cewki, aby dzięki wykorzystaniu kilku wierszy możliwe było wyświetlanie w całości również długich nazw. W poniższym przykładzie przewidziano 2 wiersze dla komentarza styku oraz 1 dla tekstu zmiennej:

Przykład: Sieć z widokiem komentarza zmiennych i adresów dla każdego styku



Sieci z podziałami (tylko w przypadku schematu styków): jeśli ta opcja jest włączona, w sieciach wstawiane są podziały, gdy tylko ustawiona szerokość okna uniemożliwia wyświetlenie wszystkich elementów sieci.

Przykład: Sieć z podziałami wierszy



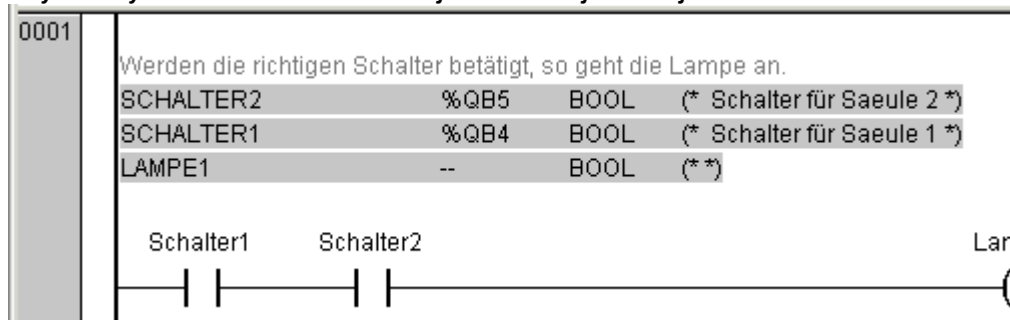
Zastąp symbolem po wprowadzeniu adresu: Jeśli opcja ta jest aktywna, można wtedy w module wzgl. przy styku lub cewce wprowadzić adres (np. „%QB4”), który bezpośrednio po wprowadzeniu zostanie zastąpiony przez nazwę zmiennej przypisanej do tego adresu. Jeśli do adresu nie jest przypisana żadna zmienna, widok pozostanie niezmienny.

Wypełnij komentarz styku komentarzem symbolu (tylko w przypadku planu styków): Jeśli ta opcja jest aktywna, w polu komentarza do styku lub cewki wyświetlany jest komentarz, który został zdefiniowany dla zmiennej podczas jej deklaracji i można go tam dalej edytować (patrz wyżej, rys. dotyczący opcji „Komentarze/styk”). Musi być jednakże w tym celu włączona również opcja „Komentarze/styk” (patrz wyżej). Uwaga: Komentarz wprowadzony już lokalnie w polu komentarza jest w takim wypadku zastępowany automatycznie przez komentarz zmiennej, ew. przez spację, jeśli w deklaracji zmiennej brak jest komentarza.

Wyświetl adres symbolu (tylko w przypadku planu styków): Jeśli zmienna wprowadzona przy styku wzgl. przy cewce przyporządkowana jest do adresu, jest on wyświetlany dodatkowo powyżej nazwy zmiennej (patrz wyżej, rys. dotyczący opcji „Komentarze/styk”).

Pokaż komentarze zmiennych dla każdej sieci w wydruku: Jeśli opcja ta jest aktywna, dla każdej sieci wyświetlany jest jeden wiersz dla każdej zmiennej użytej w sieci, przedstawiający nazwę zmiennej, adres, typ danych, jak również komentarz do zmiennej, tak jak zdefiniowano to w deklaracji zmiennej. Może to być przydatne dla dokumentacji projektu (wydruk). Przykład:

Przykład: Wyświetlanie wiersza z informacjami dla każdej zmiennej sieci



Zastosowanie opcji:

OK: Użycie tego przycisku spowoduje zastosowanie ustawionych opcji w aktualnym module i zamknięcie okna dialogowego.

Zastosuj opcje: Za pomocą tego przycisku można zastosować ustawione opcje do całego projektu. Zostanie wyświetlone okno z pytaniem, które należy jeszcze raz bezpośrednio potwierdzić.

„Wstaw” „Sieć (za)” lub „Wstaw” „Sieć (przed)”

Kombinacja klawiszy: <Shift> + <T>

Aby wstawić nową sieć w edytorze FBD lub LD, należy wybrać polecenie „Wstaw” „Sieć (za)” lub „Wstaw” „Sieć (przed)”, w zależności od tego, czy nowa sieć ma zostać wstawiona przed czy też za aktualną siecią. Aktualną sieć można zmienić, klikając na numer sieci. Można ją rozpoznać po kropkowanym prostokącie pod numerem. Za pomocą klawisza <Shift> i kliknięcia przyciskiem myszy można wybrać cały obszar sieci pomiędzy aktualną i wybraną siecią.

Edytory sieci w trybie online

W edytorach FBD i LD punkty wstrzymania można umieszczać tylko w sieciach. Numer pola sieci, w której umieszczono punkt wstrzymania, jest wyświetlany w kolorze niebieskim. Przetwarzanie zatrzymuje się wtedy przed siecią, w której znajduje się punkt wstrzymania. W takim przypadku numer pola sieci wyświetlany jest w kolorze czerwonym. W przypadku przetwarzania pojedynczych kroków (przetwarzanie krokowe) następuje przejście od sieci do sieci.

Wszystkie wartości monitorowane są na wejściach i wyjściach modułów sieci.

Podczas monitorowania wyrażeń oraz zmiennych adresowanych bitowo należy mieć na uwadze następujące kwestie: W przypadku wyrażeń, np. a AND b stanowiących warunek przejścia lub wejście bloku funkcji, wyświetlana jest zawsze wartość całego wyrażenia (a AND b wyświetlane jest na niebiesko wzgl. za pomocą :=TRUE, jeśli a i b mają wartość TRUE). W przypadku zmiennych adresowanych bitowo monitorowana jest zawsze wywoływana wartość bitu (np. a.3 niebieski wzgl. z :=TRUE, jeśli a ma wartość 4).

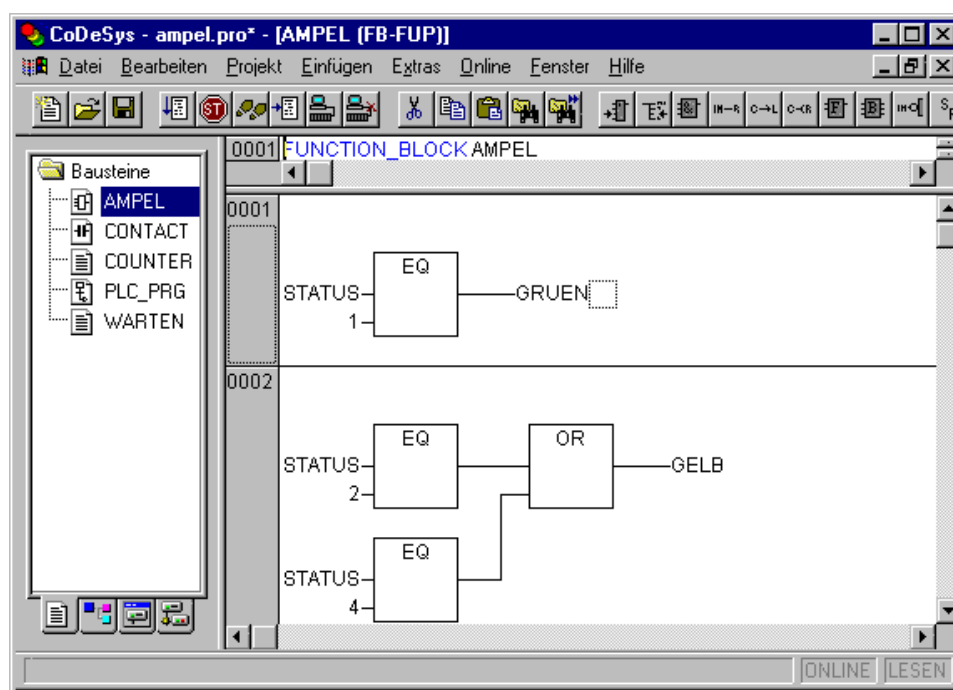
Kontrolę przebiegu można uruchomić za pomocą polecenia menu „Online” „Kontrola przebiegu”. Zapewnia ono podgląd aktualnych wartości, jakie przekazywane są w sieciach za pośrednictwem linii połączeń. Jeśli linie połączeń nie przekazują wartości boolean, wartość jest wyświetlana we wstawionym dodatkowo polu. Nieużywane pola monitoringu zmiennych (np. w przypadku funkcji SEL) są wyświetlane z szarym cieniowaniem. Jeśli linie przekazują wartości boolean, mają one – o ile przekazywane wartości dają TRUE – kolor niebieski. W ten sposób można śledzić przepływ informacji podczas działania sterownika.

Przytrzymując kursor myszy przez krótki czas nad zmienną, można wyświetlić typ, adres oraz komentarz do zmiennej w oknie etykiety.

8.4.2 Edytor schematów funkcji...

Edytor schematów funkcji to edytor graficzny. Działa on w oparciu o listę sieci, przy czym każda sieć posiada strukturę, przedstawiającą zawsze jakieś wyrażenie logiczne wzgl. arytmetyczne, wywołanie bloku funkcji, funkcji, programu, skok lub instrukcję powrotu.

Tak wygląda moduł napisany w FBD w odpowiednim edytorze CoDeSys:



Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

Należy pamiętać, że w przypadku modułów utworzonych w FBD istnieje możliwość przełączania (zarówno w trybie offline, jak i online) pomiędzy widokiem w edytorze FBD i LD (patrz „Dodatki” „Widok”). Ponadto istnieją możliwości konfiguracji komentarzy, wprowadzania adresów itp. za pośrednictwem okna dialogowego opcji, patrz rozdział 8.4.1, Komentarze, Podziały, „Dodatki” „Opcje”.

Pozycje kursora w FBD

Każdy tekst to możliwa pozycja kursora. Wybrany tekst jest podświetlony na niebiesko i można go wtedy zmienić.

W przeciwnym razie pozycję kursora wskazuje kropkowany prostokąt. Poniżej znajduje się wyliczenie wszystkich możliwych pozycji kursora wraz z przykładem:

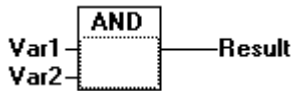
1) Każde pole tekstowe (możliwe pozycje kursora z czarnym obramowaniem):



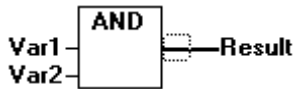
2) Każde wejście:



3) Każdy operator, funkcja lub moduł funkcji:



4) Wyjścia, jeśli po nich znajduje się przyporządkowanie lub skok:



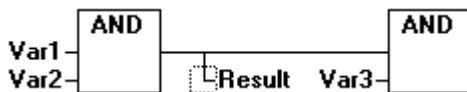
5) Skrzyżowanie linii nad przyporządkowaniem, skokiem lub inną instrukcją powrotu:



6) Za obiektem każdej sieci, najbardziej wysuniętym w prawo („ostatnia pozycja kursora”, jest to także pozycja kursora w momencie wybrania sieci):



7) Skrzyżowanie linii bezpośrednio przed przyporządkowaniem:



Ustawianie kursora w FBD

Kursor można umieścić na określonej pozycji klikając przyciskiem myszy lub przy użyciu klawiatury.

Za pomocą klawiszy strzałek można każdorazowo przejść do najbliższej pozycji kursora w wybranym kierunku. Można w ten sposób osiągnąć wszystkie pozycje kursora, włącznie z polami tekstowymi. Jeśli wybrano ostatnią pozycję kursora, za pomocą klawiszy strzałek <do góry> / <na dół> można wybrać ostatnią pozycję kursora poprzedniej wgl. następną sieci.

Pusta sieć zawiera jedynie trzy znaki zapytania „???”. Kliknięcie za tymi znakami powoduje wybranie ostatniej pozycji kursora.

„Wstaw” „Przyporządkowanie” w FBD

Symbol:  Kombinacja klawiszy: <Ctrl> + <A>

Polecenie to powoduje wstawienie przyporządkowania.

Wstawienie nastąpi, w zależności od wybranej pozycji (patrz „Pozycje kursora w FBD”), bezpośrednio przed wybranym wejściem, bezpośrednio za wybranym wyjściem lub też na końcu sieci.

W odniesieniu do wstawionego przyporządkowania można następnie wybrać wprowadzony tekst „???” i zastąpić go zmienną, do której powinien być przyporządkowany. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Należy pamiętać o możliwości wprowadzania adresów, jeśli opcja ta została ustawiona w oknie dialogowym opcji (patrz rozdział 8.4.1: Komentarze, Podziały wierszy, „Dodatki” „Opcje”).

Aby do istniejącego już przyporządkowania dodać kolejne, należy użyć polecenia „Wstaw” „Wyjście”.

„Wstaw” „Skok” w FBD

Symbol:  **Kombinacja klawiszy: <Ctrl> + <L>**

Polecenie to powoduje wstawienie skoku.

Wstawienie nastąpi, w zależności od wybranej pozycji (patrz „Pozycje kursora w FBD”), bezpośrednio przed wybranym wejściem, bezpośrednio za wybranym wyjściem lub też na końcu sieci.

W odniesieniu do wstawionego skoku można następnie wybrać wprowadzony tekst „???” i zastąpić go znacznikiem skoku, do którego powinno nastąpić przejście.

„Wstaw” „Powrót” w FBD

Symbol:  **Kombinacja klawiszy: <Ctrl> + <R>**

Polecenie to powoduje wstawienie instrukcji RETURN.

Wstawienie nastąpi, w zależności od wybranej pozycji (patrz „Pozycje kursora w FBD”), bezpośrednio przed wybranym wejściem, bezpośrednio za wybranym wyjściem lub też na końcu sieci.

„Wstaw” „Moduł” w FBD

Symbol:  **Kombinacja klawiszy: +**

Za pomocą tego polecenia można wstawiać operatory, funkcje, bloki funkcji oraz programy. Najpierw wstawiany jest zawsze operator „AND”. Wybierając go i zastępując tekst typu („AND”) można przekształcić go w każdy inny operator, w każdą funkcję, w każdy blok funkcji oraz w każdy program. Przy użyciu funkcji pomocy przy wpisywaniu (<F2>) można wybrać żądany moduł. Jeśli nowo wybrany moduł ma inną minimalną liczbę wejść, zostaną one dołączone. Jeśli nowy moduł ma mniejszą maksymalną liczbę wejść, ostatnie wejścia zostaną usunięte.

W przypadku funkcji i bloków funkcji można wyświetlać formalne nazwy wejść i wyjść.

W przypadku bloków funkcji istnieje edytowalne pole instancji ponad oknem. Jeśli w wyniku zmiany tekstu typu zostanie wywołany inny blok funkcji, który nie jest znany, zostanie wyświetlone okno operatora z dwoma wejściami oraz podanym typem. Jeśli wybrano pole instancji, za pomocą klawisza <F2> można wywołać funkcję pomocy przy wpisywaniu, zawierającą kategorie zmiennych do wyboru.

Nowy moduł zostanie wstawiony w zależności od wybranej pozycji (patrz „Pozycje kursora w FBD”).

- Jeśli zaznaczone jest wejście, moduł zostanie wstawiony przed tym wejściem. Pierwsze wejście modułu zostanie połączone z gałęzią na lewo od zaznaczonego wejścia. Wyjście nowego modułu zostanie połączone z zaznaczonym wejściem.
- Jeśli zaznaczone jest wyjście, moduł zostanie wstawiony za tym wyjściem. Pierwsze wejście modułu zostanie połączone z zaznaczonym wyjściem. Wyjście nowego modułu zostanie połączone z odgałęzieniem, z którym połączone było zaznaczone wyjście.
- Jeśli zaznaczono moduł, funkcję lub blok funkcji, wtedy stary element zostanie zastąpiony nowym modułem. Odgałęzienia zostaną połączone, o ile to możliwe, tak jak przed zastąpieniem. Jeśli poprzedni element posiadał więcej wejść niż nowy, wtedy odgałęzienia bez możliwości połączenia zostaną usunięte. To samo dotyczy wyjść.

- Jeśli zaznaczono skok lub powrót, moduł zostanie wstawiony przed tym skokiem wzgl. powrotem. Pierwsze wejście modułu zostanie połączone z gałęzią na lewo od zaznaczonego elementu. Pierwsze wyjście modułu zostanie połączone z gałęzią na prawo od zaznaczonego elementu.
- Jeśli zaznaczono ostatnią pozycję kursora w sieci, wtedy moduł zostanie wstawiony za ostatnim elementem. Pierwsze wejście modułu zostanie połączone z gałęzią na lewo od zaznaczonej pozycji.

Wszystkie wejścia modułu, których nie można połączyć, otrzymają tekst „???”. Tekst ten należy kliknąć i zamienić na żadaną stałą lub zmienną.

Jeśli z prawej strony wstawionego modułu znajduje się gałąź, zostanie ona przyporządkowana do pierwszego wyjścia modułu. W przeciwnym razie wyjścia pozostaną przyporządkowane.

„Wstaw” „Wejście”

Symbol:  **Kombinacja klawiszy: <Ctrl> + <U>**

Polecenie to powoduje wstawienie wejścia operatora. Liczba wejść jest w przypadku wielu operatorów zmienna (np. ADD może posiadać 2 lub więcej wejść).

Aby rozszerzyć taki operator o wejście, należy zaznaczyć wejście, przed którym ma zostać wstawione kolejne wejście, albo też sam operator, jeśli ma zostać wstawione najniższe wejście (patrz „Pozycje kursora w FBD”).

We wstawionym wejściu znajduje się tekst „???”. Tekst ten należy kliknąć i zamienić na żadaną stałą lub zmienną. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Należy pamiętać o możliwości wprowadzania adresów, jeśli opcja ta została ustawiona w oknie dialogowym opcji (patrz rozdział 8.4.1: Komentarze, Podziały wierszy, „Dodatki” „Opcje”).

„Wstaw” „Wyjście”

Symbol: 

Polecenie to powoduje dodanie dodatkowego przyporządkowania do istniejącego przyporządkowania. Funkcja ta służy do tworzenia tzw. ramion przyporządkowujących, czyli przyporządkowania wartości danego przewodu do wielu zmiennych.

Jeśli zostanie zaznaczone skrzyżowanie linii ponad przyporządkowaniem lub też wyjście znajdujące się bezpośrednio przed nim, za istniejącymi już przyporządkowaniami zostanie dodane kolejne.

Jeśli zostało zaznaczone skrzyżowanie linii przed przyporządkowaniem, wtedy przed tym przyporządkowaniem zostanie wstawione kolejne.

We wstawionym wyjściu znajduje się tekst „???”. Tekst ten należy kliknąć i zamienić na żadaną zmienną. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Należy pamiętać o możliwości wprowadzania adresów, jeśli opcja ta została ustawiona w oknie dialogowym opcji (patrz rozdział 8.4.1: Komentarze, Podziały wierszy, „Dodatki” „Opcje”).

„Dodatki” „Negacja”

Symbol:  **Kombinacja klawiszy: <Ctrl> + <N>**

Symbol negacji to małe kółko na połączeniu.

Jeśli zaznaczone jest wejście, zostanie ono zanegowane.

Jeśli zaznaczone jest wyjście, zostanie ono zanegowane.

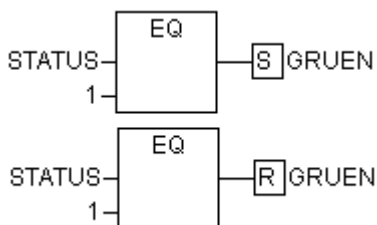
Jeśli zaznaczony jest Skok lub Powrót, zanegowane zostanie wejście tego skoku wzgl. powrotu.

Negację można usunąć poprzez ponowną negację.

„Dodatki” „Ustaw/Przywróć”

Symbol: 

Za pomocą tego polecenia można zdefiniować wyjścia jako wyjścia Set lub Reset. Bramka z wyjściem Set jest wyświetlana jako [S], zaś z wyjściem Reset jako [R].

Wyjścia Set/Reset w FBD

Wyjście Set ma ustawioną wartość TRUE, jeśli odnośna bramka zwraca TRUE. Wyjście zachowuje teraz tę wartość, nawet jeśli bramka przeskoczy ponownie na wartość FALSE.

Wyjście Reset ma ustawioną wartość FALSE, jeśli odnośna bramka zwraca TRUE. Wyjście zachowuje tę wartość, także jeśli bramka przeskoczy ponownie na wartość FALSE.

W przypadku wielokrotnego wykonania polecenia wyjście przełącza się pomiędzy Set, Reset i zwykłym wyjściem.

„Dodatki” „Widok”

Za pomocą tego polecenia dla modułu utworzonego w edytorze schematów funkcji można wybierać pomiędzy widokiem edytora schematów styków wzgl. edytora schematów funkcji. Jest to możliwe zarówno w trybie offline, jak i online.

Powiększanie wywołanego modułu**Kombinacja klawiszy: <Alt> + <Enter>**

Polecenie to jest dostępne w menu podręcznym (<F2>) lub też w menu „Dodatki”, jeśli kursor w edytorze tekstowym znajduje się na nazwie wywołanego modułu wzgl. gdy w edytorze graficznym zaznaczone jest pole danego modułu. Powiększenie powoduje otwarcie danego modułu w oknie edytora.

Jeśli jest to moduł z biblioteki, wywoływany jest Menedżer bibliotek i wyświetlany jest odpowiedni moduł.

Wycinanie, kopiowanie, wklejanie i usuwanie w FBD

Polecenia „Wytnij”, „Kopiuj”, „Wklej” lub „Usuń”, dostępne są w punkcie menu „Edycja”.

Jeśli zaznaczone jest skrzyżowanie linii, wtedy wszystkie znajdujące się poniżej przyporządkowania, skoki lub instrukcje RETURN zostaną wycięte, usunięte lub skopiowane.

Jeśli zaznaczony jest moduł, wtedy zarówno sam zaznaczony obiekt, jak również wszystkie znajdujące się na wejściach gałęzie, z wyjątkiem pierwszej (najwyższej), zostaną wycięte, usunięte lub skopiowane.

W przeciwnym razie wycięta, usunięta lub skopiowana zostanie cała gałąź znajdująca się przed pozycją kursora.

Po skopiowaniu lub wycięciu element usunięty wzgl. skopiowany znajduje się w schowku i można go stamtąd wklejać dowolną ilość razy.

Należy w tym celu najpierw wybrać pozycję wklejania. Prawidłowe pozycje wklejania to wejścia i wyjścia.

Jeśli w schowku umieszczono moduł (dla przypomnienia: w takim przypadku w schowku znajdują się również wszystkie przylegające do niego gałęzie, z wyjątkiem pierwszej), wtedy pierwsze wejście zostanie połączone z gałęzią przed pozycją wklejenia.

W przeciwnym razie zawartość schowka zastąpi całą gałąź znajdującą się przed pozycją wklejenia.

W takim przypadku ostatni wklejony element zostanie połączony z gałęzią znajdującą się na prawo od pozycji wklejenia.

Wskazówka: Poprzez wycięcie i wklejenie można rozwiązać następujący problem: W środku sieci wstawiony został nowy operator. Gałąź znajdująca się na prawo od operatora jest teraz połączona z pierwszym wejściem, powinna być jednak połączona z drugim wejściem. Należy więc zaznaczyć pierwsze wejście i zastosować polecenie menu „Edycja” „Wytnij”. Następnie należy zaznaczyć drugie wejście i zastosować polecenie menu „Edycja” „Wklej”. Teraz gałąź będzie związana z drugim wejściem.

Schemat funkcji w trybie online

W schemacie funkcji punkty wstrzymania można umieszczać tylko w sieciach. W wyniku umieszczenia punktu wstrzymania w sieci, pole numeru sieci zostanie wyświetlone w kolorze niebieskim. Przetwarzanie zatrzymuje się wtedy przed siecią, w której znajduje się punkt wstrzymania. W takim przypadku numer pola sieci wyświetlany jest w kolorze czerwonym. W przypadku przetwarzania krokowego (krok pojedynczy) następuje przejście od sieci do sieci.

W odniesieniu do każdej zmiennej wyświetlana jest aktualna wartość. Wyjątek: jeśli wejście bloku funkcji jest wyrażeniem, monitorowana jest tylko pierwsza zmienna tego wyrażenia.

Dwukrotne kliknięcie zmiennej powoduje otwarcie okna dialogowego zapisu zmiennej. Można w nim zmienić wartość aktualnej zmiennej. W przypadku zmiennych boolean okno dialogowe nie jest wyświetlane, lecz następuje przełączenie wartości.

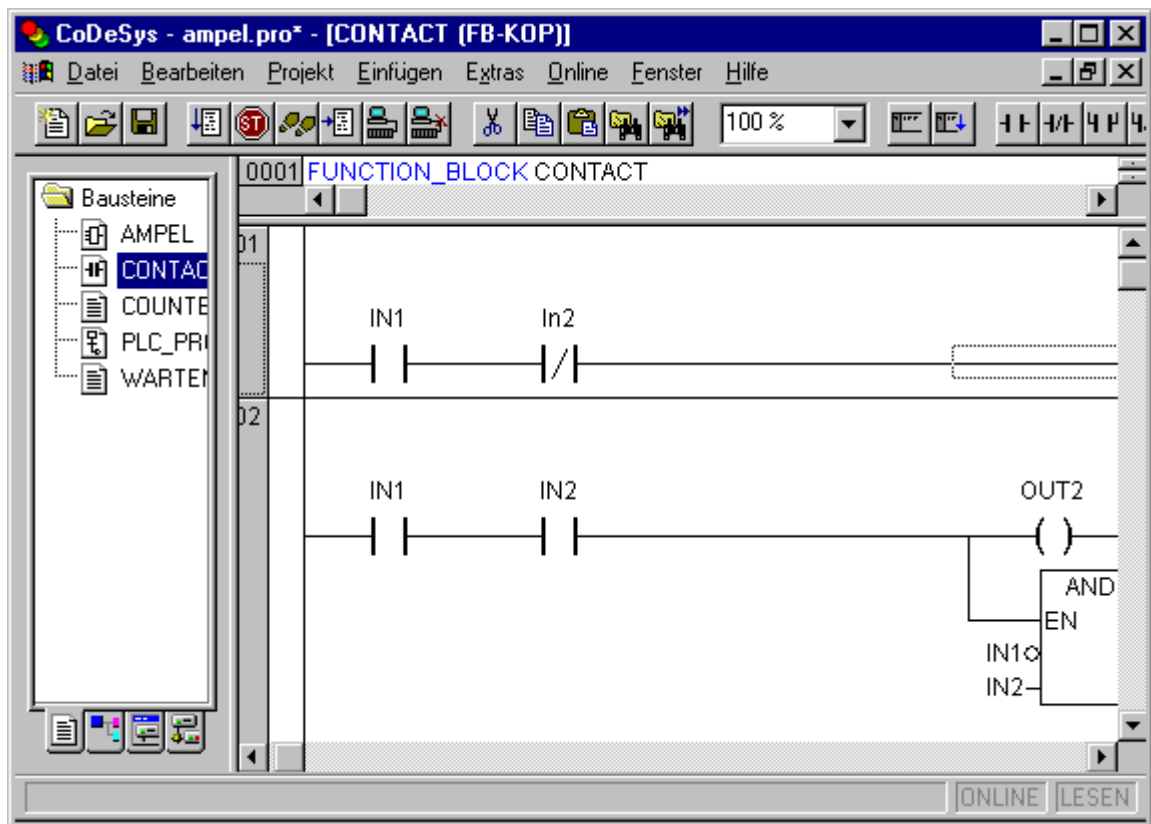
Nowa wartość ma kolor czerwony i pozostaje niezmienną. Jeśli zostanie użyte polecenie „Online” „Zapisz wartości”, wszystkim zmiennym zostaną nadane wybrane wartości i będą one ponownie wyświetlane w kolorze czarnym.

Kontrolę przebiegu można uruchomić za pomocą polecenia menu „Online” „Kontrola przebiegu”. Zapewnia ono podgląd aktualnych wartości, jakie przekazywane są w sieciach za pośrednictwem linii połączeń. Jeśli linie połączeń nie przekazują wartości boolean, wartość jest wyświetlana we wstawionym dodatkowo polu. Jeśli linie przekazują wartości boolean, mają one – o ile przekazywane wartości dają TRUE – kolor niebieski. W ten sposób można śledzić przepływ informacji podczas działania sterownika.

Przytrzymując kursor myszy przez krótki czas nad zmienną, można wyświetlić typ, adres oraz komentarz do zmiennej w oknie etykiety.

8.4.3 Edytor schematów styków...

Tak wygląda moduł napisany w LD w odpowiednim edytorze CoDeSys:



Wszystkie edytory modułów składają się z części deklaracyjnej oraz ze szkieletu. Są one od siebie oddzielone separatorem ekranu.

Edytor LD to edytor graficzny. Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

Informacje na temat dostępnych elementów, patrz rozdział 2.2.6, Schemat styków (LD).

Pozycje kursora w edytorze LD

Następujące miejsca mogą stanowić pozycje kursora, przy czym wywołania bloków funkcji oraz programów mogą być traktowane jak styki. Moduły z wejściami EN oraz moduły z nimi powiązane są traktowane tak jak w schemacie funkcji. Informacje na temat edycji tych elementów sieci, patrz „Edytor LD” (rozdział 2.2.4).

1. Każde pole tekstowe (możliwe pozycje kursora z czarnym obramowaniem)



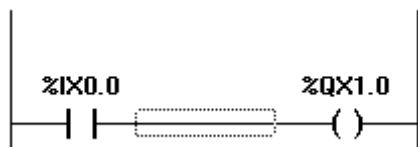
2. Każdy styk lub blok funkcji



3. Każda cewka



4. Linia połączenia pomiędzy stykami a cewkami

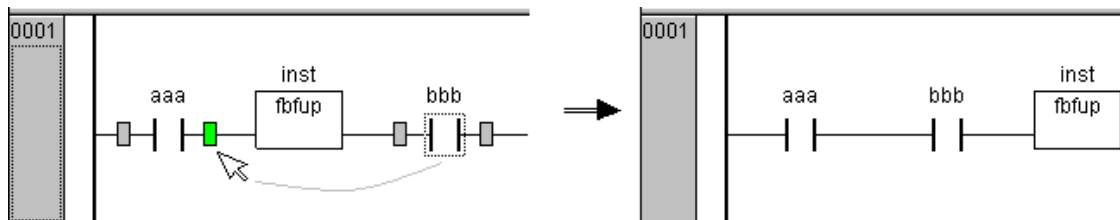


Elementy, przesuwanie nazw w edytorze LD

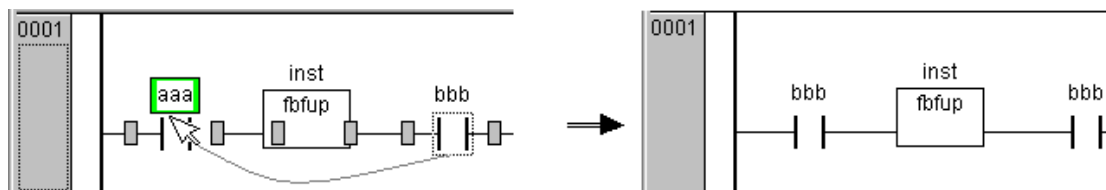
Zarówno kompletne elementy (styk, cewka, blok funkcji) modułu LD, ale również nazwy (zmienna, adres, komentarz) elementu można przesuwać przy użyciu metody „Drag&Drop” w dowolne miejsce w obrębie modułu.

W tym celu zaznacz styk lub cewkę albo blok funkcji i przytrzymując naciśnięty klawisz myszy przeciągnij go z aktualnej pozycji wstawienia. Wszystkie pozycje w obrębie sieci danego modułu, do których można przesunąć zaznaczony element, są teraz pokazywane jako szare prostokąty.

Gdy tylko element zostanie przeciągnięty na któryś z tych prostokątów, kolor jego wypełnienia zmieni się wtedy na zielony. Po zwolnieniu przycisku myszy element zostanie wstawiony na nowe miejsce.



Jeśli natomiast element po przeciągnięciu zostanie umieszczony nad opisem (nazwa zmiennej) innego elementu, będzie on wtedy wyświetlany na zielono. Po zwolnieniu przycisku myszy dotychczasowa nazwa zostanie zastąpiona nazwą „przyciągniętą”. Jeśli dodatkowo są wyświetlane adres i komentarz, to kopiowanie odnosi się również do nich.



„Wstaw” „Sieć (przed)” w LD

Symbol:


Za pomocą tego polecenia można wstawić nową sieć w edytorze. Jeśli występują już jakieś sieci, nowa sieć zostanie wstawiona przed tą siecią, w której aktualnie znajduje się kursor.

„Wstaw” „Sieć (za)” w LD

Symbol: Kombinacja klawiszy: <Ctrl> + <T>

Za pomocą tego polecenia można wstawić nową sieć w edytorze. Jeśli występują już jakieś sieci, nowa sieć zostanie wstawiona za siecią, w której aktualnie znajduje się kursor.

„Wstaw” „Styk” w LD

Symbol:  Kombinacja klawiszy: <Ctrl> + <K>

Za pomocą tego polecenia można dodać styk, który zostanie wstawiony w edytorze LD przed zaznaczonym miejscem w sieci.

Jeśli zaznaczonym miejscem jest cewka lub linia łącząca styki i cewki, wtedy nowy styk zostanie podłączony szeregowo do istniejącego układu połączeń.

Styk zawiera domyślnie tekst „???”. Klikając ten tekst można zmienić w nim nazwę żadnej zmiennej lub stałej. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Należy pamiętać o możliwości wprowadzania adresów, jeśli opcja ta została ustawiona w oknie dialogowym opcji (patrz rozdział 8.4.1: Komentarze, podziały wierszy, „Dodatki” „Opcje”).

Jeśli w oknie dialogowym Opcje została również zaznaczona opcja **Komentarze/styk**, można tam również obok żadnej liczby **wierszy do komentarzy zmiennych** podać odpowiednią liczbę **wierszy dla nazwy zmiennych**. Jest to przydatne w przypadku długich nazw zmiennych, by zachować zwartą strukturę sieci.

Warto również pamiętać o opcji **Sieci z podziałami**, którą można włączyć również za pomocą polecenia „Dodatki” „Opcje”.

„Wstaw” „Styk (zanegowany)” w LD

Symbol:  Kombinacja klawiszy: <Ctrl> + <G>

Za pomocą tego polecenia można wstawić styk zanegowany. Otrzymywany efekt jest taki sam, jak przy użyciu poleceń „Wstaw” „Styk” i „Dodatki” „Negacja”, które w kombinacji dają również styk zanegowany.

„Wstaw” „Styk równoległy” w LD

Symbol:  Kombinacja klawiszy: <Ctrl> + <R>

Za pomocą tego polecenia można dodać styk, który zostanie wstawiony w edytorze LD przed zaznaczonym miejscem w sieci.

Jeśli zaznaczonym miejscem jest cewka lub linia łącząca styki i cewki, wtedy nowy styk zostanie podłączony równolegle do istniejącego układu połączeń.

Styk zawiera domyślnie tekst „???”. Klikając ten tekst można zmienić w nim nazwę żadnej zmiennej lub stałej. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Informacje na temat wprowadzania adresów, wyświetlania nazwy zmiennej w kilku wierszach lub dodawania własnych komentarzy dla styku znajdują się wyżej w części „Wstaw” „Styk”.

„Wstaw” „Styk równoległy (zanegowany)” w LD

Symbol:  Kombinacja klawiszy: <Ctrl> + <O>

Za pomocą tego polecenia można wstawić styk równoległy zanegowany. Otrzymywany efekt jest taki sam, jak przy użyciu poleceń „Wstaw” „Styk równoległy” i „Dodatki” „Negacja”, które w kombinacji dają również styk równoległy zanegowany.

„Wstaw” „Cewka” w LD


Symbol:  Kombinacja klawiszy: <Ctrl> + <L>

Za pomocą tego polecenia można dodać cewkę, która zostanie wstawiona w edytorze LD równolegle do istniejących już cewek.

Jeśli zaznaczonym miejscem jest połączenie między stykami i cewkami, wtedy nowa cewka zostanie wstawiona jako ostatnia. Jeśli zaznaczonym miejscem jest cewka, wówczas nowa cewka zostanie wstawiona bezpośrednio nad nią.


Cewka zawiera domyślnie tekst „???”. Klikając ten tekst można go zmienić na nazwę żądanej zmiennej. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Informacje na temat wprowadzania adresów, wyświetlania nazwy zmiennej w kilku wierszach lub dodawania własnych komentarzy dla styku znajdują się wyżej w części „Wstaw” „Styk”.

„Wstaw” „Cewka «Set»” w LD

Symbol:  Kombinacja klawiszy: <Ctrl> + <I>

Za pomocą tego polecenia można wstawić cewkę ustawiającą (patrz rozdz. 2.2.6). Otrzymywany efekt jest taki sam, jak przy użyciu poleceń „Wstaw” „Cewka” i „Dodatki” „Ustaw/Przywróć”, które w kombinacji dają również cewkę ustawiającą.

„Wstaw” „Cewka «Reset»” w LD

Symbol: 

Za pomocą tego polecenia można wstawić cewkę Reset (kasującą) (patrz rozdz. 2.2.6). Otrzymywany efekt jest taki sam, jak przy użyciu poleceń „Wstaw” „Cewka” i „Dodatki” „Ustaw/Przywróć”, które w kombinacji dają również cewkę Reset.

„Wstaw” „Blok funkcji” w LD

Symbol:  Kombinacja klawiszy: <Ctrl> +

To polecenie może być użyte, by blok funkcji lub program wstawić jako osobny moduł. W tym celu należy zaznaczyć połączenie między stykami i cewkami lub jedną cewkę. Okno dialogowe pomocy przy wprowadzaniu otwiera się, skąd można wybrać standardowe lub zdefiniowane przez użytkownika moduły.

Pierwsze wejście i wyjście nowego modułu są zajęte dla połączeń wejścia/wyjścia, dlatego zmienne te muszą być koniecznie typu BOOL. Wszystkie pozostałe wejścia i wyjścia modułu są zajęte tekstami „???”. W ich miejsce można wstawić stałe, zmienne lub adresy. Można w tym celu również użyć funkcji pomocy przy wpisywaniu.

Informacje na temat wprowadzania adresów, wyświetlania nazwy zmiennej w kilku wierszach lub dodawania własnych komentarzy dla bloku funkcji znajdują się w opisie opcji schematu funkcji oraz schematu styków.

Moduły z wejściami EN

Chcąc używać sieci LD do sterowania wywołaniami innych modułów, należy wtedy do schematu wstawić moduł z wejściem EN. Moduł taki jest podłączony równolegle do cewek. Począwszy od niego sieć można dalej rozbudowywać tak jak w Schemacie Bloków Funkcji FDB. Polecenia służące do rozbudowywania modułu EN znajdują się w punkcie menu „Wstaw” „Wklej przy modułach”.

Operator, blok funkcji, program lub funkcja z wejściem EN zachowuje się jak odpowiedni moduł w Schemacie Bloków Funkcji (FDB) z tą różnicą, że jego wykonaniem steruje wejście EN. To wejście jest przyłączone do linii połączenia między cewkami i stykami. Jeśli za pośrednictwem tej linii przekazywana jest informacja „TRUE”, wtedy moduł ten jest przetwarzany.

Jeśli utworzony został moduł z wejściem EN, można wtedy za pomocą tego modułu budować sieć jak w schemacie bloków funkcji. Oznacza to, że moduł EN może przyjmować dane ze zwykłych operatorów, funkcji, bloków funkcji oraz transportować te dane do takich samych zwykłych modułów.

Aby zaprogramować w edytorze LD sieć tak jak w FBD, wystarczy na początku wstawić do nowej sieci operator EN, a następnie począwszy od tego modułu rozbudowywać dalej sieć tak jak w edytorze FBD. Utworzona w ten sposób sieć zachowuje się tak samo jak sieć utworzona w FBD.

„Wstaw” „Moduł z EN” w LD

Symbol: 

Za pomocą tego polecenia można wstawić do sieci LD blok funkcji, operator, funkcję lub program z wejściem EN.

Zaznaczone miejsce wstawienia musi być połączeniem między stykami i cewkami lub cewką. Nowy moduł jest podłączany równolegle poniżej tych samych cewek i nosi domyślną nazwę „AND”. Nazwę domyślną można dowolnie zmienić. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Do dyspozycji są moduły standardowe i zdefiniowane przez użytkownika.

„Wstaw” „Wklej przy modułach” w LD

Za pomocą tego polecenia można do istniejącego już modułu (także modułu z wejściem EN), dodawać kolejne elementy. Polecenia dostępne w tym punkcie menu są wykonywalne na tych samych położeniach kursora jak odpowiadające im polecenia w schemacie bloków funkcji.

Polecenie **Wejście** dodaje nowe wejście do modułu.

Polecenie **Wyjście** dodaje nowe wyjście do modułu.

Polecenie **Moduł** dodaje kolejny moduł. Sposób postępowania jest identyczny, jak w opisie polecenia „Wstaw” „Moduł”.

Za pomocą polecenia **Przyporządkowanie** można wstawić przyporządkowanie dla zmiennej. Po wstawieniu jest ono pokazywane jako trzy znaki zapytania „???” , które można edytować i zastąpić dowolną zmienną. Dostępna jest przy tym funkcja pomocy przy wprowadzaniu.

Informacje na temat wprowadzania adresów, wyświetlania nazwy zmiennej w kilku wierszach lub dodawania własnych komentarzy dla styku znajdują się wyżej w części „Wstaw” „Styk”.

„Wstaw” „Wykrywanie narastającego zbocza”

Symbol: 

Za pomocą tego polecenia można wstawić blok funkcji R_TRIG, który służy do wykrywania wzrastającego zbocza (FALSE -> TRUE) przychodzącego sygnału (patrz też rozdz. 13.17.3). Zasada działania jest taka sama, jak w przypadku polecenia „Wstaw” „Blok funkcji”, za pomocą którego można również wybrać i wstawić taki sam moduł.

„Wstaw” „Wykrywanie opadającego zbocza”

Symbol: 

Za pomocą tego polecenia można wstawić blok funkcji F_TRIG, który służy do wykrywania opadającego zbocza (FALSE -> TRUE) przychodzącego sygnału (patrz też rozdz. 13.17.3). Zasada działania jest taka sama, jak w przypadku polecenia „Wstaw” „Blok funkcji”, za pomocą którego można również wybrać i wstawić taki sam moduł.

„Wstaw” „Timer (TON)”

Symbol: 

Za pomocą tego polecenia można wstawić blok funkcji Timer typu „TON”. Dzięki niemu można zrealizować opóźnienie załączania (opóźnione przesłanie sygnału wejścia) (patrz też rozdz. 13.17.5). Wstawianie odbywa się tak samo, jak w przypadku polecenia „Wstaw” „Blok funkcji”, za pomocą którego można również wybrać moduł TON.

„Wstaw” „Skok” w LD

Za pomocą tego polecenia można dodać skok, który zostanie wstawiony w edytorze LD równolegle na końcu poprzednich cewek. Jeśli przewód doprowadzający dostarcza wartość „TRUE”, wówczas skok zostanie wykonany do wskazanego znacznika.

Zaznaczone miejsce wstawienia musi być połączeniem między stykami i cewkami lub cewką.

Skok zawiera domyślnie tekst „Label”. Klikając ten tekst można go zmienić na żądany znacznik skoku.

„Wstaw” „Powrót” w LD

Za pomocą tego polecenia można dodać instrukcję RETURN, która zostanie wstawiona w edytorze LD równolegle na końcu poprzednich cewek. Jeśli przewód doprowadzający dostarcza wartość „Wł.”, wówczas przetwarzanie modułu w tej sieci zostanie przerwane.

Zaznaczone miejsce wstawienia musi być połączeniem między stykami i cewkami lub cewką.

„Dodatki” „Wklej za” w LD

Za pomocą tego polecenia można dodać zawartość schowka ze stykami szeregowymi, które zostaną wstawione w edytorze LD za miejscem zaznaczenia. To polecenie jest dostępne, jeśli zawartość schowka i zaznaczone miejsce są sieciami zbudowanymi ze styków.

„Dodatki” „Wklej pod” w LD

Za pomocą tego polecenia można dodać zawartość schowka ze stykami równoległymi, które zostaną wstawione w edytorze LD pod miejscem zaznaczenia. To polecenie jest dostępne, jeśli zawartość schowka i zaznaczone miejsce są sieciami zbudowanymi ze styków.

„Dodatki” „Wklej nad” w LD

Za pomocą tego polecenia można dodać zawartość schowka ze stykami równoległymi, które zostaną wstawione w edytorze LD nad miejscem zaznaczenia. To polecenie jest dostępne, jeśli zawartość schowka i zaznaczone miejsce są sieciami zbudowanymi ze styków.

„Dodatki” „Negacja” w LD

Symbol:  Kombinacja klawiszy: <Ctrl> + <N>

Za pomocą tego polecenia można zanegować styk, cewkę, instrukcję skoku lub RETURN albo wejście i wyjście modułów EN w aktualnym położeniu kursora.


Między okrągłymi nawiasami cewki albo między prostymi liniami styku pojawia się ukośnik (/) albo ||). W przypadku skoków, instrukcji Return, wejść i wyjść modułów EN pojawia się podobnie jak w edytorze FBD małe kółko na linii połączenia.

Cewka zapisuje teraz zanegowaną wartość połączenia na wejściu w przyporządkowanej zmiennej boolean. Zanegowany styk przełącza stan wejścia na wyjście dokładnie wtedy, gdy odpowiednia zmienna boolean dostarcza wartość FALSE.

Jeśli zaznaczony jest Skok lub Powrót, zanegowane zostanie wejście tego skoku wzgl. powrotu.

Negację można usunąć poprzez ponowną negację.

„Dodatki” „Ustaw/Przywróć” w LD

Symbol: 

Wykonanie tego polecenia dla cewki spowoduje, że będzie ona cewką ustawiającą. Cewka taka nigdy nie zastępuje wartości TRUE w przyporządkowanej zmiennej boolean. To znaczy, że jeśli dla zmiennej ustawiono raz wartość TRUE, pozostanie ona zawsze TRUE. Cewka ustawiająca jest rozpoznawalna po „S” w symbolu cewki.

Ponowne wykonanie tego polecenia spowoduje, że cewka stanie się cewką kasującą. Cewka taka nigdy nie zastępuje wartości FALSE w przyporządkowanej zmiennej boolean. To znaczy, że jeśli dla zmiennej ustawiono raz wartość FALSE, pozostanie ona zawsze FALSE. Cewka kasująca jest rozpoznawalna po „R” w symbolu cewki.

W przypadku wielokrotnego wykonania polecenia cewka przełącza się na cewkę ustawiającą, Reset (kasującą) i normalną cewkę.

Schemat styków w trybie online

W trybie online wszystkie styki i cewki pozostające w stanie włączonym (TRUE), są na schematach drabinkowych

przedstawione w kolorze **niebieskim**, również wszystkie przewody przesyłające stan „włączony” są przedstawione na niebiesko. Na wejściach i wyjściach bloków funkcji pokazywane są **wartości** odpowiednich zmiennych.

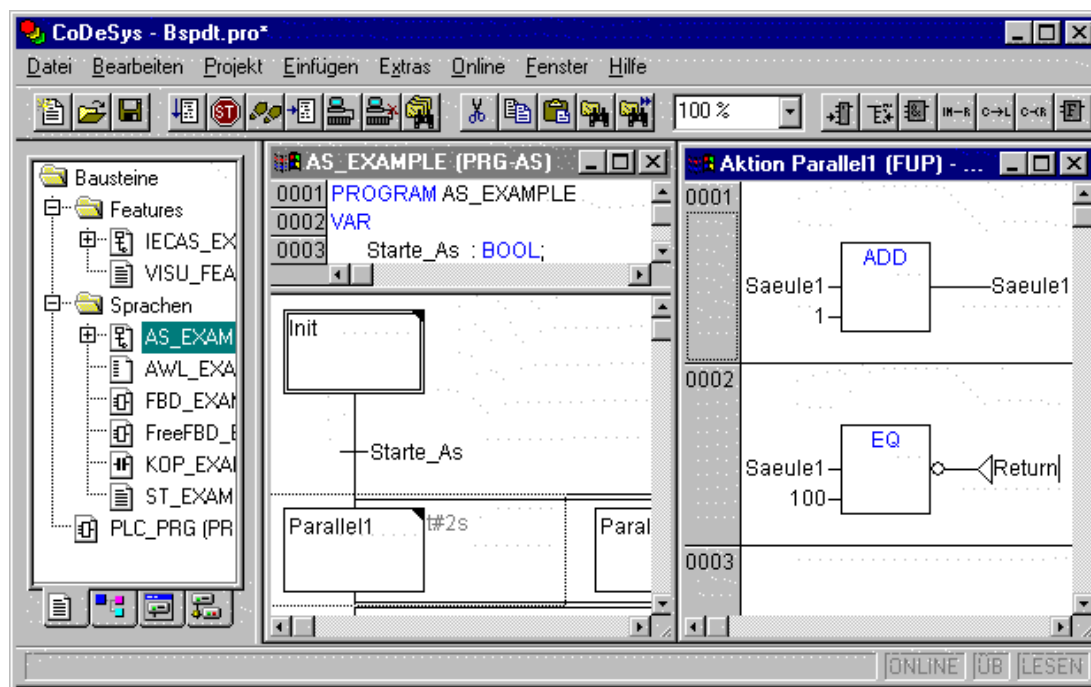
Punkty wstrzymania można ustawić tylko w sieciach; w przypadku **przetwarzania krokowego** następuje skok z jednej sieci do drugiej.

Przy włączonej **kontroli przebiegu** („Online” „Kontrola przebiegu”) pola z numerem aktualnie wykonywanych sieci są zaznaczone na zielono.

Przytrzymując kursor myszy przez krótki czas nad zmienną, można wyświetlić typ, adres oraz komentarz do zmiennej w oknie **etykiety**.

8.4.4 Edytor języka przebiegu...

Tak wygląda moduł napisany w SFC w odpowiednim edytorze CoDeSys:



Wszystkie edytory modułów składają się z części deklaracyjnej oraz ze szkieletu. Są one od siebie oddzielone separatorem ekranu.

Edytor języków przebiegu to edytor graficzny. Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy). Etykiety ekranowe pokazują użytkownikowi, czy to w trybie offline, czy online albo w stanie powiększenia pełne nazwy wzgl. wyrażenia dla kroków, przejścia, skoków, znaczników skoków, kwalifikatorów lub skojarzonych akcji.

Informacje na temat języka przebiegu, patrz rozdział 2.2.3, Język przebiegu (SFC).

Edytor języka przebiegu musi obsługiwać funkcje typowe dla SFC. Do tego celu służą następujące punkty menu:


Zaznacz blok

Zaznaczony blok składa się z kilku elementów SFC, otoczonych prostokątem narysowanym linią przerywaną.

Elementy (krok, przejście lub skok) można wybierać ustawiając kursor myszy na danym elemencie i klikając go lewym przyciskiem myszy lub używając klawiszy strzałki. By zaznaczyć grupę elementów, zaznacz blok i przytrzymując naciśnięty klawisz <Shift> wybierz element w lewym lub prawym rogu danej grupy. Dokonany wybór stanowi najmniejszą grupę wspólnie powiązanych elementów, która zawiera obydwie te elementy.


Krok można usunąć tylko z poprzedzającym lub następującym po nim przejściem!

„Wstaw” „Przejście kroku (przed)”

Symbol:  Kombinacja klawiszy: <Ctrl> + <T>

To polecenie dodaje w edytorze SFC krok i przejście, które zostaną wstawione przed zaznaczonym blokiem.

„Wstaw” „Przejście kroku (po)”


Symbol:  Kombinacja klawiszy: <Ctrl> + <E>

To polecenie dodaje w edytorze SFC krok i przejście, które zostaną wstawione za pierwszym przejściem w zaznaczonym bloku.

Usuwanie kroku i przejścia


Krok można usunąć tylko z poprzedzającym lub następującym po nim przejściem. W tym celu zaznacz krok i przejście, a następnie wykonaj polecenie „Edycja” „Usuń” albo naciśnij klawisz <Delete>.

„Wstaw” „Gałąź alternatywna (w prawo)”

Symbol:  Kombinacja klawiszy: <Ctrl> + <A>

To polecenie dodaje w edytorze SFC gałąź alternatywną jako prawe odgałęzienie zaznaczonego bloku. Zaznaczony blok musi rozpoczynać i kończyć się przejściem. Nowa gałąź składa się wtedy z jednego przejścia.

„Wstaw” „Gałąź alternatywna (z lewej)”

Symbol: 


To polecenie dodaje w edytorze SFC gałąź alternatywną jako lewe odgałęzienie zaznaczonego bloku. Zaznaczony blok musi rozpoczynać i kończyć się przejściem. Nowa gałąź składa się wtedy z jednego przejścia.

„Wstaw” „Gałąź równoległa (z prawej)”

Symbol:  Kombinacja klawiszy: <Ctrl> + <L>

To polecenie dodaje w edytorze SFC gałąź równoległą jako prawe odgałęzienie zaznaczonego bloku. Zaznaczony blok musi rozpoczynać i kończyć się krokiem. Nowa gałąź składa się wtedy z jednego kroku. Skoki do gałęzi równoległej będą możliwe po ustawieniu dla niej znacznika skoku.

„Wstaw” „Gałąź równoległa (z lewej)”

Symbol: 

To polecenie dodaje w edytorze SFC gałąź równoległą jako lewe odgałęzienie zaznaczonego bloku. Zaznaczony blok musi rozpoczynać i kończyć się krokiem. Nowa gałąź składa się wtedy z jednego kroku. Skoki do gałęzi równoległej będą możliwe po ustawieniu dla niej znacznika skoku.

„Wstaw” „Skok”

Symbol:  Kombinacja klawiszy: <Ctrl> + <U>

Za pomocą tego polecenia można w edytorze SFC dodać skok na końcu odgałęzienia, do którego należy zaznaczony blok. Odgałęzienie takie musi być gałęzią alternatywną.

Dla wstawionego skoku można następnie zaznaczyć wprowadzony tekst „Skok” i zastąpić go dowolną nazwą skoku albo znacznikiem skoku jednej lub kilku gałęzi równoległych, do których powinno nastąpić przejście.

„Wstaw” „Skok przejścia”

Symbol: 

Za pomocą tego polecenia menu można wstawić w edytorze SFC przejście z następującym po nim skokiem na końcu wybranego odgałęzienia. Odgałęzienie takie musi być gałęzią równoległą.

Dla wstawionego skoku można następnie zaznaczyć wprowadzony tekst „Skok” i zastąpić go dowolną nazwą skoku albo znacznikiem skoku gałęzi równoległej, do której powinno nastąpić przejście.

„Wstaw” „Dodaj akcję wejściową”

Za pomocą tego polecenia można dla wybranego kroku dodać akcję wejściową. Akcja wejściowa wykonywana jest tylko jeden raz, bezpośrednio po aktywowaniu kroku. Akcja wejściowa może być implementowana w dowolnym języku.

Krok z akcją wejściową jest zaznaczony literą „E” w lewym dolnym rogu.

„Wstaw” „Dodaj akcję wyjściową”

Za pomocą tego polecenia można dla wybranego kroku dodać akcję wyjściową. Akcja wyjściowa wykonywana jest tylko jeden raz, zanim krok zostanie dezaktywowany. Akcja wyjściowa może być implementowana w dowolnym języku.

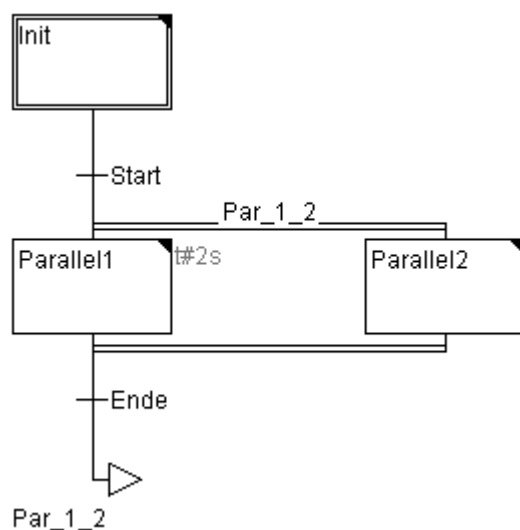
Krok z akcją wyjściową jest zaznaczony literą „X” w prawym dolnym rogu.

„Wstaw” „Dodaj gałąź równoległą (z prawej)”

To polecenie wstawia zawartość schowka jako prawą gałąź równoległą zaznaczonego bloku. Zaznaczony blok musi rozpoczynać i kończyć się krokiem. Zawartość schowka musi być również blokiem SFC rozpoczynającym i kończącym się krokiem.

„Wstaw” „Dodaj znacznik do gałęzi równoległej”

Aby przypisać dla nowej gałęzi równoległej znacznik skoku, należy zaznaczyć przejście występujące przed gałęzią równoległą i wykonać polecenie „Dodaj znacznik do gałęzi równoległej”. Gałąź równoległa otrzymuje nazwę domyślną „Równoległa” i bieżący numer, które można edytować zgodnie z zasadami nazewnictwa. Na przedstawionym niżej przykładzie nazwa „Równoległa” została zastąpiona przez „Par_1_2”, a skok występujący za przejściem „Koniec” skierowany na znacznik skoku.



Usuń znacznik skoku

Znacznik skoku można usunąć poprzez usunięcie jego opisu.

„Dodatki” „Wklej po”

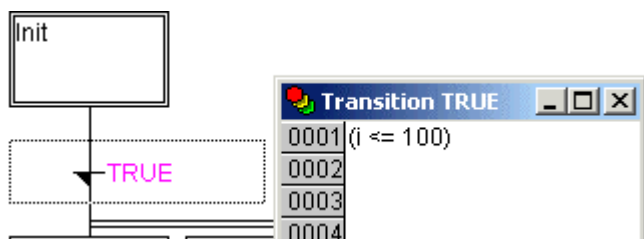
Za pomocą tego polecenia można przenieść do schowka blok SFC występujący za pierwszym krokiem albo za pierwszym przejściem zaznaczonego bloku (normalne kopiowanie wstawia go przed zaznaczonym blokiem). Polecenie zostanie wykonane tylko wtedy, gdy wynikowa struktura SFC jest zgodna z normami języka.

„Dodatki” „Powiększ akcję/przejście”

Kombinacja klawiszy: <Alt> + <Enter>

Akcja pierwszego kroku zaznaczonego bloku wzgl. szkieletu pierwszego przejścia zaznaczonego bloku zostanie załadowana do edytora w języku, w którym została napisana. Jeśli akcja lub szkielet przejścia są puste, wówczas należy wybrać język, w jakim zostaną one napisane.

W przypadku przejść należy pamiętać, że napisany w edytorze warunek ma pierwszeństwo przed znacznikiem przejścia lub warunkiem przejścia napisanym bezpośrednio do znacznika. Przykład: Jeśli $i > 100$, wtedy warunek przejścia jest w stanie FALSE, pomimo że znacznik jest TRUE!



„Dodatki” „Usuń akcję/przejście”

Za pomocą tego polecenia można usuwać akcje pierwszego kroku zaznaczonego bloku, ewentualnie pierwsze przejście zaznaczonego bloku.

Jeśli w danym kroku została zaimplementowana tylko i wyłącznie akcja zwykła, albo akcja wejściowa, albo akcja wyjściowa, wówczas wykonanie polecenia spowoduje jej usunięcie. W przeciwnym razie zostanie wyświetlone okno dialogowe, w którym należy wybrać akcje do usunięcia.

Jeśli kursor znajduje się na akcji należącej do kroku IEC, usunięte zostanie tylko to skojarzenie. Jeśli zaznaczony zostanie krok IEC ze skojarzoną akcją, usunięte zostanie tylko to skojarzenie. Jeśli natomiast krok IEC zawiera więcej akcji, wówczas wyświetlone zostanie okno dialogowe wyboru.

„Dodatki” „Atrybuty kroków”

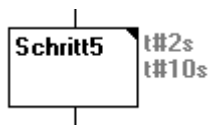
To polecenie otwiera okno dialogowe, w którym można edytować atrybuty zaznaczonego kroku.

W oknie dialogowym atrybutów kroku można wprowadzić trzy różne wpisy. W polu **Min. czas** wprowadź minimalny czas przetwarzania danego kroku. W polu **Maks. czas** wprowadź maksymalny czas przetwarzania danego kroku. Należy pamiętać, że wprowadzenia muszą być typu **TIME**, tzn. w polach tych można wprowadzać stałe typu TIME (np. T#3s) lub zmienne typu TIME.

W polu **Komentarz** można wprowadzić komentarz dla kroku. W oknie dialogowym „Opcje języków przebiegu”, otwieranym poleceniem „Dodatki” „Opcje”, można wprowadzić ustawienie, czy w edytorze SFC mają być wyświetlane komentarze i ustawione czasy przetwarzania dla kroków. Komentarz lub czasu przebiegu będzie pokazywany wówczas z prawej strony danego kroku.

Okno dialogowe edycji atrybutów kroku

W razie przekroczenia maksymalnego czasu przetwarzania ustawiane są flagi SFC, których użytkownik może używać do wysyłania kwerend.



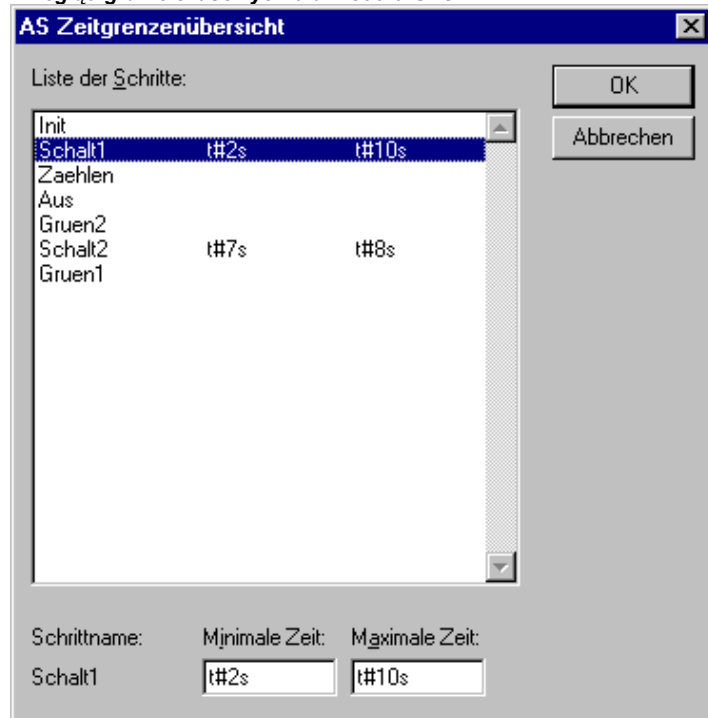
W przykładzie przedstawiono krok, którego wykonanie powinno trwać minimum 2 i maksymalnie 10 sekund. W trybie online dodatkowo pokazywana jest informacja, ile czasu dany krok jest już aktywny.

„Dodatki” „Przegląd czasów”

To polecenie otwiera okno dialogowe, w którym można edytować ustawienia czasów kroków SFC.

W przeglądzie granic czasowych wyświetlane są wszystkie kroki modułu SFC. Ustawione ograniczenia czasu przetwarzania dla kroków są pokazywane z prawej strony danego kroku (najpierw dolna, a potem górna wartość ograniczenia). Ponadto można edytować ograniczenia czasu. W tym celu kliknij w oknie przeglądu żądany krok. **Nazwa kroku** pokazywana jest u dołu okna, następnie przejdź do pól **Min. czas** lub **Maks. czas** i wprowadź tam odpowiednie ograniczenie czasu. Należy pamiętać, że wprowadzenia muszą być typu **TIME**, tzn. w polach tych można wprowadzać stałe typu TIME (np. T#3s) lub zmienne typu TIME. Po zamknięciu okna przyciskiem **OK** wprowadzone zmiany zostaną zapisane.

Przegląd granic czasowych dla modułu SFC

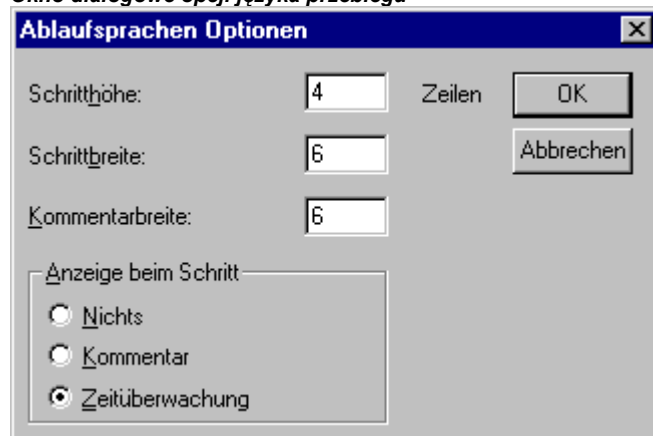


Przykład przedstawia kroki 2 i 6, dla których ustawiono ograniczenie czasowe. Schalt1 może trwać minimalnie 2 i maksymalnie 10 sekund. Schalt2 może trwać minimalnie 7 i maksymalnie 8 sekund.

„Dodatki” „Opcje”

To polecenie otwiera okno dialogowe, w którym można ustawić różne opcje dla modułu SFC.

Okno dialogowe opcji języka przebiegu



W oknie dialogowym opcji języków przebiegu można wprowadzić pięć różnych wpisów. W polu

Wys. kroku można wprowadzić, ile wierszy w edytorze SFC ma zajmować jeden krok SFC. Ustawieniem domyślnym jest 4. W polu **Szer. kroku** można wprowadzić, ile kolumn może zajmować jeden krok. Ustawieniem domyślnym jest 6. **Szer. komentarza** definiuje ilość kolumn, które będą wyświetlane dla kroków z komentarzem.

W polu **Pokaż podczas kroku** można ustawić, które wprowadzenia z okna „Dodatki” „Atrybuty kroku” mają być wyświetlane. Można też wybrać opcję **Nic**, **Komentarz** lub **Monitorowanie czasu**.

„Dodatki” „Powiązanie akcji”

Za pomocą tego polecenia można kojarzyć akcje i zmienne boolean z krokami IEC.

Z prawej strony kroku IEC został dodany kolejny dwuczęściowy prostokąt służący do kojarzenia akcji. Domyślnie w lewym polu znajduje się kwalifikator „N” i nazwa „Akcja”. Obydwa te ustawienia można zmienić. Można w tym celu również użyć funkcji pomocy przy wpisywaniu.

Dla jednego kroku IEC może być przypisanych maksymalnie dziewięć akcji!

Nową akcją dla kroków IEC można utworzyć w Organizерze obiektów dla określonego modułu SFC za pomocą polecenia „Projekt” „Dodaj akcję”.

„Dodatki” „Zastosuj kroki IEC”

Symbol: 

Jeśli polecenie jest aktywne (rozpoznawalne po zaznaczeniu w punkcie menu i wciśniętym przycisku na pasku narzędzi), wówczas każde wstawienie przejścia kroku i gałęzi równoległych spowoduje, że zamiast uproszczonych kroków będą dodawane kroki IEC.

Jeśli ta opcja jest wybrana, wówczas przy tworzeniu nowego modułu SFC krok Inicjacji będzie typu IEC.

Ustawienie to jest zapisywane w pliku „CoDeSys.ini” i odtwarzane wraz z każdym kolejnym uruchomieniem programu CoDeSys.

Język przebiegu w trybie online

W edytorze języków przebiegu aktualnie aktywne kroki w trybie online są wyświetlane na niebiesko. W zależności od ustawienia w oknie „Dodatki” „Opcje” obok kroków wyświetla się również kontrola czasu, czyli podana przez użytkownika dolna i górna granica czasu przetwarzania oraz dodatkowo informacja, jak długo dany krok jest aktywny.



Na powyższym przykładzie pokazany został krok, który jest aktywny od 8 sekund i 410 milisekund. Musi on jednak trwać minimum 7 minut, aby możliwe było jego opuszczenie.

Za pomocą polecenia „Online” „Wstaw/usuń punkt wstrzymania” można ustawić dla kroku punkt wstrzymania, a również dla akcji w miejscach dopuszczonych w zależności od stosowanego języka. Przetwarzanie zatrzyma się przed wykonaniem tego kroku albo przed lokalizacją danej akcji w programie. Kroki i miejsca w programie, w których ustawiono punkty wstrzymania, są zaznaczone kolorem jasnoniebieskim.

Jeśli w gałęzi równoległej istnieje kilka aktywnych kroków, wówczas aktywny krok, którego akcja ma być przetwarzana jako następna, będzie wyświetlany na czerwono.

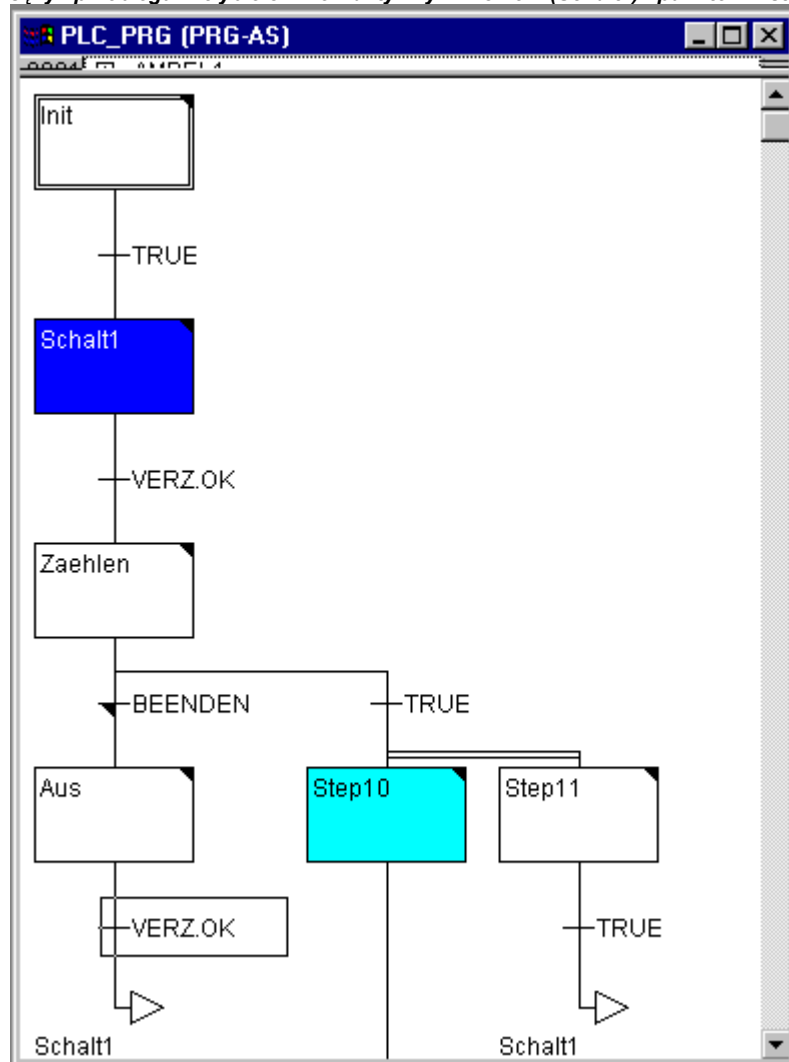
Jeśli zostały użyte kroki IEC, wszystkie aktywne akcje w trybie online będą wyświetlane na niebiesko.

Język SFC obsługuje również przetwarzanie krokowe:

Za pomocą polecenia „Online” „Krok pojedynczy przez” można w każdym momencie przejść do następnego kroku, którego akcja jest wykonywana. Jeśli bieżącą lokalizacją jest

- krok z liniowym przebiegiem modułu lub krok z najbardziej na prawo położoną gałęzią równoległą modułu, wówczas nastąpi opuszczenie modułu SFC i powrót do elementu wywołującego. Jeśli moduł jest programem głównym, rozpoczyna się następny cykl.
- krok nie znajdujący się w najbardziej na prawo położonej gałęzi odgałęzienia równoległego, wówczas następuje przeskok do aktywnego skoku w następnej gałęzi równoległej.
- ostatnim punktem wstrzymania w obrębie akcji 3S, wówczas następuje przeskok do elementu wywołującego SFC.
- ostatnim punktem wstrzymania w obrębie akcji IEC, wówczas następuje przeskok do elementu wywołującego SFC.
- ostatnim punktem wstrzymania w obrębie akcji wejściowej/wyjściowej, wówczas następuje przeskok do pierwszego aktywnego kroku.

Język przebiegu w trybie online z aktywnym krokiem (Schalt1) i punktem wstrzymania (Step10)



Za pomocą polecenia „Online” „Krok pojedynczy w” można dodatkowo wykonywać sekwencje krokowe w akcjach. Aby dostać się do akcji wejściowej, wyjściowej lub akcji IEC, należy ustawić tam punkt wstrzymania. W obrębie akcji użytkownik może korzystać ze wszystkich funkcji debugowania danego edytora.

Przytrzymując kursor myszy przez krótki czas nad zmienną w edytorze deklaracji, można wyświetlić typ, adres oraz komentarz do zmiennej w oknie **etykiety**.

Należy pamiętać: W przypadku dokonania zmiany nazwy kroku online, w czasie gdy ten krok jest aktywny, program zostanie zatrzymany w stanie nieokreślonym!

Kolejność przetwarzania elementów **sekwencji kroków**:

1. Na początku resetowane są wszystkie flagi kontrolne akcji IEC wykorzystywane w tej sekwencji kroków. (Jednak nie flagi akcji IEC, które są wywoływane w obrębie akcji).
2. Dla wszystkich kroków ujętych w sekwencji sprawdzany jest kolejno (z góry na dół i z lewej do prawej) warunek wykonania akcji wyjściowej. Jeśli warunek jest spełniony akcja wyjściowa jest wykonywana.
3. Dla wszystkich kroków ujętych w sekwencji sprawdzany jest kolejno warunek wykonania akcji wejściowej. Jeśli warunek jest spełniony akcja wejściowa jest wykonywana.

Dla wszystkich kroków ujętych w sekwencji wykonywane jest kolejno następujące działanie:

- kopiowanie czasu przetwarzania w zmiennej przyporządkowanej dla kroku.
- sprawdzenie, czy nie nastąpiło przekroczenie czasu przetwarzania i ustawienie w takim przypadku flagi SFC-Error.

- dla kroków nie będących krokami IEC wykonanie tylko przyporządkowanej akcji.
- 4. Akcje IEC, wykorzystywane w sekwencjach kroków, są wykonywane w kolejności alfabetycznej, przy czym każda lista akcji wykonuje się w dwóch przebiegach. W pierwszym przebiegu wykonywane są wszystkie akcje IEC pozostające w bieżącym cyklu w stanie nieaktywnym. W drugim przebiegu wykonywane są wszystkie akcje IEC pozostające w bieżącym cyklu w stanie aktywnym.
- 5. Warunki przejścia są sprawdzane: Jeśli krok w bieżącym cyklu był aktywny i następne przejście dostarcza wartość TRUE (i ewentualnie minimalny czas aktywności już upłynął), wówczas następny krok jest aktywowany.

Przy implementowaniu akcji należy przestrzegać:

Może się zdarzyć, że jedna akcja zostanie w danym cyklu wykonana wielokrotnie, gdyż jest skojarzona z kilkoma sekwencjami kroków. (Na przykład jedna sekwencja może mieć dwie akcje IEC A i B, obie są zaimplementowane w SFC, i obie wywołują akcję IEC C, wtedy w tym samym cyklu obie akcje A i B mogą być aktywne i w obu akcjach może być aktywna również akcja C; w takim przypadku C jest wywołana podwójnie).

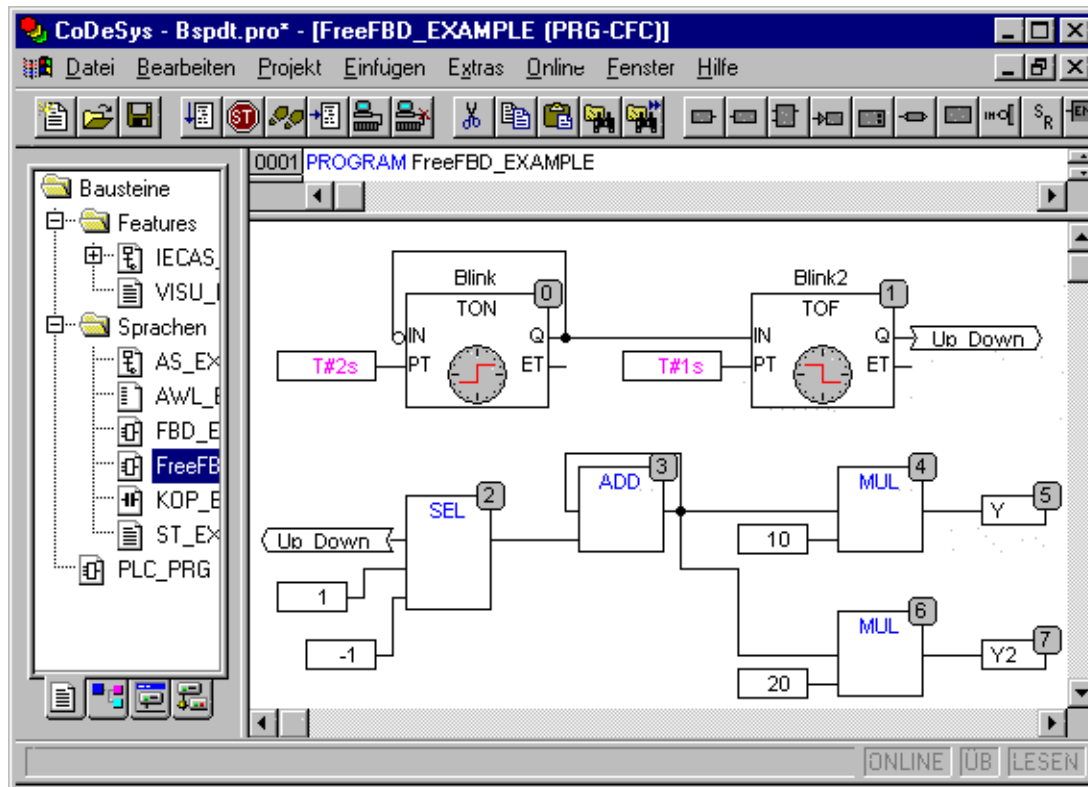
Jeśli ta sama akcja IEC jest wykorzystywana jednocześnie na kilku poziomach sekwencji, może to w opisaney wyżej sytuacji prowadzić do niepożądanych zdarzeń. Z tego powodu takie przypadki są sygnalizowane komunikatem o błędzie. Takie zdarzenia mogą wystąpić przy pracy z projektami utworzonymi w starszych wersjach CoDeSys!

Wskazówka: Monitoring wyrażeń (np. A AND B) w przejściach wyświetla tylko „wartość łączną” przejścia.

8.4.5 Graficzny edytor schematów funkcji (CFC)...

W graficznym edytorze schematów funkcji elementy nie są zestawiane w sieci, lecz mogą być dowolnie rozmieszczane. Elementami na liście do przetwarzania mogą być moduły, wejście wyjście, skok, etykieta, powrót i komentarz. Wejścia i wyjścia tych elementów można łączyć poprzez przeciąganie połączenia za pomocą myszy. Linia połączenia jest rysowana automatycznie. Przy uwzględnieniu istniejących połączeń program rysuje najkrótszą linię połączenia. Podczas przesuwania elementów linie połączenia są automatycznie optymalizowane. Jeśli linia połączenia z powodu braku miejsca nie może być narysowana, wówczas między wejściem a przynależącym wyjściem wyświetlana jest linia czerwona. Gdy tylko zwolni się miejsce, linia ta zostanie przekształcona w linię połączenia.

Przykładowy moduł utworzony w graficznym edytorze schematów funkcji (CFC):



Edytor graficzny ma tę przewagę nad zwyczajnym edytorem bloków funkcji FBD, że umożliwia bezpośrednie dodawanie sprzężeń zwrotnych.

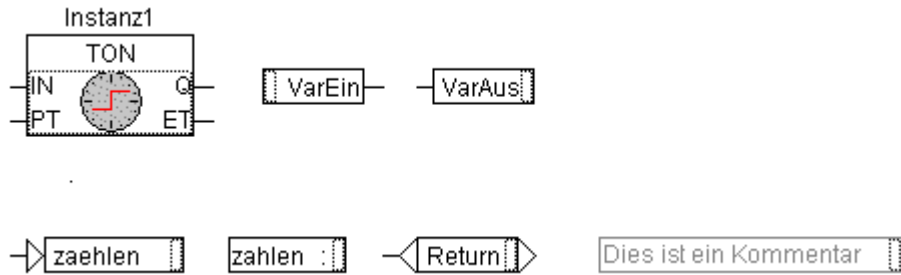
Najważniejsze polecenia znajdują się w menu podręcznym.

Pozycje kursora

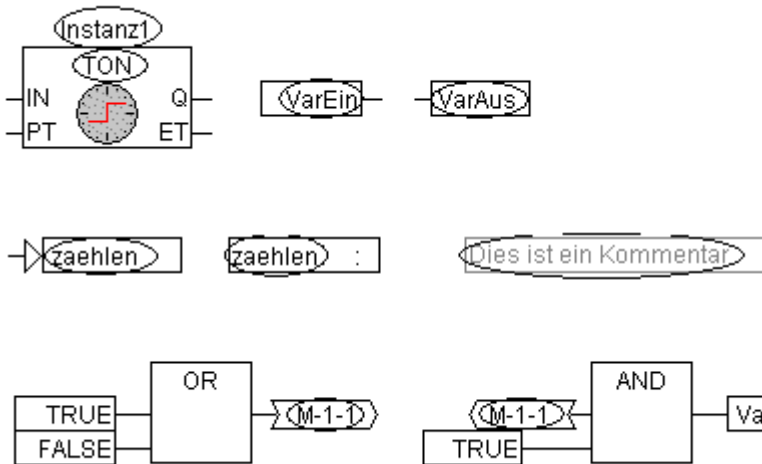
Każdy tekst to możliwa pozycja kursora. Wybrany tekst jest podświetlony na niebiesko i można go zmienić.

W przeciwnym razie pozycję kursora wskazuje kropkowany prostokąt. Poniżej znajduje się wyliczenie wszystkich możliwych pozycji kursora wraz z przykładami:

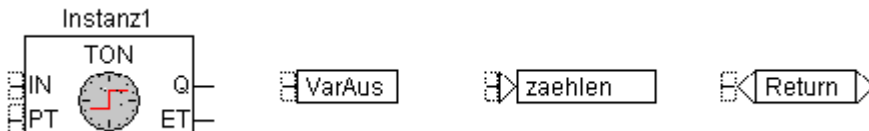
1. Szkielety elementów modułu, wejście, wyjście, skok, etykieta, powrót i komentarz:



2. Pola tekstowe elementów modułu, wejście, wyjście, skok, etykieta i komentarz, dalej pola tekstowe znaczników połączenia:




3. Wejścia elementów modułu, wyjście, skok i powrót:



4. Wyjścia elementów modułu i wejście:




„Wstaw” „Moduł”

Symbol:  Kombinacja klawiszy: <Ctrl> +

Za pomocą tego polecenia można wstawiać operatory, funkcje, bloki funkcji oraz programy. Najpierw wstawiany jest zawsze operator „AND”. Zaznaczając go i zmieniając jego nazwę można przekształcić go w każdy inny operator, w każdą funkcję, w każdy blok funkcji oraz w każdy program. Przy użyciu funkcji Asystenta deklaracji (pomocy) można wybrać żądany moduł z listy obsługiwanych modułów. Jeśli nowy moduł ma inną minimalną liczbę wejść, zostaną one dołączone. Jeśli nowy moduł ma mniejszą maksymalną liczbę wejść, ostatnie wejścia zostaną usunięte.

„Wstaw” „Wejście”

Symbol:  Kombinacja klawiszy: <Ctrl> + <E>

Za pomocą tego polecenia można wstawić wejście. Wpisany tekst „???” można zaznaczyć i zastąpić go nazwą zmiennej lub stałej. Można w tym celu również użyć funkcji pomocy przy wpisywaniu.

„Wstaw” „Wyjście”

Symbol:  Kombinacja klawiszy: <Ctrl> + <A>

Za pomocą tego polecenia można wstawić wyjście. Wpisany tekst „???” można zaznaczyć i zastąpić go nazwą zmiennej. Można w tym celu również użyć funkcji pomocy przy wpisywaniu. Wartość na wejściu tego wyjścia zostanie przypisana tej zmiennej.

„Wstaw” „Skok”

Symbol:  Kombinacja klawiszy: <Ctrl> + <J>

Za pomocą tego polecenia można wstawić skok. Wpisany tekst „???” można zaznaczyć i zastąpić go znacznikiem skoku, do którego powinno nastąpić przejście.

Znacznik skoku można wstawić za pomocą polecenia „Wstaw” „Znacznik”.

„Wstaw” „Znacznik”

Symbol:  Kombinacja klawiszy: <Ctrl> + <L>

Za pomocą tego polecenia można wstawić znacznik skoku. Wpisany tekst „???” można zaznaczyć i zastąpić go nazwą znacznika skoku. W trybie online dodawana jest automatycznie etykieta RETURN do zaznaczenia końca modułu.


Skok można wstawić za pomocą polecenia „Wstaw” „Skok”.

„Wstaw” „Powrót”

Symbol:  Kombinacja klawiszy: <Ctrl> + <R>

Za pomocą tego polecenia można wstawić instrukcję RETURN. Należy pamiętać, że w trybie online dodawany jest automatycznie znacznik skoku z nazwą RETURN w pierwszej kolumnie i za ostatnim wierszem w edytorze. Przeskok do znacznika następuje przy przetwarzaniu krokowym przed opuszczeniem modułu.

„Wstaw” „Komentarz”

Symbol:  Kombinacja klawiszy: <Ctrl> + <K>

Za pomocą tego polecenia można wstawić komentarz.

Nowy wiersz komentarza można dodać za pomocą kombinacji <Ctrl> + <Enter>.

„Wstaw” „Wejście modułu”

Kombinacja klawiszy: <Ctrl> + <U>

Polecenie to powoduje wstawienie wejścia modułu. Liczba wejść jest w przypadku wielu operatorów zmienna (np. ADD może posiadać 2 lub więcej wejść).


Aby rozszerzyć taki operator o jedno wejście, operator musi być sam (pozycja kursora 1) zaznaczony.

„Wstaw” „Pin wejścia”, „Wstaw” „Pin wyjścia”

Symbol:  

Polecenia te są dostępne, jeśli otwarte jest makro do edycji. Służą one do dodawania pinów wejścia lub wyjścia dla wejść i wyjść makra. Różnią się one od normalnych wejść i wyjść modułów graficzną formą przedstawienia i tym, że nie posiadają indeksu pozycji.

„Dodatki” „Negowanie”

Symbol:  Kombinacja klawiszy: <Ctrl> + <N>


Za pomocą tego polecenia można zanegować wejścia, wyjścia, skoki lub instrukcje RETURN. Symbol negacji to małe kółko na połączeniu.

Jeśli zaznaczone jest wejście modułu, wyjścia, skoku, lub powrotu (pozycja kursora 3), wówczas to wejście zostanie zanegowane.

Jeśli zaznaczone jest wyjście modułu lub wejścia (pozycja kursora 4), wówczas to wyjście zostanie zanegowane.

Negację można usunąć poprzez ponowną negację.

„Dodatki” „Ustaw/Przywróć”

Symbol:  Kombinacja klawiszy: <Ctrl> + <T>

To polecenie można wykonać tylko dla zaznaczonych wejść elementów wyjście (pozycja kursora 3).

Symbolem dla Ustaw (Set) jest S, a dla Przywróć (Reset) jest R.



Zmienna VarOut1 przyjmuje wartość TRUE, jeśli zmienna VarIn1 dostarcza wartość TRUE. Zmienna VarOut1 przechowuje tę wartość, nawet jeśli VarIn1 ponownie powróci do stanu FALSE.

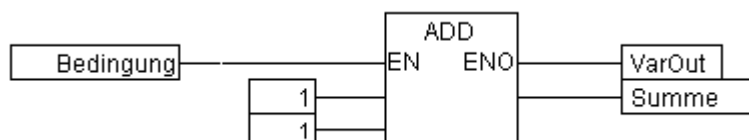
Zmienna VarOut2 przyjmuje wartość FALSE, jeśli zmienna VarIn2 dostarcza wartość TRUE. Zmienna VarOut2 przechowuje tę wartość, nawet jeśli VarIn2 ponownie powróci do stanu FALSE.

W przypadku wielokrotnego wykonania polecenia wyjście przełącza się pomiędzy Set, Reset i zwykłym wyjściem.

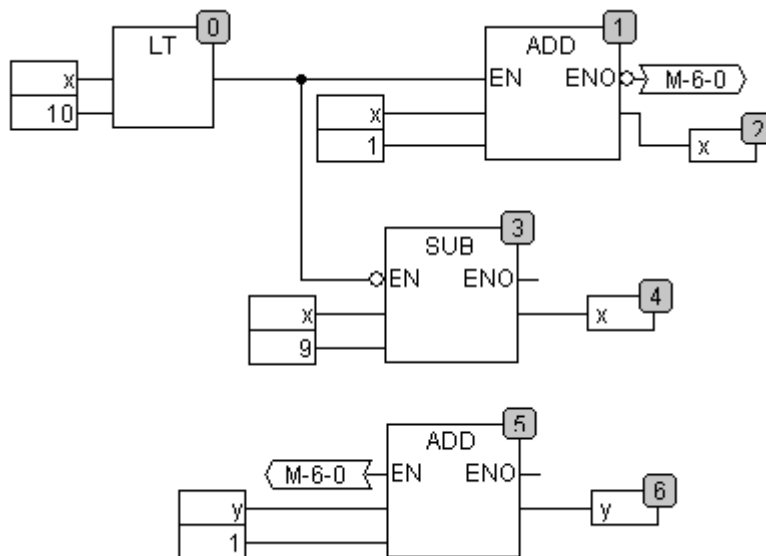
„Dodatki” „EN/ENO”

Symbol:  Kombinacja klawiszy: <Ctrl> + <O>

Za pomocą tego polecenia można dla zaznaczonego modułu (pozycja kursora 3) dodać dodatkowe wejście zwalniające EN (Enable In) typu boolean i wyjście ENO (Enable Out) również typu boolean.



W przedstawionym przykładzie ADD zostanie wykonany tylko wtedy, gdy zmienna boolean „warunek” ma wartość TRUE. Wtedy VarOut po wykonaniu ADD zostanie ustawiona również na TRUE. Jeśli jednak zmienna „warunek” zmieni przy tym wartość na FALSE, wówczas ADD nie będzie więcej przetwarzany, a VarOut zachowa wartość TRUE! Poniżej przedstawiono przykład wykorzystania wartości zmiennej ENO w innych modułach.



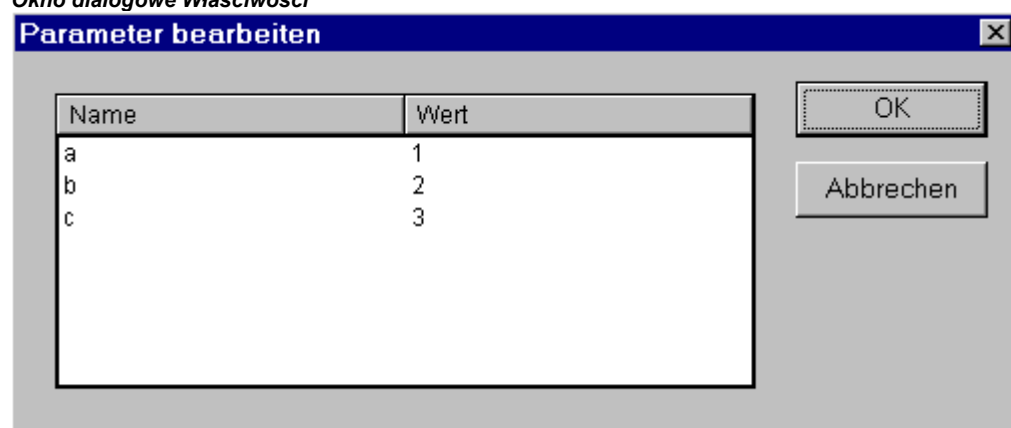
x ma być zainicjowana wartością 1, a y wartością 0. Numery w prawym rogu modułów podają numer w kolejce do przetwarzania.

x będzie zwiększany o jeden tak długo, aż jego wartość będzie równa 10. Spowoduje to, że wyjście modułu LT(0) dostarczy wartość FALSE i moduły SUB(3) i ADD(5) zostaną wykonane. Wtedy x zostanie ustawiona z powrotem na 1, a y zostanie zwiększona o 1. Następnie LT(0) jest wykonywany ponownie tak długo, jak długo wartość x będzie mniejsza niż 10. y zlicza, jak często x przyjmuje wartości od 1 do 10.

„Dodatki” „Właściwości...”

W graficznym edytorze schematów funkcji stałe parametry wejścia (VAR_INPUT CONSTANT) dla funkcji i bloków funkcji nie są pokazywane bezpośrednio. Można je wyświetlić i zmienić ich wartość zaznaczając szkielet odpowiedniego modułu (pozycja kursora 1) i wybierając polecenie „Dodatki” „Właściwości...” lub po prostu klikając dwukrotnie szkielet. Otwiera się okno dialogowe Edycja parametrów:

Okno dialogowe Właściwości



Jeśli w kolumnie Wartość zaznaczona jest wartość parametru, można ją zmienić po ponownym jej kliknięciu albo naciśnięciu klawisza spacji. Zmienioną wartość można zaakceptować naciśnięciem klawisza <Enter>; naciśnięcie klawisza <Escape> spowoduje anulowanie wprowadzonych zmian. Za pomocą przycisku **OK** wszystkie zmiany zostaną zapisane.

Wskazówka: Ta funkcjonalność i tym samym deklaracja zawierająca słowo kluczowe „VAR_INPUT CONSTANT” ma znaczenie tylko dla edytora CFC. W edytorze FBD wszystkie zmienne wejściowe są zawsze pokazywane na module, nie ma znaczenia, czy deklaracja będzie zmienną VAR_INPUT albo VAR_INPUT CONSTANT. Nie ma to również znaczenia w edytorach tekstowych. VAR_INPUT CONSTANT może być skuteczna tylko wtedy, gdy jest używana w edytorze CFC.

Zaznaczanie elementów

Aby zaznaczyć element, należy kliknąć myszą szkielet elementu (pozycja kursora 1).

Aby zaznaczyć więcej elementów, przytrzymaj klawisz <Shift> i kliknij myszą po kolei odpowiednie elementy lub przytrzymując lewy przycisk myszy, rozciągnij obszar nad elementami, które chcesz zaznaczyć.

Za pomocą polecenia „Dodatki” „Zaznacz wszystko” można zaznaczyć wszystkie elementy.

Przesuwanie elementów

Przytrzymując naciśnięty klawisz <Shift> i używając klawiszy strzałki można przesunąć jeden lub kilka zaznaczonych elementów. Elementy można również przesuwać z przytrzymanym lewym przyciskiem myszy. Po zwolnieniu lewego przycisku myszy elementy zostaną przeniesione w inne miejsce, o ile nie zasłaniają innych elementów lub nie powodują przekroczenia dopuszczalnego obszaru edytora. W takim przypadku elementy powrócą do swego pierwotnego miejsca i rozlega się dźwięk ostrzeżenia.

Kopiowanie elementów

Jeden lub kilka zaznaczonych elementów można skopiować za pomocą polecenia „Edytuj” „Kopiuj” i wkleić za pomocą polecenia „Edytuj” „Wklej”.

Tworzenie połączeń

Jedno wejście elementu może być połączone z tylko jednym wyjściem innego elementu. Jedno wyjście elementu może być połączone z kilkoma wejściami innych elementów.

Istnieje wiele możliwości połączenia wejścia elementu E2 z wyjściem elementu E1.



Kliknij lewym przyciskiem myszy wyjście elementu E1 (pozycja kursora 4), przytrzymaj naciśnięty lewy przycisk myszy, przeciągnij wskaźnik myszy na wejście elementu E2 (pozycja kursora 3) i zwolnij tam lewy przycisk myszy. Podczas przeciągania za pomocą myszy rysuje się linia połączenia od wyjścia elementu E1 do wskaźnika myszy.

Kliknij lewym przyciskiem myszy wejście elementu E2, przytrzymaj naciśnięty lewy przycisk myszy, przeciągnij wskaźnik myszy na wyjście elementu E1 i zwolnij tam lewy przycisk myszy.

Przesuń jeden z elementów E1 lub E2 (pozycja kursora 1) i zwolnij lewy przycisk myszy w takim miejscu, w którym wyjście elementu E2 i wejście elementu E1 poruszają się.

Jeśli element E2 jest modulem z wolnym wejściem, można za pomocą myszy przeciągnąć połączenie od wyjścia E1 do szkieletu E2. Po zwolnieniu przycisku myszy zostanie utworzone połączenie z najwyższym wolnym wejściem E2. Jeśli moduł E2 nie posiada wolnego wejścia lecz operator, który można rozbudować o dodatkowe wejście, wtedy nowe wejście zostanie utworzone automatycznie.

Za pomocą tych metod można również łączyć ze sobą wyjście i wejście tego samego modułu (sprzężenie zwrotne). Aby utworzyć połączenie między dwoma pinami, kliknij lewym przyciskiem myszy jeden pin, przytrzymaj naciśnięty przycisk i przeciągnij połączenie do drugiego pinu, a następnie zwolnij przycisk. Jeśli podczas przeciągania połączenia przekroczony zostanie obszar roboczy edytora, ekran zostanie automatycznie przewinięty. Podczas tworzenia połączenia odbywa się sprawdzanie typów dla prostych typów danych. Jeśli typy obydwu pinów nie są kompatybilne,

wówczas w miejscu kursora wyświetla się komunikat „zabronione”. Dla kompletnych typów danych sprawdzanie typów nie odbywa się.

Zmianianie połączeń

Połączenie między wyjściem elementu E1 a wejściem elementu E2 można w prosty sposób zmienić na połączenie między wyjściem elementu E1 a wejściem elementu E3. W tym celu kliknij myszą wejście E2 (pozycja kursora 3), przytrzymaj naciśnięty lewy przycisk myszy, przesuń kursor myszy na wejście E3 i zwolnij przycisk.

Usuwanie połączeń

Istnieje wiele możliwości usuwania połączenia między wyjściem elementu E1 a wejściem elementu E2:

Zaznacz wyjście E1 (pozycja kursora 4) i naciśnij klawisz <Delete> lub wybierz polecenie „**Edytuj**” „**Usuń**”. Jeśli wyjście E1 jest połączone z kilkoma wejściami, wówczas usunięte zostaną wszystkie te połączenia.

Zaznacz wejście E2 (pozycja kursora 4) i naciśnij klawisz <Delete> lub wybierz polecenie „**Edytuj**” „**Usuń**”.

Za pomocą myszy zaznacz wejście E2 i przytrzymując naciśnięty lewy przycisk myszy przeciągnij połączenie z wejścia E2. Po przesunięciu kursora w wolnym obszarze i zwolnieniu przycisku myszy połączenie zostanie usunięte.

„Dodatki” „Znacznik połączenia”

Połączenia mogą oprócz linii połączeń być przedstawiane graficznie za pomocą konektorów (znaczników połączenia). Wyjście i przynależące wejście są ze sobą łączone konektorem, który ma jednoznaczną nazwę.

Chcąc zmienić istniejące połączenie między dwoma elementami na połączenie z konektorem, należy najpierw zaznaczyć wyjście linii połączenia (pozycja kursora 3) i wybrać punkt menu „Dodatki” „Znacznik połączenia”. Następująca ilustracja przedstawia połączenie przed i po wybraniu ostatniego punktu menu.



Program nadaje domyślnie jednoznaczną nazwę konektora, która rozpoczyna się literą M, i którą można zmienić. Nazwa konektora jest przechowywana jako parametr wyjścia, ale można ją edytować zarówno na wejściu jak i na wyjściu:

Edycja nazwy konektora na wyjściu:

Zmiana tekstu (opisu) na konektorze spowoduje, że nowa nazwa zostanie zapamiętana na wejściach wszystkich przynależących do niego konektorów. Nie można jednak wybrać nazwy, która należy już do **innego** znacznika połączenia, gdyż w przeciwnym razie naruszona została by zasada jednoznaczności nazwy konektora. W takim przypadku wyświetla się odpowiednie ostrzeżenie.

Edycja nazwy konektora na wejściu:

Zmiana tekstu (opisu) na konektorze spowoduje, że nowy opis zostanie również zmieniony na znaczniku połączenia drugiego modułu. Połączenia pokazywane jako konektory można odwrotnie przekształcić w zwyczajne połączenia, zaznaczając wyjścia połączeń (pozycja kursora 4) i ponownie wybierając punkt menu „Dodatki” „Znacznik połączenia”.

Wstaw wejścia/wyjścia „On the fly”

Jeśli zaznaczony został dokładnie jeden pin wejścia lub wyjścia danego elementu, można poprzez bezpośrednie wprowadzenie ciągu znaków za pomocą klawiatury wstawić odpowiedni element wejścia lub wyjścia i wypełnić jego pole edycji ciągiem znaków.

Kolejność przetwarzania

W graficznym edytorze schematów funkcji CFC elementy takie jak moduł, wyjście, skok, powrót i etykieta otrzymują numer przetwarzania. Określa on kolejność, w jakiej poszczególne elementy są przetwarzane w czasie rzeczywistym.

Nowy element dodawany do schematu otrzymuje automatycznie numer w kolejności topologicznej (z lewej do prawej i z góry na dół). Jeśli kolejność została wcześniej zmieniona, wówczas nowy element przejmuje topologicznie numer swojego poprzednika, a pozostałe wyższe numery będą zwiększone o jeden.

Podczas przesuwania elementu jego numer nie zmienia się.

Kolejność ma wpływ na wynik i w określonych przypadkach można ją zmienić.

W prawym górnym rogu elementu pokazywany jest numer w kolejce do przetwarzania, o ile kolejność została ustawiona.

„Kolejność” „Pokaż”

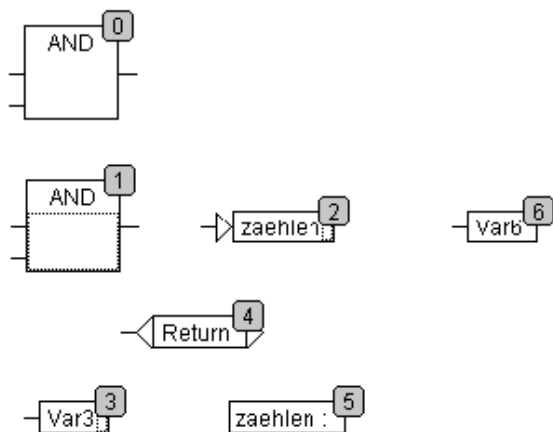
Za pomocą tego polecenia można włączać i wyłączać wyświetlanie kolejności przetwarzania w edytorze CFC. Domyślnie kolejność przetwarzania jest pokazywana (rozpoznawalne po zaznaczeniu przed poleceniem menu).

W elementach moduł, wyjście, skok, powrót i etykieta pojawia się w prawym górnym rogu numer w kolejce do przetwarzania.

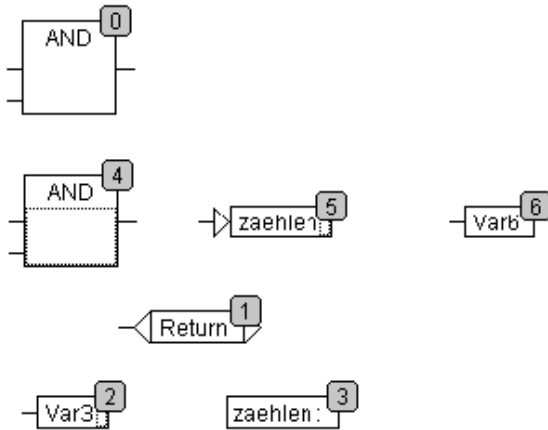
„Dodatki” „Kolejność” „Rozmieść topologicznie”

Elementy w edytorze CFC są uporządkowane w kolejności topologicznej, jeśli przetwarzanie odbywa się z lewej do prawej i z góry na dół, tzn. w przypadku elementów uporządkowanych topologicznie numery zwiększają się z lewej do prawej i z góry na dół. Połączenia nie odgrywają tu żadnej roli. Liczy się tylko położenie elementów.

Za pomocą polecenia „Dodatki” „Kolejność” „Rozmieść topologicznie” można rozmieścić topologicznie wszystkie **zaznaczone** elementy. Przy tym wszystkie elementy selekcji zostaną wyciągnięte z listy przetwarzania. Po przetworzeniu są one pojedynczo wstawiane z powrotem na pozostałą listę przetwarzania poczynawszy od prawego dolnego do lewego górnego miejsca. Każdy zaznaczony element jest przy tym wstawiany na listę przetwarzania przed następnym elementem wg kolejności topologicznej, tzn. element zostanie wstawiony przed tym elementem, który wg kolejności topologicznej zostałby przetworzony bezpośrednio po nim, jeśli wszystkie elementy edytora byłyby uporządkowane w kolejności topologicznej. Dokładnie pokazane jest to na przykładzie.



Elementy z numerami 1, 2 i 3 są zaznaczone. Jeśli teraz wybrane zostało polecenie „**Rozmieść topologicznie**”, wszystkie trzy zaznaczone elementy zostaną wyciągnięte z listy przetwarzania. Następnie zmienna Var3, skok i operator AND zostaną z powrotem wstawione jeden po drugim. Zmienna Var3 zostanie wstawiona przed etykietą i otrzymuje numer 2. Następnie wstawiony zostanie skok i otrzymuje najpierw numer 4, po wstawieniu AND numer 5. Powstała następująca nowa kolejność przetwarzania:



Przy wstawianiu nowego modułu będzie on domyślnie umieszczony na liście do przetwarzania przed swoim topologicznym następnikiem.

„Dodatki” „Kolejność” „Jeden do przodu”

Za pomocą tego polecenia wszystkie zaznaczone elementy w edytorze CFC, za wyjątkiem elementu znajdującego się na początku listy przetwarzania, zostaną przesunięte o jedno miejsce do przodu w kolejce do przetwarzania.

„Dodatki” „Kolejność” „Jeden do tyłu”

Za pomocą tego polecenia wszystkie zaznaczone elementy w edytorze CFC, za wyjątkiem elementu znajdującego się na końcu listy przetwarzania, zostaną przesunięte o jedno miejsce do tyłu w kolejce do przetwarzania.

„Dodatki” „Kolejność” „Przejdź na początek”

Za pomocą tego polecenia wszystkie zaznaczone elementy w edytorze CFC zostaną przesunięte na początek kolejki do przetwarzania, przy czym kolejność w obrębie zaznaczonych elementów zostanie zachowana. Również zachowana zostanie kolejność w obrębie elementów niezaznaczonych.

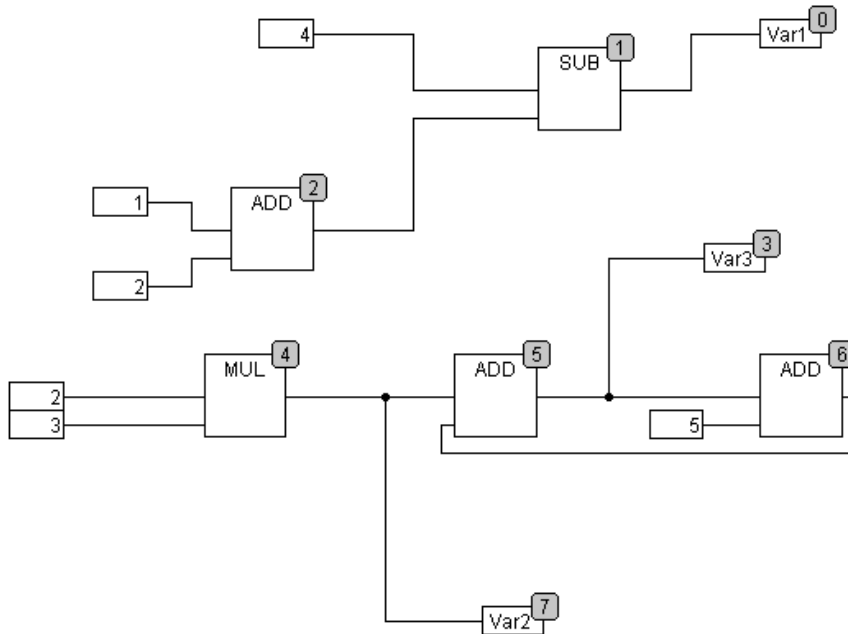
„Dodatki” „Kolejność” „Przejdź na koniec”

Za pomocą tego polecenia wszystkie zaznaczone elementy w edytorze CFC zostaną przesunięte na koniec kolejki do przetwarzania, przy czym kolejność w obrębie zaznaczonych elementów zostanie zachowana. Również zachowana zostanie kolejność w obrębie elementów niezaznaczonych.

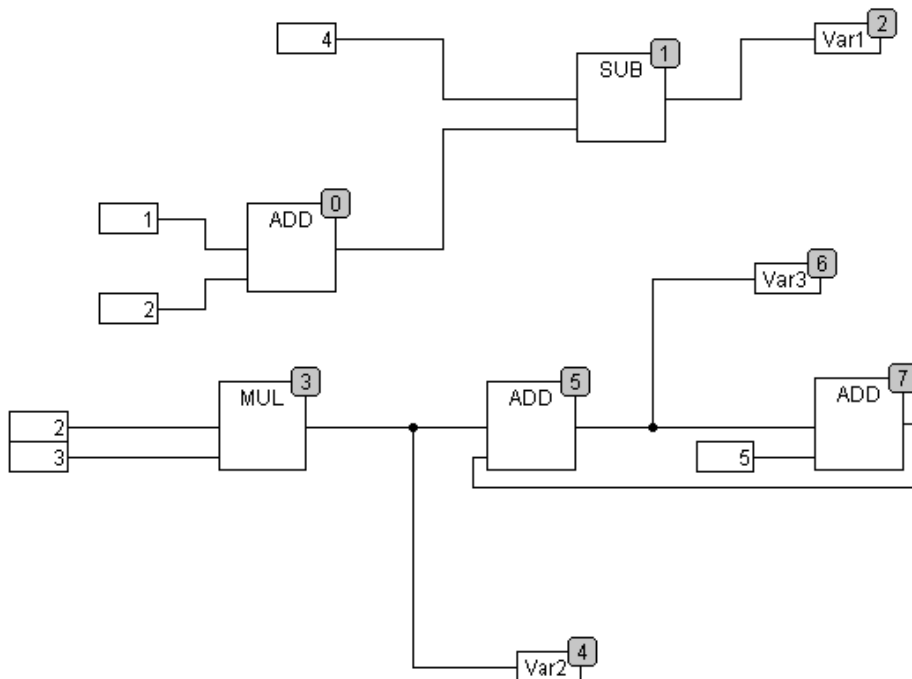
„Dodatki” „Kolejność” „Rozmieść wszystko wg przepływu danych”

To polecenie odnosi skutek **na wszystkie** elementy w edytorze CFC. O kolejności przetwarzania decyduje tutaj przepływ danych elementów, a nie ich położenie.

Następująca ilustracja pokazuje elementy, które są uporządkowane topologicznie.



Po wybraniu polecenia uzyskuje się następujące rozmieszczenie:



Po wybraniu polecenia wszystkie elementy są najpierw sortowane topologicznie. Następnie tworzona jest nowa lista przetwarzania. Począwszy od znanych wartości wejścia ustala się, które z elementów nieponumerowanych mogą być przetwarzane jako następne. W pokazanej wyżej „sieci” moduł ADD może być od razu przetwarzany, gdyż znane są wartości dostępne na jego wejściach (1 i 2). Dopiero po nim może nastąpić przetwarzanie modułu SUB, gdyż dla wykonania wymaga on wyniku modułu ADD itd.

Sprężenia zwrotne są zasadniczo wstawiane jako ostatnie.

Zaletą kolejności przetwarzania opartej na przepływie danych jest to, że element wyjściowy, połączony z wyjściem modułu jest przetwarzany bezpośrednio po nim, co w przypadku kolejności topologicznej nie zawsze jest regułą. Kolejność topologiczna daje w pewnych okolicznościach inny wynik niż kolejność wg przepływu danych, co widać na powyższych przykładach.

„Dodatki” „Utwórz makro”

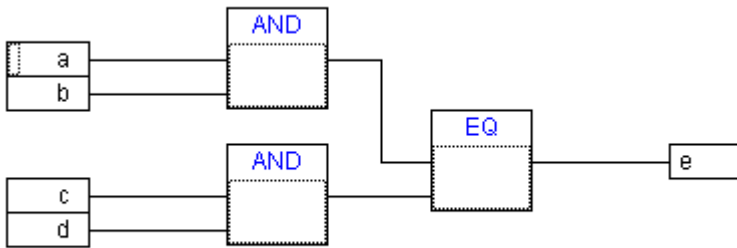
Symbol: 

Za pomocą tego polecenia można zgrupować w bloku kilka zaznaczonych modułów w edytorze CFC i utworzyć makro. Makra mogą być powielane za pomocą poleceń Kopiuj/Wklej, przy czym każda kopia stanowi osobne makro, którego nazwę można wybierać niezależnie. Makra nie mogą być zatem referencjami. Wszystkie połączenia, które zostały „wycięte” podczas tworzenia makra, zastępują piny wejścia wzgl. wyjścia przy makro. Połączenia z wejściami tworzą pin wejścia (In). Nazwa przy pinie jest nazwą domyślną w postaci In<n>. Dla połączeń z wyjściami pojawia się nazwa Out<n>. Te połączenia, które przed utworzeniem makra miały znaczniki połączenia, otrzymują znacznik połączenia na pinie makra.

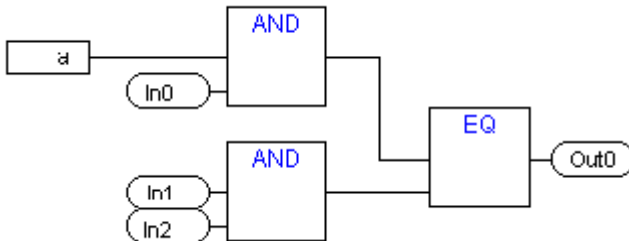
Nowe makra są tworzone z domyślną nazwą „MAKRO”. Można ją zmienić w polu nazwy Zastosowania makra. Podczas edycji nazwa makra jest wyświetlana na pasku tytułowym okna edytora obok nazwy modułu.

Przykład:

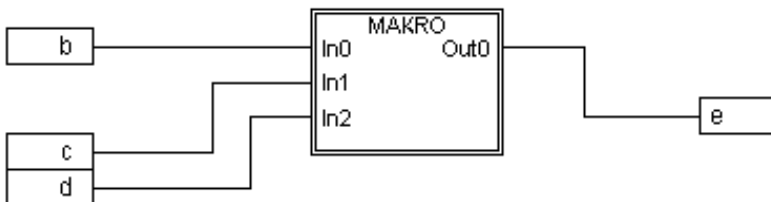
Zaznaczenie



Makro:





W edytorze:



„Dodatki” „Przejdź do makra”Symbol: 

Za pomocą tego polecenia albo podwójnego kliknięcia szkieletu makra w edytorze CFC można otworzyć makro w oknie edytora przynależącego modułu w celu edycji. Nazwa makra jest wyświetlana obok nazwy modułu na pasku tytułowym.

Utworzone podczas zakładania makra skrzynki z pinami dla wejść i wyjść makra można przesuwając, usuwać, dodawać itp. jak normalne wejścia i wyjścia modułów. Różnią się one jedynie formą graficzną oraz tym, że nie posiadają indeksu pozycji. Do dodawania nowych pinów służą przyciski 

(wejście) albo  (wyjście), dostępne na pasku narzędzi. Skrzynki z pinami mają zaokrąglone narożniki. Tekst skrzynki pinów odpowiada nazwie pinu na wyświetlanym obrazie makra.



Kolejność pinów w skrzynce odpowiada kolejności przetwarzania elementów makra. Najniższy indeks kolejności przed wyższym, górny pin przed niższym.

Kolejność przetwarzania w obrębie makra jest zamknięta, tzn. makro jest obliczane jako blok czyli na pozycji makra w nadrzędnym module. Z tego względu polecenia zmieniające kolejność przetwarzania odnoszą skutek tylko w obrębie makra.

„Dodatki” „Rozszerz makro”

Za pomocą tego polecenia zaznaczone w edytorze CFC makro zostanie rozłożone na podstawowe elementy, które zostaną wstawione w miejscu wstawienia makra w module. Połączenia do pinów makra zostaną z powrotem przedstawione jako połączenia do wejść i wyjść elementów. Jeśli rozłożenie makra w miejscu wstawienia skrzynki makra jest niemożliwe z powodu braku miejsca, wtedy makro będzie przesuwane w prawo a potem w dół tak długo, dopóki dostępna będzie wystarczająca ilość wolnego miejsca.

Wskazówka: Jeśli projekt został zapisany w wersji projektu 2.1, wtedy wszystkie makra zostaną rozłożone. Również konwersja na inne języki powoduje rozłożenie wszystkich makr.

„Dodatki” „Powrót o jeden poziom makr”, „Dodatki” „Wszystkie poziomy makr wstecz”Symbole:  

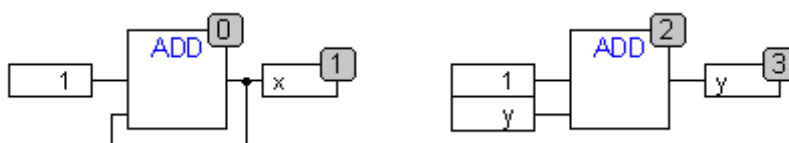
Te polecenia są dostępne również na pasku narzędzi po otwarciu makra w edytorze CFC celem edycji. Jeśli makra są zagnieżdżone jedno w drugim, można przełączać jeden poziom wyświetlania wyżej albo na najwyższy poziom wyświetlania.

Sprzężenia zwrotne

W graficznym edytorze schematów funkcji CFC można w odróżnieniu od zwyczajnego edytora wyświetlać bezpośrednio sprzężenia zwrotne. Należy przy tym pamiętać, że wyjście modułu zawiera zawsze wewnętrzną zmienną pośrednią. W przypadku operatorów zmienna pośrednia przyjmuje typ największego typu danych dla wejść.

Typ danych dla wartości stałych wynika z najmniejszego z możliwych typów danych, tzn. dla stałej „1” przyjmuje się typ danych SINT. Na przykład przy dodawaniu sprzężenia zwrotnego i stałej „1” pierwsze wejście dostarcza typ danych SINT, a drugie ze względu na sprzężenie zwrotne dostarcza typ niezdefiniowany. W takiej sytuacji zmienna pośrednia jest również typu SINT. Wartość zmiennej pośredniej jest przypisywana dopiero po zmiennych wyjściowych.

Poniższa ilustracja pokazuje dodawanie ze sprzężeniem zwrotnym i dodawanie bezpośrednio ze zmiennymi. Zmienne x i y mają być przy tym typu INT.



Między obydwoma schematami dodawania są pewne różnice:

Zmienna y może być inicjowana wartością inną niż 0, nie dotyczy to jednak zmiennej pośredniej z lewego schematu dodawania.

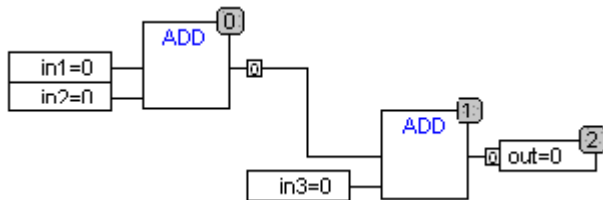
Zmienna pośrednia lewego schematu dodawania jest typu SINT, a na prawym schemacie typu INT. Zmienne x i y mają od 129-go wywołania inne wartości. Zmienna x, mimo że jest typu INT, ma wartość -127, gdyż zmienna pośrednia ma przepełnienie. Wobec tego zmienna y przyjmuje wartość 129.

CFC w trybie online

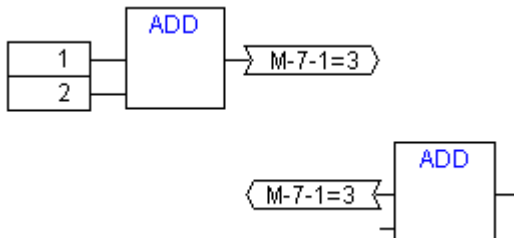
Monitoring:

Wartości dla wejść i wyjść wyświetlane są w skrzynkach wejścia wzgl. wyjścia. Stałe nie są monitorowane. Dla zmiennych nie będących zmiennymi boolean pola wyświetlane są w powiększeniu odpowiednio do pokazywanych wartości. Dla zmiennych boolean nazwy zmiennych oraz połączenia są wyświetlane na niebiesko, jeśli mają wartość TRUE, w przeciwnym wypadku są wyświetlane na czarno.

Wewnętrzne połączenia boolean w trybie online są pokazywane również na niebiesko, jeśli są w stanie TRUE, w przeciwnym razie na czarno. Wartość wewnętrznego połączenia nie będącego połączeniem boolean jest pokazywana w małym kwadracie z zaokrąglonymi narożnikami na pinie wyjścia połączenia.



Piny w makrach są kontrolowane jak skrzynki wejścia lub wyjścia.



Połączenia nieboolean ze znacznikami połączenia pokazują wartość w obrębie znacznika połączenia. Przewody oraz nazwy makr dla połączeń boolean są wyświetlane wszędzie na niebiesko, jeśli przewód dostarcza wartość TRUE, w przeciwnym razie na czarno.

Kontrola przebiegu:

Przy włączonej kontroli przebiegu połączenia są wyświetlane w kolorze ustawionym w opcjach projektu.

Punkty wstrzymania:

Punkty wstrzymania można osadzić również na elementach, które posiadają indeks kolejności przetwarzania. Przetwarzanie programu zostanie wstrzymane przed wykonaniem danego elementu, tzn. w przypadku modułów i wyjść przed przypisaniem wejść, w przypadku znaczników skoku przed wykonaniem elementu z następnym indeksem. Przy określaniu położenia punktu wstrzymania w oknie dialogowym system korzysta z indeksu kolejki do przetwarzania elementu.

Ustawianie punktu wstrzymania odbywa się dla zaznaczonego elementu za pomocą klawisza F9 lub za pomocą punktu menu „Wstaw/usuń punkt wstrzymania” w menu „Online” lub „Dodatki” lub w menu kontekstowym edytora. Jeśli na elemencie ustawiono już punkt wstrzymania, wówczas przy ponownym wykonaniu polecenia „Wstaw/usuń punkt wstrzymania” zostanie on usunięty i odwrotnie. Ponadto stan punktu wstrzymania na elemencie można przełączać dwukrotnym kliknięciem.

Punkt wstrzymania jest pokazywany w kolorze ustawionym w opcjach projektu.

Znacznik RETURN:

W trybie online znacznik skoku z nazwą „RETURN” jest wstawiany automatycznie w pierwszej kolumnie i za ostatnim elementem w edytorze. Znacznik ten oznacza koniec modułu i przy przetwarzaniu krokowym następuje przejście do niego zanim nastąpi opuszczenie modułu. W makrach nie są dodawane żadne znaczniki powrotu.

Przetwarzanie krokowe:

Przy używaniu polecenia „Krok pojedynczy przez” przeskoczenie następuje zawsze do elementu z następnym w kolejności wyższym indeksem. Jeśli bieżącym elementem jest makro lub moduł, wówczas poleceniem „Krok pojedynczy do” następuje rozgałęzienie do jego implementacji. Jeśli stamtąd zostanie wykonane polecenie „Krok pojedynczy przez”, wówczas nastąpi przejście do tego elementu, którego indeks kolejki następuje za indeksem makra.

„Dodatki” „Powiększenie”

Kombinacja klawiszy: <Alt> + <Enter>

Za pomocą tego polecenia można otworzyć implementację modułu, jeśli moduł jest zaznaczony w edytorze CFC.



9 Zasoby

9.1 Przegląd zasobów

W zakładce **Zasoby** Organizera obiektów znajdują się obiekty do organizowania projektów, do śledzenia wartości zmiennych i do konfiguracji projektu przeznaczonego do stosowania na systemie docelowym lub w sieci:

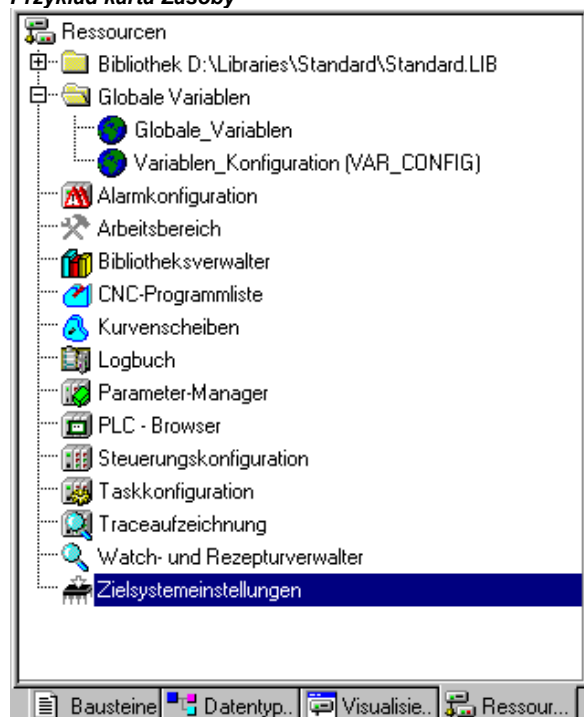
- **Zmienne globalne**, których można używać w całym projekcie albo sieci: Zmienne globalne projektu i podłączone biblioteki oraz, w zależności od ustawień systemu docelowego, także zmienne globalne w obrębie sieci.
- **Konfiguracja alarmów** do konfiguracji klas i grup alarmów, które mogą być wstawiane np. do wizualizacji w celu pokazania informacji lub obsługi.
- **Menedżer bibliotek** do zarządzania wszystkimi bibliotekami podłączonymi do projektu.
- **Dziennik** do chronologicznego rejestrowania działań, odbywających się podczas sesji online.
- **Konfiguracja sterownika** do konfigurowania urządzenia.
- **Konfiguracja zadań** do sterowania programem za pomocą zadań.
- **Menedżer podglądu zmiennych i receptur** do wyświetlania i ustawiania wartości domyślnych dla zmiennych.
- **Obszar roboczy** jako podgląd wszystkich aktualnie ustawionych opcji projektu.
- **Ustawienia systemu docelowego** do wybierania i ewentualnego parametryzowania sprzętu docelowego.

W zależności od ustawień systemu docelowego dostępne są ponadto następujące zasoby:

- **Zapis śledzenia** do graficznego przedstawienia wartości zmiennych.
- **Menedżer parametrów** (katalog obiektów) do przygotowywania zmiennych, do których dostęp mają także inni użytkownicy w sieci sterującej. (Funkcjonalność jest zależna od systemu docelowego).
- **Przeładowarka PLC** do odpytywania informacji ze sterownika w czasie rzeczywistym.
- **Narzędzia** do podłączania zewnętrznych narzędzi, które będzie można później uruchamiać z CoDeSys.
- **SoftMotion** funkcjonalność (licencjonowana) z edytorem CNC (lista programów CNC) i edytorem CAM (krzywki tarczowe).

Dodatkowo można, jeśli obiekt zmiennych globalnych jest otwarty, utworzyć i wywołać dla projektu **szablon dokumentu**, za pomocą którego w dokumentacji można dla takich samych zmiennych projektowych przygotowywać różne komentarze.

Przykład karta Zasoby



9.2 Zmienne globalne, konfiguracja zmiennych, szablon dokumentu

Obiekty w „Zmiennych globalnych”

W Organizerze obiektów znajdują się w zakładce **Zasoby** w folderze **Zmienne globalne** standardowo dwa obiekty (w nawiasach domyślne nazwy obiektów):

- Lista zmiennych globalnych (Zmienne_globalne)
- Konfiguracja zmiennych (Konfiguracja_zmiennych)

Wszystkie zmienne zdefiniowane w tych obiektach są znane w całym projekcie, zmienne globalne w sieciach mogą ponadto służyć do wymiany danych z innymi użytkownikami sieci.


Jeśli folder Zmienne globalne nie jest rozwinięty (znak plus przed folderem), można go otworzyć podwójnym kliknięciem lub naciśnięciem klawisza <Enter> po zaznaczeniu wiersza.

Wybierz odpowiednią opcję. Za pomocą polecenia „**Edytuj obiekt**” można otworzyć okno z dotychczas zdefiniowanymi zmiennymi globalnymi. Edytor wówczas pracuje jak edytor deklaracji.

Dodatkowe listy zmiennych

Zmienne globalne (**VAR_GLOBAL**), globalne zmienne sieciowe (**VAR_GLOBAL**, dostępność zależna od systemu docelowego), i konfiguracje zmiennych (**VAR_CONFIG**) muszą być zdefiniowane w osobnych obiektach.

Dla zmiennych globalnych zadeklarowanych przez użytkownika istnieje możliwość zakładania list zmiennych, w których można przechowywać je w sposób przejrzysty i uporządkowany tak jak na standardowej liście „Zmienne_globalne”.

Zaznacz w obiekcie Organizer folder **Zmienne globalne** lub jeden z istniejących już  obiektów ze zmiennymi globalnymi i wykonaj polecenie „**Projekt**” „**Wstaw obiekt**”. W wyświetlonym oknie dialogowym należy wprowadzić nazwę nowego obiektu. Pod taką nazwą zostanie utworzony kolejny obiekt ze słowem kluczowym **VAR_GLOBAL**. Jeśli jednak chcesz utworzyć obiekt będący konfiguracją zmiennych, zmień słowo kluczowe odpowiednio na **VAR_CONFIG**.

9.2.1 Zmienne globalne

Co to są zmienne globalne

Zmiennymi globalnymi mogą być „normalne” zmienne, stałe lub zmienne remanentne, znane w całym projekcie, ale także zmienne sieciowe, które służą do wymiany danych między użytkownikami sieci.

Wskazówka: Można definiować zmienną lokalną z taką samą nazwą jak zmienna globalna. W obrębie modułu zmienna lokalna ma zawsze pierwszeństwo.
Nie można utworzyć dwóch zmiennych globalnych o jednakowej nazwie; przykładowo zostanie wygenerowany komunikat o błędzie, jeśli zarówno w konfiguracji sterownika i na liście zmiennych globalnych jest zdefiniowana zmienna „var1”.

Zmienne sieciowe

Wskazówka: Korzystanie ze zmiennych sieciowych musi być oferowane przez system docelowy i aktywowane w ustawieniach systemu docelowego (kategoria Funkcje sieciowe).

Poprzez **automatyczną wymianę danych** (por. nie-automatyczną wymianę danych w Menedżerze parametrów) jest możliwe aktualizowanie wartości zmiennych na wielu sterownikach w obrębie sieci utworzonej ze sterowników kompatybilnych z CoDeSys. Nie wymaga to specjalnych funkcji sterujących, użytkownicy sieci muszą jednak w swoich projektach używać jednakowych list deklaracji i odpowiedniej konfiguracji transmisji zmiennych sieciowych.

W celu zapewnienia identyczności list, nie zaleca się ręcznego wprowadzania deklaracji zmiennych w każdej aplikacji sterownika osobno, lecz pobrania ich ze specjalnego pliku, który można przykładowo utworzyć poprzez wyeksportowanie. (zob. „Tworzenie listy zmiennych globalnych”).

By wymiana zmiennych sieciowych mogła się odbywać, muszą być one używane w cyklicznym lub dowolnym zadaniu lub w PLC_PRG. Nie wystarcza przy tym zadeklarowanie ich tylko w części z deklaracją. Jeśli zmienne są używane w różnych zadaniach/PLC_PRG, zostanie uwzględniona ta zmienna o najwyższym priorytecie.

Tworzenie listy zmiennych globalnych

By utworzyć nową listę zmiennych globalnych, przejdź do obiektu Organizator i w Zasobach zaznacz wpis „Zmienne globalne” albo jedną z istniejących tam list zmiennych globalnych. Po wybraniu polecenia „Projekt” „Obiekt” „Wstaw” otworzy się okno dialogowe **Lista zmiennych globalnych**.


To okno dialogowe otwiera się również w celu pokazania konfiguracji utworzonej już wcześniej listy zmiennych globalnych po zaznaczeniu jej w obiekcie Organizator i wybraniu polecenia „Projekt” „Obiekt” „Właściwości”.

Powiązanie pliku:

Nazwa pliku: Jeśli istnieje plik eksportu (*.exp) lub plik DCF, zawierający odpowiednie zmienne, można go tutaj wybrać. W tym celu podaj ścieżkę do pliku ew. kliknij przycisk Znajdź, który otwiera systemowe okno dialogowe „Wybierz plik tekstowy”. Pliki DCF podczas wczytywania są przekształcane w składnię IEC.

Wybierz opcję Importuj przed kompilacją, jeśli chcesz, by przed każdą kompilacją projektu podana zewnętrzna lista zmiennych była wczytywana od nowa. Opcję Eksportuj przed kompilacją można wybrać, jeśli lista zmiennych ma być zapisywana do pliku przed każdą kompilacją projektu.

Nazwa listy zmiennych globalnych: Podaj nazwę dla nowej listy.

Po zamknięciu okna dialogowego „Lista zmiennych globalnych” przyciskiem **OK**, zostanie utworzony nowy obiekt, oznaczony w Organizatorze symbolem . Można go teraz otwierać poleceniem „Projekt” „Obiekt” „Edytuj” albo dwukrotnym kliknięciem wpisu.

Za pomocą polecenia „Projekt” „Obiekt” „Właściwości” można otworzyć okno dialogowe konfiguracji „Lista zmiennych globalnych” dla wpisu zaznaczonego w Organizatorze.

Okno dialogowe tworzenia nowej listy zmiennych globalnych

Konfiguracja zmiennych sieciowych (Connection...):

Jeśli w ustawieniach systemu docelowego (patrz rozdz. 13.32) włączona jest opcja „Obsługuj zmienne sieciowe”, można używać przycisku **Dodaj połączenie sieciowe**. Za pomocą niego można wyświetlić dodatkowe opcje okna dialogowego, które wówczas wygląda jak wyżej. Jeśli opcja ta jest nieaktywna, wówczas przycisk jest niewidoczny.

Connection <n> (<typ sieci>):

W dolnej części okna dialogowego można dla łącznie czterech ($n = 1$ do 4) połączeń sieciowych na każdej karcie utworzyć wiersz konfiguracji, który definiuje, w jaki sposób można posługiwać się listą zmiennych podczas wymiany między użytkownikami sieci. By wymiana danych funkcjonowała zgodnie z określonym celem, ta sama lista zmiennych musi być odpowiednio skonfigurowana u innych użytkowników sieci.

Jeśli nie utworzono jeszcze żadnej konfiguracji, w przypadku sieci UDP będzie dostępna najpierw tylko jedna karta z opisem „**Connection 1 (UDP)**”. Każde kliknięcie przycisku „Dodaj nowe połączenie sieciowe” powoduje dodanie kolejnej karty (maksymalnie 4) z kolejnym numerem za słowem „Connection”.

Typ sieci: Wybierz z listy odpowiedni typ. Lista jest zdefiniowana zgodnie z ustawieniami systemu docelowego. Na przykład „CAN” jako skrót może oznaczać sieć CAN, albo „UDP” oznacza system transmisji UDP.

Ustawienia: Przycisk ten powoduje wyświetlenie okna dialogowego **Ustawienia dla <typ sieci>** z następującymi możliwościami konfiguracji:

UDP:

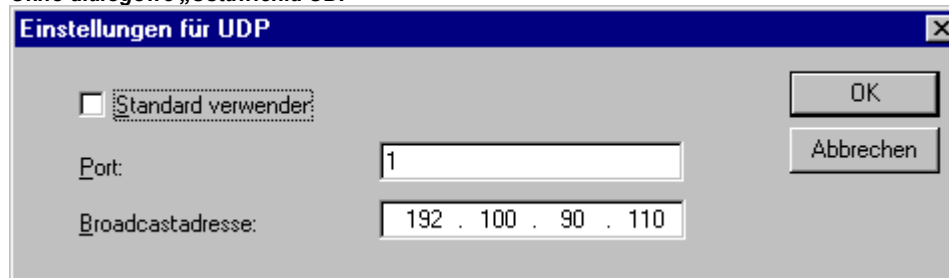
Użyj domyślnych Jeśli ten przycisk zostanie użyty, zostanie wpisany port 1202 do wymiany danych między innymi użytkownikami sieci. W adresie Broadcast/Multicast podany jest domyślnie adres „255 . 255 . 255”, co oznacza, że wymiana danych może odbywać się z wszystkimi użytkownikami sieci.

Port: Port alternatywny (patrz wyżej) (Uwaga: musi być tak samo ustawiony we wszystkich węzłach każdego z użytkowników). Dla typu sieci UDP podana tutaj wartość jest automatycznie przyjmowana dla wszystkich połączeń zdefiniowanych ew. na kolejnych kartach.

Broadcast/Multicast address: Adres alternatywny do domyślnego (patrz wyżej) ew. adres podsieci (np. „197 . 200 . 100 . 255” obejmujący wszystkich użytkowników z adresami IP 197 . 200 . 100 . x).

Należy pamiętać, że dla systemów Win32 adres Broadcast musi pasować do maski podsieci konfiguracji TCP/IP komputera!

Okno dialogowe „Ustawienia UDP”

CAN:

Indeks kontrolera: Indeks kontrolera CAN, za pomocą którego zmienne mają być przesyłane.

W konfiguracji transmisji zmiennych można aktywować lub dezaktywować następujące opcje:

Pakuj zmienne: Zmienne do transmisji są grupowane w pakiety (telegramy), których rozmiar zależy od rodzaju sieci. Jeśli opcja jest wyłączona, dla każdej zmiennej tworzony jest osobny pakiet.

Numer listy zmiennych: Numer pierwszego pakietu, w którym wysyłane są zmienne. (domyślnie = 1). Kolejne pakiety są numerowane rosnąco.

W zależności od systemu docelowego zmienne na liście mogą być zdefiniowane z prawem „do odczytu” i „do zapisu” lub wyłącznie „do odczytu” albo „do zapisu”. Do tego celu służą odpowiednie opcje „Odczytaj” i „Zapisz”:

Odczytaj: Zmienne na liście mogą być odczytywane; jeśli opcja jest wyłączona, nowe wartości wysyłane przez sieć będą ignorowane. Jeśli opcja „Odczytaj” jest zaznaczona, można dodatkowo wybrać następujące opcje:

Monit podczas bootowania: Jeśli lokalny węzeł jest węzłem „odczytującym” (opcja Odczytaj jest aktywna, patrz wyżej) i zostanie ponownie uruchomiony, wówczas w tym momencie aktualne wartości zmiennych z innych sterowników, które je zapisują, zostaną przez nie zażądane i będą dalej wysyłane, niezależnie od pozostałych warunków dla czasu i zdarzeń, ustawionych dla nich w konfiguracji wysyłania. Warunkiem jest, by w konfiguracji zmiennych zapisujących wartości sterowników była aktywna opcja Odpowiedz na monit bootowania (patrz niżej).

Zapisz: Zmienne na liście będą zapisywane, przy czym dodatkowo dostępne są następujące opcje:

Prześlij sumę kontrolną: Podczas wysyłania dla każdego pakietu dołączana jest suma kontrolna, która jest sprawdzana przy odbiorze. Dzięki temu sprawdza się, czy definicje zmiennych są identyczne przy wysyłaniu i odbiorze. Pakiety z niezgodną sumą kontrolną nie będą przyjmowane, jeśli tak ustawiono (Transfer potwierdzony, patrz niżej), z potwierdzeniem negatywnym.

Transfer potwierdzony: (nie dotyczy CAN): Każda wiadomość jest potwierdzana przez urządzenie odbierające. Jeśli stacja nadająca nie otrzyma przynajmniej jednego potwierdzenia przyjęcia w czasie jednego cyklu, wówczas zostanie wygenerowany komunikat o błędzie.

Odpowiedz na monit bootowania: Jeśli węzeł lokalny jest węzłem „zapisującym” (opcja Zapisz jest aktywna, patrz wyżej), wówczas żądania węzłów odczytujących, wysyłane podczas

bootowania (opcja Odpowiedz podczas bootowania, patrz wyżej), będą odpowiadane. Oznacza to, że aktualne wartości zmiennych będą przesyłane, nawet jeśli niektóre warunki w konfiguracji czasu lub zadań na to nie pozwalają.

Transmisja cykliczna: Zmienne będą przesyłane w odstępach zgodnie z ustawieniem w polu Odstęp (odstęp czasu np. T#70 ms).

Transmisja w przypadku zmiany: Zmienne są zapisywane tylko podczas zmiany wartości; wartość podana w polu Odstęp minimalny można ustalić jako odstęp minimalny pomiędzy kolejnymi transmisjami.

Transmisja sterowana zdarzeniem: Zmienne z listy są zapisywane, jeśli zmienna wpisana w polu Zmienna ma wartość TRUE.

Listy zmiennych globalnych sieciowych można rozpoznać w Organizeryze po symbolu .

Wskazówka: Jeśli zmienna sieciowa globalna występuje w jednym lub kilku zadaniach, czas transmisji jest kontrolowany w następujący sposób: Przy wywołaniu każdego zadania odbywa się sprawdzenie, jakie parametry obowiązują dla transmisji wartości zmiennej (konfiguracja w oknie dialogowym „Lista zmiennych globalnych”). Wartość zmiennej zostanie przesłana bądź nie w zależności od tego, czy ustawiony czas odstępu właśnie upłynął. Przy każdej transmisji licznik czasu odstępu dla danej zmiennej jest zerowany.

Wysyłanie jest zawsze wykonywane przez system czasu przebiegu danego sterownika. Nie ma więc potrzeby korzystania z dodatkowych funkcji specjalnych sterownika dla wymiany danych.

Edycja list zmiennych globalnych, zmienne globalne sieciowe

Edytor zmiennych globalnych pracuje jak edytor deklaracji. Jeśli jednak wyświetlana jest zewnętrzna lista zmiennych, to wtedy nie można jej tutaj edytować. Zewnętrzne listy zmiennych mogą być edytowane tylko zewnętrznie i są wczytywane od nowa przy każdym otwieraniu i kompilacji projektu.

Składnia:

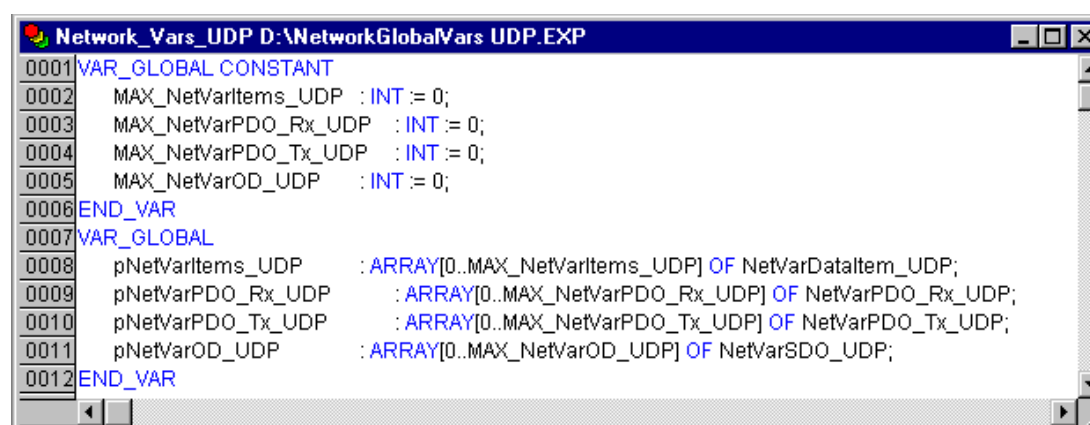
VAR_GLOBAL

(* deklaracje zmiennych *)

END_VAR

Zmiennych sieciowych można używać tylko wtedy, gdy system docelowy to umożliwi. Muszą być one również zdefiniowane w takiej samej składni.

Przykład listy zmiennych sieciowych, utworzonej na podstawie pliku eksportu *.exp zawierającej tytuł NETWORK_VARS_UDP:



```

0001 VAR_GLOBAL CONSTANT
0002     MAX_NetVarItems_UDP : INT := 0;
0003     MAX_NetVarPDO_Rx_UDP : INT := 0;
0004     MAX_NetVarPDO_Tx_UDP : INT := 0;
0005     MAX_NetVarOD_UDP : INT := 0;
0006 END_VAR
0007 VAR_GLOBAL
0008     pNetVarItems_UDP : ARRAY[0..MAX_NetVarItems_UDP] OF NetVarDataItem_UDP;
0009     pNetVarPDO_Rx_UDP : ARRAY[0..MAX_NetVarPDO_Rx_UDP] OF NetVarPDO_Rx_UDP;
0010     pNetVarPDO_Tx_UDP : ARRAY[0..MAX_NetVarPDO_Tx_UDP] OF NetVarPDO_Tx_UDP;
0011     pNetVarOD_UDP : ARRAY[0..MAX_NetVarOD_UDP] OF NetVarSDO_UDP;
0012 END_VAR
    
```

Edycja list remanentnych zmiennych globalnych

W zależności od systemu przebiegu rzeczywistego można pracować ze zmiennymi remanentnymi (patrz także rozdział 8.2.1, Zmienne remanentne, Przegląd albo Reinicjalizacja). Istnieją dwa rodzaje remanentnych zmiennych globalnych:

- Zmienne Retain zachowują swoje wartości po niekontrolowanym zakończeniu systemu czasu przebiegu (Wył./Wł.) ewentualnie po wydaniu polecenia Restet w trybie online w CoDeSys.

- Zmienne trwale zachowują swoje wartości tylko po wykonaniu pobierania programu.

Zmienne remanentne zawierają ponadto słowo kluczowe **RETAIN** albo **PERSISTENT**.

Uwaga: Zmienne trwale nie są automatycznie również zmiennymi Retain!

Składnia:

VAR_GLOBAL RETAIN
 (* deklaracje zmiennych *)
 END_VAR

VAR_GLOBAL PERSISTENT
 (* deklaracje zmiennych *)
 END_VAR

oba słowa kluczowe są używane do tworzenia kombinacji właściwości Retain i Persistent:

VAR_GLOBAL RETAIN PERSISTENT lub **VAR_GLOBAL PERSISTENT RETAIN**

Zmienne sieciowe (specyficzne dla systemu docelowego) są również definiowane w tej składni.

Stałe globalne

Stałe globalne występują dodatkowo ze słowem kluczowym **CONSTANT**.

Składnia:

VAR_GLOBAL CONSTANT
 (* deklaracje zmiennych *)
 END_VAR


9.2.2 Konfiguracja zmiennych...

W modułach funkcji można dla zmiennych, zdefiniowanych między słowami kluczowymi **VAR** i **END_VAR**, podać adresy dla wejść i wyjść, które nie są zdefiniowane kompletnie. Niekompletnie zdefiniowane adresy są oznaczone gwiazdką.

Przykład:

```
FUNCTION_BLOCK locio
VAR
  loci AT %I*: BOOL := TRUE;
  loco AT %Q*: BOOL;
END_VAR
```

W przykładzie tym zdefiniowano dwie zmienne lokalne wejścia/wyjścia, lokalna zmienna wejścia (%I*) i lokalna zmienna wyjścia (%Q*).

Jeśli chcesz zdefiniować lokalne zmienne wejścia/wyjścia, w Organizeryze na zakładce **Zasoby** dostępny jest standardowo obiekt  **Konfiguracja_zmiennych**, w którym można skonfigurować zmienne. Obiekt można utworzyć bez nazwy oraz można tworzyć kolejne obiekty dla konfiguracji zmiennych.

Edytor konfiguracji zmiennych pracuje jak edytor deklaracji.

Zmienne dla lokalnej konfiguracji wejścia/wyjścia muszą znajdować się między słowami kluczowymi **VAR_CONFIG** i **END_VAR**.

Nazwa takiej zmiennej składa się z kompletnej ścieżki instancji, przy czym nazwy poszczególnych modułów i instancji są oddzielone od siebie kropkami. Deklaracja musi zawierać adres, którego klasa (wejście/wyjście) odpowiada adresowi niekompletnemu (%I*, %Q*) w bloku funkcji. Także typ danych musi być zgodny z deklaracją w module funkcji.

Zmienne konfiguracji, których ścieżka instancji jest nieważna, gdyż instancja nie występuje, będą oznaczane jako błędy. I odwrotnie komunikat o błędzie będzie wyświetlany również wtedy, gdy dla

zmiennej typu instancja nie występuje żadna konfiguracja. By otrzymać kompletną listę wszystkich potrzebnych zmiennych konfiguracji, można skorzystać z polecenia menu „**Wszystkie ścieżki instancji**” w menu „**Wstaw**”.

Przykład konfiguracji zmiennych

```
FUNCTION_BLOCK locio
VAR
  loci AT %I*: BOOL := TRUE;
  loco AT %Q*: BOOL;
END_VAR
```

W przykładzie tym zdefiniowano dwie zmienne lokalne wejścia/wyjścia, lokalna zmienna wejścia (%I*) i lokalna zmienna wyjścia (%Q*).

W programie istnieją następujące definicje modułów funkcji:

```
PROGRAM PLC_PRG
VAR
  Hugo: locio;
  Otto: locio;
END_VAR
```

Prawidłowa konfiguracja zmiennych wygląda następująco:

```
VAR_CONFIG
PLC_PRG.Hugo.loci AT %IX1.0 : BOOL;
PLC_PRG.Hugo.loco AT %QX0.0 : BOOL;
PLC_PRG.Otto.loci AT %IX1.0 : BOOL;
PLC_PRG.Otto.loco AT %QX0.3 : BOOL;
END_VAR
```

Wskazówka: Należy pamiętać o tym, by nie zamieszczać opisu wyjścia, które występuje w konfiguracji zmiennych, ani w projekcie ani poprzez zmienną (deklaracja AT), gdyż opis ten będzie ignorowany.

„Wstaw” „Wszystkie ścieżki instancji”

Za pomocą tego polecenia tworzony jest blok **VAR_CONFIG – END_VAR**, który zawiera wszystkie ścieżki instancji występujące w projekcie. Dostępne już deklaracje nie będą wstawiane ponownie, by zachować istniejące adresy. Ten punkt menu jest dostępny w oknie konfiguracji zmiennych, gdy projekt jest kompilowany („Projekt” „Kompiluj wszystko”).

9.2.3 Szablon dokumentu

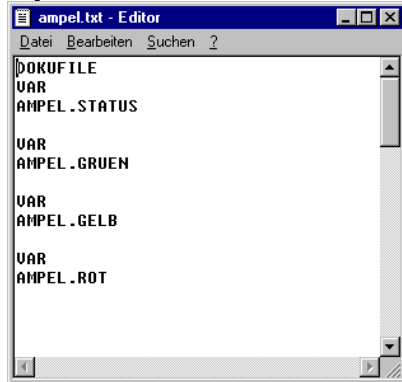
Co to jest szablon dokumentu

Niezależnie od funkcji „Projekt” „Kompiluj projekt na inny język” można używać szablonu dokumentu w razie potrzeby wielokrotnego dokumentowania projektu. Z dokumentacji można również skorzystać dla tego samego projektu zawierającego komentarze w wielu językach albo dla kilku podobnych projektów, które używają takich samych nazw zmiennych.

Wybierz polecenie „Dodatki” „Utwórz szablon dokumentacji”, które jest dostępne, gdy tylko zostanie otwarta globalna lista zmiennych.

Utworzony plik można otworzyć i edytować w dowolnym edytorze tekstowym. Plik rozpoczyna się wierszem **DOKUFILE**, następnie występuje lista zmiennych projektowych, przy czym każda zmienna zajmuje domyślnie trzy wiersze, wiersz **VAR**, który rozpoczyna definicję nowej zmiennej, następnie wiersz z nazwą zmiennej, a na końcu pusty wiersz. Ten wiersz można wykorzystać, by wstawić komentarz dla zmiennych. Zmienne, których nie ma potrzeby dokumentować, można zwyczajnie usunąć z tekstu. Dla jednego projektu można utworzyć dowolną ilość szablonów dokumentacji.

Edytor Windows z szablonem dokumentacji



W celu użycia szablonu dokumentacji należy wybrać polecenie „Dodatki” „Wybierz szablon dokumentacji”. Jeśli teraz zostanie utworzona dokumentacja dla całego projektu albo części projektu zostaną wydrukowane, wówczas w części implementacyjnej (nie w deklaracji!), czyli tam gdzie używana jest zmienna, dodawany jest tekst komentarza, który został utworzony w szablonie dokumentacji dla tej zmiennej. Ten komentarz pokazuje się tylko na wydruku!

„Dodatki” „Utwórz szablon dokumentacji”

Za pomocą tego polecenia można utworzyć szablon dokumentacji. Polecenie to jest dostępne, jeśli otwarty jest obiekt Zmienne globalne.

Otwiera się okno dialogowe zapisywania pliku pod nową nazwą. W polu **Nazwa pliku** rozszerzenie *.txt jest wstawione domyślnie. Należy wpisać w nim dowolną nazwę. Zostanie utworzony plik tekstowy, w którym zawiera się lista wszystkich zmiennych w projekcie.

„Dodatki” „Wybierz szablon dokumentacji”

Za pomocą tego polecenia można wybrać szablon dokumentacji.

Zostanie wyświetlone okno dialogowe, umożliwiające otwieranie plików. Należy wybrać żądany szablon dokumentacji i nacisnąć przycisk **OK**. Jeśli teraz zostanie utworzona dokumentacja dla całego projektu albo części projektu zostaną wydrukowane, wówczas w treści programu zostanie wstawiony komentarz do wszystkich zmiennych, który został utworzony w szablonie dokumentacji. Ten komentarz pokazuje się tylko na wydruku!

W celu utworzenia szablonu dokumentacji należy wybrać polecenie „Dodatki” „Utwórz szablon dokumentacji”.

9.3 Konfiguracja alarmów

9.3.1 Przegląd

Za pomocą zintegrowanego w CoDeSys systemu alarmów jest możliwe wykrywanie krytycznych stanów procesów, ich rejestracji albo obrazowania ich w wizualizacji. Obsługę alarmów można przeprowadzić w CoDeSys oraz opcjonalnie również w sterowniku. Należy przy tym zapoznać się z ustawieniami systemu docelowego w oknie dialogowym „Wizualizacja”.

Jeśli system docelowy tego nie zapewnia, dostępne są okna dialogowe „Konfiguracja alarmów” w Organizerze na zakładce Zasoby.

Są tutaj zdefiniowane **Klasy alarmów** i **Grupy alarmów**. Klasa alarmu służy do ustalenia typu alarmu, tzn. nadaje ona jemu określone parametry. Grupa alarmu służy do konkretnej konfiguracji jednego lub kilku alarmów (którym przypisane zostały określona klasa i dalsze parametry) w celu stosowania w projekcie. Oferuje ona również możliwość tworzenia struktury dostępnych alarmów. Użytkownik programu może dodawać i definiować różne grupy alarmów poniżej punktu rozgałęzienia „System”.

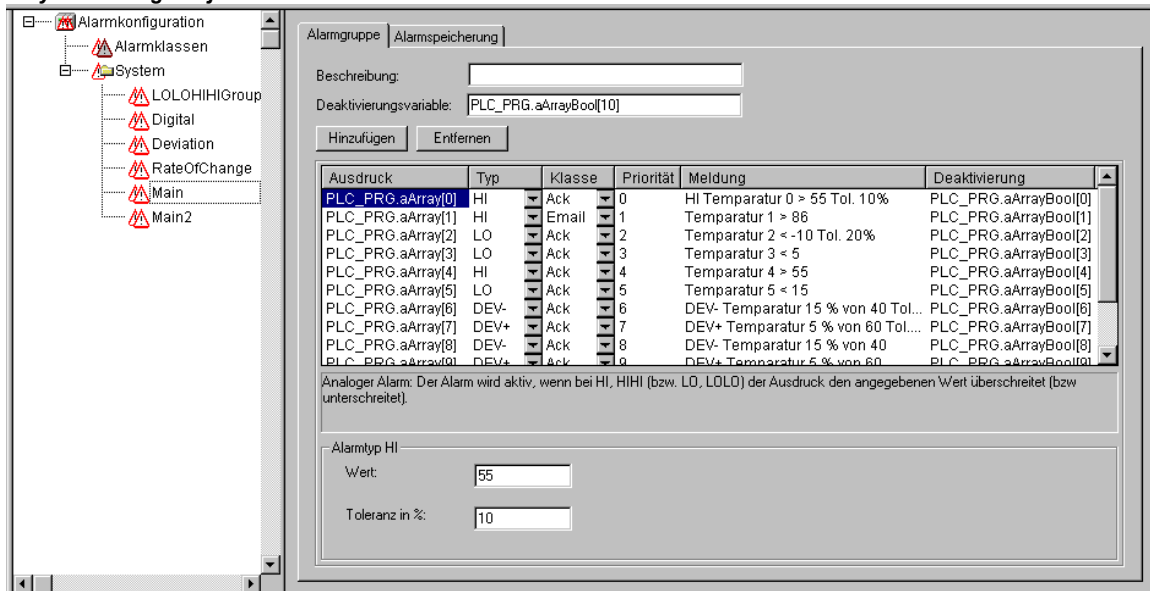
Konfiguracja alarmów

Do **wizualizacji** alarmów służy element „Tabela alarmów” w wizualizacji. W tej tabeli użytkownik może obserwować i potwierdzać alarmy.

By otrzymać **Historię**, czyli zapis zdarzeń alarmowych w pliku Log, plik taki musi być zdefiniowany dla każdej grupy wraz z warunkami zapisywania zdarzeń.

Jeśli zostanie wybrany wpis „Konfiguracja alarmów” w Zasobach, otwiera się dwuczęściowe okno dialogowe „Konfiguracja alarmów”, które obsługuje się podobnie jako okno konfiguracji sterownika albo konfiguracji zadań. Z lewej strony okna jest widoczne drzewo konfiguracji, z prawej strony znajduje się okno dialogowe konfiguracji pasujące do wpisu zaznaczonego w drzewie.

Przykład konfiguracji alarmów



Rozwiń drzewo poprzez kliknięcie myszą znaczka plus przed wpisem „Konfiguracja alarmów”. W przypadku tworzenia nowej konfiguracji drzewo zawiera jedynie wpisy „Klasy alarmów” i „System”.

9.3.2 System alarmów, definicje

Korzystanie z systemu alarmów w CoDeSys wymaga zapoznania się z następującymi ogólnie obowiązującymi opisami i definicjami dotyczącymi alarmów:

- Alarm: Zasadniczo alarm jest traktowany jako specjalny warunek (wartość wyrażenia).
- Priorytet: Priorytet albo też „istotność” alarmu opisuje jak ważny (istotny) jest warunek alarmu. Najwyższym priorytetem jest „0”, najniższą możliwą wartością jest „255”.
- Stan alarmu: Wyrażenie lub zmienna skonfigurowana w celu kontroli alarmów może przyjmować następujące stany: NORM (brak alarmu), INTO (alarm właśnie następuje, „Alarm nadchodzi”), ACK (alarm wystąpił i został potwierdzony przez użytkownika), OUTOF (stan alarmowy upłynął, „Alarm upłynął”, ale nie został jeszcze potwierdzony!)
- Stan podrzędny: Warunek alarmu może zawierać granice (Lo, Hi) i granice „ekstremalne” (LoLo, HiHi). Przykład: Wartość wyrażenia rośnie i przekracza najpierw granicę HI, po czym następuje alarm HI. Jeśli wartość dalej wzrasta i jeszcze przed potwierdzeniem alarmu HI przekroczy granicę HHI, wówczas alarm HI zostanie potwierdzony wewnętrznie automatycznie, ale na liście alarmów (wewnętrzna lista do zarządzania alarmami) pozostaje jeszcze stan alarmowy HHI. Stan HI w takim przypadku jest określany jako stan podrzędny.
- Potwierdzanie alarmów: Głównym zastosowaniem alarmów jest oznajmianie użytkownikowi sytuacji alarmowych. Bardzo często należy przy tym mieć pewność, czy użytkownik takie powiadomienie otrzymał (patrz możliwe działania w konfiguracji klas alarmów). Użytkownik musi „potwierdzić” alarm, aby ten mógł zostać usunięty z listy alarmów.

- Zdarzenie alarmowe: Zdarzenia alarmowego nie należy mylić z warunkiem alarmu. Podczas gdy warunek alarmu może trwać przez dowolnie długi czas, to zdarzenie alarmowe opisuje chwilowe wystąpienie zmiany, przykładowo ze stanu normalnego do stanu alarmowego. Możliwe zdarzenia alarmowe: INTO, OUTOF, ACK.

CoDeSys obsługuje następujące możliwości:

- Dezaktywacja generowania alarmu dla poszczególnych alarmów oraz całych grup alarmów
- Zaznaczanie wyświetlanych alarmów poprzez grupy alarmów oraz priorytet poszczególnych alarmów
- Zapisywanie wszystkich powstałych zdarzeń alarmowych
- Element wizualizacji Tabela alarmów w CoDeSys Wizualizacja

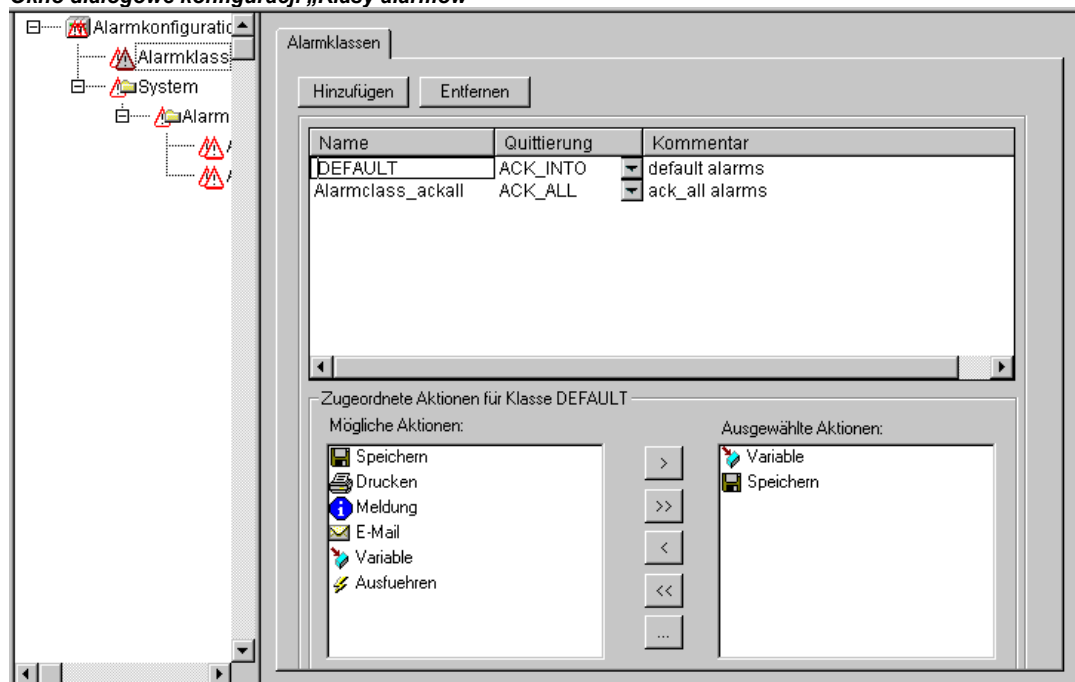
9.3.3 Klasy alarmów

Klasy alarmów służą zasadniczo do opisywania określonych kryteriów alarmów, jak np. filozofia kwitowania (potwierdzanie alarmu przez użytkownika), wykonywanie działania (jakie działanie ma być wykonane automatycznie w określonych stanach alarmowych) oraz wizualizacja tabeli alarmów. Klasy alarmów są definiowane globalnie w konfiguracji alarmów i następnie mogą być używane jako „konfiguracje podstawowe” dla każdej grupy alarmów.

Konfiguracja klas alarmów:

Zaznacz wpis „Klasy alarmów” w drzewie konfiguracji alarmów. Otwiera się okno dialogowe konfiguracji „Klasy alarmów”:

Okno dialogowe konfiguracji „Klasy alarmów”



Kliknij przycisk **Dodaj**, by utworzyć klasę alarmu. W górnym wierszu zostanie wstawiony wiersz, który domyślnie zawiera tylko ustawienia „NOACK” (no acknowledgement) w kolumnie „Potwierdzanie”. Kliknij pole w kolumnie **Nazwa** i w wyświetlonej ramce wpisz nazwę nowej klasy alarmu, a w razie potrzeby wybierz inny rodzaj potwierdzania z listy rozwijanej w kolumnie **Potwierdzanie**.

Stosowane są następujące typy potwierdzenia:

NO_ACK: Nie jest wymagane potwierdzenie alarmu przez użytkownika.

ACK_INT0: Użytkownik musi potwierdzić istniejący warunek alarmu (status „INT0”, alarm trwa).

Konfiguracja alarmów

ACK_OUTOF: Użytkownik musi potwierdzić alarm, który minął (status „OUTOF”, alarm zakończony).

ACK_ALL: Warunki dla alarmów trwających i minionych muszą być potwierdzone przez użytkownika.

W razie potrzeby można wprowadzić **Komentarz**.

Kolejne wpisy dla klas alarmów są dołączane zawsze na końcu listy.

Za pomocą przycisku **Usuń** można usunąć zaznaczony na liście wpis.

Działania przyporządkowane do klasy <nazwa klasy>:

Każdej klasie alarmu można przyporządkować listę działań, które zostaną wykonane w razie wystąpienia zdarzenia alarmowego.

Zaznacz na liście **Możliwe akcje** żądane działanie i przenieś je za pomocą przycisku „>” do pola **Wybrane akcje**. Za pomocą przycisku „>>” można jednocześnie wybrać wszystkie działania. Analogicznie za pomocą przycisków „<” lub „<<” można usunąć jedno lub wszystkie wybrane działania.

Jeśli wybrane działanie jest zaznaczone na liście, można za pomocą przycisku „...” otworzyć odpowiednie okno dialogowe, w którym można wpisać adres e-mail, ustawić drukarkę i zdefiniować tekst komunikatu dla każdego działania.

Obsługiwane są następujące typy działań:

Działanie	Opis	Ustawienia do wprowadzenia w odpowiednim oknie dialogowym:
Zapisz:	Zdarzenie alarmowe jest zapisywane wewnątrz, np. w celu zarejestrowania w pliku Log. Należy pamiętać: Plik ten musi być zdefiniowany w konfiguracji grupy alarmów!	Ustawienia te należy wprowadzić w konfiguracji zapisywania alarmów (patrz rozdz. 9.3.5).
Drukuj:	Komunikat zostanie wydrukowany na drukarce	Złącze drukarki: Wybierz jedną z drukarek zdefiniowanych w systemie lokalnym; wydruk tekstowy: tekst komunikatu (patrz niżej), który ma być wydrukowany Należy pamiętać: Ta funkcja jest niedostępna dla wizualizacji elementu docelowego!
Komunikat:	Otwiera się okienko komunikatu, w którym można wprowadzić tekst.	Komunikat: Tekst komunikatu (patrz niżej), który ma być wyświetlany w okienku komunikatu. Należy pamiętać: Ta funkcja jest niedostępna dla wizualizacji elementu docelowego!
E-Mail:	Wysyła wiadomość e-mail, zawierającą tekst komunikatu.	Od: Adres e-mail nadawcy; Do: Adres e-mail adresata; Temat: Tekst tematu; Wiadomość: Tekst komunikatu (patrz niżej); Serwer: Nazwa serwera e-mail.
Zmienna:	Status alarmu albo tekst komunikatu zostanie przypisany zmiennej z bieżącego projektu.	Zmienna: Nazwa zmiennej: Zmienną można wybrać korzystając z funkcji pomocy przy wpisywaniu (<F2>): Można wybrać zmienną boolean, jeśli status alarmu ma być pokazywany jako NORM=0 i INTO=1, zmienne typu całkowitego pokazują status alarmu NORM =0, INTO =1, ACK =2, OUTOF =4; zmiennej typu String można przypisać tekst komunikatu, który został zdefiniowany w polu

Działanie	Opis	Ustawienia do wprowadzenia w odpowiednim oknie dialogowym: Komunikat (patrz niżej).
Wykonaj:	W momencie wystąpienia zdarzenia alarmowego (patrz niżej) uruchamia zewnętrzny program.	Plik wykonywalny: Nazwa pliku, który ma być wykonany (np. notepad.exe); za pomocą przycisku „...” plik można wybrać z systemowego okna wyboru plików; Parametry: parametry odpowiadające plikowi exe, które mają być dołączone do wywołania.

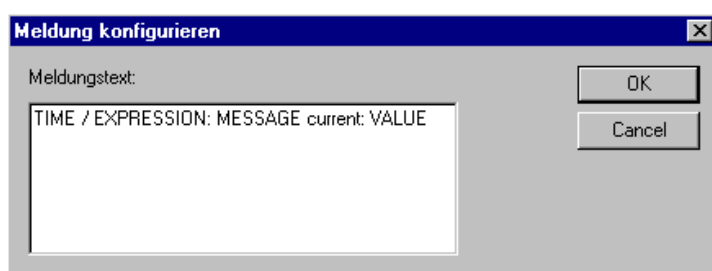
Definicja tekstu komunikatu:

Podczas konfiguracji działań „Komunikat”, „E-Mail”, „Drukuj”, „Zmienna” i ew. „Wykonaj” można zdefiniować tekst komunikatu, który w przypadku wystąpienia zdarzenia alarmowego (patrz niżej) zostanie wygenerowany. Podziały wierszy można wstawiać za pomocą kombinacji <Ctrl> + <Enter>. Możliwe jest korzystanie z następujących symboli zastępczych:

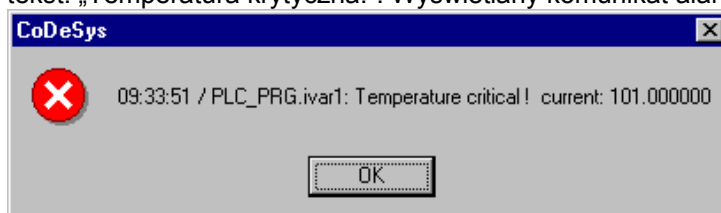
MESSAGE	Generuje komunikat zdefiniowany dla alarmu w konfiguracji grupy alarmów (kolumna Komunikat).
DATE	Data powrotu do odpowiedniego statusu (INTO)
TIME	Generuje godzinę wystąpienia alarmu
EXPRESSION	Wyrażenie (definiowane w grupie alarmów), które wygenerowało alarm
PRIORITY	Priorytet alarmu (definiowany w grupie alarmów)
VALUE	Bieżąca wartość wyrażenia
TYPE	Typ alarmu (definiowany w grupie alarmów)
CLASS	Klasa alarmu (definiowana w grupie alarmów)
TARGETVALUE	Wartość celu przy typach alarmu DEV+ i DEV- (definiowana w grupie alarmów)
DEADBAND	Tolerancja alarmu (definiowana w grupie alarmów)
ALLDEFAULT	generuje wszystkie dane dla alarmu, analogicznie jak wydruk do pliku zapisu (Historia)

Przykład:

Zdefiniuj komunikat (działanie „Komunikat”) wpisując następujący tekst w oknie komunikatów:



Ponadto wprowadź w definicji alarmu w grupie alarmów w polu tabeli „Komunikat” następujący tekst: „Temperatura krytyczna!”. Wyświetlany komunikat alarmu ma wówczas taką postać:



Wskazówka: Tekst komunikatu może być uwzględniony również przy zmianie języka projektu, jeśli występuje w pliku *.vis lub pliku kompilacji *.tlt. Aby wprowadzić tekst do pliku kompilacji *.tlt należy odpowiedni ciąg znaków – tak jak wszystkie teksty wizualizacji – wprowadzić między znaki „#” (np. dla ww. przykładu: „#Temperatura krytyczna!#” i „TIME /EXPRESSION: MESSAGE #current#: VALUE”, by odpowiednie składniki tekstu otrzymać w pliku kompilacji jako ALARMTEXT_ITEM.)

Plik zapisu dla działania „Zapisz” definiowany w konfiguracji grupy alarmów.

Zdarzenia dla akcji:

Dla każdego działania ustala się, przy jakich **zdarzeniach alarmowych** zostanie ono uruchomione.

Zaznacz żądane zdarzenia:

- INTO Występuje alarm.
- ACK Użytkownik potwierdził alarm.
- OUTOF Stan alarmowy zakończony.

Kolory/mapy bitowe dla klasy:

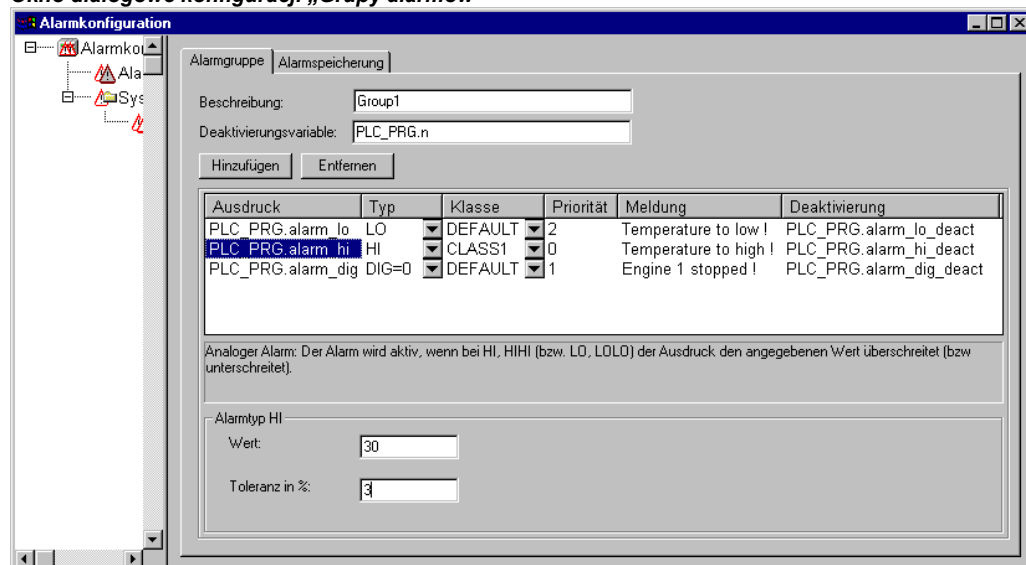
Każdej klasie alarmu można przypisać różne kolory i mapy bitowe, za pomocą których można rozróżniać alarmy przy wizualizacji tabeli alarmów. Wybierz **kolor pierwszego planu** i **kolor tła** dla możliwych zdarzeń alarmowych INTO, ACK i OUTOF (patrz wyżej). Po kliknięciu symbolu strzałki otwiera się systemowe okno dialogowe wyboru koloru albo okno wyboru mapy bitowej, jeśli kliknięty został szary kwadrat dla odpowiedniej klasy.

9.3.4 Grupa alarmów

Grupy alarmów umożliwiają tworzenie struktury różnych alarmów. Każdy alarm jest przypisany dokładnie jednej grupie alarmów i jest tam zarządzany. Wszystkie alarmy należące do jednej grupy mogą otrzymać jedną wspólną zmienną dezaktywującą i wspólne parametry odnośnie zapisywania alarmów. Grupa może więc służyć do tworzenia struktury dostępnych alarmów. Także pojedynczy alarm musi być skonfigurowany w grupie alarmów.

Hierarchiczna struktura grup alarmów w konfiguracji alarmu może być utworzona na bazie elementów drzewa folderów. Jeśli grupa alarmów jest zaznaczona w drzewie konfiguracji alarmów, automatycznie wyświetla się **okno dialogowe grupy alarmów**:

Okno dialogowe konfiguracji „Grupy alarmów”



W polu **Opis** można wprowadzić nazwę grupy alarmów.

Jako **zmienną dezaktywującą** można wpisać zmienną projektową typu boolean, która przy wzrastającym zboczu spowoduje dezaktywację wszystkich alarmów tej grupy, a przy zboczu opadającym spowoduje ich aktywowanie.

Za pomocą przycisku **Dodaj** można dodawać poszczególne alarmy należące do grupy, które są zdefiniowane przez następujące parametry:

Wyrażenie: Zmienna projektowa, do której odnosi się alarm. Przy wprowadzaniu konkretnej wartości można skorzystać z pomocy przy wpisywaniu (<F2>) lub funkcji Intellisense. Można również wpisać wyrażenie (np. „a + b”)

Typ: Możliwe jest korzystanie z następujących typów alarmów: (Należy pamiętać, że poniżej tabeli wyświetlane są komentarze dla poszczególnych wierszy).

DIG=0: alarm cyfrowy, jest aktywny, jeśli wyrażenie przyjmuje wartość FALSE.

DIG=1: alarm cyfrowy, jest aktywny, jeśli wyrażenie przyjmuje wartość TRUE.

LOLO: alarm analogowy, jest aktywny, jeśli wyrażenie ma wartość niższą niż wartość podana dla „typu alarmu LOLO”. Możliwe jest podanie tolerancji w procentach wartości. Tak długo jak wartość wyrażenia pozostaje w obszarze tolerancji nie jest generowany alarm, nawet po przekroczeniu dolnej granicy wartości LOLO.

LO: odpowiednio jako LOLO

HI: Alarm analogowy, jest aktywny, jeśli wyrażenie ma wartość wyższą niż wartość podana dla „typu alarmu HIHI”. Możliwe jest podanie tolerancji w procentach wartości. Tak długo jak wartość wyrażenia pozostaje w obszarze tolerancji nie jest generowany alarm, nawet jeśli przekroczona zostanie górna granica wartości HI.

HIHI: odpowiednio jako HI

DEV-: Dewiacja (odchylenie od wartości docelowej); alarm jest aktywny, jeśli wyrażenie ma wartość niższą niż wartość docelowa podana dla „typu alarmu DEV-” + odchylenie procentowe. $\text{Odchylenie procentowe} = \text{wartość docelowa} * (\text{odchylenie w \%}) / 100$.

DEV+: Dewiacja (odchylenie od wartości docelowej); alarm jest aktywny, jeśli wyrażenie ma wartość wyższą niż wartość docelowa podana dla „typu alarmu DEV+” + odchylenie procentowe. $\text{Odchylenie procentowe} = \text{wartość docelowa} * (\text{odchylenie w \%}) / 100$.

ROC: Rate of change (szybkość zmian w jednostce czasu). Alarm jest aktywny, gdy wyrażenie uległo zbyt dużej zmianie w stosunku do poprzedniej wartości. Wartość graniczna intensywności zmiany wywołująca alarm jest definiowana poprzez ilość zmian jednostek (zmiana wartości) na sekundę, minutę lub godzinę.

Klasa: Wybierz żadaną klasę alarmu. Z dostępnej tu listy można wybrać klasy zdefiniowane w konfiguracji klas alarmów przed ostatnim zapisaniem projektu.

Priorytet: W tym miejscu można nadać priorytet od 0 do 255, przy czym 0 ma najwyższy priorytet. Priorytet ma znaczenie przy sortowaniu tabeli alarmów.

Komunikat: W tym miejscu można zdefiniować tekst komunikatu, który będzie generowany w tabeli alarmów albo poprzez makro „MESSAGE” w poszczególnych działaniach. Okienko to musi być potwierdzone przez użytkownika przyciskiem OK, co jednak nie oznacza automatycznego potwierdzenia alarmu! W celu potwierdzenia alarmu należy otworzyć tabelę alarmów. Tabela jest dostępna poprzez element wizualizacji „Tabela alarmów”, albo poprzez datę wpisu alarmu w pliku zapisu, który może być generowany opcjonalnie.

Dezaktywacja W tym miejscu można wpisać zmienną projektową, która przy wzrastającym zboczu spowoduje dezaktywację alarmu. Należy jednak pamiętać, że to ustawienie jest nadpisywane przez wpis, który mógł zostać wprowadzony w polu „Zmienna dezaktywująca” (patrz wyżej).

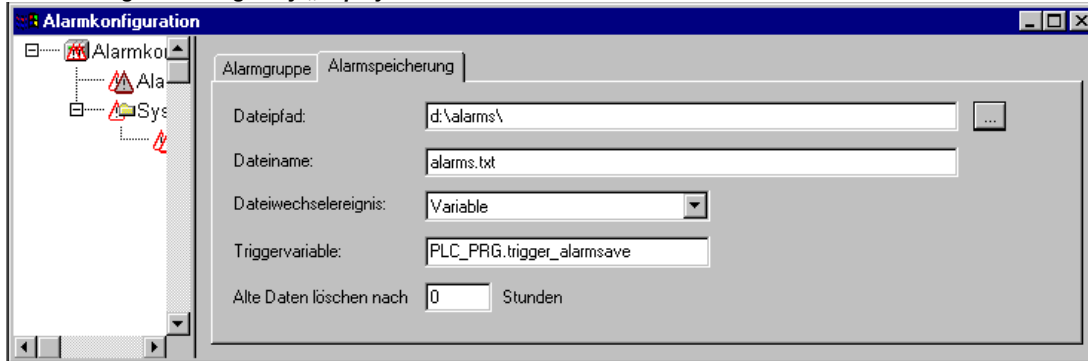
9.3.5 Zapisywanie alarmów

Dla każdej grupy alarmów można zdefiniować plik, w którym zapisywane będą zdarzenia alarmowe, o ile (!) na liście działań dla odpowiedniej klasy zostało aktywowane działanie „Zapisz”.

Konfiguracja alarmów

Zaznacz grupę alarmów w drzewie konfiguracji alarmów i wybierz zakładkę „Zapisywanie alarmów”:

Okno dialogowe konfiguracji „Zapisywanie alarmów”



Możliwe są następujące wprowadzenia:

Ścieżka: Ścieżka do katalogu zawierającego plik podany w polu Nazwa pliku; za pomocą przycisku „...” można otworzyć systemowe okno dialogowe wyboru katalogu.

Nazwa pliku: Nazwa pliku, w którym mają być zapisywane zdarzenia alarmowe (np. „alarmlog”). Plik jest tworzony wówczas automatycznie z podaną tutaj nazwą oraz dodatkową cyfrą i rozszerzeniem „.alm”. Cyfra określa wersję pliku log. Pierwszy plik zapisu jest oznaczony cyfrą 0, natomiast kolejne pliki powstające ze względu na następstwo warunków zdefiniowanych w polu „Zdarzenia zmiany pliku” otrzymują numery 1, 2 itd. (przykłady -> „alarmlog0.alm”, „alarmlog1.alm”). Format pliku zapisu można zdefiniować w oknie dialogowym „Ustawienia ram dokumentacji”.

Zdarz. zmiany pliku: W tym miejscu można wprowadzić warunek, na podstawie którego ma być utworzony nowy plik zapisu. Możliwe warunki: Nigdy, po godzinie, po jednym dniu, po tygodniu, po miesiącu, po wystąpieniu zbocza wzrastającego zmiennej w polu „Zmienna wyzwalająca”, po osiągnięciu określonej ilości zdarzeń w polu „Maksymalna liczba wpisów”.

Zmienna wyzwalająca lub Maksymalna liczba wpisów: patrz Zdarzenia zmiany pliku:

Usuń stare dane po... godzinach: Liczba dni po dacie utworzenia, po których wszystkie pliki zapisu alarmów zostaną usunięte poza plikiem aktualnym.

Plik zapisu (Historia) zawiera następujące wpisy:

Data/czas w DWORD Priorytet	Data	Czas	Zdarzenie	Wyrażenie	Typ alarmu	Wartość graniczna	Tolerancja	Wartość bieżąca	Klasa Komunikat	
1046963332	6.3.03	16:08:52	INTO	PLC_PRG.b	LO-30	5-31	Alarm_high	0	cl1	3 Komunikat1
1046963333	6.3.03	16:08:53	ACK	PLC_PRG.n	HIHI	35	Ostrzeżenie	9	cl3	0 Komunikat2

Przykład:

```
1046963332,6.3.03 16:08:52,INTO,PLC_PRG.ivar5,HIHI,,,, 9.00,a_class2,0,
1046963333,6.3.03 16:08:53,INTO,PLC_PRG.ivar4,ROC,2,,, 6.00,a_class2,2,
1046963333,6.3.03 16:08:53,INTO,PLC_PRG.ivar3,DEV,,,, -6.00,a_class2,5,
1046963334,6.3.03 16:08:54,INTO,PLC_PRG.ivar2,LOLO,-35,,3, -47.00, warning, 10, warning: low temperature !
1046963334,6.3.03 16:08:54,INTO,PLC_PRG.ivar1,HI,20,,5, 47.00,a_class1,2,temperature to high ! Acknowledge !
```

9.3.6 Menu Dodatki: Ustawienia

Kategoria Data/czas:

W tym miejscu można zdefiniować, w jakim formacie daty i godziny mają być pokazywane alarmy w pliku zapisu. Wprowadź formaty zgodnie ze składnią; myślniki i dwukropki należy wprowadzać w pojedynczych cudzysłowach:

dla daty: dd-'MM'-'yyyy -> np. „12. sty-1993”

dla godziny: hh':'mm':'ss -> np. „11:10:34”

Język:

W tym miejscu wybierz plik języka, który ma być używany do zmiany języka i zawiera również teksty konfiguracji alarmów. Patrz również następujące opisy:

- Wizualizacja, Ustawianie języka
- Kompilacja projektu na inny język

9.4 Zarządzanie biblioteką...

Menedżer bibliotek

Menedżer bibliotek pokazuje wszystkie biblioteki, które są podłączone do bieżącego projektu. Moduły, typy danych i zmienne globalne bibliotek mogą być używane jak moduły, typy danych i zmienne globalne zdefiniowane przez użytkownika.

Menedżera bibliotek można otworzyć poleceniem „Okno” „Zarządzanie biblioteką” lub w zakładce „Zasoby”. Informacja o powiązanych bibliotekach jest przechowywana razem z projektem. Aby ją odczytać, należy w menedżerze bibliotek zaznaczyć odpowiedni wpis i wykonać polecenie „Dodatki” „Właściwości”.

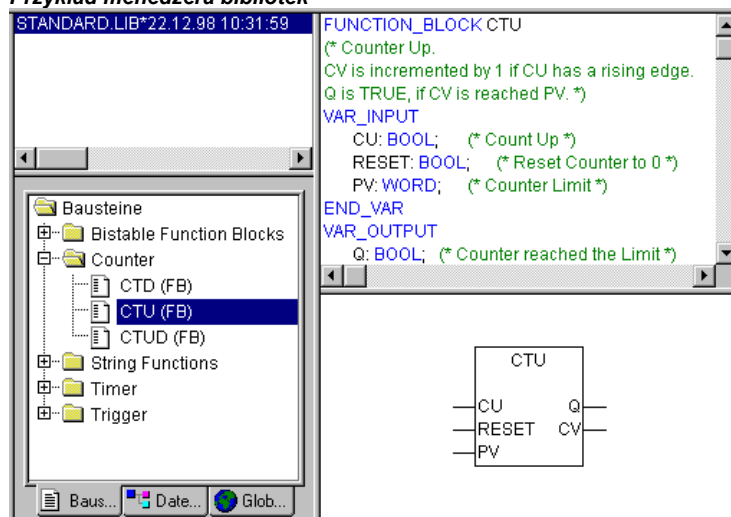
W bibliotekach utworzonych w CoDeSys można w części z deklaracją dołączyć dyrektywy pragma, które powodują, że podczas późniejszego używania biblioteki w projekcie w menedżerze bibliotek nie wyświetla się kompletna część z deklaracją. Dzięki temu niektóre deklaracje zmiennych lub komentarze mogą być niewidoczne dla użytkownika (patrz rozdział 8.2.3, Dyrektywy pragma w edytorze deklaracji).

Używanie menedżera bibliotek

Okno menedżera bibliotek jest podzielone separatorem ekranu na trzy albo cztery obszary. W lewym górnym obszarze jest wyświetlona lista podłączonych do projektu bibliotek.

W obszarze niżej – zależnie od wybranej zakładki – znajdują się **moduły**, **typy danych**, **wizualizacje** lub **zmienne globalne** należące do wyżej wybranej biblioteki.

Przykład menedżera bibliotek



Strukturę folderów można rozwijać i zwiąć również poprzez dwukrotne kliknięcie wiersza lub naciśnięcie klawisza <Enter>. Foldery zwinięte mają przed nazwą folderu znak plus, foldery rozwinięte są oznaczone znakiem minus.

Po zaznaczeniu modułu kliknięciem myszy lub za pomocą klawiszy strzałki w prawym obszarze menedżera bibliotek pojawia się na górze deklaracja modułu, a na dole obraz graficzny modułu z wejściami i wyjściami.

W przypadku typów danych i zmiennych globalnych w prawym obszarze menedżera bibliotek pokazywana jest deklaracja.

Biblioteka standardowa

Biblioteka „standard.lib” jest dostępna w standardowej wersji programu. Zawiera ona wszystkie funkcje i moduły funkcji, które są wymagane dla systemu programowania IEC jako standardowe moduły IEC61131-3. Różnica między funkcją standardową i operatorem polega na tym, że operator jest pośrednio rozpoznawalny w systemie programowania, podczas gdy moduły standardowe muszą być podłączone do projektu jako biblioteka (standard.lib).

Kod źródłowy tych modułów jest dostępny w bibliotece C i stanowi składnik systemu CoDeSys.

Biblioteki użytkownika

Projekt można zapisać jako bibliotekę używając polecenia „**Zapisz jako...**” w menu „**Plik**”. Projekt jako taki pozostaje niezmieniony; jest tworzony dodatkowy plik rozszerzeniem domyślnym „.lib”, który będzie dostępny potem jak np. biblioteka standardowa pod podaną tutaj nazwą.

Projekt można zapisać jako **wewnętrzną bibliotekę *.lib**, aby umożliwić wykorzystanie zawartych w nim modułów w innych projektach. Można ją wtedy jak np. bibliotekę standard.lib podłączyć do innych projektów przy użyciu menedżera bibliotek.

Wskazówka: Warto pamiętać, że wykorzystując dyrektywy pragma można zdefiniować, jakie części deklaracji biblioteki mają być później dostępne do wglądu w menedżerze bibliotek po podłączeniu biblioteki do projektu („Ukrywanie” deklaracji zmiennych, patrz rozdział 8.2.3).

Chcąc podłączyć do projektu jako bibliotekę moduły zaimplementowane w innych językach programowania, np. C, należy podczas zapisywania projektu wybrać typ pliku **Zewnętrzna biblioteka *.lib**. W wyniku tego dla utworzonego pliku biblioteki zostanie zapisany dodatkowy plik, noszący również nazwę pliku biblioteki, jednakże o rozszerzeniu „.h”. Plik ten ma strukturę pliku C-Header i zawiera deklaracje wszystkich modułów dostępnych w bibliotece, typów danych oraz zmiennych globalnych. Jeśli w projekcie jest używana biblioteka zewnętrzna, podczas symulacji przeprowadzana jest implementacja, zapisana w odniesieniu do modułów w programie CoDeSys. Natomiast na systemie docelowym przetwarzana jest implementacja napisana w języku C.

Jeśli korzystanie z biblioteki ma być uwarunkowane obowiązkiem uzyskania licencji, należy kliknąć przycisk **Edytuj informację o licencji** i wprowadzić w oknie dialogowym „Edytuj informacje licencyjne” odpowiednie dane. Patrz opis polecenia „Projekt” „Zapisz jako...” oraz Zarządzanie licencjami w CoDeSys.

„Wstaw” „Dodaj bibliotekę”

Za pomocą tego polecenia można podłączyć do projektu dodatkową bibliotekę.

Otworzy się okno dialogowe otwierania pliku. Jeśli ustawiony bieżący folder nie zawiera żądanej biblioteki, wówczas w polu **Katalog biblioteki** można spośród katalogów dostępnych w „Projekt/Opcje/Foldery/Biblioteki” (patrz rozdział 6.2) wybrać żądany katalog i wyświetlić zawarte tam pliki bibliotek (typ pliku „*.lib”). Wybierz jedną lub kilka bibliotek i potwierdź przyciskiem OK. Okno dialogowe zamyka się i biblioteka jest wstawiana do menedżera bibliotek. Obiekty biblioteki można teraz używać jak zdefiniowane własne obiekty.

Ścieżki biblioteki:

Niezwykle ważnym jest, by sprawdzać, jakie katalogi biblioteki są aktualnie zdefiniowane w opcjach projektu. Jeśli zostanie wstawiona biblioteka z katalogu, który nie jest tam podany, w polu będzie wpisana kompletna ścieżka dostępu do tej biblioteki.

Przykład: Podłączono bibliotekę standard.lib z folderu „D:\codesys\libraries\standard”.

- Jeśli katalog ten jest zdefiniowany w opcjach projektu, wówczas w menedżerze bibliotek będzie wpisane: „standard.lib <data i godzina utworzenia pliku>”.
- Jeśli w opcjach projektu jest zdefiniowany tylko folder „D:\codesys\libraries” wówczas będzie wpisane: „standard\standard.lib <data i godzina utworzenia pliku>”.

- Jeśli w opcjach projektu nie został zdefiniowany żaden pasujący folder, wówczas będzie wpisana kompletna bezwzględna ścieżka dostępu: „D:\codesys\libraries\standard\standard.lib <data i godzina utworzenia pliku>”.

Podczas otwierania projektu biblioteki są wyszukiwane według wpisów w menedżerze bibliotek. Przykładowo biblioteka, która została wpisana bez ścieżki dostępu, jest wyszukiwana w katalogach bibliotek zdefiniowanych w opcjach projektu.

Jeśli podczas otwierania pliku nie zostały odnalezione biblioteki, użytkownik zostanie najpierw zapytany, czy chce zmienić katalog ustawiony w opcjach projektu. W przypadku odrzucenia zapytania pojawi się okno dialogowe z informacją o nieodnalezionych bibliotekach, a odpowiadające im wpisy w menedżerze bibliotek będą wyświetlane na czerwono. W takim przypadku można zaznaczyć czerwony wpis i z menu kontekstowego wybrać polecenie **Znajdź...** Zostanie otwarte okno dialogowe otwierania pliku umożliwiające również bezpośrednie załadowanie brakującej biblioteki.

Licencja:

Podczas wstawiania biblioteki licencjonowanej w niektórych sytuacjach może pojawić się komunikat, że biblioteka jest dostępna tylko w wersji demonstracyjnej i że nie funkcjonuje dla aktualnie ustawionego systemu docelowego. Komunikat ten można zignorować albo podjąć natychmiastowe kroki w celu uzyskania licencji. Używanie bibliotek bez ważnej licencji powoduje podczas kompilacji projektu („Projekt” „Kompiluj”) powstanie błędu. Komunikat błędu można kliknąć dwukrotnie albo użyć klawisza <F4>. Wyświetli się wówczas okno dialogowe „Informacja o licencji”, w którym za pomocą Kreatora można wprowadzić odpowiednie dane niezbędne dla otrzymania licencji.

Usuwanie bibliotek

Za pomocą polecenia „Edycja” „Usuń” można usunąć bibliotekę z projektu i menedżera bibliotek.

„Dodatki” „Właściwości”

To polecenie otwiera okno dialogowe „Informacje o bibliotece wewnętrznej/zewnętrznej”. Dla bibliotek wewnętrznych zawiera ono wyłącznie statystykę danych, które są wprowadzane jako informacje o projekcie podczas tworzenia biblioteki; m.in. również zawierają informację o licencji. Dla bibliotek zewnętrznych zawiera ono nazwę i ścieżkę dostępu do biblioteki.

9.5 Dziennik...

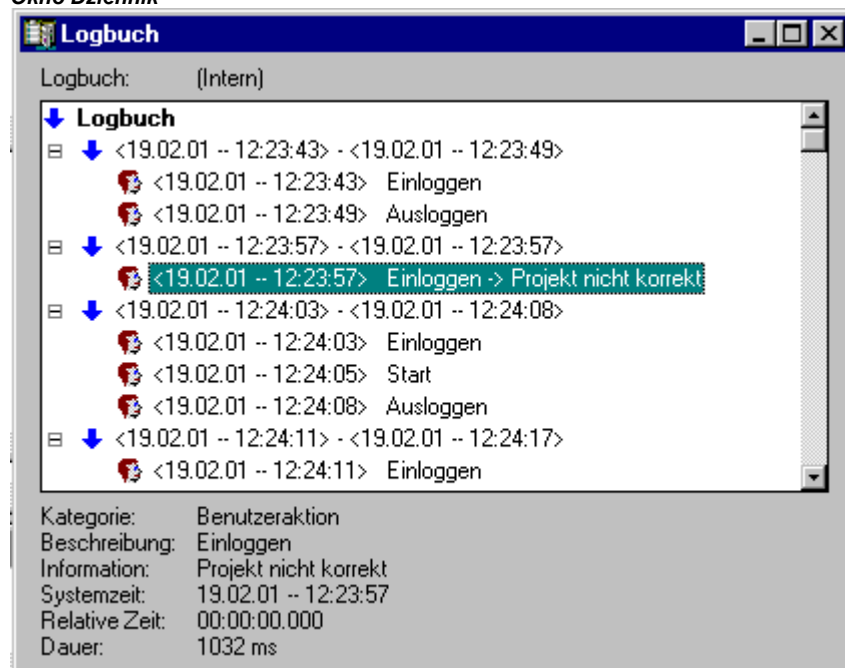
W dzienniku zapisywane są w porządku chronologicznym działania, występujące podczas sesji online. Odbywa się to w ten sposób, że do każdego projektu zakładany jest binarny plik dziennika (*.log). Ponadto istnieje możliwość zapisywania wyciągów pliku dziennika danego projektu w innym zewnętrznym pliku dziennika.

Okno dziennika można otwierać w trybie offline i online i tym samym prowadzić bezpośredni monitoring w trybie online.

„Okno” „Dziennik”

By otworzyć, wybierz punkt menu „Okno” „Dziennik” lub odpowiedni wpis w oknie zakładki Zasoby.

Okno Dziennik



W oknie wpisów dziennika za wierszem **Dziennik**: znajduje się nazwa pliku aktualnie pokazywanego dziennika. Jeśli jest to dziennik bieżącego projektu, występuje dodatkowo informacja „(Wewnętrzny)”.

W oknie wpisów wyświetlane są zarejestrowane wpisy. Najnowszy wpis znajduje się zawsze na końcu listy.

Są tam pokazywane są tylko czynności z kategorii zaznaczonych jako aktywne w menu „Projekt” „Opcje” „Dziennik” w obszarze „Filtr”!

Poniżej okna wpisów znajdują się informacje dotyczące aktualnie zaznaczonego wpisu:

Kategoria: Kategoria danego wpisu dziennika. Możliwe są następujące cztery kategorie:

- Czynność użytkownika: Użytkownik przeprowadził czynność w trybie online (typowa czynność z menu Online).
- Czynność wewnętrzna: Została wykonana wewnętrzna czynność w trybie online (np. *Delete Buffers* lub *Init Debugging*).
- Zmiana statusu: Status systemu czasu przebiegu uległ zmianie (np. ze stanu *Running* do stanu *Break*, w przypadku osiągnięcia punktu wstrzymania).
- Wyjątek: Wystąpił wyjątek, np. błąd komunikacji.

Opis: Rodzaj czynności. Czynności użytkownika nazywają się tak samo jak polecenia menu, które je uruchamiają, wszystkie czynności są zapisywane w języku angielskim i nazywają się podobnie jak odpowiadająca im funkcja `OnlineXXX()`.

Informacja: To pole zawiera opis błędu, jaki prawdopodobnie wystąpił podczas wykonywania czynności. Pole jest puste, jeśli nie wystąpił żaden błąd.

Czas systemowy: Czas systemowy opisujący chwilę początku czynności; z dokładnością do jednej sekundy.

Czas względny: Czas względny liczony do początku sesji online; z dokładnością do jednej milisekundy.

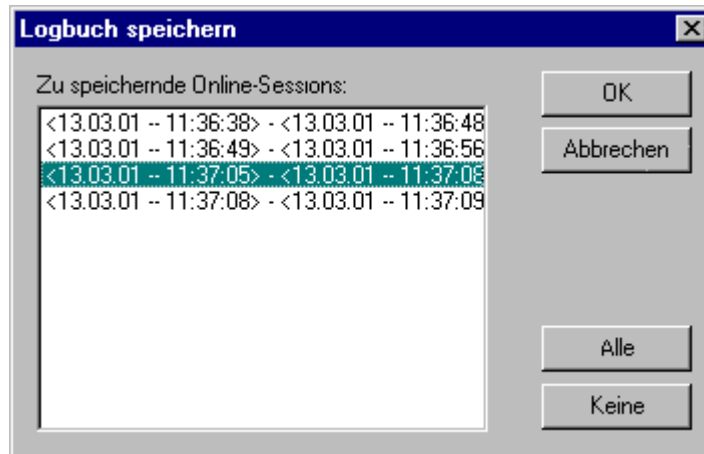
Okres: Czas trwania czynności w milisekundach.

Menu Dziennik

Jeśli w oknie dziennika znajduje się fokus wprowadzania, na pasku menu zamiast punktów „Dodatki” i „Opcje” jest widoczna opcja menu **Dziennik**.

Menu oferuje następujące punkty:

- Wczytaj...** W standardowym oknie otwierania pliku można wczytać i wyświetlić zewnętrzny plik dziennika *.log.
Użycie polecenia nie powoduje zastąpienia przez nadpisanie dziennika znajdującego się w projekcie. Jeśli okno dziennika zostanie zamknięte, a następnie ponownie otwarte albo zostanie uruchomiona nowa sesja online, wówczas wczytana wersja zostanie zastąpiona przez dziennik projektu.
- Zapisz...** Ten punkt menu można wybrać tylko wtedy, gdy aktualnie pokazywany jest dziennik projektu. Umożliwia on zapisywanie do zewnętrznego pliku wyciągu z dziennika projektu. Podczas takiej operacji wyświetla się następujące okno dialogowe, w którym można wybrać sesje online do zapisania.



Po dokonaniu wyboru otwiera się systemowe okno dialogowe zapisywania pliku („Zapisz dziennik”).

- Pokaż dziennik projektu** To polecenie można wybrać tylko wtedy, gdy aktualnie pokazywany jest zewnętrzny dziennik. Powoduje ono powrót do dziennika projektu.


Zapisywanie dziennika projektu

Niezależnie od możliwości zapisania dziennika w zewnętrznym pliku dziennik projektu jest zapisywany automatycznie w pliku binarnym <nazwa projektu>.log. Jeśli w oknie dialogowym „Projekt” „Opcje” „Dziennik” nie podano specjalnej ścieżki folderu, dziennik zapisuje się automatycznie w tym samym katalogu co projekt.

W oknie dialogowym „Projekt” „Opcje” „Dziennik” można ustawić maksymalną liczbę sesji online, które można zapisać. Jeśli podczas zapisywania liczba ta zostanie przekroczona, w miejscu najnowszej sesji z bufora zapisów usuwana jest każdorazowo najstarsza sesja.

9.6 Konfiguracja sterownika

9.6.1 Przegląd

 Konfiguracja sterownika jako obiekt znajduje się w oknie zakładki **Zasoby** w obiekcie Organizera. Za pomocą dostępnego tam edytora konfiguracji można opisać urządzenie docelowe, obsługujące otwarty projekt. Przy tworzeniu programu szczególnie ważne są liczba i położenia wejść i wyjść. Na podstawie tego opisu CoDeSys sprawdza, czy używane w programie adresy IEC rzeczywiście występują również w urządzeniu.

Konfiguracja sterownika

Niezbędne do pracy w edytorze konfiguracji są pliki konfiguracji (*.cfg, patrz niżej „Wskazówki dotyczące kompatybilności”) i pliki urządzenia (np. *.gsd, *.eds), które są przechowywane w katalogu zdefiniowanym w pliku docelowym (patrz Ustawienia systemu docelowego) względnie w opcjach projektu, a odczytywane podczas uruchamiania projektu. W katalogach tych można zapisywać pliki w czasie pracy programu.

Plik konfiguracyjne *.cfg opisuje konfigurację podstawową, która jest wyświetlana automatycznie podczas otwierania edytora i która określa, jakie parametry można jeszcze definiować w edytorze.

Uwaga: W przypadku zmiany podstawowego pliku konfiguracyjnego należy ponownie utworzyć konfigurację w CoDeSys!

Wskazówka dotycząca kompatybilności: W CoDeSys wersji 2.2 został zaimplementowany **nowy format konfiguracji sterownika** z podstawowymi plikami konfiguracyjnymi o rozszerzeniu *.cfg. Konfigurator sterownika używany przy programowaniu projektów w starszej wersji bazował na plikach konfiguracyjnych, które posiadały rozszerzenie *.con. W systemie docelowym można wprowadzić takie ustawienie, że „stary” konfigurator będzie dalej używany, nawet jeśli w wersji 2.2 lub wyższej zostanie otwarty stary projekt (patrz wskazówki niżej). Nie trzeba wtedy tworzyć nowych plików konfiguracyjnych, gdyż pliki *con mogą być dalej używane bez zmian. Jeśli jednak system docelowy nie obsługuje starego konfiguratora, wówczas starą konfigurację sterownika, która jest zapisana w projekcie, można przekonwertować do nowego formatu przy użyciu odpowiedniego pliku *.cfg. Zobacz rozdz. 9.6.3, „Dodatki” Przekształć”.

Konfigurator sterownika w CoDeSys umożliwia podłączanie prostych modułów We/Wy, jak również modułów CAN, Profibus i DeviceNet.

W zależności od stosowanego systemu docelowego istnieje możliwość odczytu ze sterownika następujących informacji: 1. skan struktury bieżącego sterownika, które można wykorzystać bezpośrednio do konfiguracji sterownika; 2. komunikaty diagnozy i statusu, które mogą być wyświetlane w CoDeSys.

Po dokonaniu wszystkich ustawień konfiguracji w edytorze i uruchomieniu pobierania projektu zostanie wysłany do sterownika binarny obraz projektu.

System docelowy może być ewentualnie skonfigurowany w taki sposób, by dalej pracować z **konfiguracją sterownika używaną w CoDeSys wersji 2.1**. W takim przypadku podczas pracy z konfiguratorem obowiązuje dokumentacja użytkownika dla CoDeSys wersja 2.1, przy czym należy pamiętać o następujących modyfikacjach konfiguratora:

Konfigurator CAN:

Opcja „Generuj wszystkie SDO” w oknie dialogowym „Parametry CAN” modułu CAN (patrz rozdz. 9.6.8)

Uwaga: Generowanie SDO odbywa się teraz zawsze według mechanizmu używanego w wersji konfiguratora 2.3, tzn. w pewnych okolicznościach rezultat może być inny niż dotychczas!

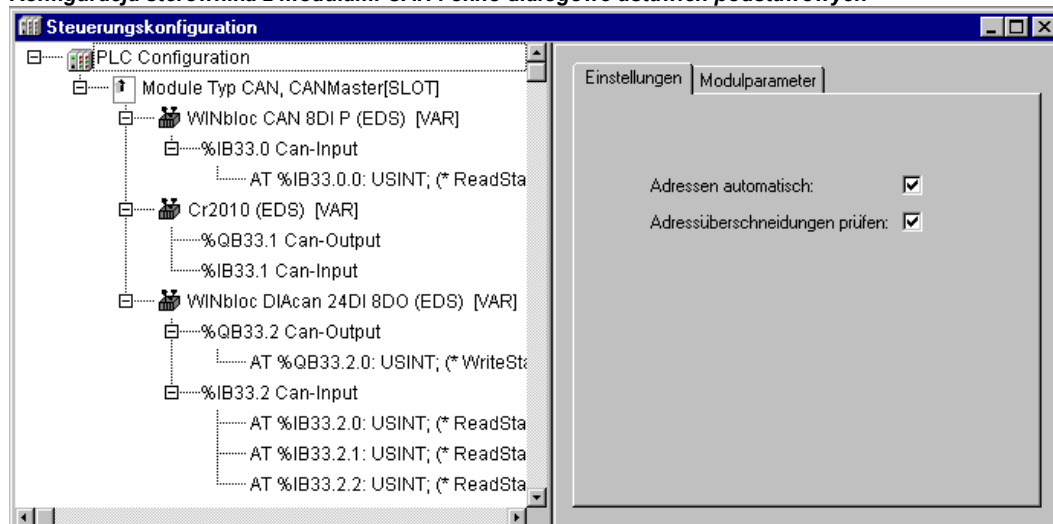
Pole edycji „Typ urządzenia” w oknie dialogowym „Ustawienia CAN” modułu CanDevice (patrz rozdz. 9.6.9).

Konfigurator Profibus:

Lista rozwijana zawierająca dostępne do wstawiania moduły Profibus (patrz rozdz. 9.6.7) jest teraz posortowana alfabetycznie wg nazw modułów.

9.6.2 Używanie konfiguratora sterownika w CoDeSys

Konfiguracja sterownika z modułami CAN i okno dialogowe ustawień podstawowych



Konfiguracja jest przedstawiona w edytorze w formie struktury drzewa i można ją edytować za pomocą poleceń menu i dialogów. Zawiera ona elementy, które mogą być używane zarówno jako wejścia lub wyjścia albo elementy do zarządzania, które ze swojej strony posiadają również elementy podrzędne (np. magistrala CAN, PROFIBUS lub cyfrowa karta wejść z 8 wejściami).

Wejścia i wyjścia są pokazywane w edytorze z adresami IEC, za pomocą których można je wybierać. Zasadniczo dla każdego wejścia i wyjścia można nadawać symboliczne nazwy, które są również pokazywane przed adresem IEC.

Jeśli otwarty projekt zawiera konfigurację sterownika utworzoną w starszej wersji CoDeSys, można ją przekształcić do nowego formatu „Jednolitej konfiguracji sterownika”.

Okno edytora konfiguracji składa się z dwóch części: Z lewej strony wyświetlane jest „**drzewo konfiguracji**”. Pierwotna struktura i zawartość wynika z definicji pliku konfiguracyjnego (konfiguracja standardowa). Podczas użytkowania programu struktura i zawartość zmieniają się poprzez dodawanie kolejnych konfiguracji w projekcie. Z prawej strony dla zaznaczonego elementu drzewa konfiguracji wyświetlane są odpowiednie **dialogi konfiguracji** na jednej lub kilku kartach zakładek.

Dialogi konfiguracji jest domyślnie aktywne, ale można je wyłączyć za pomocą polecenia „**Dodatki**” „**Właściwości**”.

W nagłówku drzewa konfiguracji znajduje się tzw. **moduł główny (root)**, który zawiera nazwę, jaka mu została nadana w pliku konfiguracji. Poniżej, z wcięciem, występują hierarchicznie kolejne elementy konfiguracji: moduły różnych typów (CAN, PROFIBUS, We/Wy), kanały lub kanały bitowe.

Zaznaczanie elementów

Zaznaczanie elementów odbywa się kliknięciem myszy odpowiedniego elementu albo poprzez przesunięcie za pomocą klawiszy strzałek wypunktowanego prostokąta w miejsce wybranego elementu.

Elementy, przed którymi znajduje się znak plus, są elementami organizacyjnymi i zawierają elementy podrzędne. W celu rozwinięcia struktury zaznacz element i kliknij dwukrotnie znak plus lub naciśnij klawisz <Enter>. W ten sam sposób można zwinąć rozwiniętą strukturę elementów (znak minus przed elementem).

Dodawanie elementów, „Wstaw” „Dodaj element”, „Wstaw” „Załącz element podrzędny”

W zależności od ustawień w plikach konfiguracyjnych i plikach urządzeń, które są wczytywane podczas otwierania projektu, pewne elementy konfiguracji sterownika występują już w drzewie

konfiguracji. Jeśli któryś z tych elementów zostanie zaznaczony, można również dodawać kolejne elementy, jeśli jest to dozwolone zgodnie z definicjami w pliku konfiguracyjnym i jeśli występują wymagane pliki urządzeń. W tym celu dostępne są następujące polecenia:

- Menu „Wstaw” „Wstaw element”: Można wybrać element i wstawić przed zaznaczonym elementem.
- Menu „Wstaw” „Załącz element podrzędny”: Można wybrać element i wstawić jako ostatni element podrzędny zaznaczonego elementu.

Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

Należy pamiętać: Jeśli system docelowy dopuszcza taką możliwość, można w celu dodania nowych modułów skorzystać ze „skanu” aktualnie włączonego urządzenia.

Zamienianie/przełączanie elementów, „Dodatki” „Zastąp element”

Jeśli definicja pliku konfiguracyjnego dopuszcza taką możliwość, można zastąpić zaznaczony element w drzewie konfiguracji innym elementem. Funkcja umożliwi również przełączanie kanałów, które są skonfigurowane w taki sposób, by można ich było używać jako wejścia lub wyjścia. Używanie polecenia „Dodatki” „Zastąp element”

Nadawanie nazw symbolicznych

Nazwy symboliczne dla modułów i kanałów mogą być zdefiniowane w pliku konfiguracji. Są one wtedy pokazywane w drzewie konfiguracji przed „AT” adresem IEC danego elementu. W pliku konfiguracyjnym jest również zdefiniowane, czy nazwa symboliczna może być edytowana względnie nadawana w edytorze konfiguracji. Po zaznaczeniu elementu i kliknięciu „AT” adresem IEC otworzy się pole edycji, w którym można wpisać nową nazwę symboliczną. Nazwy symboliczne można również edytować klikając je myszą.

Należy pamiętać, że nazwa symboliczna, jaka została nadana koresponduje z obowiązującą w projekcie deklaracją zmiennej!

Dodaj plik konfiguracyjny

Za pomocą tego polecenia menu „Dodatki” można dodać kolejny plik do plików konfiguracji. Jako pliki konfiguracyjne obowiązują w tym przypadku wszystkie pliki, jakie znajdują się w katalogach „plików konfiguracyjnych” zdefiniowanych w opcjach projektu.

W otwartym oknie dialogowym **Wybierz plik konfiguracji** można ustawić filtr wyboru plików CAN (*.eds, *. dcf), Profibus (gsd*.*) , plików konfiguracji (*.cfg) lub wszystkich plików (*.*). Po wyborze pliku odbywa się sprawdzenie, czy plik występuje już w jednym ze zdefiniowanych katalogów dla plików konfiguracyjnych. Jeśli tak, wyświetla się odpowiedni komunikat i pliku nie można dodać. Jeśli zostanie wybrany plik cfg, wyświetla się zwykle komunikat z ważnymi informacjami.

Jeśli plik można dodać, wyświetla się okno dialogowe **Wybierz katalog konfiguracyjny**, w którym znajdują się katalogi zdefiniowane dla projektu. Wybierz katalog, do którego chcesz skopiować plik. Po potwierdzeniu wyboru plik będzie bezpośrednio dostępny w konfiguracji sterownika.

Obliczanie nowych adresów modułów, „Dodatki” „Oblicz adresy”

Jeśli w ustawieniach podstawowych konfiguracji sterownika jest aktywna opcja „Przypisz adresy automatycznie”, można za pomocą polecenia „Dodatki” „Oblicz adresy” wykonać obliczenie nowych adresów dla wszystkich elementów następujących za zaznaczonym modułem.

Przywracanie konfiguracji domyślnej, „Dodatki” „Konfiguracja standardowa”

Używając polecenia „Dodatki” „Konfiguracja standardowa” można po wprowadzeniu zmian w edytorze konfiguracji przywrócić pierwotną konfigurację sterownika, która jest zapisana w projekcie jako plik konfiguracji *.cfg.

Uwaga: W pliku konfiguracji można w jednym wierszu wprowadzić ustawienie, że konfiguracja domyślna będzie **zawsze** przywracana podczas każdorazowego otwierania projektu. Wszystkie ustawienia wprowadzone przedtem w edytorze konfiguracji zostaną utracone!

Przekształcanie starych konfiguracji sterownika, „Dodatki” „Przekształć”

To polecenie jest dostępne w menu „Dodatki”, jeśli zostanie otwarty projekt zawierający konfigurację sterownika, utworzoną w wersji CoDeSys starszej niż 2.2 i jeśli w pliku docelowym nie zostało wprowadzone ustawienie, że ma być dalej używany konfigurator starszej wersji. Jeśli dostępne są wszystkie potrzebne pliki konfiguracji (**Uwaga: informacje w pliku *.con muszą być teraz zawarte w pliku konfiguracji *.cfg!**), można za pomocą polecenia „Przekształć” przekonwertować tę konfigurację do aktualnego formatu Jednolitej konfiguracji sterownika. Po wykonaniu polecenia wyświetli się okno dialogowe z zapytaniem „Czy chcesz przekształcić konfigurację sterownika na nowy format? **Uwaga:** konwersja wsteczna jest niemożliwa.”, na które trzeba odpowiedzieć Tak lub Nie. W przypadku potwierdzenia zapytania przyciskiem „Tak” edytor konfiguracji sterownika zostanie zamknięty i przy powtórnym otwarciu plik będzie wyświetlany w nowym formacie.

Uwaga: Po wykonaniu konwersji nie można przywrócić starej konfiguracji!

Eksportowanie i importowanie modułów

Jeśli w pliku konfiguracji (*.cfg) zdefiniowano, że moduł jest „eksportowalny”, wówczas w menu kontekstowym takiego modułu dostępne są polecenia „Eksportuj moduł” i „Importuj moduł”.

Po wybraniu polecenia „**Eksportuj moduł**” otwiera się okno dialogowe „**Wybierz plik eksportu**”. Można tu podać nazwę pliku XML, w jakim zostanie zapisany moduł ze wszystkimi modułami podrzędnymi i ich konfiguracjami. Plik ten można za pomocą polecenia „**Importuj moduł**” zaimportować w innej konfiguracji, jeśli jest tam zaznaczony odpowiednio zdefiniowany moduł.

W ten prosty sposób można przenosić drzewo konfiguracji poszczególnych modułów na inne konfiguracje sterowników.

9.6.3 Ustawienia podstawowe w konfiguracji sterownika

W drzewie konfiguracji zaznacz wpis „Konfiguracja sterownika” (odpowiada modułowi „root”). Otworzy się okno dialogowe **Ustawienia**:

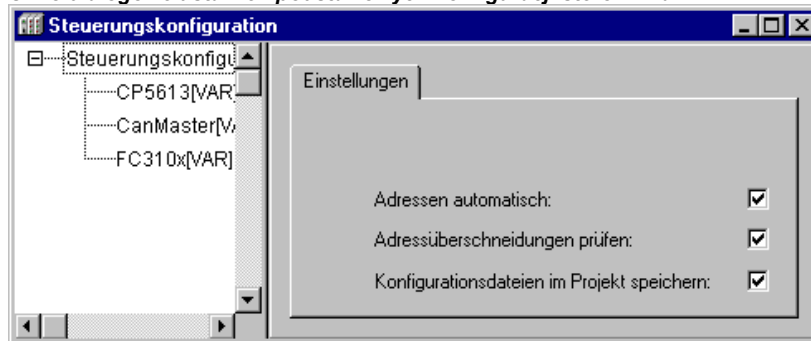
Przypisz adresy automatycznie: Każdy nowy moduł otrzymuje automatycznie adres, który wynika z adresu poprzednio wstawionego modułu i jego rozmiaru. Jeśli moduł zostanie usunięty z konfiguracji, wówczas adresy następujących po nim modułów zostaną automatycznie dopasowane. Za pomocą polecenia „Dodatki” „Oblicz adresy” można uruchomić obliczanie nowych adresów poczynawszy od bieżącego węzła (modułu).

Sprawdź skrzyżowania adresów: Podczas kompilacji projektu są sprawdzane i zgłaszane skrzyżowania adresów.

Zapisz pliki konfiguracyjne w projekcie: Informacja plików konfiguracyjnych *.cfg i plików urządzeń, na podstawie których została utworzona bieżąca konfiguracja sterownika, jest zapisana w projekcie. W ten sposób (**o ile w pliku konfiguracyjnym nie zostało zdefiniowane**, że ma być **zawsze** przywracana konfiguracja standardowa!) utworzona konfiguracja pozostaje zachowana również wtedy, jeśli podczas otwierania projektu pliki konfiguracyjne nie zostaną odnalezione. Jeśli opcja ta nie jest aktywna, wówczas w takim przypadku cała konfiguracja projektu zostanie utracona!

Zapisanie informacji o konfiguracji w projekcie gwarantuje, że zostaną one zachowane nawet w przypadku zmiany systemu docelowego. Należy jednak pamiętać o uwzględnieniu ewentualnych plików konfiguracyjnych utworzonych przez system docelowy.

Okno dialogowe ustawień podstawowych konfiguracji sterownika

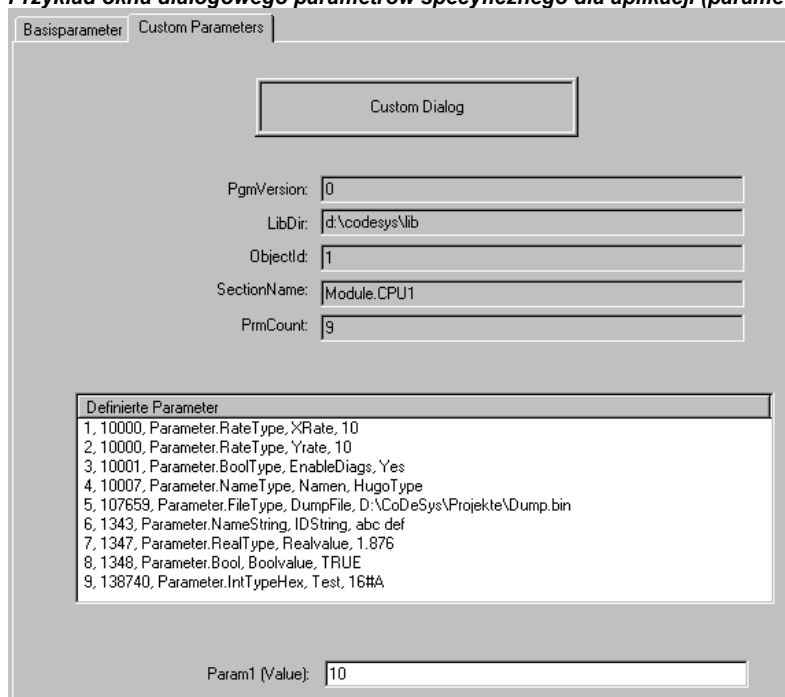


Moduł globalny służący do nadawania adresów (adresy płaskie / adresy wg Id) w obrębie konfiguracji sterownika jest zdefiniowany domyślnie w pliku konfiguracyjnym.

9.6.4 Okno dialogowe parametrów specyficzne dla aplikacji

Możliwości parametryzowania w konfiguratorze można rozbudować używając biblioteki DLL specyficznej dla aplikacji, czyli indywidualnego okna dialogowego. Używana do tego celu biblioteka dll „Hook” musi być przechowywana w tym samym katalogu co plik konfiguracyjny i zadeklarowana tam w wierszu przy opisie modułu albo klasy kanału. W takim przypadku dla konkretnego modułu albo odpowiednich kanałów domyślne okno dialogowe „Parametry modułu” zostanie zastąpione oknem dialogowym zdefiniowanym w bibliotece DLL.

Przykład okna dialogowego parametrów specyficznego dla aplikacji (parametry użytkownika)



9.6.5 Konfiguracja modułu wejścia/wyjścia...

Parametry podstawowe modułu wejścia/wyjścia

Jeśli w drzewie konfiguracji zaznaczony jest moduł wejścia/wyjścia, zostanie wyświetlone okno dialogowe „Parametry podstawowe” z następującymi wpisami:

Id modułu: Numer Id modułu jest jednoznaczna nazwą całego otoczenia konfiguracji. Jest on odczytywany z pliku konfiguracji i nie można go edytować.

Numer węzła: Numer węzła jest odczytywany z wpisu w pliku konfiguracji lub ewentualnie, jeśli tam nie ma żadnego wpisu, na podstawie lokalizacji w drzewie konfiguracji.

Adres wejściowy, adres wyjściowy, adres diagnostyczny: Adresy dla wejścia i wyjścia lub do zapisywania danych diagnostycznych.

Adresy te odnoszą się do modułu. W ustawieniach podstawowych i definicjach w pliku konfiguracyjnym określa się, które adresy będą predefiniowane, jaki tryb adresowania ma być używany w konfiguracji i czy adres można tutaj później edytować.

Okno dialogowe parametrów podstawowych dla modułu wejścia/wyjścia

The screenshot shows a dialog box titled 'Basisparameter'. It contains the following elements:

- Modul-Id: 1
- Knotennummer: 0
- Eingabeadresse: %IB0
- Ausgabeadresse: %QB0
- Diagnoseadresse: %MB4
- Kommentar: (empty text field)
- Adressen nicht automatisch ändern: (unchecked checkbox)
- Modulbeschreibung laden: (checked checkbox)

Komentarz: Można wpisać tutaj dodatkowe informacje dotyczące modułu.

Wczytaj opis modułu: Opcja ta jest dostępna tylko wtedy, jeśli zostanie tak zdefiniowana w pliku konfiguracji. W pliku cfg można również wprowadzić ustawienie domyślne. Jeśli ta opcja jest wyłączona, opis modułu nie będzie uwzględniany podczas pobierania projektu i podczas aktualizacji wejść/wyjść.

Nie zmieniaj adresów automatycznie: Opcja ta jest dostępna tylko wtedy, jeśli zostanie tak zdefiniowana w pliku konfiguracji. Jeśli jest ona aktywna, moduł nie będzie uwzględniany przy automatycznym obliczaniu adresów. (Ustawienie domyślne: opcja jest wyłączona.)

Szczegółowe informacje na temat diagnozy w konfiguracji sterownika:

W polu **Adres diagnostyczny** należy wpisać adres rozpoznawczy, na jaki będą później zapisywane automatycznie informacje diagnostyczne. W przypadku normalnych modułów wejścia/wyjścia zależy od specjalnej konfiguracji urządzenia, w jaki sposób mają być używane adresy diagnostyczne. W przypadku adresów diagnostycznych dla magistrali CAN i systemów PROFIBUS DP obowiązują następujące zasady: Musi być podany adres rozpoznawczy, od którego informacja diagnostyczna ma być zapisywana. Informacja ta jest tworzona w strukturze *GetBusState*, która zawiera się w bibliotece producenta (na przykład BusDiag.lib producenta 3S – Smart Software Solutions GmbH):

Wszystkie moduły magistrali są cyklicznie wzywane do wypełnienia struktury diagnostycznej *GetBusState*, gdy tylko zadanie IEC wyśle swoje dane procesowe do modułów we/wy albo gdy je stamtąd odczyta.

Jeśli jeden z elementów magistrali zgłosi błąd, wówczas można odczytać jego specyficzną informację diagnostyczną za pomocą modułu *DiagGetState*, który jest również zdefiniowany w ww. bibliotece. Ta funkcja jest aktywna tylko wtedy, jeśli szyna została również skonfigurowana w CoDeSys.

Poniżej przedstawiono parametry wejścia i wyjścia dla bloku funkcji **DiagGetState**, który jest wywoływany do odczytywania informacji diagnostycznych dla określonego modułu magistrali:

Zmienne wejściowe bloku DiagGetState:

Konfiguracja sterownika

ENABLE: BOOL;	Przetwarzanie modułu rozpoczyna się przy zboczu wzrastającym sygnału
DRIVERNAME:POINTER TO STRING;	Nazwa sterownika (adres nazwy), do którego ma być wysłane zlecenie diagnozy. Jeśli wpisane zostanie tutaj 0, zlecenie diagnozy zostanie wysłane do wszystkich sterowników.
DEVICENUMBER:INT;	Identyfikacja magistrali, zarządzanej przez ten moduł (sterownik). Przykładowo sterownik karty Hilscher może obsługiwać do 5 kart (magistrali). Indeks opiera się na 0.
BUSMEMBERID:DWORD ;	Jednoznaczna identyfikacja magistrali względnie modułu. (np. NodeID w karcie CANopen, adres stacji w karcie PB-DP)

Zmienne wyjściowe bloku DiagGetState:

READY: BOOL ;	TRUE: przetwarzanie zlecenia diagnozy zostało zakończone
STATE:INT;	Jeśli READY = TRUE, wówczas zmienna STATE podaje aktualny status modułu jako jedną z poniższych wartości. Są one przypisane stałym globalnym: -1: nieprawidłowy parametr wejściowy (NDSTATE_INVALID_INPUTPARAM:INT;) 0: moduł nie pracuje (NDSTATE_NOTENABLED:INT;) 1: moduł wywołuje informację o diagnozie (NDSTATE_GETDIAG_INFO:INT;) 2: informacja o diagnozie jest aktualnie dostępna (NDSTATE_DIAGINFO_AVAILABLE:INT;) 3: brak dostępnej informacji o diagnozie (NDSTATE_DIAGINFO_NOTAVAILABLE:INT;)
EXTENDEDINFO:ARRAY[0..1 29] OF BYTE;	Do 100 bitów danych diagnostycznych producenta urządzenia magistrali.

Parametry modułu / parametry niestandardowe modułu wejścia/wyjścia

Okno dialogowe parametrów modułu

Index	Name	Wert	Default	Min.	Max.
1	XRate	10	10		
2	Yrate	10	10		
3	Enable...	Yes	Yes		
4	Namen	HugoType	HugoType		
5	DumpFI...	D:\CoDeSys\Projekte\ID...	D:\CoDeSys\Proj...		
6	IDString	abc def	abc def		
7	Realval...	1.876	1.876	-1.1	2.9876
8	Boolval...	TRUE	TRUE		
9	Test	16#A	16#A	16#A	16#FF

Zawiera parametry podane w pliku urządzenia. Edytować można tylko kolumnę Wartość.

Indeks: Indeks jest to bieżący numer (i) parametru w obrębie modułu.

Nazwa: Nazwa parametru

Wartość: Wartość parametru, edytowalna

Początkowo pokazywana jest wartość domyślna. Wartości mogą być pokazywane bezpośrednio lub jako nazwy symboliczne. Jeśli wpisy w pliku konfiguracyjnym nie są ustawione tylko do odczytu „Read Only”, można je edytować. Po kliknięciu wartości otwiera się pole edycji albo lista rozwijana, z której

można wybrać inną wartość. Jeśli w polu Wartość występuje nazwa pliku, wówczas po dwukrotnym kliknięciu tej nazwy otwiera się okno dialogowe „Otwórz plik”, z którego można wybrać inny plik.

Domyślnie: Wartość domyślna parametru

Min.: minimalna wartość parametru (tylko przy wyświetlaniu bezpośrednich wartości)

Maks.: maksymalna wartość parametru (tylko przy wyświetlaniu bezpośrednich wartości)

Ewentualne dodatkowe informacje dotyczące zaznaczonego parametru są wyświetlane w okienku etykiety.

Okno dialogowe parametrów modułu można zastąpić **oknem dialogowym parametrów specyficznych dla aplikacji** (parametry niestandardowe). Ma to miejsce, jeśli dla danego modułu w pliku konfiguracji jest „podłączone” poprzez bibliotekę Hook-DLL okno dialogowe parametrów specyficznych dla aplikacji.

9.6.6 Konfiguracja kanału

Parametry podstawowe kanału

Okno dialogowe parametrów podstawowych kanału

Id kanału: Jednoznaczna nazwa kanału obowiązująca globalnie.

Klasa: Informacja, czy kanał jest używany jako wejście (I), jako wyjście (Q) lub jako wejście i wyjście (I&Q), albo czy jest to kanał z przełączaniem wejść i wyjść (I|Q). Jeśli jest to kanał przełączalny, można użyć polecenia „Dodatki” „Zastąp element”.

Wielkość: Rozmiar kanału [Byte]

Domyślny Id: Symboliczna nazwa kanału, zdefiniowana w pliku konfiguracji.

Nazwa kanału jest definiowana w pliku konfiguracyjnym. Tylko w przypadku gdy moduł macierzysty jest odpowiednio skonfigurowany, nazwę można edytować w drzewie konfiguracji.

Komentarz: Informacje dodatkowe dotyczące kanału

W polu edycji można edytować wstawiony komentarz lub dodać nowy.

Adres: To pole edycji wyświetla się tylko wtedy, jeśli w pliku konfiguracyjnym jest aktywne odpowiednie ustawienie. Tutaj można wprowadzić żądany adres kanału.

Parametry kanału

To okno dialogowe służy do prezentacji i modyfikacji wartości parametrów kanału podobnie jak okno dialogowe parametrów modułu: **Indeks**, **Nazwa**, **Wartość**, **Domyślnie**, **Min.**, **Maks.**. Podobnie jak w modułach można je zastąpić oknem dialogowym „Parametry niestandardowe” specyficznym dla aplikacji.

Kanały bitowe

Kanały bitowe są automatycznie dodawane, jeśli w pliku konfiguracyjnym dany kanał ma ustawienie `CreateBitChannels=TRUE`.

Parametry podstawowe w kanałach bitowych otrzymują pole edycji **Komentarz**.

9.6.7 Konfiguracja modułów Profibus...

CoDeSys obsługuje konfigurację sprzętową zgodnie ze standardem PROFIBUS DP. Do tego potrzebny jest plik konfiguracyjny *.cfg, który zezwala na wstawianie modułów Profibus.

System PROFIBUS DP składa się z jednego lub kilku modułów master i slave. By urządzenia magistrali mogły wzajemnie wymieniać się danymi, należy je najpierw skonfigurować. Następnie po uruchomieniu każdy moduł master parametryzuje przydzielone mu podczas konfiguracji moduły slave. Podczas pracy moduł master wysyła dane do każdego z modułów slave i/lub żąda od nich danych.

Konfiguracja urządzeń master i slave w CoDeSys bazuje na **plikach GSD**. Pliki urządzeń podstawowych są dostarczane przez producenta urządzeń i zawierają standardowy opis właściwości charakterystycznych urządzenia PROFIBUS DP. Należy jednak pamiętać, że potrzebne pliki GSD w czasie uruchamiania programu CoDeSys muszą być umieszczone w katalogu zdefiniowanym dla plików konfiguracyjnych.

Urządzenia opisywane plikami gsd można wtedy dodawać w drzewie konfiguracji i ustawiać dla nich parametry. Do urządzenia master można podłączyć jedno lub kilka urządzeń slave.

Jeśli w drzewie konfiguracji jest zaznaczone urządzenie master DP, na odpowiednich kartach zakładek dostępne do wyboru są następujące okna dialogowe (w zależności od definicji w pliku konfiguracji): Parametry podstawowe, Parametry DP, Parametry magistrali, Parametry modułu. Druga zakładka posiada domyślną nazwę „Parametry DP”, jednak może być ona zmieniona w pliku konfiguracyjnym.

Jeśli w drzewie zaznaczone jest urządzenie slave DP, podłączone do urządzenia master DP, dostępne są następujące okna dialogowe: Parametry podstawowe, Parametry DP, Wejścia/Wyjścia, Parametry użytkownika, Przyporządkowanie do grupy, Parametry modułu.

Jeśli natomiast urządzenie slave DP jest konfigurowane do pracy na poziomie master, do konfiguracji będą potrzebne następujące okna dialogowe: Parametry podstawowe, Parametry DP, Wejścia/Wyjścia, Parametry modułu.

Parametry podstawowe urządzenia master DP

Okno dialogowe parametrów podstawowych urządzenia master DP odpowiada oknu parametrów innych modułów, patrz rozdział 9.6.5, Parametry podstawowe modułu wejścia/wyjścia.

Parametry modułu urządzenia master DP

Okno dialogowe parametrów modułu master DP ma podobną budowę jak w przypadku innych modułów: Parametry, które zostały przydzielone do urządzenia master i dodatkowo wpisane do pliku konfiguracji obok parametrów DP i parametrów magistrali, są tutaj wyświetlane i wartości ich mogą być w normalnym przypadku edytowane.

Parametry DP urządzenia master DP

To okno dialogowe zawiera następujące parametry urządzenia DP pobrane z pliku urządzenia (ewentualnie okno dialogowe nosi nazwę zdefiniowaną w pliku konfiguracji):

Info	Producent, Wersja GSD, Id (numer identyfikacyjny), HW Release i SW Release (wersja sprzętu i programu), Nazwa pliku GSD
Nazwa modułu	Nazwę domyślną można w tym miejscu edytować.

Adresy

Adres stacji: Dopuszczalne wartości zawierają się w obszarze od 0 do 126. Każde nowe urządzenie wstawione do magistrali otrzymuje automatycznie kolejny wyższy adres. (uwaga: adres 126 jest domyślnym adresem slave DP). Wprowadzenia ręczne są możliwe; odbywa się sprawdzanie na zdublowane adresy.

Najwyższy adres stacji: Wyświetlany jest najwyższy przydzielony adres stacji (HSA) magistrali. Można tutaj wprowadzić również niższy adres, by zmniejszyć obszar GAP (tzn. obszar adresów, w którym wyszukiwane są nowe aktywne urządzenia).

Okno dialogowe parametrów DP dla urządzenia master DP

Za pomocą przycisku **Plik GSD** można otworzyć należący do urządzenia plik.

Przycisk **Grupy** otwiera okno dialogowe „Właściwości grupy”. Właściwości grupy odnoszą się do urządzeń slave przyporządkowanych do urządzenia master.

Parametry DP urządzenia master DP / Właściwości grupy

Gruppenname	Sync. Mode	Freeze Mode
Gr 1	X	
Gr 2		X
Gr 3		X
Gr 4	X	
Gr 5	X	X
Gr 6	X	X
Gr 7		X
Gr 8		X

Można ustawić maksymalnie 8 grup. Dla każdej grupy można wprowadzić ustawienie, czy grupa ma pracować w **trybie wstrzymania** i/lub w **trybie synchronicznym**. Poprzez przyporządkowanie urządzeń slave (patrz „Właściwości urządzenia slave DP”, „Przyporządkowanie do grupy”) do różnych grup można sprawić, że wymiana danych z urządzenia master będzie odbywała się synchronicznie przy użyciu polecenia Global-Control. Polecenie wstrzymania powoduje, że urządzenie master nakazuje urządzeniu slave lub grupie wstrzymać chwilowo wejścia i przesyłać te dane w następnym cyklu wymiany danych. Polecenie synchronizacji powoduje, że dane odebrane przez urządzenie master bezpośrednio po wydaniu komendy zostaną przesyłane synchronicznie przez urządzenia slave z chwilą wydania następnej komendy synchronizacji.

W celu włączenia/wyłączenia opcji wstrzymywania i synchronizacji dla grupy, kliknij lewym przyciskiem myszy odpowiednie miejsce w tabeli i wstaw/usuń znacznik „X” albo za pomocą prawego

przycisku myszy aktywuj/dezaktywuj opcję w menu kontekstowym. Ponadto można tutaj edytować nazwy grup.

Parametry magistrali urządzenia master DP

Parametry magistrali opisują ustawienia komunikacji w czasie. Wartości poszczególnych parametrów są obliczane automatycznie z danych w plikach GSD w zależności od **prędkości transmisji** ustawionej przez użytkownika, jeśli zaznaczona jest aktywna opcja **Optymalizuj automatycznie**.

Uwaga: Wartości obliczone automatycznie stanowią tylko wartości przybliżone!

Parametry magistrali urządzenia master DP

Parameter	Wert	Einheit
Slot Time (TSL)	0	tBit
Min.Station Delay (min TSDR)	0	tBit
Max.Station Delay (max TSDR)	0	tBit
Quiet Time (TQUI)	0	tBit
Setup Time (TSET)	0	tBit
Target Rotation Time (TTR)	0	tBit
Gap Update Factor	0	
Max. Retry Limit	0	
Min. Slave Interval	0	100 µs
Poll Timeout	0	10 ms
Data Control Time	0	ms
Watchdog Time (TWD)	0	ms

Wszystkie parametry można także edytować ręcznie.

- Prędkość transmisji** Dostępne do wyboru są ustawienia w pliku GSD; ustawiona może być jednak tylko jedna prędkość transmisji, której mogą używać wszystkie urządzenia slave
- optymalizuj automatycznie** Jeśli ta opcja jest aktywna, wówczas ustawienia wprowadzone w oknie dialogowym „Parametry magistrali” zostaną zoptymalizowane na podstawie plików GSD; edycja wartości jest możliwa tylko wtedy, jeśli opcja jest nieaktywna.
Uwaga: Wartości obliczone automatycznie stanowią tylko wartości przybliżone!
- Slot Time** Maksymalny czas oczekiwania na odbiór pierwszego znaku odpowiedzi z urządzenia slave liczony od momentu wysłania wywołania przez urządzenie master.
- Min.Station Delay** min. TSDR (w Tbit): Minimalny czas reakcji, po którym stacja magistrali może odpowiedzieć (min. 11 TBit).
- Max.Station Delay** max. TSDR (w Tbit): Maksymalny przedział czasu, w jakim urządzenie slave musi odpowiedzieć.
- Quiet Time** TQUI (w Tbit): czas spoczynku, który musi być uwzględniony podczas przetwarzania sygnałów NRZ (Non Return to Zero) na inne kodowanie (czas przełączania dla repeatera).
- Target Rotation Time** TTR (w Tbit): żądany czas obiegu tokena; projektowany przedział czasu, w jakim urządzenie master ma otrzymać tokena. Wynika z sumy czasów wstrzymania tokena wszystkich urządzeń master na magistrali.
- Gap Update Factor** Współczynnik aktualizacji G: Liczba cykli magistrali, po których w obszarze GAP urządzenia master (obszar między adresem własnym magistrali a adresem następnej aktywnej stacji) następuje wyszukiwanie kolejnej, nowej, aktywnej stacji.

Max. Retry Limit	Maksymalna liczba ponownych prób wywołania urządzenia master, które nie odebrało od urządzenia slave żadnej odpowiedzi.
Min. Slave Interval	Czas między cyklami magistrali, w których urządzenie slave może przetwarzać żądanie urządzenia master (baza czasowa 100 μ s). Podana tutaj wartość musi być zgodna z odpowiednimi ustawieniami w plikach GSD urządzenia slave.
Poll Timeout	Maksymalny czas, po którym musi zostać odebrana odpowiedź urządzenia master przez urządzenie requester w komunikacji master-master (DP_Master klasa 2) (baza czasowa 1 ms).
Data Control Time	Czas, w którym urządzenie master przesyła do przyporządkowanych urządzeń slave informację o swoim czasie pracy. Jednocześnie urządzenie master kontroluje, czy w tym czasie nastąpiła chociaż jedna użyteczna wymiana danych między urządzeniami slave i aktualizuje listę Data_Transfer_List.
Watchdog Time	Czas dla kontroli działania procesu (watchdog). Ustawienie jest niedostępne w tej wersji programu (ustawienie stałe 400 ms).

Parametry podstawowe urządzenia slave DP


Okno dialogowe parametrów urządzenia slave DP ma podobną budowę jak w przypadku innych modułów: patrz rozdział 9.6.5, Parametry podstawowe modułu wejścia/wyjścia.

parametry DP urządzenia slave DP

To okno dialogowe zawiera następujące parametry urządzenia slave DP pobrane z pliku urządzenia (ewentualnie okno dialogowe nosi nazwę zdefiniowaną w pliku konfiguracji):

Info	Producent, Wersja GSD, HW Release i SW Release (wersja sprzętu i programu), Nazwa pliku GSD, Typ slave
Parametry domyślne	Nr identyfikacyjny: jednoznaczny numer identyfikacyjny dla tego typu urządzenia nadany przez PNO. Tworzy jednoznaczny referencję między urządzeniem DP slave a przynależącym plikiem GSD. TSDR (Tbit*): Time Station Delay Responder: czas reakcji, po którym urządzenie slave może najwcześniej wysłać odpowiedź do urządzenia master (min. 11 TBit). * TBit: jednostka czasu dla transmisji jednego bitu przez PROFIBUS; wartość odwrotna do prędkości transmisji; np. 1 TBit przy 12 MBaud = 1/12 000 000 Bit/sek = 83 ns Zablokuj/odblokuj: Blokowanie i odblokowywanie urządzenia slave dla innych urządzeń master: 0: min.TSDR i parametry specyficzne dla urządzenia slave mogą być nadpisywane; 1: urządzenie slave dostępne dla innych urządzeń master, 2: urządzenie slave zablokowane dla innych urządzeń master, wszystkie parametry są akceptowane; 3: urządzenie slave ponownie dostępne dla innych urządzeń master
Identyfikacja	Adres stacji (patrz „Parametry DP urządzenia master DP”), nazwa stacji (odpowiada nazwie urządzenia, edytowalna)
Aktywacja	Urządzenie slave jest aktywne/nieaktywne w bieżącej konfiguracji. Jeśli nie zostało aktywowane, dane konfiguracji urządzenia slave zostaną pobierane podczas pobierania przesłane do sprzęgu, jednak bez wymiany danych przez magistralę.
Watchdog	Jeśli kontrolka Watchdog jest aktywna, obowiązuje ustawiony czas watchdog (kontrola działania procesu, baza 10 ms). Jeśli w tym czasie urządzenie master nie skomunikuje się z urządzeniem slave, wówczas przejdzie ono z powrotem w stan inicjalizacji.

Okno dialogowe parametrów DP dla urządzenia slave DP

Anwenderparameter	Gruppenzuordnung	Modulparameter
Basisparameter	DP Parameter	Ein-/Ausgänge
<p>Info</p> <p>Hersteller: WAGO Kontakttechnik GmbH Revision: W3 HW Release: HW 01 SW Release: SW 1.03 Dateiname: wagob752.gsd Slavetyp: 3@WAGO-IO-SYSTEM@BINARY</p>		 <input type="button" value="GSD Datei..."/>
<p>Identifikation:</p> <p>Stationsadresse: <input style="width: 50px;" type="text" value="2"/></p> <p>Stationsname: <input style="width: 150px;" type="text"/></p>	<p>Standardparameter</p> <p>Identnummer: 0xB752</p> <p>TSDR (TBit): <input style="width: 50px;" type="text" value="11"/></p> <p>Lock/Unlock: <input style="width: 50px;" type="text" value="2"/></p>	
<p>Aktivierung</p> <p>Slave in aktueller Konfiguration aktiv: <input checked="" type="checkbox"/></p>	<p>Watchdog</p> <p>Watchdog Control: <input checked="" type="checkbox"/></p> <p>Zeit (ms): <input style="width: 50px;" type="text" value="1000"/></p>	

Za pomocą przycisku **Plik GSD** można otworzyć należący do urządzenia plik GSD.

Wejścia/wyjścia urządzenia slave DP

Sposób postępowania przy konfiguracji urządzenia slave zależy od tego, czy w danym przypadku chodzi o urządzenia „modularne” czy niemodularne „stałe” urządzenie slave.

Wybór modułów dla **modularnego urządzenia slave** odbywa się w następujący sposób:

Na liście po lewej stronie okna dialogowego zaznacz kliknięciem myszy moduł wejścia lub wyjścia i skopiuj go do prawego okna za pomocą przycisku **Wybierz**. Niepotrzebne moduły można usunąć z prawego okna za pomocą przycisku **Usuń**. Wstawione moduły są od razu pokazywane w drzewie konfiguracji. Jeśli zostaną tam zaznaczone, wyświetla się odpowiednie okno dialogowe **Moduł Profibus**, w którym pokazane są adresy wejścia, wyjścia i diagnozy modułu. Jeśli w tym module zostanie zaznaczony odpowiedni kanał, otworzy się okno dialogowe **Kanał Profibus**, w którym jest pokazywany adres kanału. Nazwy obydwu tych okien dialogowych można zmienić wprowadzając nowe nazwy w pliku konfiguracyjnym.

Ze względu na ograniczenia długości danych (**Maks. długość wejścia**, **Maks. długość wyjścia**, **Maks. długość wejścia/wyjścia**) i liczby modułów (**Maks. liczba modułów**) w plikach GSD informacje te są dostępne na obydwu listach modułów. W lewym bloku znajdują się maksymalnie dopuszczalne wartości dla urządzenia, w prawym bloku podawana jest suma wartości obliczona na podstawie wybranej konfiguracji. W przypadku przekroczenia wartości maksymalnych zostanie wyświetlony komunikat o błędzie.

Okno dialogowe konfiguracji wejść/wyjść urządzenia slave DP

Anwenderparameter		Gruppenzuordnung		Modulparameter	
Basisparameter		DP Parameter		Ein-/Ausgänge	
Max. Länge Input:	244 Byte	Länge Input:	0 Byte		
Max. Länge Output:	244 Byte	Länge Output:	0 Byte		
Max. Länge In-/Output:	368 Byte	Länge In-/Output:	0 Byte		
Max. Modulanzahl:	24	Anzahl Module:	0		

Eingabemodule	Ausgewählte Module
... 1 byte input con (0x90)	
... 2 byte input con (0x91)	
... 3 byte input con (0x92)	
... 4 byte input con (0x93)	
... 8 byte input con (0x97)	
... 12 byte input con (0x9B)	
... 16 byte input con (0x9F)	
... 20 byte input con (0x40,0x90)	
... 32 byte input con (0x40,0x9f)	
... 64 byte input con (0x40,0xb1)	
... 1 word input con (0xD0)	
... 2 word input con (0xD1)	
... 3 word input con (0xD2)	
... 4 word input con (0xD3)	
... 8 word input con (0xD7)	
... 12 word input con (0xDB)	

Okno dialogowe zawiera w lewym obszarze listę wszystkich modułów wejścia i wyjścia urządzenia slave zapisanych w pliku GSD, okno z prawej strony zawiera aktualnie wybraną dla urządzenia konfigurację wejść i wyjść.

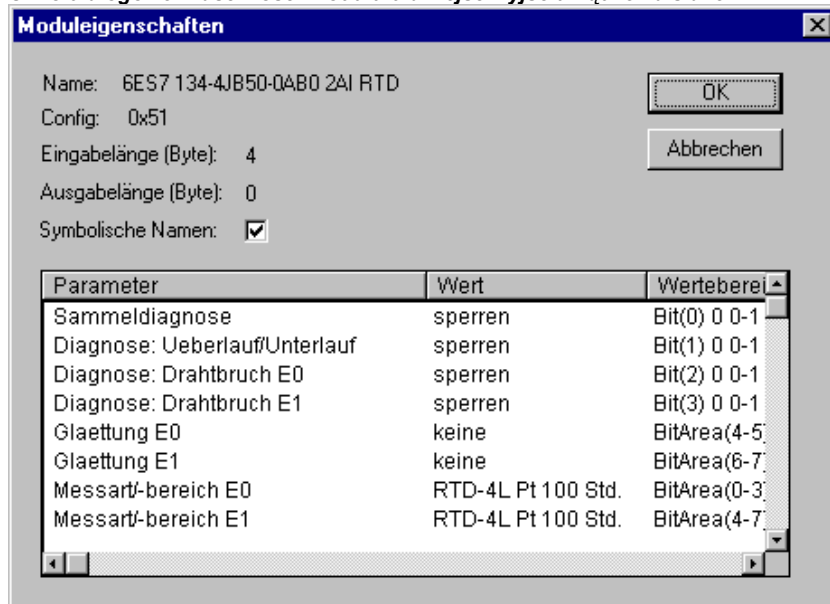
W przypadku modularnego urządzenia slave (urządzenie, które może być wyposażone w różne moduły wejścia i wyjścia), wyboru dokonuje się w następujący sposób: Na lewej liście zaznacz kliknięciem myszy moduł wejścia lub wyjścia i skopiuj go do prawego okna za pomocą przycisku >>. Niepotrzebne moduły można usunąć z prawego okna za pomocą przycisku **Usuń**.

Taki sposób wybierania nie jest możliwy w przypadku **niemodularnych urządzeń slave**. Wejścia i wyjścia tych urządzeń są wyświetlane bezpośrednio w prawym oknie. Niepotrzebne moduły można usunąć po zaznaczeniu i kliknięciu przycisku **Usuń**.

Przycisk **Właściwości** otwiera okno dialogowe „Właściwości modułu” dla modułu aktualnie zaznaczonego na lewej lub prawej liście wybranego modułu wejścia lub wyjścia.

Okno pokazuje **nazwę, konfigurację** (kod opisu modułu wg standardu PROFIBUS) i długość **wejścia** i **wyjścia** modułu w **bitach**. Jeśli opis modułu w pliku GSD zawiera dodatkowe specyficzne parametry, będą one tutaj wyświetlane razem z wartościami i zakresem wartości. Jeśli zaznaczona jest aktywna opcja **Nazwy symboliczne**, będą przy tym używane nazwy symboliczne.

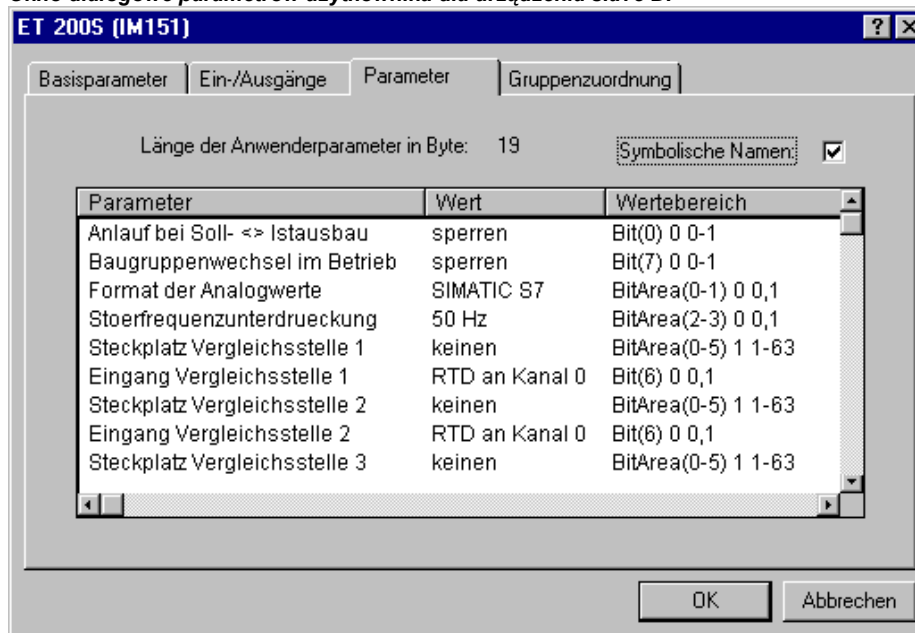
Okno dialogowe właściwości modułu dla wejść/wyjść urządzenia slave DP



Parametry użytkownika urządzenia slave DP

W tym miejscu znajduje się lista zawierająca rozszerzone parametry urządzenia slave DP zdefiniowane w pliku GSD. Kolumna **Parametr** pokazuje nazwę parametru. Wartości parametrów w kolumnie **Wartość** można zmieniać klikając dwukrotnie muszą albo za pomocą prawego przycisku myszy. Poza tym podany jest **Zakres wartości**.

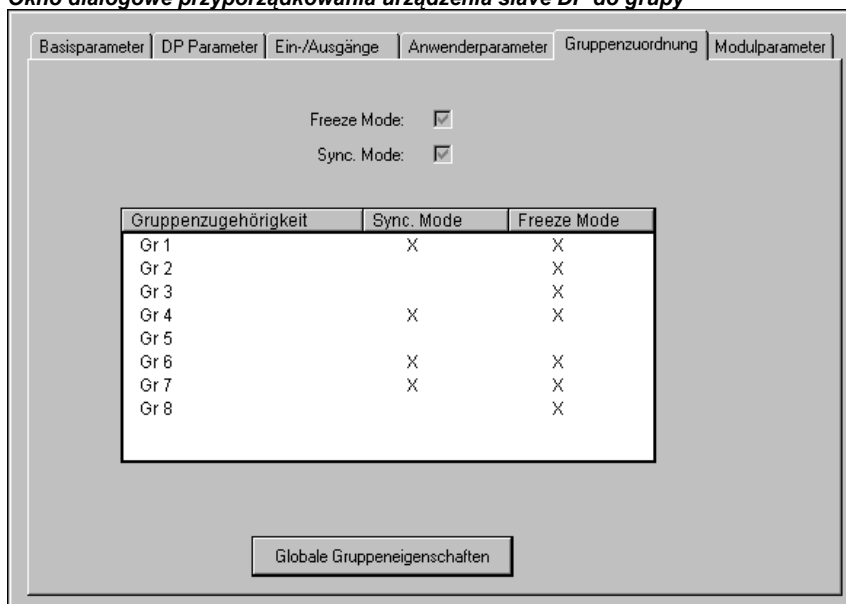
Okno dialogowe parametrów użytkownika dla urządzenia slave DP



Jeśli w pliku GSD zostały zdefiniowane również nazwy symboliczne dla parametrów, można zaznaczyć opcję **Nazwy symboliczne**, by wyświetlić je razem z wartościami. Jako informacja podana jest również w tabeli **długość parametrów użytkownika**.

Przyporządkowanie urządzenia slave DP do grupy

Okno dialogowe przyporządkowania urządzenia slave DP do grupy



To okno dialogowe umożliwia przyporządkowanie urządzenia slave do jednej lub kilku spośród ośmiu możliwych grup. Ogólnie obowiązujące właściwości grupy (**tryb synchronizacji** i/lub **tryb wstrzymania**) są natomiast definiowane w konfiguracji właściwości master (patrz „Parametry DP urządzenia master DP”, „Właściwości grupy”). Za pomocą przycisku **Ogólne właściwości grupy** otwiera się również takie samo okno dialogowe.

Grupy, dla których zostały przyporządkowane urządzenia slave są oznaczane znakiem plus. Przyporządkowanie względnie usunięcie urządzenia slave z grupy jest możliwe poprzez zaznaczenie nazwy grupy w kolumnie **Przynależność do grupy** i wybraniu z menu kontekstowego polecenia „Dodaj slave do grupy” albo „Usuń slave z grupy” lub ponownym kliknięciem z lewej strony obok nazwy grupy.

Urządzenie slave można przyporządkować tylko do grup, których właściwości może ono obsługiwać. Właściwości danego urządzenia slave (**tryb synchroniczny** / **tryb wstrzymania**) są pokazywane u góry tabeli. Tryb obsługiwany przez urządzenie jest zaznaczony haczykiem.

Parametry modułu urządzenia slave DP

Okno dialogowe parametrów modułu urządzenia slave DP ma podobną budowę jak w przypadku innych modułów: Parametry, które zostały przydzielone do urządzenia slave i dodatkowo wpisane do pliku konfiguracji obok parametrów DP i parametrów użytkownika, są tutaj wyświetlane i wartości ich mogą być w normalnym przypadku edytowane.

Właściwości urządzenia slave DP w trybie slave magistrali PROFIBUS

Jeśli magistrala PROFIBUS pracuje w trybie slave, wówczas urządzenie slave znajduje się na poziomie master. Konfigurację umożliwiają cztery karty zakładek:

1. Parametr podstawowy
2. Parametry DP
3. Parametry modułu
4. Wejścia/wyjścia

9.6.8 Konfiguracja modułów CANopen...

CoDeSys obsługuje konfigurację urządzeń zgodnie ze standardem CANopen Draft Standard 301. Do tego potrzebny jest plik konfiguracyjny *.cfg, który zezwala na wstawianie modułów CAN.

Pliki EDS (Electronic Data Sheet) i DCF (Device Configuration File) znajdujące się w folderze plików konfiguracyjnych można wykorzystać w konfiguracji w ramach, jakie zostały zdefiniowane w pliku konfiguracyjnym *.cfg. W pliku EDS są opisane możliwości ustawień modułu CAN. Jeśli zostanie wstawiony moduł, który jest opisany w pliku DCF, wtedy można zmieniać tylko adresy IEC, gdyż cały moduł został już skonfigurowany w konfiguratorze CAN.

Moduły CAN zawierają konfigurację, która opisuje ich zachowanie w czasie oraz w razie wystąpienia błędu przesyłania wiadomości (okno dialogowe „Parametry CAN modułu CAN”). Ponadto dla każdego modułu definiowane jest mapowanie PDO (Process Data Objects), służących do wysyłania i odbierania wiadomości między urządzeniami (okna dialogowe „Odbiór mapingu PDO”). Wartości dostępnych SDO (Service Data Objects – Obiekty danych serwisowych) można dopasowywać (okno dialogowe „Obiekty danych serwisowych”).

Dodatkowe parametry podane w pliku urządzenia modułu CAN ew. Master CAN można skonfigurować w oknie dialogowym „Parametry modułu”.

By podłączyć sterownik programowalny CoDeSys do sieci CAN jako urządzenie CANopen Slave (zwane również węzłem CANopen, a w dalszej części rozdziału nazywane urządzeniem CAN Device), można go skonfigurować w konfiguratorze sterownika CoDeSys i zapisać konfigurację jako plik EDS. Taki plik EDS można później używać w dowolnej konfiguracji CANopen Master.

W następnych akapitach opisano okna dialogowe i polecenia menu konfiguratora CANopen. Szczegółowe informacje znajdują się w dokumencie CANopen dla systemów czasu przebiegu 3S V<numer wersji>.pdf.

Parametry podstawowe modułu Master CAN

Okno dialogowe parametrów podstawowych urządzenia Master CAN wygląda tak jak w przypadku innych modułów (patrz rozdział 9.6.5, Parametry podstawowe modułu wejścia/wyjścia).

Parametry CAN modułu Master CAN

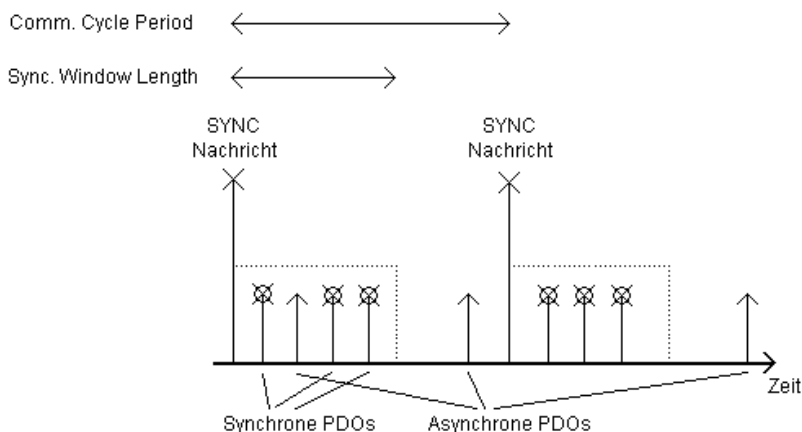
Okno dialogowe parametrów CAN dla urządzenia Master CAN

W tym miejscu można zdefiniować ustawienia globalne i parametry kontrolne dla magistrali CAN:

Wybierz z listy rozwijanej żadaną **prędkość transmisji** danych w magistrali.

Spośród PDO (Process Data Object) rozróżnia się tryby transmisji: synchroniczny i asynchroniczny (patrz: Właściwości PDO). **Cykl komunikacji** [µs] jest to przedział czasu w mikrosekundach, w którym wysyłany jest komunikat synchronizacji i jego unikalny numer **Sync.COB-Id** (Communication Object Identifier).

aktywuj: Tylko wtedy gdy ta opcja jest zaznaczona, odbywa się wysyłanie komunikatów synchronizacji między stacjami master i slave.



Synchroniczne PDO są przesyłane bezpośrednio po komunikacji synchronizacji w czasie podanym w polu **Sync.Window Length** [μ s]. Jeśli pola Comm. Cycle Period i Sync. Window Length zawierają wartość 0, wówczas nie będą wysyłane komunikaty synchronizacji.

NodeID: jednoznaczny identyfikator modułu CAN odpowiada numerowi modułu od 1 do 127, jaki został nadany modułowi automatycznie. Numer ID musi być podany w notacji dziesiętnej. (Nie należy go mylić z „numerem węzła”, który jest też używany w konfiguracji sterownika!)

Jeśli opcja **Uruchom automatycznie** jest aktywna, wtedy magistrala CAN jest uruchamiana i inicjalizowana automatycznie podczas pobierania i uruchamiania sterownika. Jeśli opcja jest nieaktywna, wtedy magistralę należy uruchomić w projekcie.

Jeśli aktywna jest opcja **Obsługuj DSP301,V3.01 i DSP306**, można korzystać z modułowych urządzeń CAN slave i kilku dodatkowych modyfikacji wg normy DSP301 V3.01 i DSP306. W takim przypadku można ustawiać także takt funkcji Heartbeat kontrolującej obecność urządzeń (**Heartbeat Master [ms]:**), której można używać alternatywnie do funkcji Nodeguarding. W przeciwieństwie do niej takty Heartbeat mogą wysyłać wszystkie moduły CAN niezależnie od siebie. Zwyczajnie takty Heartbeat są wysyłane z urządzeń master do urządzeń slave.

Parametry modułu Master CAN

Okno dialogowe parametrów modułu Master CAN wygląda tak jak w przypadku innych modułów: W tym miejscu są zilustrowane parametry przydzielone master dodatkowo oprócz parametrów CAN w pliku konfiguracji i w standardowym przypadku można je edytować.

Parametry podstawowe modułu CAN

Okno dialogowe parametrów podstawowych modułu CAN wygląda podobnie jak w przypadku innych modułów (patrz rozdział 9.6.5, Parametry podstawowe modułu wejścia/wyjścia).

W polach **Adres wyjściowy** i **Adres wejściowy** są pokazywane adresy IEC wprowadzone dla konfiguracji CAN, których używają PDO (Process Data Object) w projekcie, by mogły działać, przy czym kierunek (wejście lub wyjście) ustalany jest z punktu widzenia modułu.

Parametry CAN modułu CAN

Parametry modułu CAN, który pełni globalnie funkcję kontrolną w magistrali nie będąc urządzeniem master, różnią się od tych, jakie można ustawić dla urządzenia Master CAN.

Okno dialogowe parametrów CAN dla modułu CAN

Sekcja Ogólne:

Node ID służy jako jednoznaczny identyfikator modułu CAN, odpowiada numerowi modułu od 1 do 127, jaki został mu nadany automatycznie. Numer ID musi być podany w notacji dziesiętnej.

Jeśli opcja **Zapis DCF** jest aktywna, wówczas po wstawieniu pliku EDS w katalogu plików kompilacji tworzony jest plik DCF, którego nazwa składa się z nazwy pliku EDS i numeru Node-Id.

Opcja **Generuj wszystkie SDO** umożliwi tworzenie SDO dla wszystkich obiektów, nie tylko dla tych, które zostały zmienione.

Opcja **Resetuj węzeł** powoduje reset urządzenia slave w momencie ładowania konfiguracji do sterownika.

Opcja **Urządzenie opcjonalne** (dostępność w oknie dialogowym jest zależna od systemu docelowego) powoduje, że urządzenie master przeprowadza tylko jedną próbę odczytu z danego węzła. Węzeł jest ignorowany w przypadku braku odpowiedzi, tzn. urządzenie master przechodzi dalej do normalnego trybu pracy.

Opcja **Nie inicjuj** (dostępność w oknie dialogowym jest zależna od systemu docelowego) powoduje, że urządzenie master włącza węzeł do pracy natychmiast bez wysyłania mu konfiguracji przez SDO. (Dane SDO są jednak pomimo to generowane i zapisywane w sterowniku.)

Jeśli system docelowy oferuje taką możliwość, generowanie SDO można pominąć poprzez dezaktywację w trzech etapach następujących opcji, co przykładowo może być konieczne, jeśli pamięć konfiguratora jest ograniczona limitem:

UWAGA: Ustawień tych nie należy zmieniać bez dokładnej znajomości systemu!

CreateCommSDOs: Parametry komunikacji SDO

CreateMappingSDOs: Konfiguracja mapowania SDO

CreateBasicSDOs: Parametry podstawowe SDO (Nodeguarding, Sync itp.)

SDO są generowane tylko dla aktywowanych typów. Opisywana wyżej opcja „Generuj wszystkie SDO” odnosi się również tylko do aktywowanych tutaj typów SDO.

Sekcja Nodeguard: (funkcja alternatywna do Heartbeat)

Jeśli opcja **Nodeguarding** jest aktywna, wówczas w czasie podanym w polu **Czas nadzoru** w milisekundach zostanie wysłana wiadomość do modułu. Jeśli wtedy moduł o identyfikatorze podanym w polu **Nadz. COB-ID** (Communication Object Identifier) nie zgłosi się, wówczas otrzyma on status Timeout. Jeśli ilość prób osiągnie wartość podaną w polu **Wsp. czasu życia**, wówczas moduł otrzymuje status „nie OK”. Status modułu jest przechowywany w adresie diagnostycznym. Jeśli w polach Czas nadzoru i Wsp. czasu życia nie wprowadzono żadnej wartości (0), wówczas kontrola modułu nie odbywa się.

Sekcja Ustawienia Heartbeat: (funkcja alternatywna do Nodeguard)

Opcja **Włącz generowanie Heartbeat** powoduje, że moduł wysyła sygnał obecności w odstępach określonych w polu **Czas generacji Heartbeat: ms**.

Opcja **Włącz zużycie Heartbeat** powoduje, że moduł nasłuchuje sygnałów obecności wysyłanych przez urządzenie master. Jeśli te nie zostaną odebrane, wówczas moduł wyłącza wejścia/wyjścia.

Sekcja Telegram alarmowy:

W razie wystąpienia wewnętrznych błędów moduł wysyła wiadomości **alarmowe** i jednoznaczny numer **COB-Id**. Są one w zależności od wiadomości przynależących do modułu przechowywane w adresie diagnostycznym.

Po kliknięciu przycisku **Info** wyświetla się okno z wierszami „FileInfo” i „DeviceInfo” pliku EDS albo DCF producenta modułu.

Wybór modułu CAN dla modularnych urządzeń slave

W lewej kolumnie znajdują się **dostępne moduły** dla modularnego urządzenia slave. Za pomocą przycisków Dodaj i Usuń można dokonać wyboru modułów, które następnie są widoczne w prawej kolumnie (**Wybrane moduły**). Zgodnie z tym zmienia się również lista wyboru PDO i SDO.

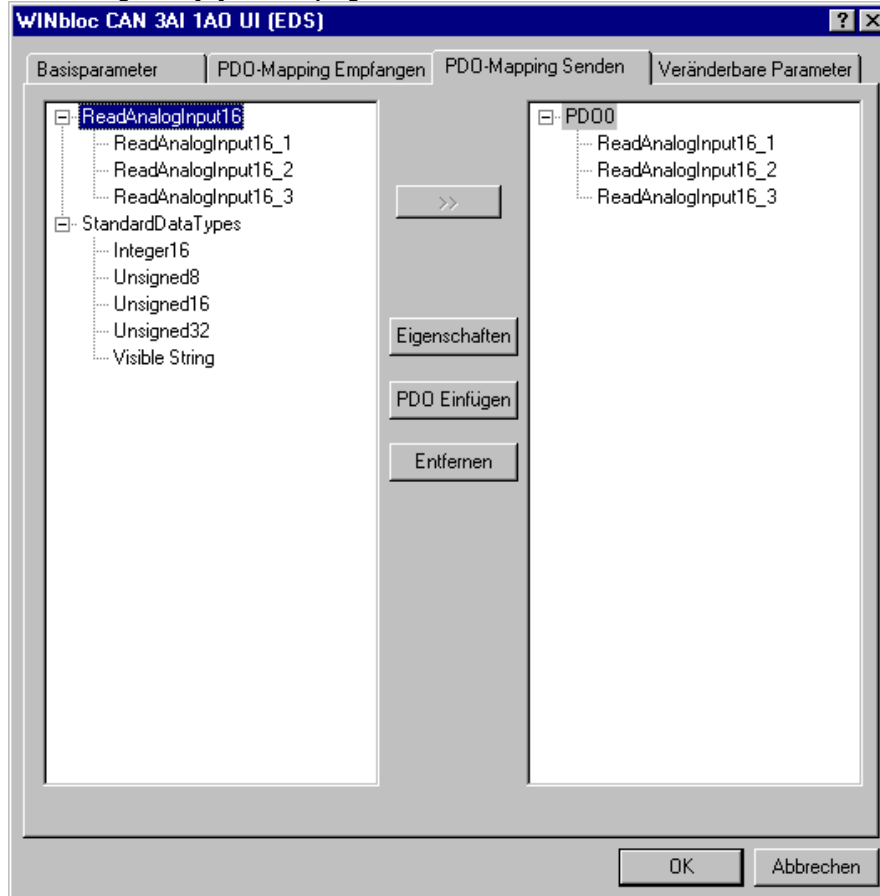
Mapowanie PDO modułu CAN

Zakładki **Odbiór mapingu PDO** i **Wysyłanie mapingu PDO** w oknie dialogowym konfiguracji modułu CAN umożliwiają modyfikację opisanego w pliku EDS „Mapowania” modułu.

Po lewej stronie są dostępne wszystkie „mapowalne” obiekty pliku EDS i można je dodać do PDO (Process Data Object) po prawej stronie (przycisk >>) lub ponownie je usunąć (przycisk **Usuń**). Standardowe typy danych można wstawić w celu utworzenia w PDO pustych przestrzeni.

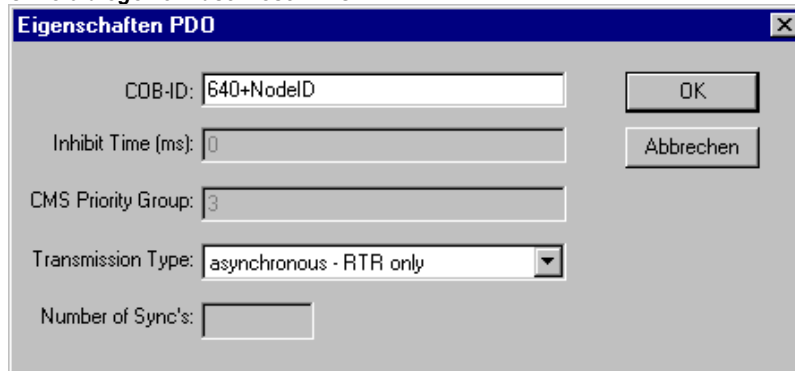
Za pomocą przycisku **Wstaw PDO** można utworzyć dalsze PDO i obłżyć je odpowiednimi obiektami. Za pomocą wstawionych PDO następuje przyporządkowanie wejść i wyjść do adresów IEC. W konfiguracji sterownika po wyjściu z okna dialogowego widoczne są dokonane ustawienia. Można tam za pomocą nazw symbolicznych obłżyć poszczególne obiekty.

Okno dialogowe wysyłania mapingu PDO



Za pomocą **Właściwości** można w oknie dialogowym edytować zdefiniowane w obrębie normy właściwości PDO.

Okno dialogowe właściwości PDO



Każdy komunikat PDO wymaga jednoznacznego **COB-ID** (Communication Object Identifier).

Jeśli opcja nie jest obsługiwana przez moduł lub nie można zmienić wartości, pole jest szare i nie można go edytować.

Inhibit Time (100 μ s) to minimalny czas między dwoma komunikatami tego PDO. Zapobiega to zbyt częstemu rozsyłaniu PDO po zmianie wartości. Jednostkę stanowi 100 μ s.

CMS Priority Group nie może być zmieniane i opisuje względną ważność PDO przy transmisji CAN. Wyświetlane są wartości od 0 do 7, przy czym 0 ma najwyższy priorytet.

Transmission Type daje wybór możliwych rodzajów transmisji dla tego modułu:

acykliczny – synchroniczny: PDO podlega wprawdzie synchronicznej transmisji, lecz nie okresowo.

cykliczny – synchroniczny: PDO podlega synchronicznej transmisji, przy czym **Number of Sync's** podaje ilość komunikatów synchronizacji, między dwoma transmisjami tego PDO.

synchroniczny – wyłącznie RTR: PDO jest aktualizowane po komunikacie synchronizacji, lecz nie jest rozsyłane. Transmisji podlega ono tylko po wyraźnym zapytaniu (**Remote Transmission Request**)

asynchroniczny – wyłącznie RTR: PDO jest aktualizowane i podlega transmisji tylko po wyraźnym zapytaniu (**Remote Transmission Request**)

asynchroniczny – właściwy dla profilu jednostki i asynchroniczny – właściwy dla producenta: PDO podlega transmisji tylko po określonych zdarzeniach.

Number of Syncs: Zależnie od typu transmisji (Transmission Type) pole to można edytować w celu wpisania ilości komunikatów synchronizacji (definicja w oknie dialogowym parametrów CAN, Com. Cycle Period, Sync Window Length, Sync. COB-Id), po których PDO ma zostać ponownie rozesyłane.

Event-Time: Zależnie od typu transmisji w tym miejscu jest podawany okres w milisekundach (**ms**), który powinien upłynąć między dwoma transmisjami PDO.

Obiekty danych serwisowych

W tym miejscu znajduje się lista obiektów z pliku EDS albo DCF, których indeksy leżą w obszarze od 0x2000 do 0x9FFF i są zdefiniowane z prawem do zapisu.

Dla każdego obiektu dostępne są pola **Indeks**, **Nazwa**, **Wartość**, **Typ** i **Domyślnie**. Każda wartość może być zmieniana. Zaznacz wartość i naciśnij klawisz <spacja>. Wprowadzone zmiany można potwierdzić klawiszem <Enter> albo anulować klawiszem <Escape>.

Okno dialogowe konfiguracji obiektów danych serwisowych (SDO)

Index	Name	Wert	Typ
2100sub1	1. digital output block		Unsigned
2100sub2	2. digital output block		Unsigned
2100sub3	3. digital output block		Unsigned
2100sub4	4. digital output block		Unsigned
2100sub5	5. digital output block		Unsigned
2100sub6	6. digital output block		Unsigned
2100sub7	7. digital output block		Unsigned
2100sub8	8. digital output block		Unsigned
2100sub9	9. digital output block		Unsigned
2300sub8	8 1byte output		Unsigned

Podczas inicjalizacji magistrali CAN ustawione wartości w postaci SDO (obiekty danych serwisowych) są przesyłane do modułów CAN.

Wskazówka: Wszystkie niekompatybilne typy danych między CANopen a IEC-61131 są zastępowane w CoDeSys przez następny pod względem wielkości typ danych standardu IEC-61131.

9.6.9 Konfiguracja urządzenia CanDevice (CANopen Slave)

Sterownik programowalny w CoDeSys może być rozpoznawany w sieci CAN jako urządzenie CANopen Slave (zwane również węzłem CANopen, dalej nazywane jako urządzenie CanDevice).

W tym celu można go skonfigurować w konfiguratorze sterownika CoDeSys i zapisać konfigurację jako plik urządzenia (plik EDS). Taki plik EDS można później używać w dowolnej konfiguracji CANopen Master.

Warunek utworzenia urządzenia CanDevices w konfiguratorze sterownika CoDeSys:

1. Biblioteki

- 3S_CanDrv.lib
- 3S_CanOpenManager.lib
- 3S_CanOpenDevice.lib

są podłączone w projekcie CoDeSys. Są one niezbędne, by używać sterownika jako urządzenia CANopen Device.

2. W podstawowym pliku konfiguracji (*.cfg) występuje odpowiedni wiersz dla urządzenia CanDevice, by konfigurator sterownika mógł utworzyć element podrzędny „CanDevice” i by można było przypisać dla niego parametry w następujących trzech oknach dialogowych konfiguracji:

- – Ustawienia podstawowe
- – Ustawienia CAN
- – Domyślne mapowanie PDO

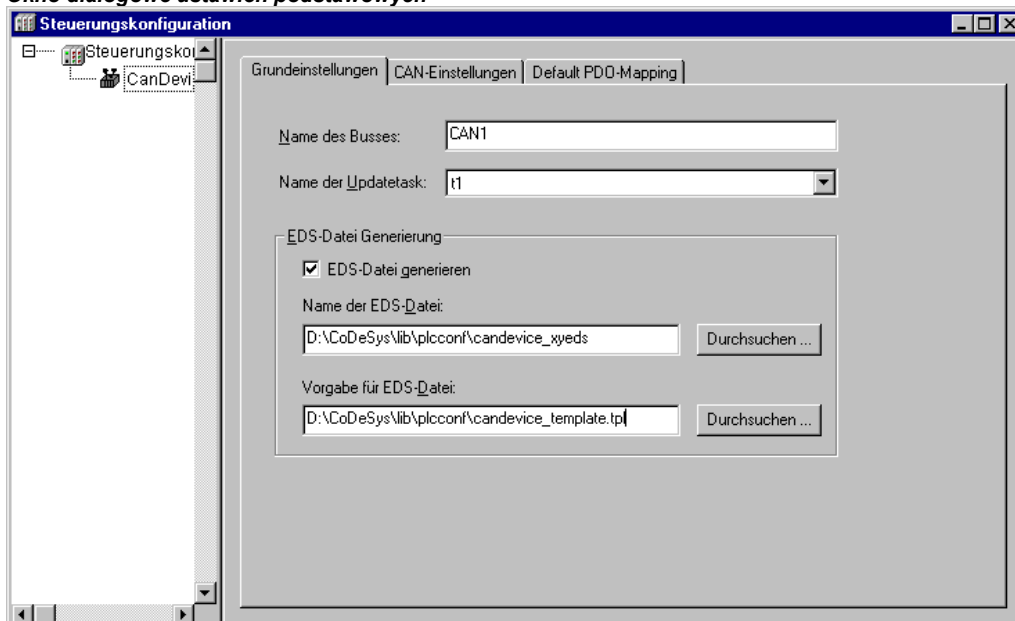
Ustawienia podstawowe urządzenia CanDevice

Nazwa magistrali: nie używana w tym momencie.

Nazwa zadania aktualizacji: nazwa zadania, w którym następuje wywołanie urządzenia CanDevice.

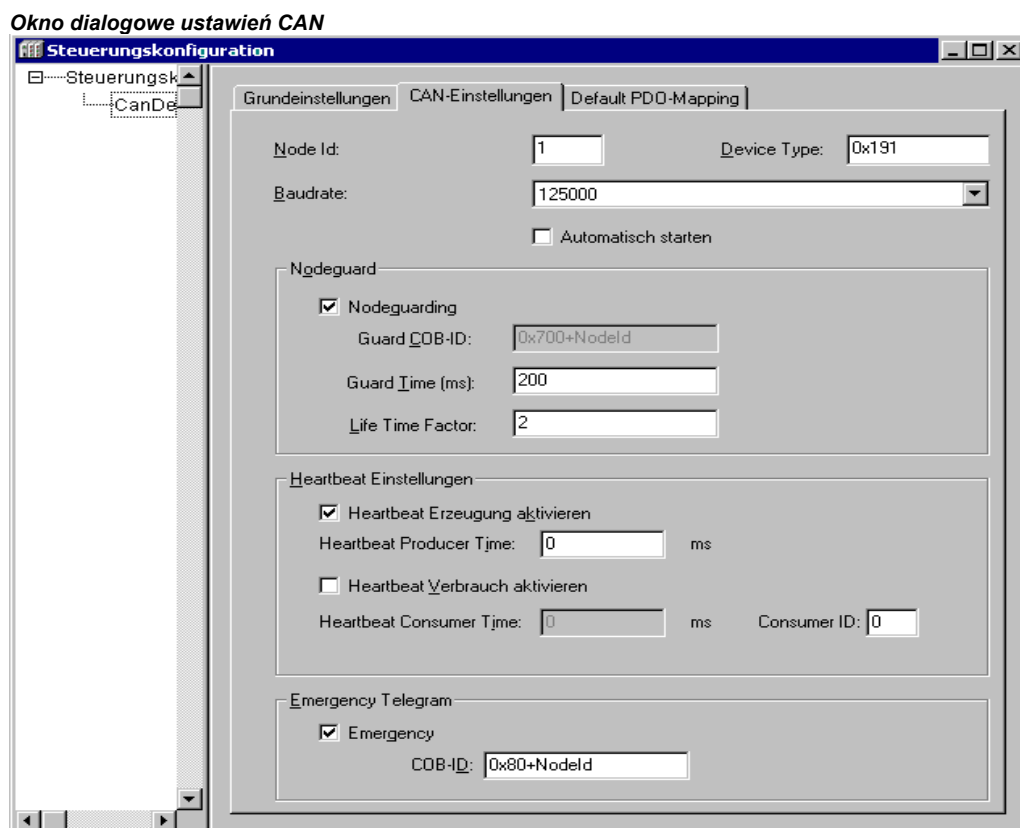
Generuj plik EDS: Jeśli ta opcja jest zaznaczona i wypełnione jest pole **Nazwa pliku EDS**, wówczas z dostępnych tu ustawień można utworzyć plik EDS, by móc wstawić urządzenie CANDevice do dowolnej konfiguracji master.

Okno dialogowe ustawień podstawowych



Opcjonalnie można jeszcze podać nazwę utworzonego ręcznie pliku szablonu (**Szablon dla pliku EDS**), w którym zostaną wstawione ustawienia urządzenia CanDevice. Przykładowo niektóre wpisy, istotne dla pewnych plików EDS, można zapisać w pliku tekstowym pod nazwą „EDS_template.txt” i wybrać tutaj ten plik jako szablon. Jeśli potem z bieżącego projektu tworzona jest konfiguracja urządzenia CanDevice jako plik „device_xy.eds”, wówczas w pliku „device_xy.eds” zostaną zapisane wpisy projektu razem z wpisami z pliku „EDS_template.txt”. (Należy pamiętać, by nie używać dla szablonów rozszerzenia „.eds”!) Jeśli z projektu tworzone są wpisy, które są już zdefiniowane w szablonie, wówczas wartości z szablonu nie będą nadpisywane.

Ustawienia CAN urządzenia CanDevice



W tym miejscu można przesłać do systemu czasu przebiegu wartości parametrów CAN zgodnie z normą CANopen. Od używanego systemu czasu przebiegu zależy, jak zostaną one zinterpretowane. Szczegółowe informacje nt. używania obsługiwanej przez systemy czasu przebiegu 3S implementacji CANopen w połączeniu z odpowiednimi bibliotekami znajdują się w dokumentacji „*CanOpen dla systemu czasu przebiegu 3S*”.

Node Id: Numer węzła (1–127), przez który magistrala master komunikuje się z urządzeniem CanDevice podłączonym do sieci CANopen.

Prędkość transmisji: Żądaną prędkość transmisji w magistrali można wybrać z dostępnej tu listy rozwijanej.

Typ urządzenia: W tym miejscu pokazuje się automatycznie typ urządzenia, jaki został zwrócony podczas kwerendy obiektu 0x1000, czyli typ sterownika skonfigurowanego w projekcie. Typ można tutaj edytować.

Jeśli opcja **Uruchom automatycznie** jest aktywna, wtedy magistrala CAN jest uruchamiana i inicjalizowana automatycznie podczas pobierania i uruchamiania sterownika. Jeśli opcja jest nieaktywna, wówczas urządzenie CanDevice czeka na odpowiednie polecenie.

Można również skonfigurować funkcje **Nodeguard** i **Telegramu alarmowego**. W tym celu należy zapoznać się z odpowiednimi opisami dla konfiguracji pozostałych modułów CAN i master (rozdział 9.6.8).

Grupa Ustawienia Heartbeat:

Włącz generowanie Heartbeat: Jeśli ta opcja jest aktywna, urządzenie CanDevice wysyła sygnał obecności w odstępach określonych w polu **Czas generacji Heartbeat:** ms.

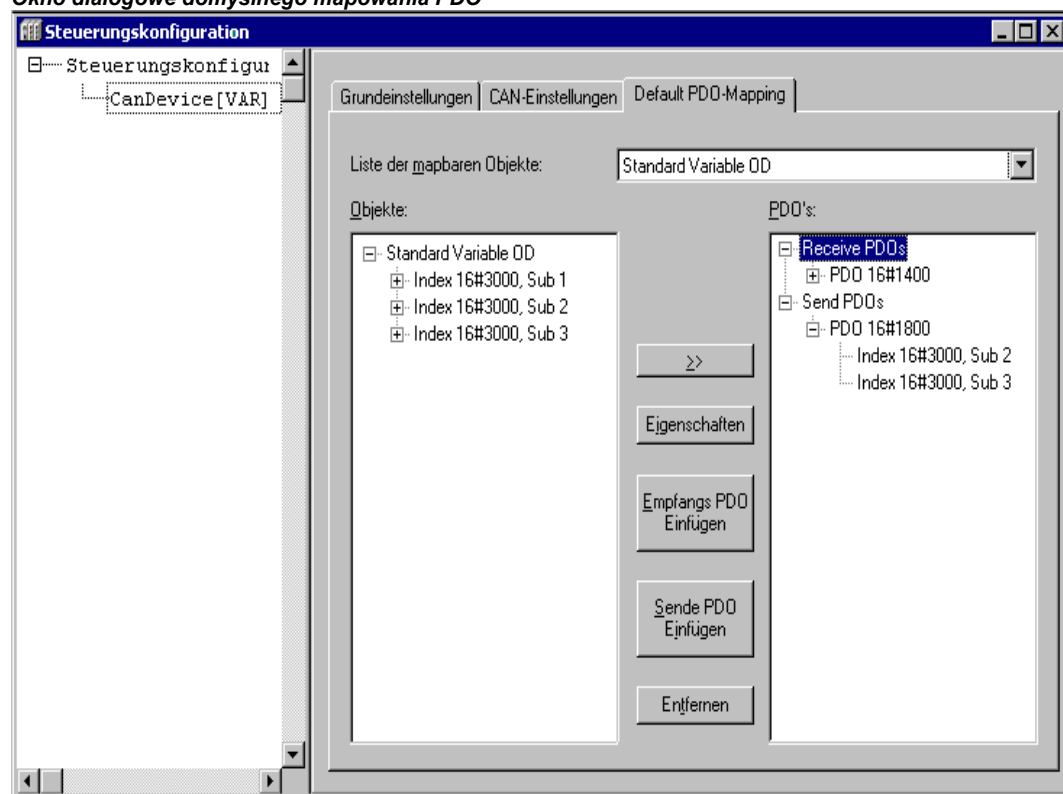
Włącz zużycie Heartbeat: Jeśli ta opcja jest aktywna, urządzenie CanDevice nasłuchuje sygnałów obecności wysyłanych przez moduł zdefiniowany jako ID odbiorcy. **Czas odbiorcy Heartbeat** w milisekundach podaje, po jakim czasie od momentu stwierdzenia braku obecności urządzenia ustawiana jest flaga błędu.

Domyślne mapowanie PDO urządzenia CanDevice

W tym miejscu można zdefiniować przyporządkowanie dla wpisów z lokalnego folderu obiektów (Menedżer parametrów) i PDO wysyłanych/odbieranych przez urządzenie CanDevice. PDO można wtedy mapować w każdej konfiguracji master, w której urządzenie CanDevice jest używane.

W Menedżerze parametrów wpisy znajdujące się na listach (zmiennie) otrzymują połączenie do zmiennych aplikacji poprzez indeks/subindeks. **Należy pamiętać**, że subindeks 0 w indeksie, który zawiera więcej niż jeden subindeks, jest używany do przechowywania informacji o liczbie subindeksów. Obiekty (parametry) indeksu muszą być wpisywane w Menedżerze parametrów w kolejności rosnącej (według subindeksów).

Okno dialogowe domyślnego mapowania PDO



Lista mapowalnych obiektów: Wybierz listę zmiennych z **Menedżera parametrów**, dla których urządzenie CanDevice ma generować PDO. Jeśli system docelowy oferuje taką możliwość, można wybierać listy parametrów typu „mapping”, które zarządzają zmiennymi procesowymi przewidzianymi wyłącznie dla tego mapowania. W takim przypadku do wyboru będą dostępne tylko takie listy parametrów. W innym przypadku będą dostępne wszystkie listy parametrów typu „zmiennie” i „instancje”.

Uwaga: Jeśli w ustawieniach systemu docelowego dla Menedżera parametrów został zdefiniowany „zakres indeksu dla mapowania”, urządzenie CanDevice będzie podczas mapowania uwzględniało **tylko** ten wybrany zakres i żaden inny, ewentualnie jeszcze zdefiniowane zakresy indeksu!

Obiekty wybranej listy są pokazywane w lewym oknie wyboru. W prawym oknie jest widoczna utworzona konfiguracja (**PDO**). Za pomocą przycisków **Wstaw odbierające PDO** albo **Wstaw nadające PDO** można tam pod elementami folderu utworzyć „PDO odbierające” i „PDO nadające”, które mają służyć do odbierania i nadawania wiadomości. By przyporządkować PDO odbierającemu i PDO nadającemu obiekt z listy po lewej stronie, zaznacz obiekt w lewym oknie oraz PDO w prawym oknie i kliknij przycisk >>. Obiekt zostanie dołączony do PDO w prawym oknie. **Właściwości** PDO można zdefiniować w oknie dialogowym używanym do konfiguracji innych modułów grupy master.

Za pomocą przycisku **Usuń** można usunąć z konfiguracji aktualnie zaznaczony PDO w prawym oknie.

Przykład:

Cel: Na pierwszym PDO odbierającym (COB-Id = 512 + NodeId) urządzenia CanDevice ma być odebrana zmienna PLC_PRG.a.

W związku z tym indeks/subindeks znajdujący się na liście zmiennych w katalogu obiektów (Menedżer parametrów) musi być skojarzony ze zmienną PLC_PRG.a: By otworzyć Menedżera parametrów, należy w ustawieniach systemu docelowego w punkcie „Funkcje sieciowe” aktywować Menedżera parametrów i ustawić właściwe zakresy indeksu/subindeksu.

Następnie w oknie dialogowym „Domyślne mapowanie PDO” urządzenia CanDevice zostanie dodany wpis indeksu/subindeksu jako wpis mapowania przyporządkowany do PDO.

9.6.10 Konfiguracja modułów DeviceNet

System CoDeSys obsługuje konfigurację sprzętową dla systemu magistrali, używającego międzynarodowego i znormalizowanego **protokołu DeviceNet (EN50325)**. Za pomocą DeviceNet są realizowane przeważnie sieci master-slave obsługujące właściwości Plug&Play, czyli wyposażone w magistralę z możliwością bezpośredniego podłączania czujników i elementów wykonawczych (wyłączniki zbliżeniowe, zawory).

Protokół komunikacyjny DeviceNet bazuje na sieci **CAN** (Controller Area Network). By transmisja danych mogła się odbywać, musi zaistnieć połączenie pomiędzy komunikującymi się modułami. Konfigurator DeviceNet dostępny w CoDeSys umożliwia zdefiniowanie urządzenia DeviceNet Master, które kontroluje transmisję danych jego urządzeń DeviceNet Slave w sieci. Obsługiwane są przy tym różne rodzaje komunikacji dla transmisji danych wejściowych i wyjściowych pomiędzy modułami slave. W normalnym przypadku urządzenie DeviceNet Master przejmuje funkcję „UCMM” (Unconnected Message Manager dla wielokrotnych połączeń odbywających się w tym samym czasie) i obsługuje kwerendy swoich urządzeń slave (kompatybilnych z UCMM) wysyłane przez inne urządzenia master.

By konfiguracja DeviceNet mogła być zrealizowana w konfiguratorze sterownika systemu CoDeSys potrzebny jest **plik konfiguracyjny** zezwalający na wstawianie urządzeń DeviceNet Master i modułów slave. Plik konfiguracyjny jest wyszukiwany automatycznie w katalogu ustawionym dla plików konfiguracyjnych (patrz rozdział 6.2, Projekt – Opcje – Katalogi).

Zgodnie z definicjami w pliku konfiguracyjnym *.cfg **pliki EDS** (pliki urządzeń, Electronic Data Sheet), które również znajdują się w katalogu aktualnie ustawionym dla plików konfiguracyjnych, mogą być używane w konfiguracji. W pliku EDS są opisane możliwości ustawień modułu DeviceNet. Należy pamiętać, że również pliki urządzeń CAN posiadają rozszerzenie .EDS, nie są jednak przeznaczone do używania w konfiguracji DeviceNet!

Warto jeszcze pamiętać o możliwości osobnego dodawania plików konfiguracji podczas pracy z projektem.

Jeśli w drzewie konfiguracji jest zaznaczone urządzenie DeviceNet Master, na odpowiednich kartach zakładki dostępne do wyboru są następujące **okna dialogowe** (w zależności od definicji w pliku konfiguracji): Parametry podstawowe, Parametry Device Net, Parametry modułu.

Jeśli w drzewie zaznaczone jest urządzenie DeviceNet Slave, podłączone do urządzenia DeviceNet Master, dostępne są następujące okna dialogowe: Parametry podstawowe, Parametry Device Net, Konfiguracja połączenia We/Wy, Parametry, Parametry modułu.

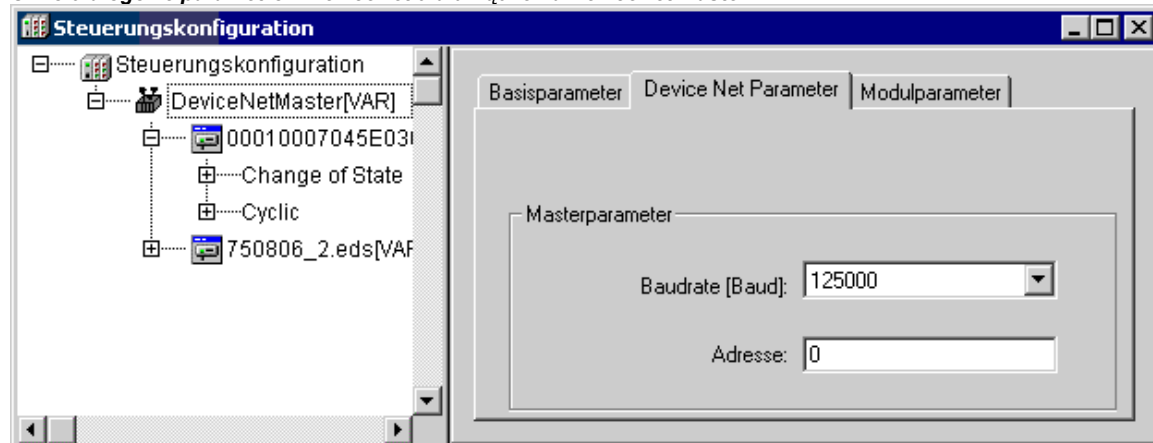
W następnych częściach rozdziału zostaną opisane poszczególne okna dialogowe konfiguracji:

Parametry podstawowe urządzenia DeviceNet Master

Okno dialogowe Parametry podstawowe urządzenia DeviceNet Master odpowiada pod względem punktów menu (**Id modułu**, **Numer węzła**, **Adres wejściowy**, **Adres wyjściowy**, **Adres diagnostyczny**) oknu dialogowemu pozostałych modułów (patrz rozdział 9.6.5, Parametry podstawowe modułu we/wy).

Parametry Device Net urządzenia DeviceNet Master

Okno dialogowe parametrów DeviceNet dla urządzenia DeviceNet Master



W polu **Adres** wprowadź nazwę urządzenia DeviceNet Master, która wynika z konfiguracji modułu. Znaczenie tej nazwy odpowiada numerowi Id węzła modułu CAN i nie należy jej mylić z numerem węzła lub adresem wprowadzonym w oknie dialogowym parametrów podstawowych! Musi być ona podana w notacji dziesiętnej jako wartość od 0 do 63; ustawienie domyślne: 0.

Ponadto należy zdefiniować **prędkość transmisji [bod]** danych w sieci. Dostępne są następujące ustawienia: 125000 (domyślnie), 250000, 500000.

Parametry modułu DeviceNet-Master

Okno dialogowe parametrów modułu DeviceNet Master odpowiada oknom dialogowym w innych modułach (patrz rozdział 9.6.5, Podstawowe parametry modułu we/wy). W tym miejscu są zilustrowane parametry przydzielone dla urządzenia master dodatkowo oprócz parametrów DeviceNet i parametrów magistrali w pliku konfiguracji i można je w zwykły sposób edytować.

Parametry podstawowe urządzenia DeviceNet Slave

Okno dialogowe Parametry podstawowe urządzenia DeviceNet Slave odpowiada pod względem punktów menu (**Adres wyjściowy** i **Adres wejściowy**) oknu dialogowemu pozostałych modułów (patrz rozdział 9.6.5, Parametry podstawowe modułu we/wy). Kierunek, wejście albo wyjście, jest ustalany z punktu widzenia modułu.

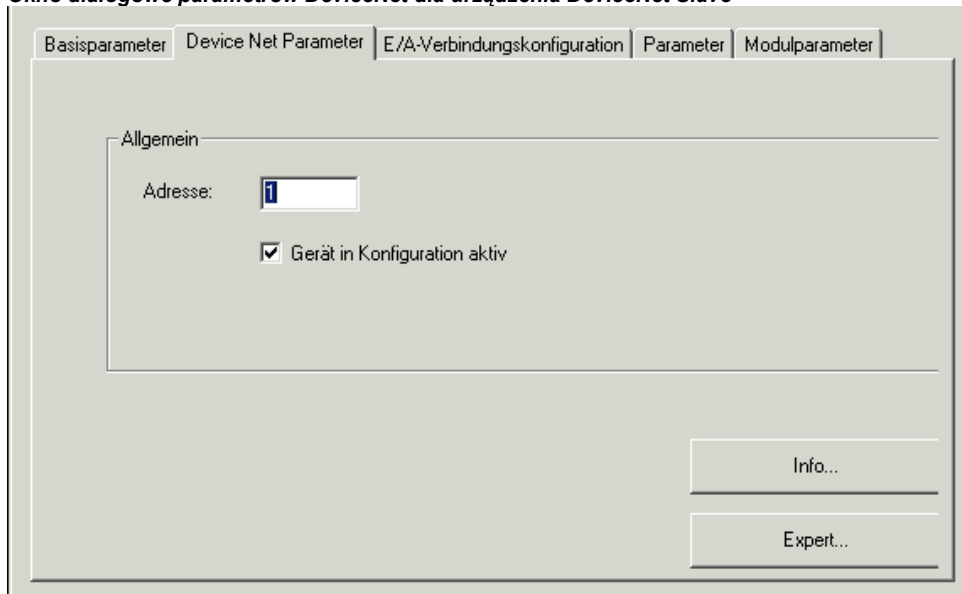
Parametry DeviceNet urządzenia DeviceNet Slave

W tym miejscu można zdefiniować parametry ogólne modułu slave:

Adres: Nazwa urządzenia DeviceNet Slave, która została nadana modułowi automatycznie. Znaczenie tej nazwy odpowiada numerowi Id węzła modułu CAN i nie należy jej mylić z numerem węzła lub adresem wprowadzonym w oknie dialogowym parametrów podstawowych! Musi być ona podana w notacji dziesiętnej jako wartość od 0 do 63.

Urządzenie aktywne w konfiguracji: Należy zaznaczyć tę opcję, by zdefiniować urządzenie jako aktywną stację w komunikacji w sieci.

Info...: Ten przycisk otwiera okno z treścią **pliku EDS** urządzenia. Należy pamiętać, że również pliki urządzeń CAN mogą posiadać rozszerzenie „.eds”, nie są one jednak przeznaczone do używania w konfiguracji DeviceNet!

Okno dialogowe parametrów DeviceNet dla urządzenia DeviceNet Slave

Expert...: Ten przycisk otwiera okno dialogowe **Ustawienia rozszerzone**, w którym można skonfigurować następujące punkty::

UCMM (Unconnected Message Manager for multiple connections): Jeśli ta opcja jest aktywna (ustawienie domyślne), urządzenie slave może odbierać komunikaty UCMM. Możliwe klasyfikacje: Grupa1, Grupa2 lub Grupa3 (ustawienie domyślne)

Kontrolę, odbywającą się domyślnie podczas **startu** sieci, można tutaj dezaktywować. Podczas kontroli urządzenia wartość podana w używanym pliku eds jest porównywana z wartością, która została odszukana w module:

Sprawdź ID dostawcy, Sprawdź typ urządzenia Device, Sprawdź kod produktu, Sprawdź wersję produktu

Okno dialogowe Ustawienia rozszerzone

Konfiguracja połączenia We/Wy urządzenia DeviceNet Slave

W tym miejscu można skonfigurować wejścia i wyjścia urządzenia, za pomocą których ma odbywać się transmisja danych (wartości parametrów). Definiowany jest typ połączenia i na podstawie oferowanych przez urządzenia możliwości konfiguracyjnych dla wejść i wyjść (plik EDS, wejścia, wyjścia) zestawiane są żądane wejścia i wyjścia.

Okno dialogowe konfiguracji połączenia We/Wy urządzenia DeviceNet Slave

The screenshot shows the configuration window for a DeviceNet Slave connection. The selected connection type is 'Cyclic'. The total I/O is 11 input bytes and 3 output bytes. The input configuration lists available connections and the configured ones, including 'Analog_Digital_Status_Input' (6 Bytes), 'Digital_Input_1' (1 Byte), and 'Digital_Input_1' (1 Byte, Set: 4 Bytes). The output configuration lists available connections and the configured ones, including 'PLC_Input_1_USINT' (1 Byte) and 'PLC_Input_1_USINT' (1 Byte, Set: 2 Bytes).

Wybrane połączenie We/Wy: Wprowadź tutaj żądany typ komunikacji, który ma obowiązywać dla realizowanej niższej konfiguracji połączenia we/wy urządzenia DeviceNet Slave:

Poll: Urządzenie master wysyła cyklicznie zapytania o dane urządzenia slave (komunikacja master-slave)

Bit Strobe: Urządzenie DeviceNet Master wysyła telegram broadcast do wszystkich urządzeń slave z żądaniem przesłania aktualnych danych. Stacje odpowiadają po kolei zaczynając od węzła 1. Dane zwracane przez urządzenia w odpowiedzi na polecenie Bit Strobe mają ograniczenie długości do 8 bajtów.

Change of State: Przy każdej zmianie wartości na wejściu urządzenie slave wysyła automatycznie dane do urządzenia master. Zapytania wysyłane przez urządzenie master nie są konieczne.

Cyclic: Urządzenie slave przesyła samodzielnie swoje dane po upływie czasu cyklu (funkcja „Heartbeat”).

Multicast Poll: funkcja nie jest zaimplementowana

We/Wy łącznie: W tym miejscu jest wyświetlana informacja, ile **bajtów wejściowych** i **bajtów wyjściowych** jest w danej chwili używanych dla wszystkich niżej skonfigurowanych wejść i wyjść. Wartości wynikają z długości zdefiniowanych dla wejść i wyjść w polach „Wejścia” i „Wyjścia”.

Zaawansowane: Ten przycisk otwiera okno dialogowe **Dodatkowe ustawienia**, które oferuje możliwość zmiany ustawień domyślnych dla bieżącego połączenia:

Expected Packet Rate: Domyślnie: 75, oczekiwany takt (w milisekundach), w którym urządzenie slave ma wysłać dane przez dostępne połączenie.

Limit czasu fragmentacji: [ms]: Ustawienie domyślne 1600 ms; Jeśli przesyłane dane przekraczają rozmiar 8 bajtów, zostaną one przesłane we fragmentach (w kilku telegramach). Limit czasu fragmentacji określa w milisekundach czas oczekiwania urządzenia master na potwierdzenie otrzymania fragmentu telegramu zanim master uruchomi działanie określone w opcji „Czynność w przypadku błędu monit. czasu”.

Czynność w przyp. błędu monit. czasu: W tym miejscu można ustawić, jakie działanie ma zostać wykonane w przypadku przekroczenia czasu kontrolnego:

Transition to timed out: (Ustawienie domyślne) To działanie jest działaniem definiowanym specjalnie dla urządzenia slave.

Auto delete: Połączenie We/Wy jest usuwane.

Auto reset: Połączenie pozostaje zachowane, urządzenie master wykonuje ponownie konfigurację urządzenia slave, Watchdog jest resetowany.

Dodatkowe opcje dla typu połączenia „Change of state”:

Czas blokady nadaw.: (Ustawienie domyślne: 1) Minimalny przedział czasu w milisekundach między dwiema wiadomościami, bez względu na to czy w tym czasie nastąpiła zmiana danych. Ta metoda zapobiega zbyt szybkiemu obciążeniu urządzenia przychodzącymi zapytaniami. Wartość 0 oznacza brak blokady czasu nadawania; w takim przypadku osiąga się najszybsze z możliwych czasu transmisji danych.

Timeout [ms]: (Ustawienie domyślne: 16) Jeśli czas „Heartbeatrate” został przekroczony o podaną tutaj wartość (w milisekundach) i nie zostały wysłane żadne dane, generowany jest błąd czasu kontrolnego.

Heartbeatrate [ms]: (Ustawienie domyślne: 250) Czas w milisekundach, po upływie którego urządzenie slave musi koniecznie wysłać swoje dane, nawet jeśli nie zostały one zmienione.

Dodatkowe opcje dla typu połączenia „Bit Strobe”:

Użyj bitu wyjścia: (Ustawienie domyślne: nieaktywne) Urządzenie slave podczas generowania odpowiedzi wykorzystuje bit wyjścia, który został przesłany w telegramie żądania przez urządzenie master.

Dodatkowe opcje dla typu połączenia „Cyclic”:

Intervall [ms]: (Ustawienie domyślne: 100) Przedział czasu (w milisekundach), w przeciągu którego urządzenie slave powinno automatycznie wysłać swoje dane (Heartbeat).

Timeout [ms]: (Ustawienie domyślne: 16) Jeśli czas Heartbeatrate został przekroczony o podaną tutaj wartość (w milisekundach) i nie zostały wysłane żadne dane, generowany jest błąd kontroli czasu.

Wejścia:

Wybierz z pola **Dostępne połączenia** żądane wejścia i przenieś je za pomocą przycisku >> do pola **Skonfigurowane połączenia wejściowe**. Za pomocą przycisku << można je również stamtąd usunąć.

By zdefiniować długość skonfigurowanego wpisu (w bajtach), kliknij dwukrotnie żądany wpis. Zostanie wyświetlone okno dialogowe **Długość połączenia**. Wybierz żadaną **długość w bajtach** i potwierdź przyciskiem OK. Ustawiona długość zostanie wtedy wyświetlona w nawiasach za skonfigurowanym wpisem.

Konfigurowane połączenia wejściowe są od razu pokazywane w drzewie konfiguracji. Poniżej urządzenia slave został dołączony wpis zawierający nazwę typu połączenia wraz z odpowiednimi połączeniami wejściowymi i wyjściowymi.

Wyjścia:

Skonfiguruj wyjścia zgodnie z opisem konfiguracji wejść.

Parametry DeviceNet Slave

Parametry ujęte na poniższej liście są zdefiniowane zgodnie z plikiem EDS urządzenia. Odpowiednio do konfiguracji wejść/wyjść odbywa się transmisja bieżących wartości w sieci.

Obiekt: Identyfikator parametru (obiektu), za pomocą którego jest on zarządzany na liście parametrów (folder obiektów). Ten numer obiektu jest generowany z numeru parametru podanego przy odpowiednim opisie parametru (dział [Params], „Param<nummer>”) w pliku EDS.

Typ: Typ danych parametru

Dostęp: Prawa dostępu: rw = odczyt/zapis, ro = tylko odczyt

Min., Maks.: Zakres wartości parametru, ograniczony za pomocą wartości minimalnej i maksymalnej

Domyślnie: wartość domyślna parametru

Wartość: Jeśli jest to zdefiniowane w pliku EDS, w tym miejscu można zmienić wartość parametru. Służy do tego lista wyboru zadanych wartości lub po kliknięciu na pole tabeli otwiera się pole wprowadzania danych.

Parametry modułu DeviceNet Slave

Okno dialogowe parametrów modułu DeviceNet Slave odpowiada oknom dialogowym w innych modułach (patrz rozdział 9.6.5, Podstawowe parametry modułu we/wy).

W tym miejscu są zilustrowane parametry przydzielone slave dodatkowo oprócz parametrów zdefiniowanych w oknie dialogowym parametrów w pliku konfiguracyjnym i w standardowym przypadku można je edytować.

9.6.11 Konfiguracja sterownika w trybie online

W trybie online przedstawiane są stany wejść i wyjść sterownika w edytorze konfiguracji. Jeśli wejście lub wyjście są typu boolean i przyjmują wartość „TRUE”, wówczas pole przed wejściem albo wyjściem w drzewie konfiguracji jest wyświetlane na niebiesko, a wartości nieboolean są dodawane na końcu wpisu (np. „=12”). Wartości boolean można przełączać kliknięciem myszy, w przypadku pozostałych wejść pokazuje się okno dialogowe wprowadzania nowej wartości, jeśli zostanie kliknięty początek wiersza. Nowa wartość jest natychmiast ustawiana w sterowniku po potwierdzeniu przyciskiem **OK**.

Ponadto należy pamiętać, że możliwości diagnozy online są zależne od systemu docelowego.

9.6.12 Skan/status/diagnoza urządzenia z systemu docelowego

W zależności od używanego systemu docelowego i ustawień w pliku konfiguracyjnym można uzyskać z systemu docelowego informacje o konfiguracji i aktualnym statusie albo informacje diagnostyczne dostępnych modułów sprzętowych w celu wykorzystania/wyświetlenia ich w konfiguracji sterownika w CoDeSys:

Skanuj konfigurację modułu

W zależności od systemu docelowego i ustawienia w pliku konfiguracyjnym można dla aktualnie zaznaczonego w Konfiguracji sterownika modułu używać polecenia menu kontekstowego **Skanuj konfigurację modułu**. Polecenie to jest dostępne wyłącznie w trybie online i powoduje wczytanie ze sterownika aktualnej konfiguracji sprzętowej tego modułu i ewentualnie zaznaczenie w drzewie konfiguracji pozycji podrzędnych w celu ich wstawienia. Jeśli plik konfiguracyjny dopuszcza taką możliwość, można wtedy w prosty sposób odtworzyć w CoDeSys dostępną konfigurację modułu.

Wczytaj status modułu

W zależności od systemu docelowego i ustawienia w pliku konfiguracyjnym można dla aktualnie zaznaczonego w Konfiguracji sterownika modułu używać polecenia menu kontekstowego **Wczytaj status modułu**. Polecenie to jest dostępne wyłącznie w trybie online i powoduje wczytanie ze sterownika aktualnego statusu tego modułu i zaznaczenie go w drzewku konfiguracji kolorem:

Kolor czarny: Moduł istnieje i ma poprawne parametry.

Kolor niebieski: Moduł istnieje, lecz ma niepoprawne parametry.

Kolor czerwony: Brak modułu.

Ta ilustracja statusu następuje automatycznie również po każdym pobraniu.

Wyświetl komunikaty diagnostyczne

W zależności od systemu docelowego i ustawienia w pliku konfiguracyjnym można dla aktualnie zaznaczonego w Konfiguracji sterownika modułu używać polecenia menu kontekstowego **Wyświetl komunikaty diagnostyczne**. Polecenie to jest dostępne wyłącznie w trybie online i powoduje wczytanie ze sterownika aktualnej informacji diagnostycznej tego modułu i wyświetlenie jej w oknie.

9.7 Konfiguracja zadań...

Przegląd

Do sterowania przetwarzaniem projektu można poza specjalnym programem PLC_PRG skorzystać również z Menedżera zadań.

Zadanie jest to jednostkowy czas przetwarzania w programie IEC. Jest ono definiowane za pomocą nazwy, priorytetu i typu, który określa, jaki warunek powoduje jego uruchomienie. Warunek ten może być zdefiniowany czasowo (cyklicznie, dowolnie) albo poprzez wewnętrzne lub zewnętrzne zdarzenie, w przypadku którego zadanie zostanie wykonane; na przykład wzrastające zbocze zmiennej globalnej lub zdarzenie Interrupt sterownika.

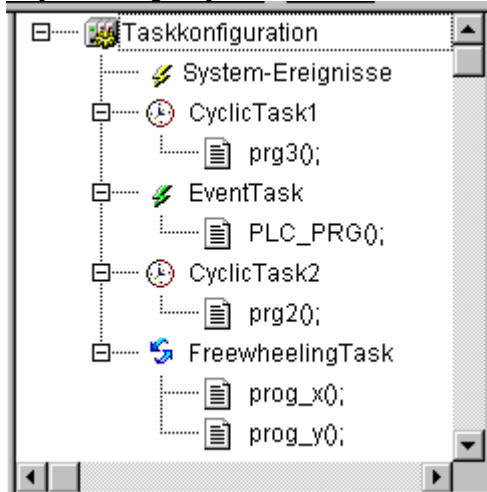
- Każdemu zadaniu można przyporządkować ciąg programów, które powinny zostać przetworzone podczas wykonywania zadania.
- Poprzez wzajemną kombinację priorytetu i warunku ustala się, w jakiej kolejności chronologicznej zadania zostaną wykonane.
- Dla każdego zadania można skonfigurować monitorowanie czasu (Watchdog); dostępne ustawienia zależą od systemu docelowego.
- W trybie online przetwarzanie zadania można obserwować na obrazie graficznym.
- Ponadto istnieje możliwość łączenia zdarzeń systemowych (np. start, stop, reset) bezpośrednio z wykonaniem modułu projektu.



Konfiguracja zadań jako obiekt znajduje się w oknie zakładki „Zasoby” w obiekcie Organizera. Okno edytor zadań składa się z dwóch części.

W lewej części okna zadania są przedstawione w postaci **drzewa konfiguracji**. W pierwszym wierszu występuje „Konfiguracja zadań”, poniżej występuje wiersz „Zdarzenia systemowe” i wpisy dla poszczególnych zadań, które są reprezentowane przez nazwę zadania. Do każdego wpisu są przyłączone odpowiednie wywołania programu.

Przykład konfiguracji zadań



W prawej części okna otwiera się okno dialogowe **Właściwości** dla wpisu zaznaczonego w drzewie konfiguracji. Można tutaj zdefiniować poszczególne zadania, wywołania programów albo zdarzenia systemowe. Możliwości konfiguracji dostępne w oknach dialogowych Właściwości są zależne od systemu docelowego i są definiowane w **pliku opisu** w formacie XML, którego referencję zawiera plik docelowy. Jeśli ustawienia domyślne zostaną tam uzupełnione przez użytkownika, wówczas będą one widoczne i dostępne do konfiguracji w dodatkowym oknie zakładki „Parametry” w prawej części okna.

Wskazówka: Nie należy używać takich samych funkcji łańcucha znaków w kilku zadaniach (patrz biblioteka standardowa standard.lib), gdyż w takim przypadku istnieje niebezpieczeństwo nadpisania wartości podczas przetwarzania zadania.

9.7.1 Obsługa konfiguratora zadań

Najważniejsze polecenia znajdują się w menu podręcznym (prawy przycisk myszy).

W nagłówku konfiguracji zadania znajduje się napis „Konfiguracja zadań”. Jeśli przed napisem znajduje się znak plus, wówczas lista zadań jest zwinięta. Listę można rozwinąć poprzez dwukrotne kliknięcie lub naciśnięcie klawisza <Enter>. Wtedy pokazuje się znak minus i ponowne dwukrotne kliknięcie powoduje zwinięcie listy.

Do każdego zadania dołączona jest lista wywołań programu, którą również można zwijać i rozwijać.

- Za pomocą polecenia „Wstaw” „Dodaj zadanie” można dodać zadanie za zaznaczonym wpisem.
- Za pomocą polecenia „Wstaw” „Dołącz zadanie” można dodać zadanie na końcu drzewa konfiguracji.
- Za pomocą polecenia „Wstaw” „Dołącz wywołanie programu” można dodać wywołanie programu do zadania zaznaczonego w drzewie konfiguracji.

Konfiguracja wpisu zaznaczonego w drzewie konfiguracji odbywa się w oknie dialogowym **Właściwości** w prawej części okna poprzez włączenie/wyłączenie opcji albo wpisów w polach edycji. Możliwości konfiguracji są zależne od systemu docelowego.

Może zostać otwarte albo okno dialogowe definiowania właściwości zadania (patrz „Dodaj zadanie”), okno dialogowe wpisywania wywołania programu (patrz „Dodaj wywołanie programu”) albo tabela zdarzeń systemowych. Wprowadzone ustawienia są od razu przenoszone do drzewa konfiguracji i tam wyświetlane po wstawieniu kursora.

Nazwę zadania lub programu można również edytować bezpośrednio w drzewie konfiguracji. W tym celu należy kliknąć nazwę lub zaznaczyć wpis i nacisnąć klawisz spacji, po czym wpisać nową nazwę w wyświetlonej ramce edycji.

Za pomocą klawiszy strzałek można zaznaczać następny albo poprzedni wpis w drzewie konfiguracji.

„Wstaw” „Dodaj zadanie” lub „Wstaw” „Dołącz zadanie”

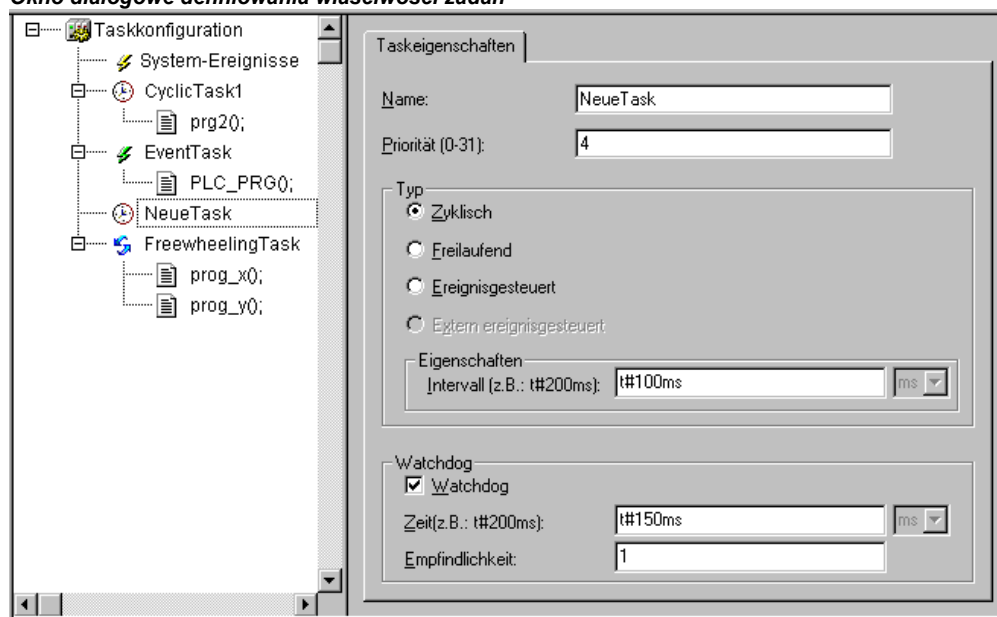
Za pomocą tego polecenia można dodać nowe zadanie w konfiguracji zadań. Każdy wpis składa się z symbolu i nazwy zadania.

Polecenie „**Dodaj zadanie**” jest dostępne, jeśli w drzewie konfiguracji jest zaznaczony wpis zadania lub wpis „zdarzenia systemowe”. Nowe zadanie jest wstawiane za zaznaczonym zadaniem. Jeśli zaznaczony jest wpis „Konfiguracja zadań”, wówczas można użyć polecenia „**Dołącz zadanie**”, by nowe zadanie wstawić na końcu listy.

Maksymalnie dopuszczalna liczba zadań zależy od systemu docelowego. Należy pamiętać, że pewna ilość zadań jest zarezerwowana w konfiguracji sterownika dla określonych modułów (definicja w bieżącym pliku cfg).

Po dodaniu nowego zadania otwiera się okno dialogowe w celu zdefiniowania **właściwości zadania**.

Okno dialogowe definiowania właściwości zadań



Wprowadź wymagane właściwości:

Nazwa: nazwa zadania, pod którą zadanie występuje w drzewie konfiguracji; nazwę można także tam edytować w polu edycji wyświetlanym po kliknięciu nazwy lub po naciśnięciu klawisza spacji.

Priorytet (0–31): liczba z przedziału od 0 do 31, przy czym 0 określa najwyższy, a 31 najniższy priorytet zadania

Typ:

Cyklicznie (🕒): Zadanie jest uruchamiane cyklicznie w odstępie czasu wprowadzonego w polu Odstęp.

Dowolne (🔄): Zadanie uruchamia się podczas startu programu i ponownie po każdym przetworzeniu. Nie określa się czasu cyklu.

Sterowanie zdarzeniami (🚦): Zadanie uruchamia się natychmiast przy pojawieniu się rosnącej krawędzi zmiennej zdefiniowanej w polu Zdarzenie.

Zewn. sterowane zdarzeniami (⚡): Zadanie uruchamia się, gdy tylko nastąpi zdarzenie systemowe zdefiniowane w polu Zdarzenie. Zdarzenia dostępne do wyboru na liście rozwijanej zależą od systemu docelowego i muszą być również zdefiniowane w pliku docelowym (nie należy ich mylić ze zdarzeniami systemowymi).

Właściwości:

Odstęp (dla typu zadania „cyklicznego” albo dla „zewnątrznie sterowanego zdarzeniami”, o ile zadanie wymaga wprowadzenia odstępu czasu): odstęp czasu, po upływie którego zadanie ma być ponownie uruchomione. Jeśli zostanie tu wprowadzona wartość, wówczas w następnym polu

można wybrać jednostkę – milisekundy [ms] lub mikrosekundy [µs]. W przypadku wprowadzeń w milisekundach wystarczy wprowadzenie liczby, właściwy format TIME (np. „t#200ms”) wyświetla się wtedy automatycznie po odświeżeniu ekranu. W przypadku wprowadzeń w mikrosekundach będzie wyświetlana dalej tylko liczba (np. „300”). Ewentualnie system docelowy może definiować **zdarzenia Singleton**. Są to zdarzenia, które dopuszczają uruchomienie tylko jednego pojedynczego zadania. Sprawdzenie, czy w projekcie takie zdarzenie uruchamia kilka zadań odbywa się podczas kompilacji. W tym celu wykorzystywany jest adres danych zmiennej zdarzenia, a nie jej nazwa.

Przykład: Jeśli system docelowy zdefiniował %MX1.1 i %IB4 jako zdarzenie Singleton, wtedy użycie następujących zmiennych jako zmiennych zdarzenia w konfiguracji zadań spowoduje wygenerowanie dwóch błędów (a i b jak również c i d mają za każdym razem ten sam adres). Jeśli system docelowy zdefiniował %MX1.1 i %IB4 jako zdarzenie Singleton, wtedy użycie następujących zmiennych jako zdarzenia spowoduje wygenerowanie dwóch błędów (a i b jak również c i d mają za każdym razem ten sam adres).

```
VAR_GLOBAL
  a AT %MX1.1: BOOL;
  b AT %MX1.1: BOOL;
  c AT %MB4: BOOL;
  d AT %MD1: BOOL;
END_VAR
```

Zdarzenie (dla typu „sterowane zdarzeniami” lub „zewnętrznie sterowane zdarzeniami”): zmienna globalna, która po wystąpieniu zbocza wznoszącego ma spowodować wykonanie zadania. Za pomocą tego przycisku lub klawisza <F2> można uruchomić funkcję pomocy przy wprowadzaniu zawierającą dostępne zmienne globalne, które można wybrać.

Jeśli zarówno w polu Odstęp, jak również w polu Zdarzenie nie wprowadzono żadnego wpisu, wówczas odstęp przetwarzania zależy od używanego systemu czasu przebiegu (patrz dokumentacja systemu przebiegu; w tym przypadku dla systemu CoDeSys SP NT od wersji V2.2 odstęp wynosi 10 ms).

Watchdog: (🚫):(🚫):

W zależności od systemu docelowego można skonfigurować funkcję kontroli czasu:

Watchdog: Zaznacz tę opcję () , jeśli zadanie ma kończyć się statusem błędu, w przypadku gdy przetwarzanie zadania przy ustawionej czułości (patrz niżej) trwa dłużej niż czas Watchdog zdefiniowany w polu „Czas” (mechanizm Watchdog).

Uwaga: System docelowy CoDeSys SP 32 Bit Full dezaktywuje funkcję Watchdog, dopóki aktywna jest funkcja kontroli przebiegu lub gdy przetwarzanie programu zatrzyma się w punkcie wstrzymania.

Czas (np.: t#200ms): Po upływie tego czasu następuje aktywowanie mechanizmu Watchdog zgodnie z ustawioną czułością (patrz niżej), o ile zadanie nie zakończyło się samoistnie. Określanie jednostki – patrz wyżej punkt „Odstęp”. Ewentualnie system docelowy żąda również podania czasu Watchdog jako procent odstępu czasu ustawionego dla zadania. W takim przypadku okno wyboru jednostki jest w kolorze szarym i zawiera symbol procentu „%”.

Czułość: W tym miejscu można podać jako liczbę całkowitą, po którym przekroczeniu czasu Watchdog sterownik ma zostać ustawiony w stanie błędu. Ustawieniem domyślnym jest „1”. Uwaga: Jeśli zostanie tu wprowadzona wartość „0”, wówczas funkcja Watchdog nie będzie działać!

Właściwości specyficzne dla producenta:

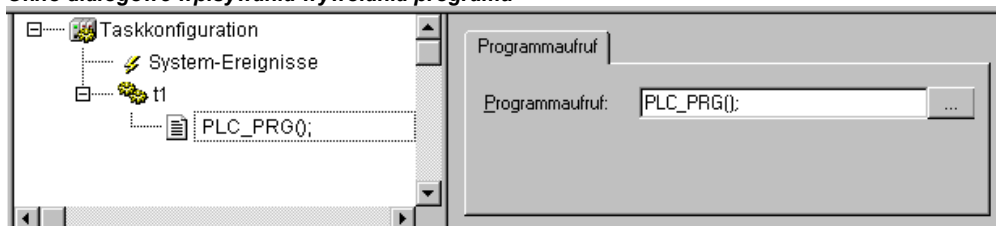
W uzupełnieniu do standardowych właściwości dla wybranego zadania na drugiej zakładce „Parametry” mogą pojawić się właściwości specyficzne dla producenta. Muszą być one przedtem zdefiniowane w pliku opisu końcowego przeznaczenia.

„Wstaw” „Dodaj wywołanie programu” lub „Wstaw” „Dołącz wywołanie programu”

To polecenie otwiera okno dialogowe, w którym można wpisać wywołanie programu dla zadania w konfiguracji zadań. Wpis w drzewie konfiguracji składa się z symbolu (📄) i nazwy programu.

W przypadku użycia polecenia „**Dodaj wywołanie programu**” nowe wywołanie programu jest dodawane przed zaznaczonym wywołaniem programu, a w przypadku użycia polecenia „**Dołącz wywołanie programu**” nowy wpis zostanie wstawiony na końcu listy wpisów programów.

Okno dialogowe wpisywania wywołania programu



W polu Wywołanie programu podaj nazwę programu ze swojego projektu albo za pomocą przycisku ... lub klawisza <F2> uruchom funkcję pomocy przy wprowadzaniu, by wybrać program z listy. Nazwę programu można również edytować w drzewie konfiguracji po zaznaczeniu wpisu programu. W tym celu należy albo kliknąć nazwę, albo zaznaczyć ją i nacisnąć klawisz spacji, po czym zmienić nazwę w polu edycji. Jeśli wybrany program żąda zmiennej wejściowej, wtedy należy ją wprowadzić z zwyczajnej postaci wraz z deklarowanym typem (np. prg(invar:=17)).

Przetwarzanie wywołań programu odbywa się później w trybie online zgodnie z kolejnością, w jakiej występują na liście, od góry do dołu.

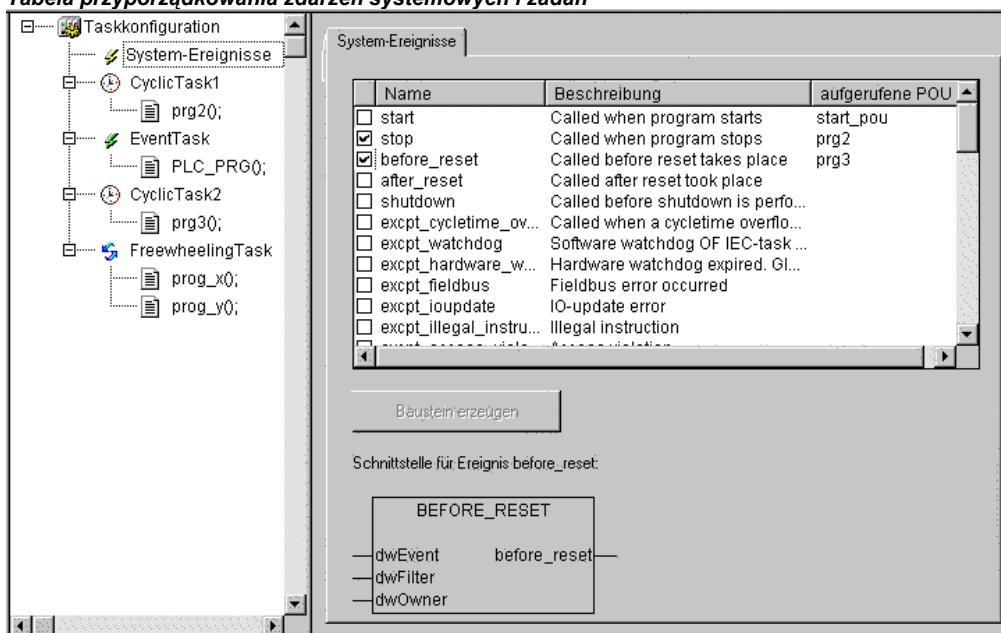
Wskazówka: Nie należy używać takich samych funkcji łańcucha znaków w kilku zadaniach (patrz rozdział 13.17.1, Elementy biblioteki standardowej), gdyż w takim przypadku istnieje niebezpieczeństwo nadpisania wartości podczas przetwarzania zadania.

9.7.2 Zdarzenia systemowe

Zdarzenie systemowe może zamiast zadania wywołać również moduł projektu i spowodować jego przetwarzanie. Używane do tego celu zdarzenia systemowe są zdefiniowane w systemie docelowym (definicja jest zawarta w pliku opisu w formacie XML; referencja do tego pliku znajduje się w pliku docelowym). Składają się one z listy obsługiwanych standardowych zdarzeń systemowych sterownika i ewentualnie z dodatkowych zdarzeń specyficznych dla producenta. Dostępne zdarzenia to np. Stop, Start, Zmiana online.

Przyporządkowanie zdarzeń systemowych do każdego z wywoływanych modułów odbywa się w oknie dialogowym **Zdarzenia**, które wyświetla się, jeśli w drzewie konfiguracji jest zaznaczony wpis „Zdarzenia systemowe”:

Tabela przyporządkowania zdarzeń systemowych i zadań



Każde zdarzenie jest przedstawiane w osobnym wierszu tabeli:

Nazwa i **Opis** pochodzą z opisu systemu docelowego, w kolumnie **wywołane POU** można wpisać moduł projektu, którego przetwarzanie rozpocznie się w chwili wystąpienia zdarzenia.

Nazwę istniejącego modułu można wprowadzić za pomocą funkcji pomocy przy wprowadzaniu (<F2>) lub ręcznie (np. „PLC_PRG” lub „PRG.ACT1”); można również wpisać nazwę jeszcze nie istniejącego modułu. **Uwaga w przypadku systemów docelowych RISC i Motorola 68K:** Nazwa funkcji skojarzonej ze zdarzeniem systemowym (funkcja callback) **musi** rozpoczynać się słowem „callback”!

By utworzyć nowo zdefiniowany moduł w projekcie, należy nacisnąć przycisk **Utwórz moduł <nazwa>**. Wtedy w Organizерze obiektów pojawi się moduł (funkcja), który w części z deklaracją zawiera automatycznie definicje parametrów transmisji, które ewentualnie mogą być potrzebne do przesłania zdarzenia.

Parametry zdarzenia w razie potrzeby wyświetlane są również graficznie jako moduł poniżej listy przyporządkowań po zaznaczeniu odpowiedniego wpisu w tabeli.

Wywołanie modułu poprzez zdarzenie nastąpi tylko wtedy, gdy wpis będzie aktywny, tzn. gdy w polu wyboru w pierwszej kolumnie jest wstawione zaznaczenie ()

9.7.3 Konfiguracja zadań w trybie online

W trybie online status zadania jest pokazywany w drzewie konfiguracji, a przebieg czasowy można obserwować na obrazie graficznym. Warunkiem jest, by biblioteki **SysTaskInfo.lib** i **SysLibTime.lib** były podłączone w projekcie. Funkcje biblioteki są używane wewnętrznie do obliczania czasów realizacji zadania.

Wskazanie statusu w drzewie konfiguracji:

W trybie online dla każdego zadania pokazywany jest aktualny status zadania (w nawiasach ostrych za wpisem zadania) oraz liczba wykonanych już cykli przetwarzania: Cykl aktualizacji dla tego wskazania jest taki sam jak cykl monitoringu wartości sterownika. Możliwe stany:

Idle	nie wystartował od ostatniej aktualizacji; używany szczególnie dla zadań wywoływanych zdarzeniem
Running	wystartował przynajmniej jeden raz od chwili ostatniej aktualizacji
Stop	zatrzymany
Stop on BP	zatrzymany z powodu osiągnięcia przez zdarzenie punktu wstrzymania
Stop on Error	błąd, np. dzielenie przez zero, błąd strony itp.
Stop Watchdog	przekroczenie czasu cyklu

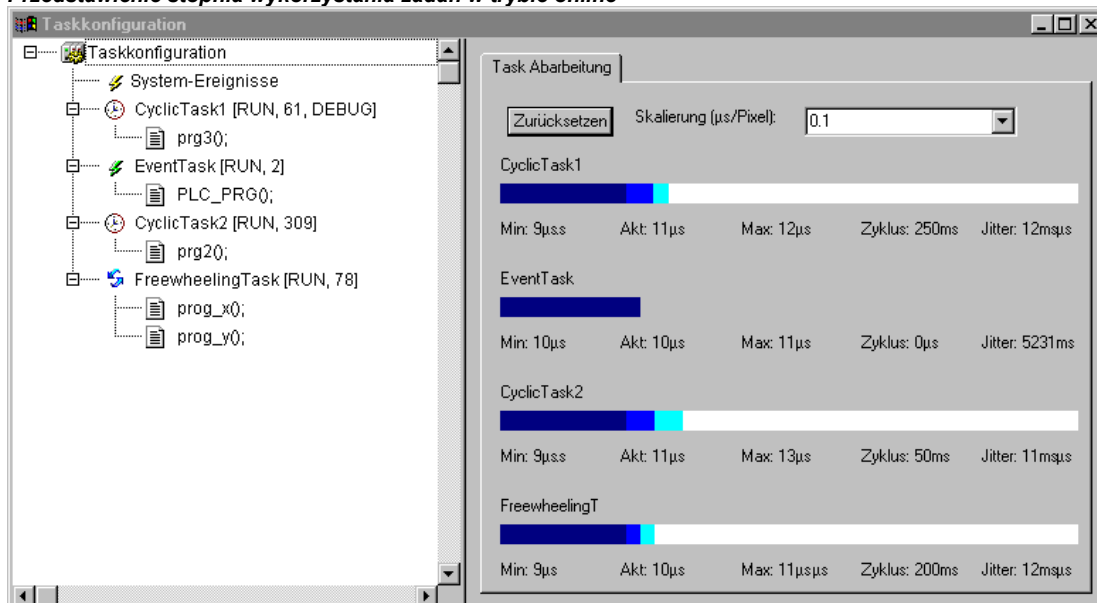
Wpis zadania w stanie „Stop on Error” i „Stop Watchdog” jest wyświetlany na czerwono.

Przedstawienie graficzne czasowego przebiegu zdarzenia

Stopień wykorzystania wszystkich zadań jest przedstawiany na wykresie słupkowym w prawej części okna po zaznaczeniu wpisu „Konfiguracja zadań” w drzewie konfiguracji.

Dla każdego zadania jest wyświetlany słupek, którego długość całkowita określa czas trwania cyklu. Pod słupkiem oraz poprzez odpowiednie kolory w słupku pokazywane są następujące wartości pomiarowe z lewej do prawej:

Przedstawienie stopnia wykorzystania zadań w trybie online



Min.: najkrótszy zmierzony czas przebiegu w μs

Akt: ostatnio zmierzony czas przebiegu w μs

Maks.: najdłuższy zmierzony czas przebiegu w μs

Cykl: całkowita długość cyklu w μs

Jitter: najdłuższe zmierzone opóźnienie (czas między uruchomieniem zadania a wyświetleniem przebiegu zadania przez uruchomiony system) w μs

Za pomocą przycisku **Reset** można wyzerować wartości Min., Maks. i Jitter.

Podziałkę wykresu (mikrosekundy/piksel) można wybrać z listy rozwijanej w punkcie **Skalowanie [$\mu\text{s}/\text{Pixel}$]**.

Dodatkowe funkcje online znajdują się w menu kontekstowym albo w menu „Dodatki”:

Jakie zadanie jest przetwarzane?

Podczas wykonywania obowiązują następujące reguły:

- Wykonywane jest zadanie, którego warunek jest spełniony, tzn. jeśli czas podany w polu Odstęp upłynął lub pojawiła się krawędź rosnąca zmiennej warunkowej zdefiniowanej dla zdarzenia.
- Jeśli kilka zadań mają spełniony warunek, wówczas wykonywane jest zadanie z najwyższym priorytetem.
- Jeśli kilka zadań mają spełniony warunek i taki sam priorytet, wówczas wykonywane jest zadanie, które najdłużej oczekuje na wykonanie.
- Przetwarzanie wywołań programu dla każdego zadania w trybie online odbywa się zgodnie z kolejnością w edytorze zadań, od góry do dołu.
- W zależności od systemu docelowego można wprowadzić takie ustawienie, że PLC_PRG będzie w każdym przypadku przetwarzany jako zadanie dowolne bez konieczności podłączania go w konfiguracji zadań.

„Dodatki” „Ustalanie debugowania zadania”

W systemach docelowych, które oferują funkcję „wielozadaniowości z wyłączeniem” można używając tego polecenia w trybie online zdefiniować zadanie, w którym ma odbywać się „debugowanie”. W drzewie konfiguracji występuje ono z tekstem „[DEBUG]”. Funkcje debugowania

odnoszą się wtedy tylko do tego zadania, tzn. przerwanie programu w punkcie wstrzymania nastąpi tylko wtedy, jeśli program „przeszedł” przez ustawione zadanie.

Definicja zadania debugowania jest przechowywana w projekcie i odświeżana automatycznie podczas logowania/pobierania.

„Dodatki” „Włączanie/wyłączanie zadania”

Za pomocą tego polecenia można włączać i wyłączać zadanie zaznaczone aktualnie w konfiguracji zadań. Zadanie, które zostało wyłączone nie jest uwzględniane w przetwarzaniu programu. W drzewie konfiguracji jest ono pokazywane jasnoszarą czcionką jako nieaktywne.

„Dodatki” „Hierarchia wywołań”

Jeśli podczas debugowania nastąpi zatrzymanie programu w punkcie wstrzymania, wówczas można użyć tego polecenia, by wyświetlić hierarchię wywołania odnośnego modułu. W takim przypadku zadanie debugowania musi być zaznaczone w drzewie konfiguracji zadań. Otworzy się okno **Hierarchia wywołań zadania <nazwa zadania>** z nazwą modułu, w którym znajduje się punkt wstrzymania (np. „prog_x (2)” w wierszu 2 w module prog_x). Następne pozycje w hierarchii wywołań pokazywane są w kolejności odwrotnej. Po potwierdzeniu przyciskiem **Idź do**, nastąpi przejście do zaznaczonej pozycji na liście.

9.8 Menedżer podglądu zmiennych i receptur...

9.8.1 Przegląd

Za pomocą Menedżera podglądu zmiennych i receptur można wyświetlać wartości wybranych zmiennych. Menedżer podglądu zmiennych i receptur umożliwia również inicjowanie zmiennych określonymi wartościami i przesyłanie ich od razu do sterownika („**Zapisz instrukcję**”). Tak samo bieżące wartości sterownika mogą być wczytane i przechowywane w Menedżerze podglądu zmiennych i receptur jako ustawienia predefiniowane („**Odczytaj instrukcję**”). Funkcje te są pomocne np. przy ustawianiu i wprowadzaniu parametrów regulacji.

Wszystkie utworzone listy kontrolne („**Wstaw**” „**Nowa lista kontrolna**”) są wyświetlane w lewej kolumnie Menedżera podglądu zmiennych i receptur i można je wybrać kliknięciem myszy lub za pomocą klawiszy strzałki. W prawym obszarze Menedżera podglądu zmiennych i receptur są pokazywane przynależne zmienne.

By rozpocząć pracę w Menedżerze podglądu zmiennych i receptur, otwórz obiekt  **Menedżer podglądu zmiennych i receptur** na zakładce **Zasoby** w obiekcie Organizera.

9.8.2 Menedżer podglądu zmiennych i receptur w trybie offline

W *trybie offline* można utworzyć jedną lub więcej list kontrolnych używając w Administratorze list kontrolnych i instrukcji polecenia „**Wstaw**” „**Nowa lista kontrolna**”.

W celu utworzenia zbioru kontrolowanych zmiennych można wywołać listę wszystkich zmiennych używając funkcji pomocy przy wprowadzaniu lub wprowadzić zmienne za pomocą klawiatury („funkcja Intellisense”) w następującej notacji:

```
<nazwa modułu>.<nazwa zmiennej>
```

W przypadku zmiennych globalnych nie trzeba podawać nazwy modułu. Na początku należy wstawić kropkę. Nazwa zmiennej może być wielopoziomowa. Adresy można podawać bezpośrednio.

Przykład zmiennej wielopoziomowej:

```
PLC_PRG.instancja1.instancja2.struktura.nazwa_składnika
```

Przykład zmiennej globalnej:

```
.global1.component1
```

Menedżer podglądu zmiennych i receptur w trybie offline

Watch	Value	Variable Name
Watch_2	0001	PLC_PRG.AMPEL1
Watch_1	0002	PLC_PRG.AMPEL2
Standard	0003	PLC_PRG.ZAEHLER:=6
	0004	
	0005	
	0006	
	0007	
	0008	
	0009	
	0010	

Zmienne na liście kontrolnej mogą być predefiniowane wartościami stałymi, tzn. w trybie online można te wartości za pomocą polecenia „Dodatki” „Zapisz instrukcję” zapisać jako zmienne. W takim przypadku wartość stałą należy przypisać do zmiennej używając operatora :=

Przykład:

```
PLC_PRG.TIMER:=50
```

Zmienna PLC_PRG.ZAEHLER pokazana na przykładzie została zdefiniowana wstępnie wartością 6.

W przypadku zmiennych typu **tablicowego**, **strukturalnego** lub **instancji bloku funkcji**: By przypisać im wartości, poszczególne elementy należy wprowadzać bezpośrednio. Przykład: Została zdefiniowana struktura STRU zawierająca składniki *a*, *b*, *c* oraz zadeklarowana zmienna strukturalna *struvar* w PLC_PRG. By teraz przypisać wartości dla *a*, *b*, *c*, należy wprowadzić je na listę kontrolną w następujący sposób:

```
PLC_PRG.struvar.a:=<wartość>
```

```
PLC_PRG.struvar.b:=<wartość>
```

```
PLC_PRG.struvar.c:=<wartość>
```

W analogiczny sposób należy przypisać wartości dla elementów tablicy: Przykład zmiennej typu tablicowego *arr_var* typu ARRAY[0...6]

```
PLC_PRG.arr_var[0]:=<wartość>
```

```
PLC_PRG.arr_var[1]:=<wartość>
```

...

Jeśli blok funkcji fb zawiera zmienne *x*, *y* i w PLC_PRG jest zadeklarowana instancja fb_inst typu fb, wartości początkowe dla zmiennych *x* i *y* można zdefiniować w następujący sposób:

```
PLC_PRG.fb_inst.x:=<wartość>
```

```
PLC_PRG.fb_inst.y:=<wartość>
```

„Wstaw” „Nowa lista kontrolna”

Za pomocą tego polecenia można w trybie offline dodać nową listę kontrolną. W wyświetlonym oknie dialogowym należy wprowadzić nazwę nowej listy kontrolnej.

„Dodatki” „Zmień nazwę listy kontrolnej”

Za pomocą tego polecenia można zmienić nazwę listy kontrolnej w Menedżerze podglądu zmiennych i receptur.

W wyświetlonym oknie dialogowym należy wprowadzić nową nazwę listy kontrolnej.

„Dodatki” „Zapisz listę kontrolną”

Za pomocą tego polecenia można zachować listę kontrolną. Otworzy się okno dialogowe zapisywania pliku. Nazwa pliku zawiera domyślnie nazwę listy kontrolnej oraz rozszerzenie „*.wtc”.

Zapisaną listę kontrolną można później otworzyć za pomocą polecenia „Dodatki” „Wczytaj listę kontrolną”.

„Dodatki” „Wczytaj listę kontrolną”

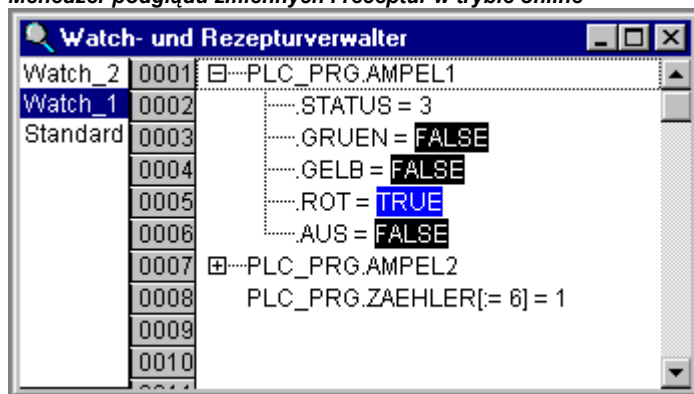
Za pomocą tego polecenia można wczytać zapisaną listę kontrolną. Otworzy się standardowe okno dialogowe otwierania pliku. Następnie należy wybrać wymagany plik z rozszerzeniem „*.wtc”. W wyświetlonym oknie dialogowym można wprowadzić nową nazwę listy kontrolnej. Domyślnie wyświetla się nazwa pliku bez rozszerzenia.

Za pomocą polecenia „Dodatki” „Zapisz listę kontrolną” można zachować listę kontrolną.

9.8.3 Menedżer podglądu zmiennych i receptur w trybie online

W trybie online pokazywane są wartości wprowadzonych zmiennych.

Menedżer podglądu zmiennych i receptur w trybie online



Wartości w formie struktury (tablice, struktury lub instancje bloków funkcji) posiadają znak plus przed identyfikatorem zmiennej. Klikając znak plus lub naciskając klawisz <Enter> zmienną można rozwinąć albo zwinąć.

Jeśli zmienna bloku funkcji jest zaznaczona na liście kontrolnej, wówczas menu kontekstowe zawiera dwa dodatkowe punkty „Powiększenie” i „Otwórz instancję”.

W celu wprowadzenia nowych zmiennych można wyłączyć wyświetlanie za pomocą polecenia „Dodatki” „Monitoring aktywny”. Po wprowadzeniu zmiennych wyświetlanie wartości można ponownie aktywować przy użyciu tego samego polecenia.

W trybie *offline* zmienne można zainicjować wartościami stałymi (poprzez wprowadzenie operatora := <wartość> po nazwie zmiennej). W trybie *online* można teraz te wartości zapisać w zmiennych używając polecenia „Dodatki” „Zapisz instrukcję”.

Więcej informacji na temat zmiennych typu tablicowego, strukturalnego i bloków funkcji znajduje się w rozdziale 9.8.2, „Menedżer podglądu zmiennych i receptur w trybie offline”.

Jeśli dla zmiennej przypisano wstępnie wartość w trybie *offline*, można przy użyciu polecenia „Dodatki” „Odczytaj instrukcję” zastąpić tamto ustawienie aktualną wartością zmiennej.

Wskazówka: Wczytywane są tylko wartości tej listy kontrolnej, która została wybrana w Menedżerze podglądu zmiennych i receptur!

„Dodatki” „Monitoring aktywny”

Za pomocą tego polecenia można włączać i wyłączać wyświetlanie list kontrolnych w Menedżerze podglądu zmiennych i receptur. Jeśli wyświetlanie jest aktywne, przed punktem menu znajduje się zaznaczenie.

By wprowadzić nowe zmienne tak jak w trybie *offline* lub przypisać wstępnie wartość, wyświetlanie należy wyłączyć używając tego polecenia. Po wprowadzeniu zmiennych wyświetlanie wartości można ponownie aktywować przy użyciu tego samego polecenia.

„Dodatki” „Zapisz instrukcję”

Za pomocą tego polecenia można w trybie online Menedżera podglądu zmiennych i receptur zapisać w zmiennych wartości predefiniowane (patrz tryb offline).

Wskazówka: Wczytywane są tylko wartości tej listy kontrolnej, która została wybrana w Menedżerze podglądu zmiennych i receptur!

„Dodatki” „Odczytaj instrukcję”

Za pomocą tego polecenia można w trybie online Menedżera podglądu zmiennych i receptur zastąpić predefiniowaną wartość zmiennej (patrz tryb offline) bieżącą wartością zmiennej.

Przykład:

```
PLC_PRG.Zaehler [:= <bieżąca wartość>] = <bieżąca wartość>
```

Wskazówka: Wczytywane są tylko wartości tej listy kontrolnej, która została wybrana w Menedżerze podglądu zmiennych i receptur!

Wymuszanie wartości i zapisywanie w Menedżerze list kontrolnych

W Menedżerze podglądu zmiennych i receptur można również używać poleceń „Wymuszanie wartości” i „Zapisz wartości”. Po kliknięciu wartości zmiennej otwiera się okno dialogowe, w którym można wprowadzić nową wartość zmiennej.


9.9 Zapis śledzenia...**9.9.1 Przegląd i konfiguracja**

Zapis śledzenia jest dostępny w CoDeSys po uruchomieniu odpowiedniej opcji w ustawieniach systemu docelowego (kategoria „Ogólne”).

Zapis śledzenia oznacza zapis przebiegu wartości zmiennych w ustalonym okresie. Wartości te są zapisywane w pamięci pierścieniowej (**Bufor śledzenia**). Po zapelnieniu pamięci „najstarsze” wartości od początku pamięci zostają nadpisane.

Maksymalnie można rejestrować równocześnie 20 zmiennych. Dla danej zmiennej można zarejestrować maksymalnie 500 wartości. Ponieważ bufor nie posiada w sterowniku stałej wartości, w przypadku wielu lub bardzo szerokich zmiennych (DWORD) zarejestrować poniżej 500 wartości.

Przykład: jeśli ma zostać zapisane 10 zmiennych WORD i pamięć w sterowniku ma długość 5000 bajtów, z każdej zmiennej można zapisać 250 wartości.

Aby zapisać śledzenie, należy otworzyć obiekt  **Zapis śledzenia** w zakładce **Zasoby** w Organizery obiektów. Należy utworzyć lub wczytać odpowiednią konfigurację śledzenia i zdefiniować zmienne śledzenia do zapisu (patrz „Dodatki” „Konfiguracja śledzenia” i „Wybór ilustrowanych zmiennych”).

Po utworzeniu konfiguracji w oknie dialogowym konfiguracji śledzenia i uruchomieniu zapisu w sterowniku („Uruchom śledzenie”) następuje zapis wartości zmiennych. Za pomocą polecenia „Wczytaj śledzenie” ostatnio zapisane wartości można przedstawić graficznie jako krzywe.

Zapis śledzenia (wartości zmiennych i konfiguracja) można zachować w formacie projektu (*.trc) lub formacie XML (*.mon) i ponownie wczytać. Samą konfigurację można zapisać i ponownie wczytać z pliku *.tcf.

Różne zapisy mogą być do obejrzenia w projekcie. Można je znaleźć na liście wyboru („**Śledzenie**”) w górnym prawym rogu okna śledzenia. Można z nich wybrać aktualną konfigurację śledzenia do zastosowania.

„Dodatki” „Konfiguracja śledzenia”

Polecenie to otwiera okno dialogowe do wpisania zmiennych do zapisu oraz różnych parametrów do zapisu śledzenia. Dialog ten można również otworzyć klikając dwukrotnie na szarym polu w oknie dialogowym Zapis śledzenia.

Konfiguracji należy najpierw nadać nazwę (**Trace Name**). Po potwierdzeniu i zamknięciu okna dialogowego konfiguracji za pomocą OK nazwa ta pojawi się w oknie „Zapis śledzenia” po prawej stronie u góry listy wyboru „Śledzenie”.

W polu **Komentarz** można ponadto umieścić dowolny tekst.

Okno dialogowe konfiguracji śledzenia

Lista zapisywanych **zmiennych** jest początkowo pusta. W celu wstawienia zmiennej należy ją podać w polu poniżej listy. Następnie można ją wstawić w listę za pomocą ikony **Wstaw** lub klawisza **<Enter>**. Można również skorzystać z **Asystenta deklaracji** (pomoc przy wpisywaniu). Możliwe jest również zastosowanie zmiennych enumeracji.

Zmienną można usunąć z listy po wybraniu jej i kliknięciu na ikonie **Usuń**.

W pole **Zmienna wyzwalająca** można wpisać zmienną boolean lub analogową (również zmienną enumeracji). W tym miejscu również można skorzystać z Pomocy przy wpisywaniu (<F2>). Zmienna wyzwalająca opisuje warunki przerwania śledzenia.

W **Poziomie wyzwalania** można podać, przy jakiej wartości analogowej zmiennej wyzwalającej następuje zdarzenie wyzwalania. Wartość tę można również podać za pomocą stałej ENUM.

Jeśli została wybrana **krawędź wyzwalająca dodatnia**, wówczas zdarzenie wyzwalające jest uruchamiane krawędzią wzrastającą zmiennej wyzwalającej typu boolean albo gdy analogowa zmienna wyzwalająca przekracza poziom wyzwalania z dołu do góry. W przypadku **ujemnej** krawędzi

wyzwalającej odpowiednio wyzwalanie następuje po krawędzi opadającej lub przekroczeniu z góry do dołu. W **obu przypadkach** po opadającej lub wzrastającej krawędzi lub dodatnim i ujemnym przebiegu następuje wyzwolenie, w przypadku **braku** nie następuje zdarzenie wyzwalania.

W **Pozycji wyzwalania** należy podać w procentach, ile wartości pomiarowych ma zostać zarejestrowanych przed wystąpieniem zdarzenia wyzwalania. Jeśli zostanie tu wprowadzona liczba 25, wówczas zostanie pokazanych 25% wartości pomiarowych zarejestrowanych przed wystąpieniem zdarzenia wyzwalania, a 75% wartości pomiarowych po zdarzeniu wyzwalania, po czym następuje przerwanie zapisu śledzenia.

W polu **Częstość próbkowania** można podać odstęp czasu pomiędzy kolejnymi chwilami zapisu w milisekundach albo mikrosekundach, o ile system docelowy dopuszcza taką możliwość. Wstępne ustawienie „0” oznacza: jedno próbkowanie na cykl.

Wybrać tryb przywołania zapisanych wartości (**Zapis**): W **Pojedyncze** jest ilustrowana raz zadana **ilość pomiarów**. W przypadku opcji **Ciągłe** odczyt zapisu śledzenia zadanej ilości wartości pomiarowych jest za każdym razem uruchamiany od początku. Przykładowo jeśli ilość wartości pomiarowych wynosi „35”, pierwszy zapis zawiera pierwszych 35 wartości pomiarowych (od 1 do 35), następny zapis kolejnych 35 wartości pomiarowych (36–70) itd. W przypadku opcji **Ręcznie** wczytywanie zapisu śledzenia odbywa się poleceniem menu „Dodatki” „Wczytaj śledzenie”.

Tryb przywołania działa niezależnie od ustawienia zmiennej wyzwalającej. Jeśli nie jest podana zmienna wyzwalająca, bufor śledzenia zostaje wypełniony ilością zadanych wartości pomiarowych i po przywołaniu zostaje wczytana i zilustrowana zawartość bufora.

Za pomocą ikony **Zachowaj** konfiguracja śledzenia zostaje zachowana w pliku (*.tcf). Służy do tego standardowy dialog „Zapisz plik jako”.

Za pomocą ikony **Wczytaj** można ponownie załadować zachowaną konfigurację śledzenia. Służy do tego standardowy dialog „Otwórz plik”.

Wskazówka: Należy pamiętać, że polecenia Zapisz i Wczytaj z okna dialogowego konfiguracji dotyczą tylko konfiguracji, nie zaś wartości zapisu śledzenia (w przeciwieństwie do poleceń menu „Dodatki” „Zachowaj śledzenie” i „Dodatki” „Wczytaj śledzenie”).

Jeśli pole **Zmienna wyzwalająca** jest puste, zapis śledzenia trwa bez końca i można go zakończyć za pomocą polecenia menu „Dodatki” „Zatrzymaj śledzenie”.

Wskazówka: Jeśli do sterowania przebiegu programu jest stosowana konfiguracja zadania, funkcja śledzenia odnosi się do zadania debugowania (patrz rozdział 9.7, Konfiguracja zadania).

Wybór zmiennych do wyświetlania

Kontrolki combobox po prawej stronie okna ilustracji krzywych zawierają wszystkie zdefiniowane w konfiguracji śledzenia zmienne śledzenia. Po wybraniu z listy zmiennej jej wartości po wczytaniu bufora śledzenia są wyświetlane w odpowiednim kolorze (Var 0 – kolor zielony itd.). Zmienne można wybierać również po wyświetleniu krzywych.

W oknie śledzenia można obserwować równocześnie maksymalnie osiem zmiennych.

9.9.2 Uruchamianie zapisu śledzenia

„Dodatki” „Uruchom śledzenie”

Symbol: 

Za pomocą tego polecenia konfiguracja śledzenia jest przesyłana do sterownika i tam rozpoczyna się zapis śledzenia.

„Dodatki” „Wczytaj śledzenie”

Symbol: 

Za pomocą tego polecenia aktualny bufor śledzenia zostaje automatycznie wczytany ze sterownika i są ilustrowane wartości wybranych zmiennych.

„Dodatki” „Wczytaj śledzenie automatycznie”

Za pomocą tego polecenia aktualny bufor śledzenia zostaje automatycznie wczytany ze sterownika i wartości są ilustrowane kolejno.

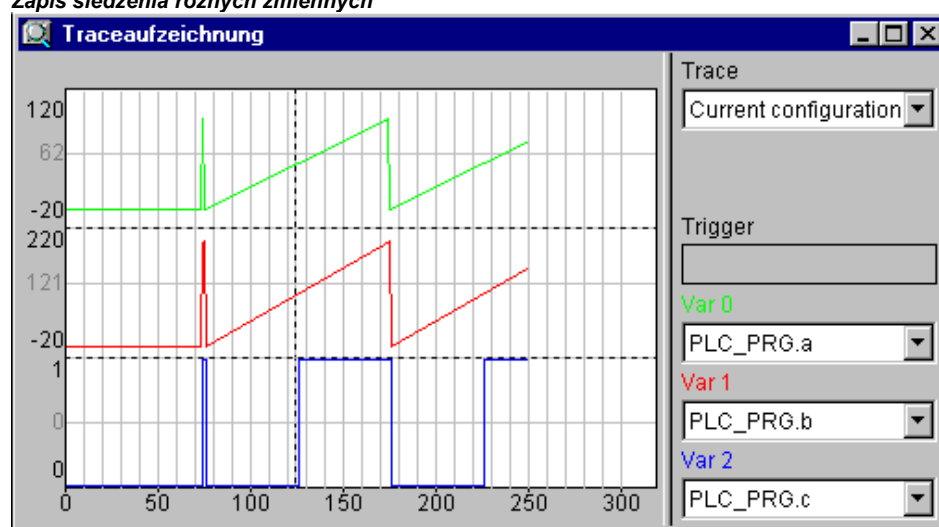
„Dodatki” „Zatrzymaj śledzenie”

Symbol: 

Polecenie to zatrzymuje w sterowniku zapis śledzenia.

9.9.3 Czytanie zapisu śledzenia

Zapis śledzenia różnych zmiennych



W oknie śledzenia u góry po prawej stronie („Śledzenie”) jest wyświetlana nazwa aktualnie stosowanej Konfiguracji śledzenia, zaś po prawej stronie może być pokazywany komentarz.

W przypadku wczytania bufora śledzenia („Dodatki” „Uruchom śledzenie”) można wczytać wartości wszystkich ilustrowanych zmiennych („Dodatki” „Wczytaj śledzenie” lub „Dodatki” „Wczytaj śledzenie automatycznie”) i w odpowiedni sposób zilustrować je w oknie śledzenia. Jeśli nie jest ustawiona Częstość próbkowania oś X zostaje opisana kolejnym numerem zarejestrowanej wartości. Bufor śledzenia zostaje wykasowany po zatrzymaniu zapisu („Dodatki” „Zatrzymaj śledzenie”).

Wskazanie statusu okna śledzenia ukazuje, czy **Bufor śledzenia** nie jest jeszcze pełny i czy zapis śledzenia jeszcze trwa lub został już zakończony.

Po podaniu wartości **Częstości próbkowania** oś X ukazuje czas dla danej wartości pomiarowej. „Najstarszej” zarejestrowanej wartości pomiarowej jest przyporządkowany czas 0. W przykładzie są ukazane wartości z ostatnich 250 ms.

Oś Y jest opisana wartościami odpowiedniego typu danych. Skalowanie jest ustawione w taki sposób, by najniższa i najwyższa wartość pasowały do zakresu ilustracji. Na przykładzie Var 0 ma najniższą wartość 0, jako najwyższa wartość jest przyjmowane 100, stąd również ustawienie skali po lewej stronie.

Po spełnieniu Warunku wyzwolenia na interfejsie między wartościami przed nastąpieniem warunku wyzwolenia i po nim pojawia się pionowa przerywana linia.

„Dodatki” „Wyświetl kursor”

Najszybszą metodą ustawienia kursora w oknie śledzenia jest kliknięcie lewym przyciskiem myszy obszaru okna. Kursor można w dowolny sposób przesuwac za pomocą myszy. Za pomocą okna

graficznego można odczytać aktualną pozycję x kursora. Oprócz Var0, Var1, ... , VarN jest ilustrowana wartość danej zmiennej.

Kolejną możliwością stanowi polecenie „Dodatki” „Wyświetl kursor”. Za pomocą tego polecenia w zapisie śledzenia są wyświetlane dwie pionowe początkowo nałożone na siebie. Jedną z tych linii można za pomocą klawiszy strzałek przesunąć w prawo i lewo. Naciśnięcie skrótu klawiszowego **<Ctrl> + <w lewo>**, lub **<Ctrl> + <w prawo>** powoduje zwiększenie prędkości przesuwania o współczynnik 10.

Dodatkowe naciśnięcie klawisza **<Shift>** powoduje przesunięcie innej linii wskazującej różnicę do pierwszej.

„Dodatki” „Kanał wielokrotny”

Za pomocą tego polecenia można przełączać między jedno- i wielokanałową prezentacją graficzną zapisu śledzenia. W przypadku ilustracji wielokanałowej punkt menu jest odpowiednio oznaczony.

Wstępnie ustawiona jest ilustracja wielokanałowa. W tym przypadku okno ilustracji jest podzielone na maksymalnie osiem ilustrowanych krzywych. Dla każdej z nich przy krawędzi jest podawana wartość maksymalna i minimalna.

W ilustracji jednokanałowej wszystkie krzywe są ilustrowane z tym samym współczynnikiem skalowania i nakładane na siebie. Może to być przydatne do ilustracji odchyłek krzywych.

„Dodatki” „Siatka współrzędnych”

Za pomocą tego polecenia można w oknie graficznego przedstawienia zapisu śledzenia włączyć i wyłączyć siatkę współrzędnych. Jeśli jest ona włączona, komenda menu jest zaznaczona.

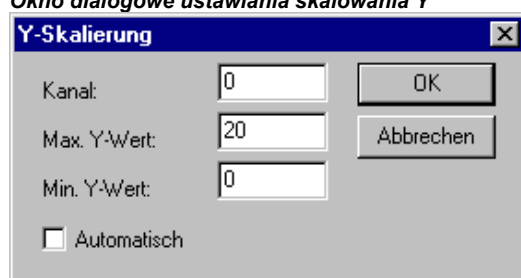
„Dodatki” „Skalowanie Y”

Za pomocą tego polecenia można zmodyfikować zadane skalowanie Y krzywej na graficznym obrazie zapisu śledzenia. Również po kliknięciu na krzywej otwiera się dialog „Skalowanie Y”.

Gdy jest aktywna opcja **Automatycznie**, stosowane jest skalowanie domyślne zależne od typu odpowiedniej zmiennej. W przypadku enumeracji jako opis skali są wyświetlane odpowiednie wartości enumeracji. W celu zmiany skalowania należy wyłączyć opcję „Automatycznie” i podać numer wymaganej krzywej (**kanał**) i nową najwyższą (**wartość maks. Y**) oraz nową wartość najniższą (**wartość maks. Y**) na osi y.

Również po kliknięciu na krzywej otwiera się opisany dialog.

Okno dialogowe ustawiania skalowania Y



„Dodatki” „Rozciągnij”

Symbol:

Za pomocą tego polecenia wydane wartości Zapisu śledzenia można rozciągać (zoomować). Pozycję początkową ustawia się za pomocą poziomego paska przesuwania. W przypadku wielokrotnego, następującego po sobie rozciągania w oknie wyświetlany jest wciąż krótszy wycinek śledzenia.

Polecenie to jest odwrotne do polecenia „Dodatki” „Kompresuj”.

„Dodatki” „Kompresuj”

Symbol: 

Za pomocą tego polecenia można skompresować wydane wartości Zapisu śledzenia tzn. po tym poleceniu można obserwować przebieg zmiennych śledzenia w dłuższym okresie. Możliwe jest wielokrotne wykonywanie tego polecenia.

Polecenie to jest odwrotne do polecenia „Dodatki” „Rozciągnij”.

9.9.4 „Dodatki” „Zapisz wartości śledzenia”

Polecenia tego menu służą do zachowania konfiguracji + wartości zapisu śledzenia w pliku w formacie projektu lub do ich wczytania z niego. Ponadto zapis można zachować w pliku ASCII.

Wskazówka: Można korzystać w alternatywnych możliwości zachowywania i wczytywania znajdujących się w menu „Dodatki” „Zewnętrzne konfiguracje śledzenia” (format XML, plik *.mon)!

„Zapisz wartości”

Za pomocą tego polecenia można zachować zapis śledzenia (wartości + konfiguracja). Otwarty zostaje dialog zachowywania pliku. Nazwa pliku zawiera rozszerzenie „*.trc”.

Uwaga: w tym miejscu w formacie projektu zostają zachowane zarówno wartości pomiarowe jak i konfiguracja śledzenia, podczas gdy Zapisz w oknie dialogowym konfiguracji dotyczy wyłącznie konfiguracji.

Ponadto wartości pomiarowe + konfigurację można zachować również w pliku w formacie XML, patrz menu „Zewnętrzne konfiguracje śledzenia”.

Zachowany zapis śledzenia można ponownie wczytać za pomocą „Dodatki” „Zapisz wartości śledzenia” „Wczytaj wartości”.

Wskazówka: Alternatywne możliwości zachowywania dają polecenia menu „Dodatki” „Zewnętrzne konfiguracje śledzenia”.

„Wczytaj wartości”

Za pomocą tego polecenia można ponownie wczytać zachować zapis śledzenia (wartości + konfiguracja). Otworzy się okno dialogowe otwierania pliku. Następnie należy wybrać wymagany plik z rozszerzeniem „*.trc”. Zapis jest ilustrowany w oknie śledzenia i konfiguracja zostaje zastosowana w projekcie jako aktualna.

Za pomocą „Dodatki” „Zapisz śledzenie” można zachować zapis śledzenia w pliku *.trc.

„Wartości w pliku ASCII”

Za pomocą tego polecenia można zachować zapis śledzenia w pliku ASCII. Otwarte zostaje okno dialogowe zapisywania pliku. Nazwa pliku zawiera rozszerzenie „*.txt”. Wartości zostają zachowane w pliku na podstawie następującego schematu:

```
CoDeSys Trace
D:\CODESYS\PROJECTS\AMPEL.PRO
Cykl PLC_PRG.ZAEHLER PLC_PRG.LIGHT1
0 2 1
1 2 1
2 2 1
.....
```

Jeśli w Konfiguracji śledzenia nie została ustawiona częstość próbkowania, w pierwszej komunie znajduje się cykl, tzn. za każdym razem dla danego cyklu jest rejestrowana jedna wartość. W innym

przypadku w tym miejscu jest wpisywany czas w [ms], w którym zostały zachowane wartości zmiennych od uruchomienia zapisu śledzenia.

W kolejnych kolumnach zachowywane są odpowiednie wartości zmiennych śledzenia. Wartości są rozdzielone spacjami.

Odpowiednie nazwy zmiennych są ilustrowane kolejno w trzeciej linii (PLC_PRG.ZAEHLER, PLC_PRG.LIGHT1).

9.9.5 „Dodatki” „Zewnętrzne konfiguracje śledzenia”

Polecenia tego menu służą do zachowania konfiguracji śledzenia + wartości lub do ich wczytania ze sterownika do projektu. Ponadto można ustawić jedną z konfiguracji jako tę do zastosowania w projekcie.

Wskazówka: Można przy tym korzystać w alternatywnych możliwości zachowywania i wczytywania znajdujących się w menu „Dodatki” „Zapisz wartości śledzenia” (format projektu, plik *.trc, ASCII)!

Polecenia tego menu służą do zachowania konfiguracji śledzenia + wartości lub do ich wczytania ze sterownika do projektu. Ponadto można ustawić jedną z konfiguracji jako tę do zastosowania w projekcie.

Wskazówka: Można przy tym korzystać w alternatywnych możliwości zachowywania i wczytywania znajdujących się w menu „Dodatki” „Zapisz wartości śledzenia” (format projektu, plik *.trc, ASCII)!

„Zapisz do pliku”

Za pomocą tego polecenia zapis śledzenia (konfiguracja + wartości) można zachować w pliku w formacie XML. Otwarte zostaje okno dialogowe zachowywania pliku. Automatycznie jest stosowane rozszerzenie „*.mon”.

Plik *.mon można wczytać do projektu za pomocą polecenia „Wczytaj z pliku”.

„Wczytaj z pliku”

Za pomocą tego polecenia można wczytać do projektu zapis śledzenia (konfiguracja + wartości) istniejące w pliku w formacie XML. W tym celu okno dialogowe do otwierania pliku automatycznie obsługuje wyszukiwanie plików w rozszerzeniem „*.mon”. Załadowany zapis śledzenia jest ilustrowany w oknie śledzenia i jest dołączany do listy wyboru w polu „Śledzenie” okna dialogowego konfiguracji. Aby stał się on aktualną Konfiguracją projektu, należy wybrać polecenie „Zastosuj jako konfigurację projektu”.

Plik *.mon można utworzyć za pomocą polecenia „Zapisz do pliku”.

Wskazówka: Alternatywne możliwości zachowywania i wczytywania dają polecenia menu „Dodatki” „Zapisz wartości śledzenia”.

„Zapisz w sterowniku”

Za pomocą tego polecenia w trybie online można wczytać do sterownika zapis śledzenia w pliku w formacie XML. W tym celu otwiera się standardowy dialog wyboru pliku, przy czym domyślnie są wyświetlane pliki z rozszerzeniem *.mon. Należy przy tym uwzględnić możliwość zachowania w plikach *.mon konfiguracji śledzenia w formacie XML („Dodatki” „Zapisz do pliku”).

„Wczytaj ze sterownika”

Za pomocą tego polecenia można wczytać do projektu aktualnie znajdujący się w sterowniku zapis śledzenia (konfiguracja + wartości, plik w formacie XML). Są one ilustrowane w oknie śledzenia i można je zastosować jako aktualną Konfigurację projektu.

„Zastosuj jako konfigurację projektu”

Za pomocą tego polecenia konfigurację śledzenia wybraną w oknie „Śledzenie” w dialogu konfiguracji można zastosować w projekcie jako aktualnie aktywną konfigurację śledzenia. Lista wyboru zawiera

oprócz chwilowo aktywnych (na najwyższym miejscu) przykładowy przegląd wszystkich pozostałych konfiguracji wczytanych do projektu za pomocą polecenia „Wczytaj z pliku” z plików (*.mon).

9.10 Obszar roboczy

Ten węzeł w zakładce „Zasoby” zawiera obraz ustawionych opcji projektu (patrz rozdział 6.2, Opcje projektu). Po jego otwarciu pojawia się okno dialogowe **Opcje** ze znanymi już kategoriami.

9.11 Menedżer parametrów...

Menedżer parametrów do właściwy dla systemu docelowego składnik systemu programowania **CoDeSys**, który należy włączyć w ustawieniach systemu docelowego.

Menedżer parametrów może być stosowany do udostępniania **parametrów** wszystkim kompatybilnym z CoDeSys systemom w sieci w celu **wymiany danych** (w typowy sposób za pomocą magistrali polowej). W tym celu w edytorze można wygenerować, edytować, wczytać do systemu docelowego i odczytać z niego listy parametrów.

Wskazówka: Listy parametrów można również bezpośrednio wygenerować lub wypełnić za pomocą instrukcji pragma w obrębie deklaracji.

Co to są parametry?

W związku z tym parametry zostały podzielone na następujące typy:

- zmienne procesu projektu IEC w CoDeSys
- parametry niezależne od procesu
- specyficzne parametry systemowe, wstępnie zdefiniowane przez system docelowy
- instancje bloku funkcji lub zmienne strukturalne, tablice

Każdy parametr jest oznaczony za pomocą określonego zestawu **Atrybutów**, takich jak np. „Wartość”, „Wartość domyślna”, „Prawa dostępu” i w specjalny sposób za pomocą **jednoznacznego klucza dostępu** („Indeks”, „SubIndeks”, „Nazwa”), przez który można w celu odczytu i zapisu danych uzyskać dostęp do wpisu na listę parametrów. Ta wymiana danych może nastąpić za pomocą **usług komunikacyjnych** przy czym nie ma potrzeby znać adresów zmiennych lub stosować dodatkowe funkcje. Przez to stosowanie menedżera parametrów stanowi funkcjonalnie alternatywę stosowania zmiennych sieciowych.

Co to są listy parametrów?

Listy parametrów służą do zarządzania parametrami przy czym można je zapisać w projekcie i wczytać w aktualnie powiązany z programem IEC system docelowy. Dla każdego typu parametrów (patrz wyżej) istnieje odpowiedni typ listy.

Każdy wpis parametru jest zilustrowany w linii listy parametrów. Każda **kolumna** listy reprezentuje jeden z atrybutów parametru (np. indeks, wartość domyślna...). Dodatkowo do zdefiniowanego zestawu atrybutów standardowych mogą być dostępne również właściwe dla producenta atrybuty opisu danego parametru.

Od definicji w **pliku opisu właściwym dla systemu docelowego** (odconfig.xml) zależy, które atrybuty, czyli kolumny w edytorze menedżera parametrów są widoczne i edytowalne i jaki jest ich układ na liście parametrów. Jeśli brak jest pliku opisu, prezentowany jest cały standardowy zestaw atrybutów, wstępnie wypełniony wartościami standardowymi.

Oprócz listy zmiennych projektu i stałych projektu za pomocą menedżera parametrów można zarządzać również listami parametrów systemowych. Są one na stałe zdefiniowane przez system docelowy. Ponadto można utworzyć listy tablic, instancji bloków funkcji lub zmiennych strukturalnych, nałożone na zdefiniowane przez użytkownika **szablony**, które można utworzyć również w menedżerze parametrów.

Ponieważ dane są zarządzane niezależnie od programu IEC, lista parametrów może być stosowana np. do zapisu „Instrukcji”, które zostają zachowane również w przypadku zastąpienia programu inną jego wersją. Ponadto aktualny sterownik można zasilać różnymi „Instrukcjami” bez potrzeby pobierania programu.

Wskazówka: Od systemu docelowego zależy, czy zawartość menedżera parametrów przy tworzeniu **projektu bootowalnego** ma być w nim zastosowana.

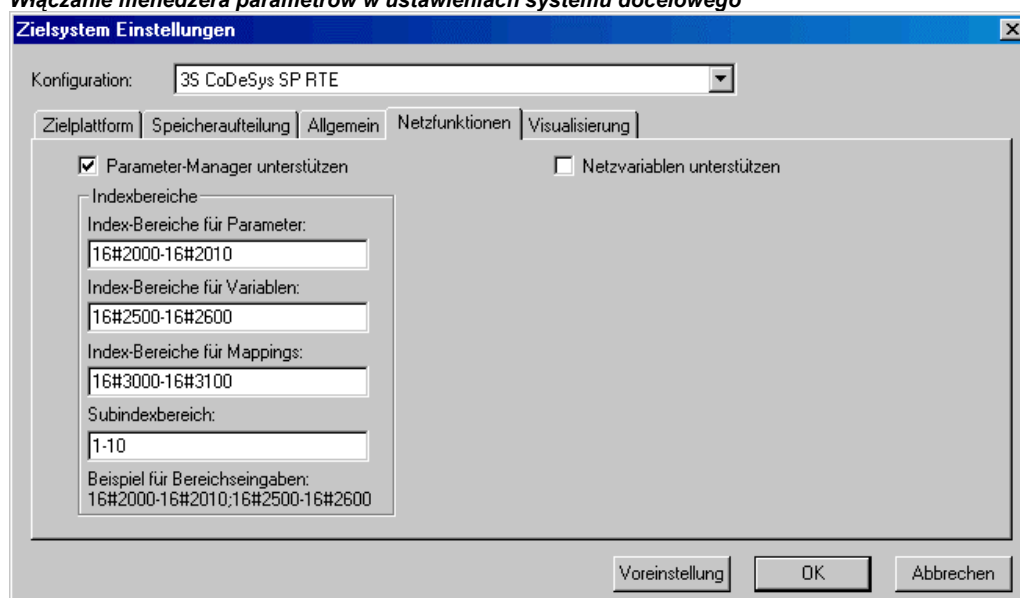
9.11.1 Włączanie menedżera parametrów

Menedżer parametrów musi być włączony w **Ustawieniach systemu docelowego**, kategoria **Funkcje sieciowe**.

W tym miejscu należy zdefiniować również zakres indeksów i subindeksów dla wpisów w listach parametrów dla typów parametrów i zmiennych oraz jeśli obsługuje to system docelowy mapowania (dla CAN Device PDO).

Zależy od systemu docelowego, w jakim stopniu ustawienia te są widoczne i edytowalne dla użytkownika.

Włączanie menedżera parametrów w ustawieniach systemu docelowego



9.11.2 Edytor menedżera parametrów, przegląd

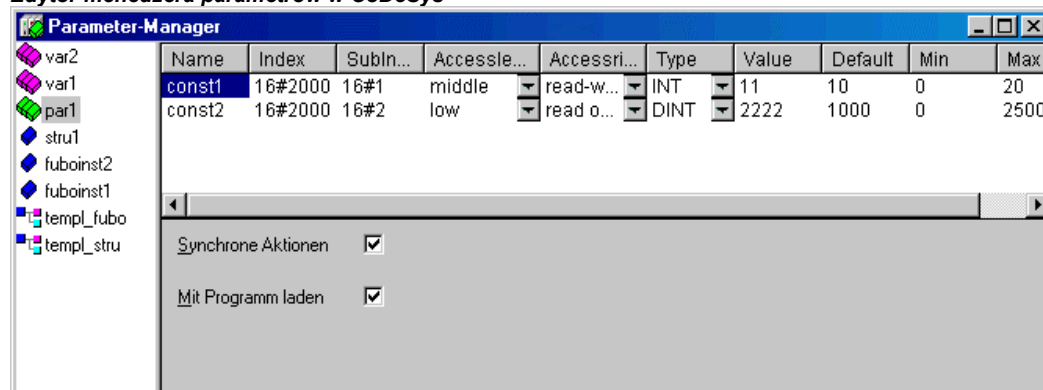
Aby otworzyć edytor, należy wybrać wpis „Menedżer parametrów” w zakładce Zasoby w CoDeSys. W tym miejscu można utworzyć, edytować i zapisać listy parametrów i w trybie online wymienić je z aktualnie podłączonym sterownikiem.

Wskazówka: Aby w projekcie CoDeSys udostępnić funkcjonalność menedżera parametrów, w ustawieniach systemu docelowego musi być włączona opcja „Obsługuj menedżera parametrów” oraz zdefiniowane odpowiednie zakresy indeksów i subindeksów.

Okno edytora jest podzielone na dwie części. Lewa część służy do nawigacji i ukazuje aktualnie wczytane w menedżera parametrów listy parametrów. Prawa część zawiera edytor listy, kolumny są zatytułowane zgodnie z nazwami atrybutów parametrów.

W **oknie nawigacji** można wstawiać, usuwać, sortować lub zmieniać nazwę list parametrów różnego typu (zmiennie, parametry (stałe), szablony, instancje, parametry systemowe).

Edytor menedżera parametrów w CoDeSys



W edytorze listy należy dla danego wpisu parametru wstawić linię zawierającą atrybuty parametru. Każdy typ listy zadaje specjalny wybór atrybutów (kolumn), które można edytować lub są one jedynie wyświetlane. Wybór ten można zdefiniować za pomocą **właściwego dla systemu docelowego pliku opisu**, w przeciwnym razie stosowane są ustawienia standardowe.

Za pomocą <F6> można przechodzić między oknem nawigacji a edytorem listy.







Wskazówka: Listy parametrów można również bezpośrednio wygenerować lub wypełnić za pomocą instrukcji pragma w obrębie deklaracji.

W trybie online można wczytać do aktualnie podłączonego systemu docelowego utworzone w edytorze listy lub można skorzystać z istniejących w systemie docelowym list parametrów, by w ten sposób przeprowadzić wymianę danych z innymi systemami w sieci (przesyłanie, zapis wartości). Ponadto w oknie edytora są wyświetlane bieżące wartości parametrów czasu przebiegu (monitoring).

W przypadku braku komunikacji z systemem docelowym listy parametrów można jedynie tworzyć i zapisywać lokalnie w projekcie.

9.11.3 Listy parametrów: Typy i atrybuty

Menedżer parametrów może zarządzać następującymi typami list parametrów:

-  **Zmienne:** Wpisy na listę parametrów tego typu reprezentują zmienne procesu projektu.
-  **Parametry:** Wpisy na listę parametrów tego typu reprezentują stałe, których wartości są niezależne od procesu.
-  **Parametry systemu:** Wpisy na listę parametrów tego typu reprezentują specjalne stałe niezależne od procesu zadane przez system docelowy. List parametrów systemowych nie można usuwać oraz zmieniać ich nazwy.
-  **Szablon:** Szablon nie zawiera wpisów parametrów, z których można bezpośrednio skorzystać w celu wymiany danych. Wpisy służą raczej jako konfiguracja podstawowa dla składników określonego bloku funkcji lub struktury. Ta konfiguracja podstawowa może być następnie stosowana w trakcie tworzenia list parametrów typu „Instancja”.
-  **Instancja:** Wpisy na listę parametrów tego typu reprezentują wpisy parametrów dla zmiennych typu bloku funkcji lub struktury, czyli dla instancji i zmiennych strukturalnych. W celu ułatwienia tworzenia list instancji stosowany jest szablon (patrz powyżej) utworzony wcześniej w menedżerze parametrów.
-  **Mapowanie:** Ten typ listy jest dostępny, jeśli jest on obsługiwany przez system docelowy. Wpisy składają się z odsyłacze do zmiennych procesu, które można „mapować” w urządzeniu CAN. W zasadzie chodzi o listę zmiennych pracująca jednak we własnym zakresie indeksów/subindeksów. Zakres ten musi być zdefiniowany w Ustawieniach systemu docelowego, kategoria Funkcje sieciowe. Urządzenie CAN stosuje w tym przypadku **wyłącznie** wpisy w listy typu „Mapowanie”, podczas gdy poza tym w konfiguracji sterownika są oferowane wszystkie wpisy z list zmiennych i instancji w oknie dialogowym „Domyślne mapowanie PDO”.

Prezentacja różnych typów list w edytorze menedżera parametrów jest definiowana za pomocą pliku opisu w formacie XML, lub jeśli jest go brak za pomocą ustawień domyślnych.

Instancje i szablony

Lista parametrów typu „Instancja” ...

... zarządza wpisami parametrów reprezentującymi określony **blok funkcji, zmienną strukturalną** lub **tablicę**. Każda lista instancji dla bloku funkcji lub zmiennej strukturalnej opiera się na szablonie, który specjalnie dla tego bloku funkcji lub struktury musi być zdefiniowany również w menedżerze parametrów.

Lista parametrów typu „Szablon” ...

... nie zawiera wpisów parametrów, z których można bezpośrednio skorzystać w celu wymiany danych. Jednak w tym miejscu są wstępnie zdefiniowane offsety indeksów i subindeksów oraz określone atrybuty dla wpisów parametrów reprezentujące określone bloki funkcji lub strukturę. Szablon ten można następnie stosować na liście parametrów typu „Instancja” (patrz wyżej), co oznacza ułatwienie w tworzeniu list parametrów dla wielu zmiennych projektu przedstawiających instancje tego samego bloku funkcji lub struktury.

Tworzenie szablonu:

W polu edycji **POU podstawowe** należy wpisać nazwę bloku funkcji lub struktury, do którego ma się odnosić dany szablon. Przy wybieraniu odpowiedniego spośród dostępnych modułów projektu można skorzystać z pomocy przy wpisywaniu (<F2>). Aby zastosować składniki wybranego modułu w edytorze listy, należy nacisnąć **Zastosuj**. Następnie należy wyedytować wpisy atrybutów i zamknąć listę, by udostępnić ją do stosowania w liście instancji.

Polecenie **Brakujące wpisy** w menu kontekstowym lub w menu „Dodatki” powoduje aktualizację wpisów zgodnie z bieżącym stanem modułu stanowiącego tego podstawę (POU podstawowe). Ewentualnie może to być konieczne lub wymagane, jeśli niektóre wpisy zostały usunięte z listy lub w przypadku dokonania zmian w podstawowym module.

Jeśli jest włączona opcja **Czynności synchroniczne**, wszystkie próby dostępu do innych POU, zdefiniowanych dla wpisów listy parametrów są przeprowadzane przez system docelowy synchronicznie wraz z przetwarzaniem bieżącego wpisu.

Aby móc utworzyć listy parametrów instancji dla tablic, nie jest wymagane tworzenie szablonu w menedżerze parametrów. Typ szablonu ARRAY jest dostępny pośrednio.

Tworzenie listy parametrów instancji:

W polu **Szablon** należy ustawić wymagany szablon. Lista wyboru zawiera wszystkie dostępne chwilowo w menedżerze parametrów szablony dla bloków funkcji i struktur oraz typ szablonu „ARRAY”.

W polu edycji **Zmienna podstawowa** należy wpisać dokładnie zmienną projektu, dla której składników należy utworzyć wpisy parametrów. Typ tej zmiennej musi odpowiadać typowi bloku funkcji, struktury lub tablicy, do których odnosi się wybrany szablon.

Następnie należy wpisać **Indeks podstawowy** i **Subindeks podstawowy** dla instancji. Wpisane w tym miejscu wartości należy traktować jako offsety dodawane automatycznie do wartości indeksów lub subindeksów zdefiniowanych dla odpowiedniego składnika w szablonie (w przypadku tablic wychodzi się zawsze z 0). Wynik dodawania zostaje również automatycznie wpisany w pole atrybutów „Indeks” lub „Subindeks”. Przykładowo po wpisaniu w tym miejscu dla składnika jako indeks podstawowy „3”, jeśli w szablonie dla tego składnika jest zdefiniowany offset indeksu 3000, składnik ten zostanie umieszczony w indeksie 3003.

Aby zastosować wstępnie skonfigurowane składniki w edytorze listy, należy nacisnąć **Zastosuj**.

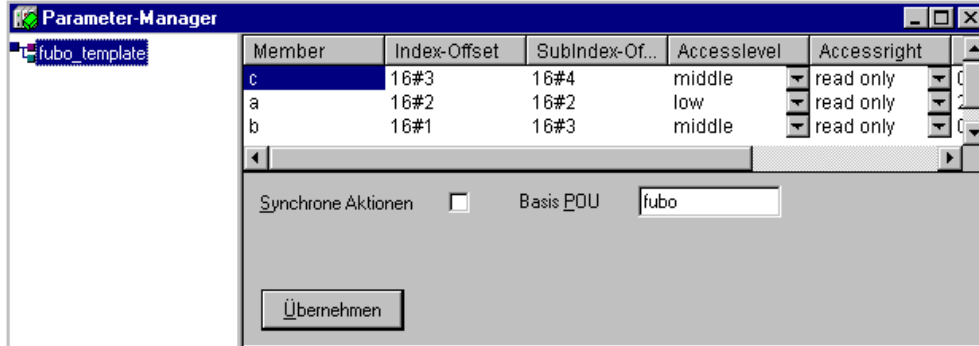
Odnosnie opcji Czynności synchroniczne patrz wyżej: Tworzenie szablonu.

Polecenie **Brakujące wpisy** w menu kontekstowym lub w menu „Dodatki” powoduje aktualizację wpisów zgodnie z bieżącym stanem zastosowanego szablonu. Może to być przydatne, jeśli wpisy na liście parametrów zostały usunięte lub szablon został zmodyfikowany.

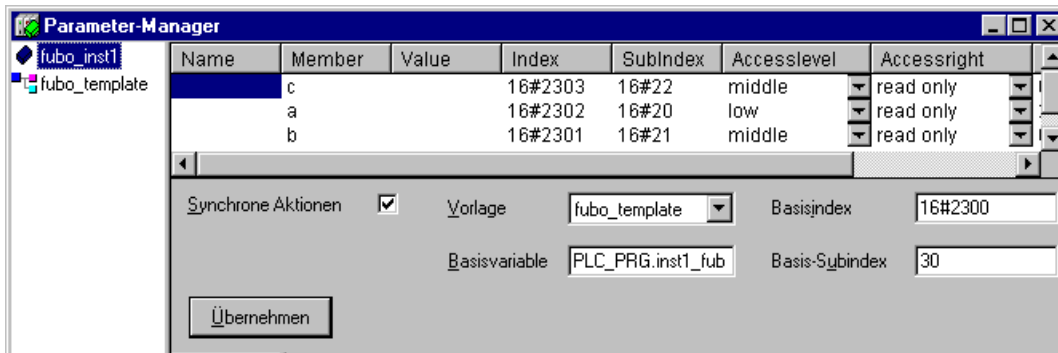
Przykład tworzenia listy parametrów instancji:

W projekcie należy utworzyć blok funkcji fubo z następującymi zmiennymi: a, b, c. W PLC_PRG należy zdefiniować następujące instancje bloku funkcji: inst1_fubo:fubo; inst2_fubo:fubo;. Następnie projekt należy skompilować.

Otworzyć menedżera parametrów w celu utworzenia list parametrów dla zmiennych inst1_fubo.a, inst1_fubo.b, inst1_fubo.c i inst2_fubo.a, inst2_fubo.b, inst2_fubo. W tym celu najpierw wstawić listę typu „Szablon” i nadać jej nazwę „fubo_template”. Zdefiniować podstawowe POU: „fubo”. Nacisnąć „Zastosuj” i zdefiniować kilka atrybutów dla składników a, b, c. M. in. wpisać offsety indeksów: dla a: 16#1, dla b: 16#2, dla c: 16#3. Również offsety subindeksów, np. a: 16#2, b: 16#3, c: 16#4.



Potem wstawić nową listę parametrów typu „Instancja”. Wybrać szablon „fubo_template”. Wpisać zmienną podstawową „inst1_fubo”. Zdefiniować indeks podstawowy: np. 16#2300 i subindeks podstawowy wynoszący 30 (należy przy tym zwrócić uwagę na zakresy indeksów zadane w ustawieniach systemu docelowego). Następnie nacisnąć „Zastosuj”, by zaktualizować we wpisach na liście indeksy automatycznie obliczone dla składników a, b, c przez dodanie podstawowego offsetu i offsetów zdefiniowanych w szablonie: Wskaźniki: 16#2301, 16#2302, 16#2303; subindeksy: 16#23, 16#33, 16#43.



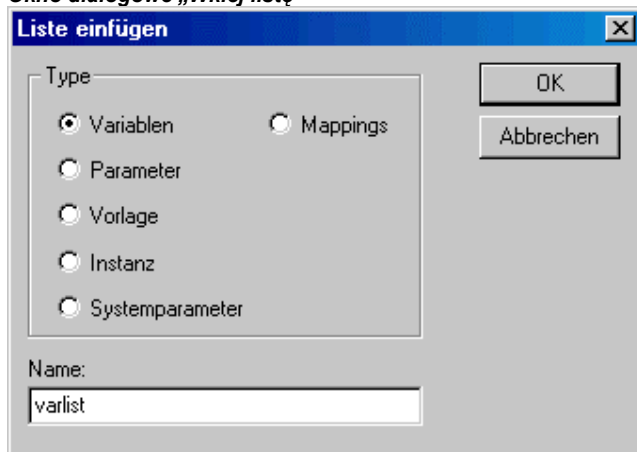
Na podstawie tych automatycznie utworzonych wpisów można w dalszym ciągu edytować listę parametrów.

9.11.4 Zarządzanie listami parametrów

Wklej listę

Skrót klawiszowy: Ins

Okno dialogowe „Wklej listę”



W celu wklejenia nowej listy parametrów w menedżera parametrów należy zastosować polecenie menu „Wstaw” lub „Wstaw nową listę” z menu kontekstowego. Polecenia te są dostępne, jeśli aktywne jest puste okno nawigacji lub już tam umieszczony wpis.

Otwarte zostaje okno dialogowe „Wklej listę”:

Należy wtedy wpisać **nazwę** nowej listy parametrów (musi być ona jednoznaczna w zakresie typu listy) i wybrać jeden z poniższych typów:

Zmienne	Wpisy zmiennych procesu.
Parametr	Wpisy danych, których wartości są niezależne od procesu.
Szablon	Szablon zestawu atrybutów dla składnika bloku funkcji lub struktury (do zastosowania w listach typu „Instancja” (patrz niżej).
Instancja	Wpisy zmiennych typu blok funkcji lub struktura (instancje) oparte na odpowiednim szablonie (patrz wyżej).
Mapowanie	Wpisy zmiennych procesu, które mogą być stosowane do mapowania PDO w urządzeniu CAN. Typ ten jest do dyspozycji, jeśli jest on obsługiwany przez system docelowy.
Parametry systemu	Wpisy parametrów, których wartości są niezależne od procesu lub wstępnie zdefiniowane przez system docelowy.

Po potwierdzeniu wpisów za pomocą **OK** i zamknięciu okna dialogowego jako wpis w oknie nawigacji pojawi się nowo utworzona lista. Typ listy jest wskazywany za pomocą umieszczonego przed wpisem przycisku.

W edytorze listy odpowiednie atrybuty są wyświetlane jako nagłówki kolumn. Wybór i układ kolumn jest zadany przez właściwy dla systemu docelowego plik opisu, jeśli zaś brak takiego, stosowane są ustawienia standardowe.

Nowa lista może być edytowana jedynie przez wstawienie dla każdego wymaganego wpisu parametru nowej linii (patrz rozdział 9.11.5, Edycja listy parametrów).

Zmień nazwę listy

Za pomocą polecenia „Zmień nazwę listy” w menu „Dodatki” lub menu kontekstowym można zmienić nazwę listy parametrów zaznaczonej w oknie nawigacji. Polecenie to otwiera ramkę edycji tworzoną za pomocą podwójnego kliknięcia na nazwie listy.

Wytnij/kopiuj/wstaw listę

Skrót klawiszowy: <Ctrl> + <X>, <Ctrl> + <C>, <Ctrl> + <V>

Rozmieszczenie list parametrów w oknie nawigacji można modyfikować za pomocą poleceń „Wytnij”, „Kopiuj” i „Wstaw” (menu „Edycja” lub menu kontekstowe).

Polecenie „Wytnij” lub „Wytnij listę” przenosi zaznaczoną listę do bufora, z którego można ją wstawić w innym miejscu za pomocą polecenia „Wstaw” lub „Wstaw listę”. Przed wstawieniem należy zaznaczyć listę powyżej której ma zostać wstawiona dana lista. Polecenie „Kopiuj” lub „Kopiuj listę” również stosuje tymczasowy bufor, przy czym jednak zostaje zachowany pierwotny wpis i za pomocą „Wstaw” lub „Wstaw listę” w drzewko nawigacji zostaje dodatkowo wstawiona kopia.

Usuń listę

Skrót klawiszowy: <Ctrl> +

Aktualnie zaznaczoną w oknie nawigacji listę można usunąć za pomocą polecenia „Usuń” (menu „Edycja”) lub „Usuń listę” (menu „Dodatki” lub menu kontekstowe).

Należy pamiętać: W trybie online za pomocą tego polecenia odpowiednia lista jest usuwana z systemu czasu przebiegu.

9.11.5 Edycja listy parametrów

Które kolumny (atrybuty) są wyświetlane / szerokość kolumny:

Aktualnie zaznaczona w oknie nawigacji lista parametrów jest prezentowana w edytorze tabel w sposób zadany za pomocą właściwego dla systemu docelowego pliku opisu lub ustawień standardowych.

Oznacza to, że wartości atrybutów każdego z parametrów zawartego na liście są opisane w **wierszu** odpowiednio do właściwego dla typu listy rozmieszczenia i wyboru kolumn.

Poszczególne kolumny można **wyświetlać i ukrywać** za pomocą menu kontekstowego, gdy kursor znajduje się na wierszu kolumny z nagłówkami kolumn.

Do zmiany szerokości kolumny służą oprócz przesuwanych linii rozdzielających kolumny również dwa polecenia umieszczone w menu kontekstowym i widoczne po skierowaniu wskaźnika myszy na nagłówek kolumny. **Standardowa szerokość kolumny** jest ustalana w taki sposób, by w oknie były widoczne wszystkie kolumny. Polecenie **Maksymalizuj kolumnę** odnosi się do aktywnej kolumny i powoduje jej poszerzenie, by w całości był widoczny każdy wpis.

Polecenia edycji wpisów parametrów:

W **menu kontekstowym** lub menu „**Wstaw**” lub „**Dodatki**” są dostępne następujące polecenia edycji:

Wstaw / usuń wiersz:

Wstaw wiersz lub Nowy wiersz	Nowy wpis (wiersz) jest wstawiany powyżej aktywnego wpisu.
Wiersz za lub Nowy wiersz za skrót klawiszowy:<Ctrl> + <Enter>	Nowy wpis (wiersz) jest wstawiany poniżej bieżącego wpisu.
Usuń wiersz	Usuwanie aktywnego wiersza. Skrót klawiszowy: <Shift> +
Wytnij wiersz, Kopiuj, Wstaw	Aktywny wiersz zostaje wycięty, wstawiony lub skopiowany. Wstawianie następuje powyżej aktywnego wiersza.

Edycja atrybutów:

Po wstawieniu nowego wiersza na wpis parametru pola atrybutów są automatycznie wypełniane właściwymi dla celu wartościami standardowymi. W celu wpisania lub edycji danej wartości należy kliknąć na odpowiednim polu. Jeśli jest ono edytowalne, pojawia się ramka edycji. W polach, w których można wpisać zmienną projektu CoDeSys, do dyspozycji jest pomoc przy wpisywaniu (<F2>).

Aby zakończyć wpis, należy nacisnąć **<Enter>**.

Za pomocą klawiszy strzałek można przejść w inne pole.

Za pomocą **** zostaje usunięta zawartość pola, w którym znajduje się kursor.

Do przełączania formatu wpisywania z „dziesiętnego” na „szesnastkowy” służy polecenie **Format dec/hex** w menu „Dodatki”.

Za pomocą **<F6>** można wchodzić i wychodzić z okna nawigacji.

Opcje:

Poniżej części edytora z tabelą można zależnie od typu listy włączyć następujące opcje:

Wczytaj z programem: Przy pobieraniu programu lista jest automatycznie wczytywana do sterownika.

Czynności synchroniczne: Wszystkie próby dostępu do innych modułów do odczytu/zapisu zdefiniowane we wpisach na liście są przez system docelowy przeprowadzane synchronicznie wraz z wywołaniem wpisu.

Sortowanie listy parametrów

Kolejność wierszy (układ wpisów) w obrębie listy parametrów można posortować odnośnie atrybutu (kolumna) w porządku rosnącym lub malejącym wartości atrybutów. Działa to zarówno w trybie offline, jak i online.

W tym celu należy kliknąć w pole z nagłówkiem kolumny zawierającej wymagany atrybut. Tabela jest wtedy sortowana na nowo i trójkąt umieszczony w polu z nazwą atrybutu wskazuje aktualne sortowanie: trójkąt skierowany wierzchołkiem w dół = kolejność rosnąca, w górę = kolejność malejąca.

9.11.6 Menedżer parametrów w trybie online

Transfer listy między edytorem a sterownikiem

Jeśli obsługuje to system docelowy, w trybie online listy parametrów utworzone w menedżerze parametrów można wczytać w sterownik (**pobieranie**), oraz tam istniejące wczytać do edytora (**przesyłanie**). Maksymalna wielkość list typów parametrów i zmiennych jest również ustalona w zależności od danego systemu docelowego.

Należy pamiętać: Przy logowaniu następuje automatyczne pobieranie wszystkich list, dla których w edytorze menedżera parametrów jest włączona opcja „**Wczytaj z programem**”.

Ponadto możliwy jest **zapis** poszczególnych wartości do sterownika.

Do **Wyświetlania bieżących wartości** każdego parametru (monitoring) służy w trybie online dodatkowa kolumna (pierwsza) w menedżerze parametrów:

608	v1
608	v2

właściwe dla danego systemu docelowego jest, czy indeks i subindeks lub RefID i Offset są stosowane do monitoringu.

W menu „**Dodatki**” do dyspozycji są następujące polecenia:

Usuń listę

Usuwanie ze sterownika zaznaczonej w oknie nawigacji listy.

Zapisz listę

Otwiera się okno dialogowe „**Kopiowanie obiektów**” gdzie z dostępnych list można wybrać te do wczytania w sterownik. Pobieranie następuje po potwierdzeniu za pomocą OK. Od systemu docelowego zależy, czy w trakcie enumeracji są przesyłane tylko wartości numeryczne lub dodatkowo również symboliczne.

Ustawienia systemu docelowego

Wczytaj listę	Wczytywanie ze sterownika no menedżera parametrów wszystkich list typu „Parametr”. „Pobieranie” list typu „Zmienne” jest przeprowadzane tylko, jeśli obsługuje to system docelowy.
Zapisz wartości	Zapis wszystkich wartości z kolumny „Wartość” w listę parametrów w sterowniku. W celu zapisania poszczególnych wartości, należy dwukrotnie kliknąć na odpowiednim polu kolumny, co powoduje otwarcie okna dialogowego „Zapisz wartości” jak w przypadku polecenia „Online” „Zapisz wartości”.
Zapisz wartości domyślne	Zapis wartości z kolumny „Domyślne” w odpowiednią listę parametrów w sterowniku.
Zastosuj wartości	Odczyt bieżących wartości parametrów ze sterownika i ich zapis w kolumnie „Wartość”.

Do przełączania formatu wyświetlania z dziesiętnego na szesnastkowy również online działa polecenie **Format dec/hex**.

Listy parametrów w projekt bootowalny

Od systemu docelowego zależy, czy zawartość menedżera parametrów przy tworzeniu projektu bootowalnego ma być w nim zastosowana.

9.11.7 Eksport / import list parametrów

„Dodatki” „Eksportuj”

Za pomocą polecenia „Eksportuj” menu „Dodatki” można wyeksportować listy parametrów do pliku XML, który z kolei można za pomocą polecenia „Dodatki” „Importuj” (przykładowo w innym projekcie) ponownie wstawić. Służy do tego standardowe okno zapisu pliku z wstępnie ustawionym rozszerzeniem *.prm. Do pliku eksportu są zawsze zapisywane wszystkie istniejące w menedżerze parametrów listy.

Zawartość menedżera parametrów można wyeksportować również za pomocą ogólnej funkcji eksportu („Projekt” Eksportuj”).

„Dodatki” „Importuj”

Za pomocą polecenia „Importuj” z menu „Dodatki” można zaimportować zawartość pliku XML opisującego listy parametrów. Plik ten można na przykład utworzyć za pomocą polecenia „Dodatki” „Eksportuj” i w standardowym przypadku ma on rozszerzenie *.prm.

Jeśli plik importu zawiera listę z nazwą, która już występuje w menedżerze parametrów, otwiera się okno dialogowe z zapytaniem o nadpisanie istniejącej listy.

9.12 Ustawienia systemu docelowego

Ustawienia systemu docelowego można znaleźć jako obiekt w zakładce Zasoby. W tym miejscu jest ustalone, na jakim sterowniku (system docelowy, cel) i z jakimi ustawieniami ma pracować projekt. Po poleceniu „Projekt” „Nowy” pojawia się wezwanie bezpośredniego wyboru „Celu”, tzn. wstępnie zdefiniowanej konfiguracji.

Lista wyboru zależy od zainstalowanego na komputerze pakietu Target Support Packages (TSP). Opisuje on właściwe dla platformy konfiguracje podstawowe i równocześnie ustala, w jakim stopniu mogą one być jeszcze dostosowywane przez użytkownika w oknach dialogowych ustawień systemu docelowego.

Wskazówka: Jeśli nie jest dostępny pakiet TSP, w wyborze systemu docelowego istnieje jedynie ustawienie „Brak”, które nie pozwala na ustawienia i automatycznie przełącza na tryb symulacji.

Pakiet Target-Support-Package

Pakiet Target Support Package (TSP) należy zainstalować przed uruchomieniem programu za pomocą programu instalacyjnego **InstallTarget**. Może go zawierać program instalacyjny CoDeSys.

W pakiecie TSP są ujęte wszystkie pliki konfiguracji i rozszerzeń potrzebne do obsługi określonego sterownika (system docelowy, cel) za pomocą aplikacji. Konfiguracja obejmuje: generator kodu, układ pamięci, zakres funkcji sterownika oraz moduły wejścia/wyjścia. Ponadto należy podłączyć biblioteki, sterowniki bramek, pliki Error i Ini do przeglądarki PLC. Centralny element pakietu TSP stanowią jeden lub więcej plików docelowych. **Plik docelowy** odsyła do dodatkowo potrzebnych do konfiguracji celu plików, może je jednak dzielić w innymi plikami docelowymi.

Plik docelowy zasadniczo ma rozszerzenie ***.trg** i jest formatu binarnego. Wpisy konfiguracji są opatrzone dodatkowymi definicjami ustalającymi, czy użytkownik może te pliki oglądać w oknie dialogowym ustawień systemu docelowego i czy można je tam edytować.

W trakcie instalacji pakietu TSP dla każdego systemu docelowego zostaje w oddzielnym katalogu utworzony odpowiedni plik docelowy, zaś jego ścieżka zostaje zapisana. Odpowiednie pliki zgodnie z dodatkowo zawartym w pakiecie TSP **plikiem Info *.tnf** są również kopiowane na komputer. Nazwa katalogu docelowego odpowiada nazwie celu. Proponowane jest umieszczenie go w katalogu z nazwą producenta.

Zainstalowane za pomocą pakietu Target Support Package pliki są wczytywane przy uruchomieniu programu CoDeSys. Podjęte w oknach dialogowych systemu programowania ustawienia systemu docelowego zostają zapisane z odpowiednim projektem.

Wskazówka: W przypadku korzystania z nowego pliku docelowego lub modyfikacji istniejącego pliku należy ponownie uruchomić CoDeSys, co spowoduje udostępnienie zaktualizowanej wersji.

Okno dialogowe Ustawień systemu docelowego

Okno dialogowe **Ustawienia systemu docelowego** otwiera się automatycznie po utworzeniu nowego projektu. Poza tym można w nie wejść z punktu menu „Ustawienia systemu docelowego” w zakładce „Zasoby”.

W **Konfiguracji** należy wybrać jedną z oferowanych konfiguracji systemu docelowego.

Jeśli pakiet Target Support Package nie jest zainstalowany, można wybrać jedynie ustawienie „**Brak**” prowadzące automatycznie do trybu symulacji. W przypadku wybrania jednej z zainstalowanych konfiguracji wstępnych, od użytkownika zależą możliwości ostatecznego dopasowania wpisów w pliku docelowym stanowiącym podstawę. W przypadku wybrania konfiguracji systemu docelowego z pakietu TSP, dla którego na komputerze brak jest licencji, pojawia się wezwanie do wybrania innego celu.

Po wybraniu konfiguracji, która w pliku docelowym jest opatrzona wpisem „Ukryj ustawienia”, pojawia się jedynie nazwa konfiguracji. Ponadto do dyspozycji jest pięć zakładek służących do ostatecznego dopasowania lub prezentacji ustawień systemu docelowego (patrz również Załącznik I: Okna dialogowe ustawień systemu docelowego):

- Platforma docelowa
- Podział pamięci
- Ogólne
- Funkcje sieciowe
- wizualizacja

Uwaga: Należy uwzględnić, iż każda modyfikacja ustawionej wstępnie konfiguracji systemu docelowego może mieć znaczny wpływ na zachowanie się systemu docelowego.

Za pomocą przycisku **Ustawienie domyślne** po zmianie ustawienia można wyzerować do wartości z konfiguracji standardowej.

9.13 Przeglądarka PLC...

W przypadku przeglądarki PLC chodzi o oparty na tekście monitor sterownika (terminal). Polecenia zapytań o określone informacje ze sterownika wpisuje się w odpowiedniej linii po czym są one jako string przesyłane do sterownika. Zwrócony string odpowiedzi jest prezentowany w oknie rezultatów przeglądarki. Funkcjonalność ta służy do celów diagnozowania i debugowania.

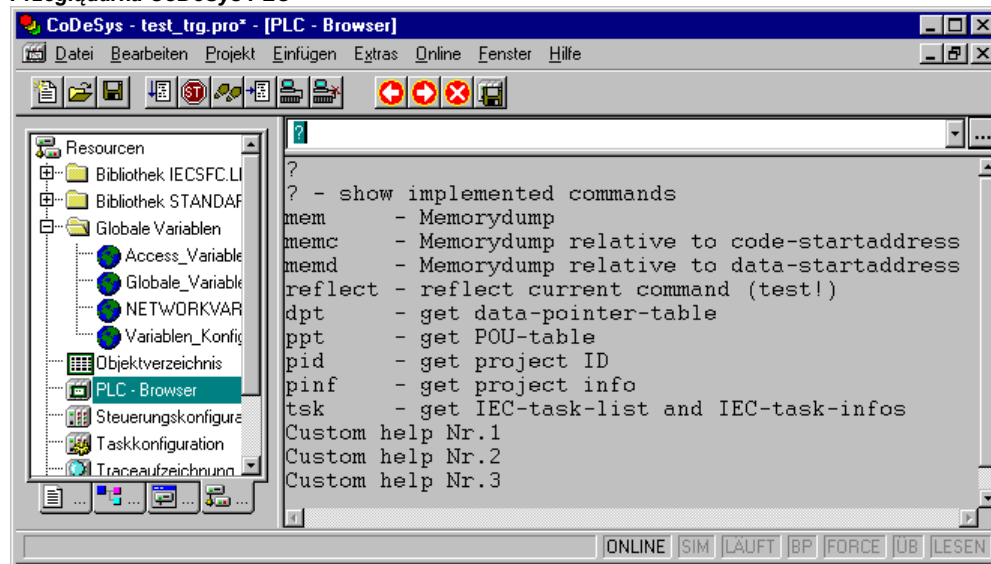
Polecenia do dyspozycji dla ustawionego systemu docelowego składają się ze standardowego zestawu CoDeSys oraz możliwego zestawu rozszerzeń pochodzącego od producenta sterownika. Są one zarządzane w pliku ini i są odpowiednio zaimplementowane w systemie czasu przebiegu.

9.13.1 Uwagi ogólne o obsłudze przeglądarki PLC

W zakładce **Zasoby** należy wybrać wpis **Przeglądarka PLC**. (Dostępność zależy od ustawień systemu docelowego.)

Przeglądarka składa się z linii wpisywania poleceń oraz okna wyników/wskazań.

Przeglądarka CoDeSys PLC




Linia wpisywania oferuje w polu wyboru listę wszystkich poleceń wpisanych od momentu uruchomienia projektu (historia wpisów). Do chwili zamknięcia projektu można je ponownie wybrać. Na listę trafiają wyłącznie polecenia różniące się od tam już istniejących.

Za pomocą <Enter> wpisane polecenie jest przesyłane do sterownika. Jeśli brak jest połączenia online, polecenie jest wyświetlane w oknie wyników w taki sposób jak zostało ono przesłane do sterownika, w przeciwnym razie pojawia się tam odpowiedź sterownika. W przypadku przesłania do sterownika nowego polecenia, zostaje usunięta zawartość okna wyników.

Polecenia można wpisywać w formie stringów, możliwe jest jednak również stosowanie makr.

9.13.2 Wpisywanie poleceń w przeglądarce PLC

Przeglądarka PLC zasadniczo daje do dyspozycji zakodowane na stałe w systemie czasu przebiegu **standardowe polecenia 3S**. Chodzi tu o funkcje bezpośredniej manipulacji pamięcią, do wydawania funkcji projektu i statusu oraz do nadzorowania czasu przebiegu. Są one opisane w **pliku ini** przeglądarki, który stanowi część pakietu Target Support Packages. Te standardowe polecenia można poszerzyć o dalsze, specjalne polecenia, np. własne funkcje diagnozowania lub inne komunikaty o statusie aplikacji sterownika. Poszerzenie listy poleceń należy przeprowadzić zarówno w interfejsie klienta w systemie czasu przebiegu jak i za pomocą dodatkowych wpisów w pliku ini przeglądarki.

W chwili otwarcia projektu z wpisów w pliku ini przeglądarki zostaje wygenerowana dostępna w przeglądarce PLC **lista poleceń**. Można ją wywołać jako pomoc przy wpisywaniu przyciskiem  w oknie dialogowym **Wstaw polecenie standardowe** (menu Wstaw) lub za pomocą <F2>. Polecenie można wpisać ręcznie lub wybrać je za pomocą dwukrotnego kliknięcia na odpowiedniej pozycji listy.

Ogólna składnia poleceń:

<KEYWORD><LEER><KEYWORD-DEPENDEND PARAMETERS>

Słowo kluczowe to **polecenie**. W odpowiedniej wskazówce w oknie pomocy przy wpisywaniu znajduje się opis, za pomocą jakich **parametrów** można je rozszerzyć.

Przesłane polecenie zostaje powtórzone w oknie wyjściowym, poniżej pojawia się odpowiedź sterownika.

Przykład: Zapytanie o Id projektu ze sterownika za pomocą polecenia „pid”

Wpis w linii poleceń:

```
pid
```

Dane wyjściowe w oknie wyników:

```
pid
Project-ID: 16#0025CFDA
```

Dla każdego polecenia standardowego można za pomocą ?<SPACJA><KEYWORD> uzyskać **tekst pomocy**. Również on jest zdefiniowany w pliku ini.

Poniższe polecenia są na stałe zintegrowane w systemie czasu przebiegu i zawierają w pliku ini odpowiednie wpisy dotyczące pomocy przy wpisywaniu, etykiетки i pomoc:

Polecenie	Opis
?	System czasu przebiegu daje listę dostępnych poleceń dla przeglądarki.
mem	Zrzut szesnastkowy obszaru pamięci. Składnia: mem <start-address> <end-address> Adresy można wpisywać dziesiętnie, szesnastkowo (prefiks 16#) lub jako makro.
memc	Zrzut szesnastkowy względnie do adresu początkowego kodu w sterowniku; jak mem, dane są dodawane do adresu początkowego zakresu kodu.
memd	Zrzut szesnastkowy względnie do podstawowego adresu danych w sterowniku; jak mem, dane są dodawane do adresu początkowego zakresu kodu.
reflect	W celu testu odwróć bieżącą linię poleceń.
dpt	Odczyt i wydawanie tabeli wskaźników do danych.
ppt	Odczyt i wydawanie tabeli wskaźników POU.
pid	Odczyt i wydawanie ID projektu.
pinf	Odczyt i wydawanie tabeli wskaźników informacji o projekcie (patrz „Projekt” „Informacja o projekcie”).
tsk	Wydawanie listy zdefiniowanych w projekcie zadań IEC z informacjami o zadaniu.
startprg	Uruchomienie programu sterującego („Online” „Start”).
stopprg	Zatrzymanie programu sterującego („Online” „Stop”).
resetprg	Zerowanie programu sterującego („Online” „Reset”).
resetprgcold	Zimny reset programu sterującego („Online” „Reset (zimny”).
resetprgorg	Reset programu sterującego do źródła („Online” „Reset (źródło”).
reload	Ponowne wczytywanie projektu bootowalnego.

Polecenie	Opis
getprgprop	Wczytanie i wyświetlenie właściwości programu (nazwa, tytuł, wersja, autor, data).
getprgstat	Wczytanie i wyświetlenie statusu programu (np. „Run”, „Stop”, ostatni błąd, flagi).
filedir	Wyświetlanie plików z katalogu sterownika.
filecopy	Kopiowanie pliku [z] [do]. Przykład: „filecopy filename.txt filename2.txt”.
filerename	Zmiana nazwy pliku w sterowniku; przykład: „filerename oldname.txt newname.txt”.
filedelete	Usuwanie pliku ze sterownika; np. „filedelete file.xml”.
saveretain	Zapis zmiennej retain. Nazwa kopii zapasowej jest podawana przez przeglądarkę.
restoreretain	Wczytywanie zmiennej retain. Nazwa kopii zapasowej, z której jest przeprowadzane odzyskiwanie, jest podawana przez przeglądarkę.
setpwd	Ustawianie hasła dla sterownika; składnia: setpwd <password> [level], np. „setpwd abcde” możliwe wartości dla <level>: „0” (domyślne) obowiązuje tylko przy logowaniu z systemu programowania „1” obowiązuje dla logowania ze wszystkich aplikacji
delpwd	Usuwanie hasła ze sterownika.

Należy pamiętać:

- Pierwsze słowo wpisanej sekwencji poleceń jest interpretowane jako **słowo kluczowe** (<KEYWORD>).
- Jeśli pierwsze słowo wpisanego polecenia **nie zostanie rozpoznane** przez sterownik, w oknie wyników pojawia się odpowiedź „Nie znaleziono słowa kluczowego”.
- Jeśli przed słowem kluczowym jest umieszczony **znak zapytania i spacja** (np. „? mem”), plik INI jest przeszukiwany pod kątem istnienia pomocy do tego słowa kluczowego. Jeśli pomoc jest dostępna, nic nie zostaje przesłane do sterownika, jedynie w oknie wyjściowym pojawia się tekst pomocy.

9.13.3 Stosowanie makr przy wpisywaniu poleceń w przeglądarce PLC

W przypadku wpisania do linii poleceń polecenia połączonego z makrem linia ta zostaje rozszerzona przed przesłaniem jej do sterownika. W oknie wyników odpowiedź pojawia się również w formie rozszerzonej.

Składnia wpisywania to: <KEYWORD> <Makro>

<KEYWORD> to polecenie,

makro to:

- %P<NAME> Jeśli NAME to nazwa POU, wyrażenie jest rozszerzane w <Indeks POU>, poza tym nie ma zmian.
- %V<NAME> Jeśli NAME to nazwa zmiennej, wyrażenie jest rozszerzane w #<INDEX>:<OFFSET>, poza tym nie ma zmian (pisownia #<INDEX>:<OFFSET> jest interpretowana przez sterownik jako adres pamięci)
- %T<NAME> Jeśli NAME to nazwa zmiennej, wyrażenie jest rozszerzane w <VARIABLENTYP> poza tym nie ma zmian.
- %S<NAME> Jeśli NAME to nazwa zmiennej, wyrażenie jest rozszerzane w <SIZEOF(VAR)> poza tym nie ma zmian.

Znak % jest ignorowany, jeśli są przed nim umieszczone symbol Escape \ (Backslash). Symbol Escape jako taki jest przesyłany, jeśli jest zapisane\\.

Przykład:

Wpis w linii poleceń: (zrzut pamięci zmiennej .testit ?)



```
mem %V.testit
```

Dane wyjściowe w oknie wyników:


```
mem #4:52
03BAAA24 00 00 00 00 CD CD CD CD ....íííí
```

9.13.4 Dalsze opcje przeglądarki PLC

W menu „Dodatki” lub w pasku narzędzi przeglądarki PLC znajdują się następujące polecenia obsługi wpisywania poleceń lub historii:

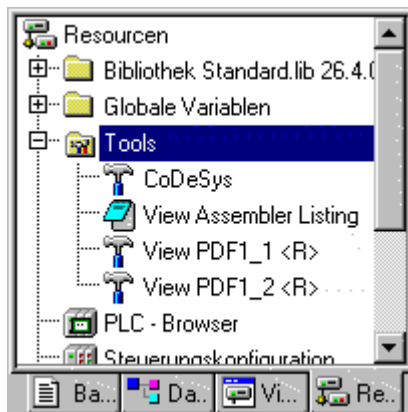
Za pomocą **Historia do przodu**  i **Historia do tyłu**  można przeglądać już istniejące wyniki zapytań. Zapis historii jest kontynuowany do chwili wyjścia z projektu.

Za pomocą **Anuluj polecenie**  można przerwać rozpoczęte zapytanie.

Za pomocą **Zapisz historię**  można zapisać uzyskane dotychczas wyniki zapytań w pliku zewnętrznym. Otwiera się wtedy okno dialogowe „Zapisz plik jako”, w którym można wpisać nazwę pliku z rozszerzeniem „.bhl” (Browser History List). Polecenie **Drukuj bieżące polecenie** powoduje otwarcie standardowego okna dialogowego wydruku. Wydrukować można bieżące zapytanie i dane wyjściowe z okna komunikatu.

9.14 Narzędzia

Obiekt „Narzędzia” jest dostępny z zakładki Zasoby, jeśli zostało to odpowiednio skonfigurowane w pliku docelowym ustawionego systemu docelowego. W obiekcie „Narzędzia” są przedstawione skróty do plików exe narzędzi zewnętrznych, które w ten sposób można wywołać z CoDeSys. W pliku docelowym jest również zdefiniowane, które i ile skrótów jest możliwe. Odpowiednio do tej definicji użytkownik ma również możliwość dodawania i usuwania skrótów w „Narzędziach”. Prezentacja w Organizery obiektów jest przykładowo taka:



W przedstawionym przykładzie są utworzone cztery skróty za pomocą narzędzi, jeden do otwierania CoDeSys, drugi do otwierania Assembler Listings w edytorze tekstowym oraz dwa za pomocą których są otwierane pliki PDF. Oznaczone za pomocą „<R>” skróty nie mogą być edytowane w CoDeSys.

Logicznym zastosowaniem byłby w tym miejscu skrót do edytora, przykładowo do notepad.exe, lub do określonego pliku pdf. Wtedy podwójne kliknięcie na odpowiednim wpisie powodowałoby otwarcie Assembler Listing w programie notepad lub pliku pdf w programie Acrobat Reader.

Dodatkowo można również określić pliki, które zostaną wczytane do sterownika po włączeniu skrótu.

9.14.1 Właściwości istniejących skrótów (Właściwości obiektu)

Kliknięcie na znaku plus przed wpisem „Narzędzia” powoduje otwarcie listy aktualnych skrótów. Jeśli został utworzony nowy projekt, wyświetlane są tylko skróty zdefiniowane wstępnie w pliku docelowym jako wpisy stałe. Jeśli pracowano już na katalogu narzędzi, ewentualnie są widoczne dodatkowe skróty, dodane przez użytkownika w CoDeSys.

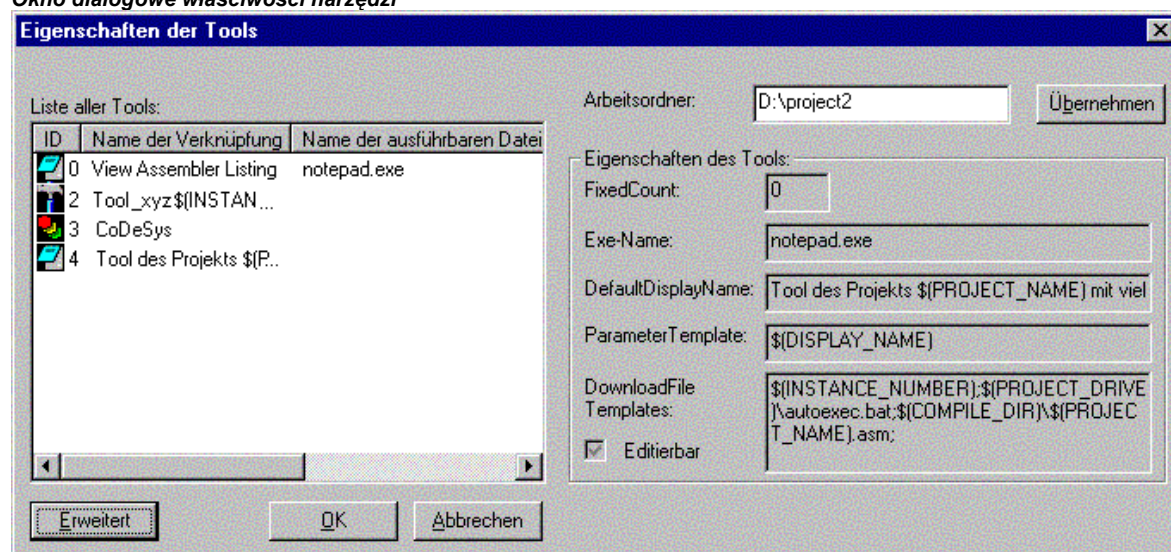
W tym miejscu można obejrzeć nadrzędne właściwości „Narzędzi” oraz poszczególnych skrótów:

1. Właściwości „Narzędzi”:

Jeśli w drzewku jest zaznaczone „Narzędzia”, w menu kontekstowym lub w menu „Projekt” „Obiekt” nad poleceniem „**Właściwości**” pojawia się okno dialogowe „Właściwości narzędzi”:

W tabeli znajdują się wszystkie stosowane dla aktualnego celu narzędzia z parametrami: Kolumna **Id** ukazuje jednoznaczny numer identyfikacyjny narzędzia, ponadto są podane **nazwa skrótu**, wyświetlana w Organizерze obiektów, oraz nazwa pliku exe (**nazwa pliku wykonywalnego**). Przycisk „**Zaawansowane**” powoduje poszerzenie okna dialogowego w prawo lub zamknięcie otwartego poszerzenia:

Okno dialogowe właściwości narzędzi



Po poszerzeniu w prawej części okna znajdują się zasadnicze właściwości połączenia zdefiniowane w pliku docelowym. Ponadto dostępne jest pole edycji, w którym

można określić **katalog roboczy** stosowany do czynności z plikiem exe. Podana ścieżka jest zapisywana za pomocą przycisku **Zastosuj**, przy czym nie powoduje to zamknięcia okna dialogowego.

Właściwości narzędzi:

- | | |
|---------------------|---|
| FixedCount | Stała ilość skrótów do tego narzędzia, automatycznie wstawiana do katalogu narzędzi. Tylko jeśli w tym miejscu jest wpisane „0”, użytkownik ma możliwość samodzielnie utworzyć dowolną ilość skrótów. Należy pamiętać: W przypadku skrótów wstawionych na stałe z pliku docelowego określona jest nie tylko ilość, modyfikować w CoDeSys nie można również właściwości (widoczne po „<R>” w Organizерze obiektów). |
| Nazwa pliku exe: | Nazwa pliku lub pełna ścieżka pliku wykonywalnego narzędzia. W tym miejscu można wpisać również ścieżkę rejestru pliku exe: „[ścieżka rejestru].<Wpis wskazujący plik exe>”. Jeśli nie pojawia się żaden wpis, oznacza to, że rozszerzenie pliku podanego w „Szablone parametrów” wywołuje automatycznie w systemie Windows plik exe odpowiedniego narzędzia.
Przykłady: „C:\programme\notepad.exe”, „345.pdf” |
| DefaultDisplayName: | Nazwa, z którą dane narzędzie w CoDeSys jest wpisane w Organizерze obiektów. Ewentualnie może być w tym miejscu stosowany szablon \$(INSTANCE NUMBER) (patrz „Szablon parametrów”). |

9 - Zasoby

Szablon parametrów: Szablon służący do określania plików otwieranych za pomocą danego narzędzia. Zawarte w tym miejscu mogą być następujące szablony połączone odpowiednim znakiem specjalnym:

\$(PROJECT_NAME) Nazwa bieżącego projektu
(nazwa pliku bez rozszerzenia „.pro”).

\$(PROJECT_PATH) Ścieżka katalogu, w którym jest umieszczony plik projektu
(bez napędu).

\$(PROJECT_DRIVE) Napęd dysku, w którym jest umieszczony bieżący projekt.

\$(COMPILE_DIR) Katalog kompilacji projektu (z napędem)

\$(TOOL_EXE_NAME) Nazwa pliku exe danego narzędzia.

\$(DISPLAY_NAME) Nazwa bieżącego skrótu, stosowana w „Narzędziach”.

\$(INSTANCE_NUMBER) Numer skrótu
(numer instancji, numer kolejny rozpoczynający się od „1”)

\$(CODESYS_EXE_DIR) Ścieżka katalogu, w którym jest umieszczone Codesys Exe
(z napędem).

Kompilacja szablonu jest widoczna w oknie dialogowym właściwości danego skrótu
(patrz poniżej).

Przykład:

„\$(PROJECT_NAME)_\$(INSTANCE_NUMBER).cfg”

⇒ Plik cfg z nazwa <Nazwa bieżącego projektu CoDeSys>_<Numer skrótu>.cfg
otwierany za pomocą narzędzia

DownloadFile
Templates:

Pliki, ścieżki do plików, które po pobraniu należy wczytać do sterownika. Jeśli opcja
Edytowalne jest włączona, można w oknie właściwości skrótu edytować listę tych
plików. Jeśli nazwa pliku jest podana bez ścieżki, plik jest szukany w katalogu, w
którym jest umieszczony plik CoDeSys exe.

„a.up;\$(PROJECT_NAME).zaw;\$(INSTANCE_NUMBER).upp”

⇒ Pliki a.up, <Bieżący projekt CoDeSys>.pro oraz <Numer skrótu>.upp są po pobraniu
wczytywane do sterownika

2. Właściwości skrótu:

Skrót należy zaznaczyć w drzewku „Narzędzia” i wybrać w menu kontekstowym lub w menu „Projekt”
„Obiekt” punkt „**Właściwości**”. Pojawia się wtedy okno dialogowe „Właściwości skrótu” zawierające
następujące punkty:

Wywołanie Wywołanie narzędzia; ścieżka pliku exe i pliku wyświetlonego w „Parametry”,
w „Szablone parametrów” (patrz wyżej)

Parametr Ścieżka pliku do wywołania przez dane narzędzie. Wynika ona z opisu celu i można ją
w tym miejscu edytować, jeśli włączona jest opcja „Edytowalne” (patrz poniżej).

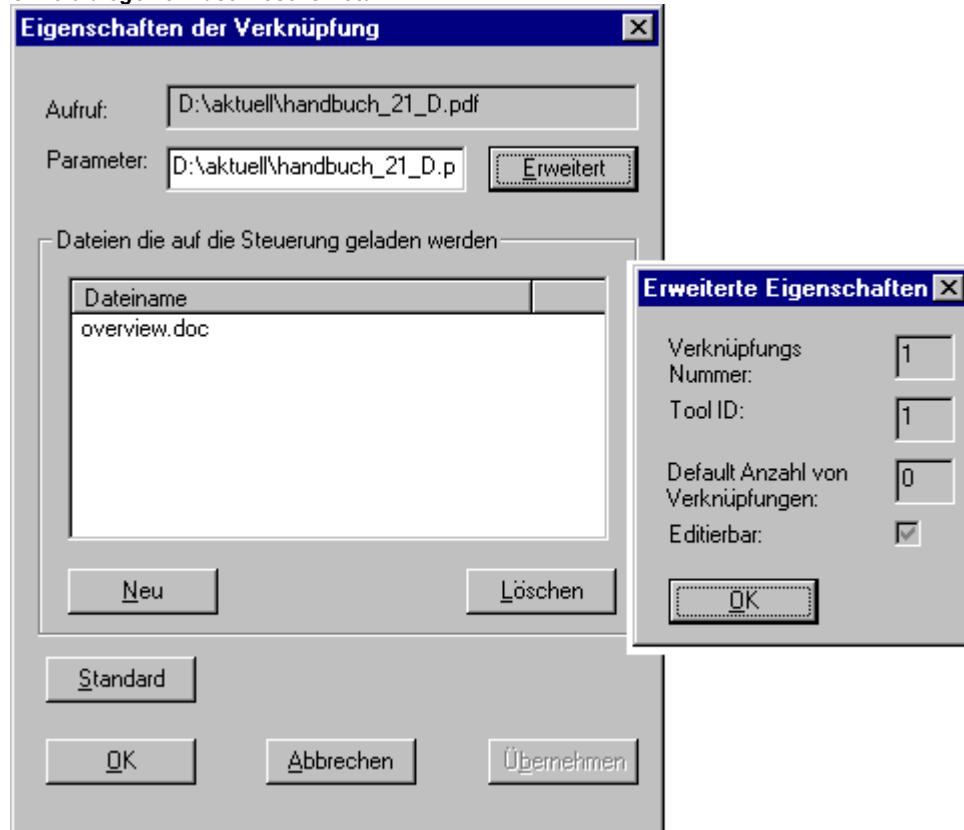
Pliki, które zostaną Automatycznie wczytane na listę są
wczytane do sterownika **nazwy plików**, wynikające z opisu celu i opisane już we właściwościach narzędzia
(patrz wyżej). Jeśli jest włączona opcja „Edytowalne” (patrz niżej „Rozszerzone okno
dialogowe”), listę można w tym miejscu edytować. W tym celu należy otworzyć za
pomocą przycisku **Nowy** okno dialogowe „**Wpisz nazwę pliku**”, w którym można wpisać
kolejny plik lub ścieżkę pliku. Jeśli pliku jest podany bez ścieżki, jest on szukany w
katalogu, w którym jest umieszczony plik CoDeSys exe. Za pomocą przycisku **Usuń**
można usunąć aktualnie zaznaczony wpis na liście.

Przycisk „**Domyślne**” zeruje wpisy w oknie dialogowym do wartości domyślnych zadanych w pliku
docelowym.

Przycisk „**Zastosuj**” powoduje zapis dokonanych ustawień bez zamykania okna dialogowego
właściwości.

Przycisk „**Zaawansowane**” powoduje rozszerzenie okna dialogowego w prawo, przy czym wygląda
ono tak:

Okno dialogowe właściwości skrótu



Numer skrótu	Numer kolejny, rozpoczynający się od 1. Dla nowych skrótów z aktualnym narzędziem stosowany jest odpowiedni wyższy numer]. Po usunięciu skrótu numery istniejących skrótów zostają jednak zachowane. Numer skrótu można za pomocą szablonu $\$(INSTANCE_NUMBER)$ stosować w innych definicjach (patrz np. wyżej, „Szablon parametrów”).
ID narzędzia	Jednoznaczny identyfikator narzędzia wynikający z jego definicji w pliku docelowym.
Domyślna liczba skrótów	Ilość instancji narzędzia odpowiadająca FixedCount zdefiniowanemu w pliku docelowym. Patrz wyżej, Właściwości narzędzi.
Edytowalne	Jeśli opcja ta jest włączona, w polu „Parametry” lub w tabeli można dokonać zmian w zakresie plików do wczytania do sterownika.

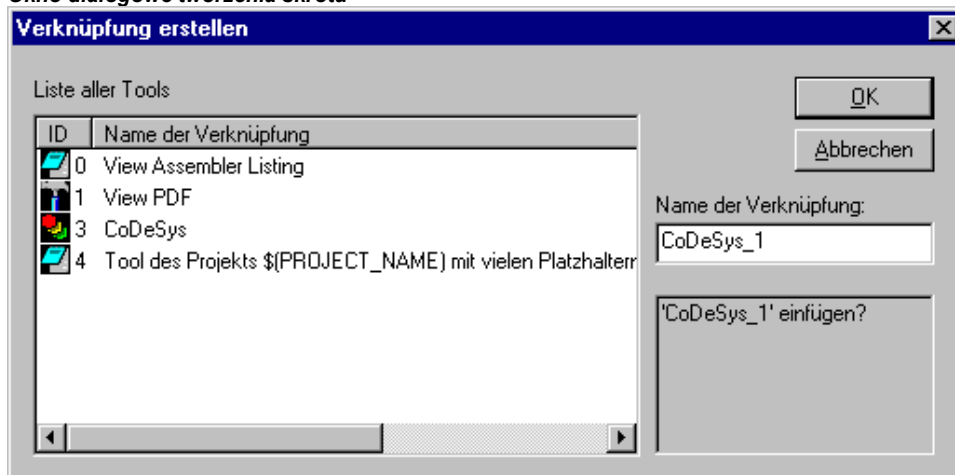
Przycisk **OK** powoduje zastosowanie dokonanych ustawień i zamknięcie okna dialogowego właściwości.

9.14.2 Zarządzanie skrótami

Tworzenie nowych skrótów

Po wybraniu węzła „Narzędzia” lub istniejącego skrótu w drzewku zasobów należy wybrać z menu kontekstowego polecenie „Wstaw obiekt” lub w menu „Projekt” „Obiekt”. Następnie pojawia się okno dialogowe „Utwórz skrót”:

Okno dialogowe tworzenia skrótów



W tabeli są zawarte wszystkie narzędzia, do których można utworzyć nowe skróty. Podane są odpowiednio do wpisów w pliku docelowym **identyfikator** narzędzia, domyślna **nazwa skrótu** oraz nazwa pliku exe danego narzędzia (**nazwa pliku wykonywalnego**).

W celu utworzenia (kolejnego) skrótu do oferowanego narzędzia należy je wybrać klikając na kolumnie „Id”. W polu **Nazwa skrótu** można zmienić domyślną nazwę nowego skrótu i potwierdzić ją za pomocą OK. Jest to możliwe po wpisaniu nazwy, która nie została jeszcze przydzielona.

OK powoduje zamknięcie okna dialogowego i nowy skrót ukazuje się w drzewku zasobów z odpowiednią nazwą i numerem wyższym od już zastosowanego dla instancji tego narzędzia.

Poniżej pola z nazwa pojawiają się wskazówki odnośnie wpisów użytkownika.

Usuwanie skrótów

Usunięcie skrótu jest wykonywane za pomocą polecenia „**Usuń**” z menu kontekstowego (prawy przycisk myszy) lub z menu „Projekt” „Obiekt”. Polecenie to jest dostępne tylko, jeśli skrót do narzędzia jest wybrany w drzewku konfiguracji i dla tego narzędzia nie jest zadana stała ilość skrótów. W związku z usunięciem danego skrótu nie zmieniają się numery pozostałych skrótów.

Wykonywanie skrótów

Skrót jest wykonywany po podwójnym kliknięciu na odpowiednim wpisie w drzewku zasobów lub wykonaniu polecenia „Edytuj obiekt” z menu „Projekt” „Obiekt” lub z menu kontekstowego (prawy przycisk myszy).

Jeśli nie powiodło się wykonanie pliku podanego pod parametrami, pojawia się odpowiedni komunikat błędu. Jeśli plik parametrów nie został znaleziony, jest wykonywany plik exe danego narzędzia i pojawia się okno dialogowe z zapytaniem, czy plik należy utworzyć.

Jeśli plik exe danego narzędzia nie został znaleziony pod podaną ścieżką lub nie została ona podana, otwiera się okno dialogowe wyboru pliku i użytkownik jest wzywany do podania ścieżki pliku exe. Ścieżka zostaje zapisana przy zamykaniu tego okna dialogowego i jest ona dostępna dla tego narzędzia również w innych projektach.

Zapis skrótów

Przy zapisie projektu zostaje zapisany również pełny stan węzła „Narzędzia” z drzewka zasobów.

Wskazówka: Jeśli projekt zostaje zapisany za pomocą „Zapisz jako” pod nową nazwą, przy stosowaniu szablonu \${PROJECT_NAME} w definicji pliku parametrów do wczytania do sterownika należy uwzględnić:

W przypadku skrótów (FixedCount=0) wstawionych przez użytkownika w starym projekcie należy w nowym projekcie ręcznie zmienić nazwy plików odpowiednio do nowej nazwy projektu. Natomiast szablon narzędzia, dla którego jest zadana stała ilość skrótów, jest automatycznie interpretowany z bieżącą nazwą projektu.

9.14.3 Najważniejsze pytania dotyczące narzędzi

Dlaczego w zakładce „Zasoby” nie pojawia się wpis „Narzędzia”?

Tylko jeśli przewiduje to definicja ustawionego systemu docelowego (plik docelowy), w zakładce „Zasoby” w CoDeSys pojawia się punkt „Narzędzia”.

Dla jakich narzędzi są już dostępne skróty, jakie można jeszcze utworzyć?

Za pomocą dwukrotnego kliknięcia na znaku plus należy otworzyć węzeł „Narzędzia” w zakładce „Zasoby”. Ukazują się wtedy narzędzia podłączone do bieżącego projektu. Po utworzeniu nowego projektu i przy braku zmian w narzędziach, chodzi w tym miejscu wyłącznie o narzędzia na stałe zadane w pliku docelowym. W innym przypadku otrzymywana jest już dostosowana do projektu lista narzędzi. Aby stwierdzić, czy listę tę można jeszcze poszerzyć, należy wybrać polecenie „Wstaw obiekt”. Otwiera się wtedy okno dialogowe ze wszystkimi narzędziami, dla których można utworzyć dodatkowe skróty.

Jakie zasadnicze właściwości mają dostępne narzędzia?

Należy zaznaczyć wpis „Narzędzia” w Organizерze obiektów i wybrać z menu kontekstowego za pomocą prawego przycisku myszy polecenie „Właściwości obiektu”. Otwierające się wtedy okno dialogowe należy poszerzyć w prawo za pomocą „Poszerz”. Z lewej strony jest wtedy widoczna lista dostępnych narzędzi zaś z prawej odpowiednie parametry. Dane narzędzie należy zaznaczyć za pomocą kliknięcia na symbolu ID po lewej stronie, by przykładowo zobaczyć w polu FixedCount do ilu skrótów jest ograniczone dane narzędzie, jaki pliki są wczytywane do sterownika przy uruchomieniu skrótu itp. Dane plików są w tym miejscu w razie potrzeby umieszczone w szablonach, których indywidualna interpretacja dla każdego skrótu jest opisana w kolejnym punkcie:

Jakie indywidualne właściwości mają już istniejące skróty?

Poniżej „Narzędzi” w Organizерze obiektów należy zaznaczyć jeden z wpisów i wybrać z menu kontekstowego za pomocą prawego przycisku myszy polecenie „Właściwości obiektu”. Po naciśnięciu przycisku „Zaawansowane” pojawiają się parametry wybranego skrótu odpowiadające częściowo wyżej opisanym zasadniczym właściwościom narzędzia. Parametry można w tym miejscu modyfikować, jeśli w pliku docelowym są ona ustawione jako „Edytowalne”.

W jaki sposób tworzy się jeden lub więcej skrótów do narzędzia?

Należy zaznaczyć wpis „Narzędzia” w Organizерze obiektów i wybrać z menu kontekstowego polecenie „Wstaw obiekt”. Pojawia się wtedy lista dostępnych narzędzi, zawierająca jednak tylko te, których maksymalna ilość zastosowań (FixedCount) nie została jeszcze osiągnięta. Należy wybrać jedno z nich i nacisnąć OK. Narzędzie to zostaje następnie wyświetlone w Organizерze obiektów. W przypadku próby kolejnego podłączenia może się to udać jedynie, jeśli podana zostanie zmieniona nazwa narzędzia, tzn. nowy wpis zostanie oznaczony jako kolejna instancja tego samego narzędzia. Przykładowo instancje narzędzia Toolxy można nazwać Toolxy_1, Toolxy_2 itd.

W jaki sposób można modyfikować parametry narzędzia?

Aby zmienić parametry instancji narzędzia, należy zaznaczyć w Organizерze obiektów dany skrót i wybrać z menu kontekstowego polecenie „Właściwości obiektu”. Od domyślnych ustawień narzędzia w pliku docelowym zależy, w jakim stopniu można edytować parametry w polach tekstowych (patrz w rozszerzonym oknie dialogowym, czy jest włączona opcja „Edytowalne”). Za pomocą przycisku „Standard” przechodzi się do ustawienia domyślnego.

W jaki sposób wykonuje się skrót do narzędzia?

W Organizерze obiektów należy dwukrotnie kliknąć na wpisie skrótu lub wybrać z menu kontekstowego polecenie „Edytuj obiekt” lub w menu „Projekt” po zaznaczeniu wpisu.

10 Zarządzanie wersjami ENI

10.1.1 Co to jest ENI

Interfejs **ENI** („Engineering Interface”) umożliwia dostęp z systemu programowania do zewnętrznej bazy danych, w której są zarządzane dane powstałe w trakcie tworzenia projektu automatyzacji. Stosowanie zewnętrznej bazy danych gwarantuje spójność danych, które mogą być następnie wspólnie używane przez wielu użytkowników, projekty i programy oraz umożliwia następujące rozszerzenia funkcjonalności CoDeSys:

- Zarządzanie wersjami projektów CoDeSys i odpowiednich zasobów (obiekty używane wspólnie): Jeśli dany obiekt został wyrejestrowany z bazy danych, zmodyfikowany i ponownie zarejestrowany, w bazie danych jest tworzona nowa wersja obiektu, jednak stare wersje zostają zachowane i w razie potrzeby można je ponownie wywołać. Dla każdego obiektu i dla całego projektu jest rejestrowana historia zmian. Wersje można sprawdzać pod kątem różnic (nie działa to przy stosowaniu jako bazy danych lokalnego systemu plików).
- Tryb z wieloma użytkownikami: Najnowszą wersję zbioru modułów, np. moduły projektu można udostępnić grupie użytkowników. Wyrejestrowane przez danego użytkownika moduły są dla innych użytkowników oznaczone jako „Edytowane” i nie można ich modyfikować. W ten sposób wielu użytkowników może równolegle pracować nad tym samym projektem, nie nadpisując przy tym sobie wzajemnie wersji obiektów.
- Dostęp za pomocą programów zewnętrznych: Oprócz systemu programowania CoDeSys ze wspólnej bazy danych mogą korzystać również inne narzędzia dysponujące interfejsem ENI. Mogą to być przykładowo wizualizacje, systemy ECAD itp., dla których są potrzebne wygenerowane w CoDeSys dane lub które same generują dane.

Aby baza danych w celu umożliwienia trybu z wieloma użytkownikami mogła się znajdować również na innym komputerze, interfejs ENI składa się z klienta i części serwera. System programowania CoDeSys jest również klientem samodzielnego **Procesu serwera ENI** oraz w razie potrzeby inną aplikacją potrzebującą dostępu do bazy danych. Odnośnie instalacji, konfiguracji i obsługi serwera ENI należy posłużyć się odpowiednią dokumentacją.

Aktualnie interfejs ENI obsługuje bazy danych „Visual SourceSafe 6.0”, „MKS Source Integrity”, „PVCS Version Manager” od wersji V7.5 oraz lokalny system plików. Obiekty mogą być tam przechowywane w różnych „Folderach” (kategorie z różnymi właściwościami dostępu), wyrejestrowywane do edycji i przez to blokowane dla innych użytkowników. Aktualny status obiektów można wywołać z bazy danych. Równocześnie obiekty mogą być przechowywane tylko lokalnie, czyli w projekcie. Plik *.pro to lokalna kopia robocza projektu zarządzanego w bazie danych.

10.1.2 Założenia pracy z bazą danych projektów ENI

Aby móc wykorzystać interfejs ENI w systemie programowania do zarządzania obiektów projektu w zewnętrznej bazie danych, należy spełnić poniższe punkty:

Wskazówka: Przy instalacji i użytkowaniu standardowego serwera ENI firmy 3S – Smart Software Solutions GmbH należy korzystać z odpowiedniej dokumentacji serwera. Należy uwzględnić możliwość stosowania w związku z serwerem ENI eksplorera, za pomocą którego można przeprowadzać zarządzanie połączonej z serwerem ENI bazy danych niezależnie od stosowanego systemu bazy danych.

- Połączenie pomiędzy CoDeSys a serwerem ENI musi być dostępne dla protokołu TCP/IP, ponieważ serwer ENI stosuje protokół HTTP.
- *Serwer ENI* musi być zainstalowany i uruchamiany lokalnie lub na innym komputerze. By były dostępne sterowniki bazy danych, wymagana jest ważna licencja. Bez licencji można stosować tylko lokalny sterownik systemu plików.
- W administracji serwera ENI – *ENI Admin* należy skonfigurować:

- użytkownik musi być zarejestrowany z prawami dostępu (User Management)
- prawa dostępu do katalogów w bazie danych muszą być poprawnie ustawione (Access Rights)
- zalecenie: Hasło administratora do dostępu do programów *ENI Admin* i *ENI Control* należy zdefiniować zaraz po instalacji.
- W programie kontrolnym serwisu *ENI Control* należy poprawnie skonfigurować skrót do wymaganej bazy danych (Database).
- Baza danych projektu, dla której istnieje sterownik obsługiwany przez serwer ENI Server musi być zainstalowana; logiczne jest wykonanie tego na komputerze, na którym pracuje również serwer. Jako alternatywę można stosować lokalny system plików, dla którego zawsze jest dostępny sterownik z serwerem ENI.
- W administracji bazy danych należy w razie potrzeby zarejestrować zarówno użytkownika (w kliencie) jak i serwer ENI jako użytkownika z prawami dostępu. Odnosi się to do każdego przypadku stosowania jako bazy danych SourceSafe, odnośnie innych sterowników baz danych i wymaganej konfiguracji użytkownika należy skorzystać z odpowiedniej dokumentacji.
- Dla bieżącego projektu CoDeSys musi być włączony interfejs ENI (wykonuje się to w oknie dialogowym CoDeSys „Projekt” „Opcje” „Baza danych projektu”).
- Dla bieżącego projektu CoDeSys należy wykonać konfigurację połączenia z bazą danych; wykonuje się to w oknach dialogowych CoDeSys w punkcie „Projekt” „Opcje” „Baza danych projektu”.
- W aktualnym projekcie użytkownik musi się logować na serwerze ENI na podstawie nazwy użytkownika i hasła; odbywa się to w oknie dialogowym logowania otwieranym za pomocą polecenia „Projekt” „Baza danych projektu” „Login” lub otwieranym automatycznie przy próbie dostępu do bazy danych.

(Patrz krótkie wprowadzenie w dokumencie „Serwer ENI – przegląd i szybkie uruchomienie”)

10.1.3 Praca w CoDeSys za pomocą bazy danych projektu

Polecenia bazy danych (przywołaj, wyrejestruj, zarejestruj, historia wersji, etykiety itp.) do zarządzania modułami projektu w bazie danych projektu ENI są do dyspozycji w bieżącym projekcie CoDeSys po włączeniu i poprawnej konfiguracji połączenia z bazą danych. Patrz „Założenia pracy z bazą danych projektów ENI”. Polecenia można znaleźć w menu „Baza danych projektu”. Można je uzyskać jako submenu z menu „Projekt” lub z menu kontekstowego pojedynczego obiektu zaznaczonego w Organizery obiektów.

Przyporządkowanie obiektu do kategorii bazy danych jest wyświetlane we właściwościach obiektu i tam można je modyfikować.

Właściwości kategorii bazy danych (parametry połączenia, prawa dostępu, zachowanie przy rejestracji i wyrejestrowaniu) można modyfikować w oknach dialogowych opcji bazy danych projektu.

10.1.4 Kategorie w zakresie bazy danych projektu

Rozróżnia się cztery grupy obiektów projektu CoDeSys:

Interfejs ENI rozróżnia trzy kategorie obiektów (kategorie obiektów ENI) zarządzane w systemie przechowywania danych: obiekty projektu, obiekty wspólne, obiekty kompilacji.

Obiekt może jednak należeć do kategorii „Lokalne” jeśli nie jest przechowywany w bazie danych, lecz w typowy sposób ma być zapisywany wyłącznie z projektem.

W związku z tym w systemie programowania moduł CoDeSys można przyporządkować do jednej z kategorii – obiekty projektu, obiekty wspólne lub lokalne; pliki kompilacji nie istnieją jeszcze w projekcie jako obiekty do przyporządkowania. Przyporządkowanie obiektu do jednej z kategorii następuje bezpośrednio po jego utworzeniu odpowiednio do ustawienia domyślnego w opcjach bazy danych projektu, lub pośrednio za pomocą polecenia „Projekt” „Baza danych projektu” „Ustal” lub

„Ustal wielokrotnie”, przyporządkowanie to można jednak w każdej chwili zmodyfikować w oknie dialogowym właściwości projektu.

Każda kategoria obiektu ENI jest oddzielnie konfigurowana w oknie dialogowym ustawień ENI (Opcje projektu, Kategoria Baza danych projektu). Oznacza to, iż uzyskuje ona własne parametry połączenia z bazą danych (Katalog, Port, Nazwa użytkownika, Prawa dostępu itp.) i zachowania w chwili Przywoływania, Wyrejestrowania i Rejestrowania. Ustawienia te odnoszą się następnie do wszystkich obiektów należących do danej kategorii. Również dane dostępu (nazwa użytkownika, hasło) przy łączeniu się w bazą danych należy wpisać oddzielnie odpowiednio dla każdej kategorii. Służy do tego okno dialogowe logowania.

W odpowiedniej bazie danych dla każdej z kategorii obiektów ENI należy utworzyć odrębny katalog na obiekty, jest jednak również możliwe przechowywanie obiektów wszystkich kategorii w tym samym katalogu, ponieważ przyporządkowanie do kategorii stanowi właściwość obiektu a nie katalogu.

Trzy możliwe kategorie obiektów ENI to:

- | | |
|-------------------|--|
| Projekt: | Dla obiektów przedstawiających właściwe dla danego projektu informacje o źródle np. wspólnie używane moduły w obrębie jednego projektu ważne w trybie z wieloma użytkownikami. Za pomocą polecenia „Przywołaj wszystko” w CoDeSys wszystkie obiekty tej kategorii są automatycznie pobierane z katalogu projektu bazy danych do projektu lokalnego, w tym również te, które jeszcze nie zostały tam utworzone. |
| Obiekty wspólne: | Dla ogólnie obowiązujących, niezależnych od projektu obiektów takich jak biblioteki modułów zwykle używane przez wielu użytkowników w zakresie różnych projektów. Uwaga: Za pomocą polecenia „Przywołaj wszystko” w CoDeSys tylko już tam utworzone obiekty tej kategorii są kopiowane z katalogu projektu do projektu lokalnego. |
| Pliki kompilacji: | Dla automatycznie utworzonych przez CoDeSys właściwych dla projektu informacji o kompilacji (np. pliki symboli), potrzebnych również dla innych narzędzi. Przykładowo wizualizacja wymaga zmiennych systemu programowania łącznie z adresami przydzielanymi jednak dopiero przy kompilacji. |

Do wyboru moduły projektu można również wykluczyć z zarządzania w bazie danych projektu i zapisać je tylko lokalnie w typowy sposób wraz z projektem.

11 Komunikacja DDE

CoDeSys dysponuje interfejsem DDE (dynamic data exchange). W ten sposób CoDeSys udostępnia treść zmiennych sterownika i adresy IEC innym aplikacjom dysponującym również interfejsem DDE.

W przypadku stosowania zorientowanego na symbole serwera GatewayDDE można wczytać wartości zmiennych ze sterownika niezależnie od systemu programowania CoDeSys i również przedstawić je za pomocą aplikacji dysponującym interfejsem DDE.

Uwaga: Za pomocą serwera DDE nie można wczytać adresów bezpośrednich. W tym przypadku należy utworzyć w CoDeSys zmienne z odpowiednimi przydziałami adresu (AT).

Uwaga: Interfejs DDE został przetestowany w programach Word 97 oraz Excel 97 pod Windows NT 4.0. Za błędy w zakresie komunikacji DDE wywołane wskutek zastosowania innych wersji lub dodatkowo zainstalowanych programów firma 3S – Smart Software Solutions nie ponosi żadnej odpowiedzialności.

11.1 Interfejs DDE systemu programowania CoDeSys...

Uruchamianie interfejsu DDE

Interfejs DDE jest uruchamiany po zalogowaniu do sterownika (lub symulacji).

Ogólne wywoływanie danych

Zapytanie DDE jest podzielone na 3 części:

1. Nazwa programu (tutaj: CoDeSys),
2. nazwy plików oraz
3. nazwy zmiennych do wczytania.

Nazwa programu: CoDeSys

Nazwa pliku: pełna ścieżka projektu, z którego należy wczytywać (C:\beispiel\bsp.pro).

Nazwa zmiennej: Nazwa zmiennej w brzmieniu, jak w menedżerze podglądu zmiennych i receptur.

Jakie zmienne można wczytać?

Wczytać można wszystkie adresy i zmienne. Wpisywanie zmiennych lub adresów zgodnie z procedurą w menedżerze podglądu zmiennych i receptur.

Przykłady:

%IX1.4.1	(* wczytuje wejście 1.4.1 *)
PLC_PRG.TEST	(* wczytuje zmienną TEST z modułu PLC_PRG *)
.GlobVar1	(* wczytuje zmienną globalną GlobVar1 *)

Łączenie zmiennych za pomocą WORD

Aby w Microsoft WORD otrzymać za pomocą interfejsu DDE aktualną wartość zmiennej TEST z modułu PLC_PRG należy wypełnić w programie WORD dowolne pole („Wstaw” „Pole”), na przykład z datą. Po kliknięciu na polu prawym przyciskiem myszy i wybraniu polecenia „Pokaż funkcję pola” można zmienić funkcję pola na wymagany tekst, w poniższym przykładzie jest to:

```
{ DDEAUTO CODESYS "C:\\CODESYS\\PROJECT\\IFMBS.PRO" "PLC_PRG.TEST" }
```

Następnie należy ponownie kliknąć na tym polu prawym przyciskiem myszy i wpisać polecenie „Aktualizuj pole”. W tekście pojawi się wtedy wymagana treść zmiennej.

Łączenie zmiennych za pomocą programu EXCEL

Aby w programie EXCEL przyporządkować do komórki zmienną, należy wpisać w EXCELA:

```
=CODESYS | 'C:\CODESYS\PROJECT\IFMBSP.PRO' ! 'PLC_PRG.TEST'
```

W „Edycja” „Połączenia” powstaje wtedy następujące połączenie:

Typ: CODESYS

Plik źródłowy: C:\CODESYS\PROJECT\IFMBSP.PRO

Element: PLC_PRG.TEST

Aktywowanie zmiennej za pomocą Intouch

Dla projektu należy ustalić nazwę dostępu DDE <AccessName> z nazwą aplikacji CODESYS i nazwą tematu DDE C:\CODESYS\PROJECT\IFMBSP.PRO

Następnie można ustalić zmienną typu DDE wraz z nazwą dostępu <AccessName>. Jako nazwę jednostki należy ponownie wpisać nazwę zmiennej (np. PLC_PRG.TEST).

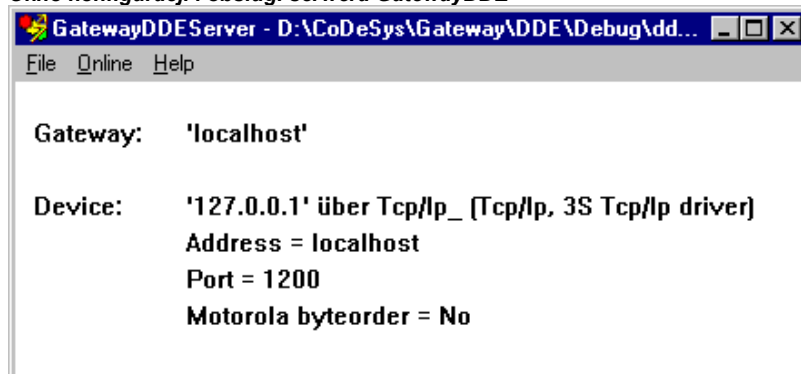
11.2 Komunikacja DDE za pomocą serwera GatewayDDE...

Obsługa serwera GatewayDDE

Serwer GatewayDDE może być stosowany do komunikacji z innymi klientami lub sterowania utworzonych w projekcie CoDeSys symboli (patrz „Projekt” „Opcje” „Konfiguracja symboli”). Może on obsługiwać interfejsy DDE aplikacji takich jak np. Excel. W ten sposób można na przykład ilustrować w innych aplikacjach wartości zmiennych ze sterownika.

Po uruchomieniu serwera GatewayDDE otwiera się okno, w którym można przeprowadzić konfigurację parametrów uruchomienia i połączenia. W tym celu można wywołać już istniejący plik konfiguracji lub na nowo ustawić parametry.

Okno konfiguracji i obsługi serwera GatewayDDE



Za pomocą polecenia „Plik” „Otwórz” można wywołać już zapisaną w pliku konfigurację. W tym celu otwiera się standardowe okno dialogowe wyboru pliku. Domyślnie szuka się plików o rozszerzeniu „.cfg”. Po wybraniu pliku konfiguracyjnego pojawiają się parametry konfiguracji bramki oraz odpowiednie urządzenie docelowe (device).

Jeśli jest aktywna opcja „Plik” „Autowczytywanie”, serwer GatewayDDE otwiera się automatycznie wraz z konfiguracją aktywną przed ostatnim zakończeniem pracy.

W przypadku uruchomienia serwera bez konfiguracji i ustawienia Autowczytywanie w oknie pojawiają się tylko hasła 'Bramka:' oraz 'Jednostka:'. Należy wtedy ponownie przeprowadzić konfigurację.

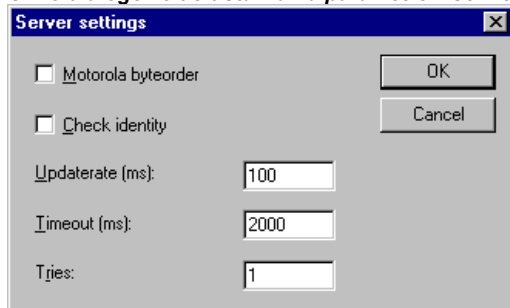
Polecenie „Plik” „Ustawienia” otwiera się okno dialogowe „Ustawienia serwera”, w którym można ustawić następujące parametry:

Motorola byteorder Stosowane jest Motorola Byteorder.

Check identity Testowanie, czy podany w pliku symboli identyfikator projektu jest zgodny z identyfikatorem w sterowniku.

Updaterate [ms]	Interwał wczytywania ze sterownika wszystkich wartości symboli.
Timeout [ms]	Limit czasu komunikacji dla stosowanego sterownika.
Próby	Ilość prób przesłania bloku danych przez sterownik komunikacji (nie jest obsługiwane przez wszystkie sterowniki).

Okno dialogowe do ustawiania parametrów serwera GatewayDDE.



Do ustawienia połączenia z aktualną bramką służy okno dialogowe „**Parametry komunikacji**” otwierane za pomocą polecenia „Online” „Parametry”. Pojawia się wtedy to samo okno dialogowe co w CoDeSys w „Online” „Parametry komunikacji”. Ustawienia muszą być zgodne z odpowiednimi ustawieniami dokonanymi w projekcie CoDeSys.

Aktualną konfigurację serwera GatewayDDE można zapisać do pliku za pomocą polecenia „**Plik**” „**Zapisz**”. W tym celu otwiera się standardowe okno dialogowe zapisu pliku, przy czym domyślnie jest przewidziane rozszerzenie „.cfg”.

Jeśli ma być włączona bramka, należy się zalogować za pomocą polecenia „**Online**” „**Login**” (zostaje wtedy podświetlona ikona bramki na pasku stanu.) W ten sposób zostaje utworzone ustawione połączenie i można wywołać odpowiednie symbole. Najpierw jednak należy je utworzyć w projekcie CoDeSys.

Do wylogowania służy polecenie „**Online**” **Logout**”.

Wywoływanie danych udostępnionych przez serwer GatewayDDE

Zapytanie DDE jest podzielone na 3 części:

1. nazwa programu,
2. nazwy plików oraz
3. nazwy zmiennych do wczytania.

Nazwa programu: GatewayDDEServer

Nazwa pliku: nazwa projektu, z którego należy wczytywać (np. BSP.PRO).

Nazwa zmiennej: Nazwa zmiennej w brzmieniu jak w menedżerze podglądu zmiennych i receptur (np. PLC_PRG.TEST).

Jakie zmienne można wczytać?

Można wczytać wszystkie zmienne. Wpisy są dokonywane w menedżerze podglądu zmiennych i receptur. Nie można wczytywać adresów bezpośrednich.

Przykłady:

```
PLC_PRG.TEST      (* wczytuje zmienną TEST z modułu PLC_PRG *)
.GlobVar1         (* wczytuje zmienną globalną GlobVar1 *)
```

Łączenie zmiennych za pomocą programu EXCEL przez serwer GatewayDDE

Wskazówka: Przed włączeniem zapytania w programie EXCEL należy uruchomić serwer GatewayDDE na podstawie odpowiednich ustawień konfiguracji.

Odpowiednio do wyżej opisanego postępowania w linii, która ma prezentować odpowiednią wartość zmiennej, należy wpisać następujące wyrażenie:

```
=GATEWAYDDESERVER|<nazwa pliku>!<nazwa zmiennej>
```

Przykład:

```
=GATEWAYDDESERVER|'bsp.pro'!'PLC_PRG.TEST'
```

Po aktualizacji pola pojawia się treść zmiennej.

W „Edycja” „Połączenia” powstaje wtedy następujące połączenie:

```
Typ: GATEWAYDDESERVER  
Plik źródłowy: BSP.PRO  
Element: PLC_PRG.TEST
```

Łączenie zmiennych za pomocą programu WORD przez serwer GatewayDDE

Wskazówka: Przed włączeniem zapytania w programie WORD należy uruchomić serwer GatewayDDE na podstawie odpowiednich ustawień konfiguracji.

Aby w Microsoft WORD otrzymać za pomocą interfejsu DDE aktualną wartość zmiennej TEST z modułu PLC_PRG należy wypełnić w programie WORD dowolne pole („Wstaw” „Pole”), na przykład z datą. Tekst funkcji pola można edytować po kliknięciu na nim prawym przyciskiem myszy i wybraniu polecenia „Pokaż funkcję pola”. Jeśli ma zostać wyświetlona wartość zmiennej TEST z modułu PLC_PRG projektu BSP.pro należy wpisać:

```
{ DDEAUTO GATEWAYDDESERVER "BSP.PRO" "PLC_PRG.TEST" }
```

Opcje linii poleceń dla serwera GatewayDDE

Po uruchomieniu serwera GatewayDDE za pomocą linii poleceń można wpisać następujące opcje:

/n	nie otwiera się wtedy automatycznie okno dialogowe z informacjami		
/s	wyświetlone zostaje okno dialogowe	/s=h	brak
		/s=i	zminimalizowane (ikona)
		/s=m	zmaksymalizowane
		/s=n	zwykła wielkość
/c	plik konfiguracyjny, który powinien zostać wczytany automatycznie	/c=<config-file>	
/o	praca online na podstawie wybranej konfiguracji (autoload lub podana za pomocą "/c=")		

Przykład:

Wpis w linii poleceń:

```
GATEWAYDDE /s=i /c="D:\DDE\conf_1.cfg"
```

Zostaje uruchomiony serwer GatewayDDE, przy czym okno dialogowe pojawia się jako ikona i zostaje automatycznie wczytana konfiguracja serwera zapisana w pliku conf_1.cfg.

12 Zarządzanie licencjami w CoDeSys

12.1 3S Licensing Manager

Zarządzanie licencjami 3S za pomocą *3S Licensing Manager* daje użytkownikowi przyjazne narzędzie do zarządzania licencjami modułów 3S na komputerze lokalnym. W CoDeSys można utworzyć projekty i zapisać je jako biblioteki wymagające licencji. W trakcie instalacji modułu 3S wymagającego licencji jest również instalowany Licensing Manager.

12.1.1 Tworzenie biblioteki wymagającej licencji

Jeśli projekt CoDeSys ma być zapisany jako biblioteka zabezpieczona licencją, należy wpisać informacje o licencji w **oknie dialogowym edycji informacji licencyjnych** otwieranego za pomocą polecenia „Plik” „Zapisz jako...”. W ten sposób zostają zapisane informacje o projekcie i można je potem stosując daną bibliotekę przejrzeć we właściwościach obiektów w menedżerze bibliotek.

Okno dialogowe: Edytuj informacje licencyjne

Informacje ogólne:

Nazwa: Nazwa modułu biblioteki, za pomocą której jest ona zarządzana w menedżerze licencji. Wpis obowiązkowy.

ID partnera: Identyfikator producenta zależny od właściwego dla niego programu do zarządzania licencjami.

Tryb demo: Opcję tę należy włączyć, jeśli dany moduł ma być używany w trybie demo, tzn. bez ID licencji, przy czym należy wpisać liczbę dni, po upływie których ta „licencja demo” ma wygasnąć. Wpisana liczba dni jest automatycznie zaokrąglana do kolejnej dziesiątki (10, 20, 30...). Jeśli moduł ma być stosowany bez ograniczeń, należy wpisać „bez ograniczeń” (dostępne na liście wyboru).

Cele: Należy wpisać ID celów systemu lub systemów docelowych, dla których ma działać licencja. Większą liczbę identyfikatorów należy wpisać jako listę ze średnikiem jako separator lub jako zakres. Przykład: "12;15-19;21"

Kontakt:

Licencja przez telefon: / Licencja e-mailem: W tym miejscu należy wpisać numer telefonu lub adres e-mail, na który użytkownik ma uzyskać identyfikator licencji dla danego modułu. Wpisy obowiązkowe.

Informacje dodatkowe:

W prawym oknie dla każdego z niżej wymienionych punktów można wpisać dowolny tekst: opis, producent, źródło zakupu, informacje o cenie

Należy pamiętać:

- Uzasadnione jest zabezpieczenie hasłem zapisanej biblioteki zawierającej informacje o licencji. Jeśli plik ma być zapisany bez hasła, pojawia się odpowiedni komunikat.
- Dla bibliotek 3S nie jest wymagane przechowywanie informacji o licencji w oddzielnym pliku opisu modułu, ponieważ są one zapisane wewnętrznie i przy zastosowaniu danej biblioteki zostają zapisane automatycznie w komputerze. Dla innych np. również zewnętrznie (nie za pomocą 3S) utworzonych modułów musi jednak istnieć plik opisu w formacie kompatybilnym z XML, który można wczytać do menedżera licencji 3S.

Patrz: dokumentacja 3S Licensing Manager.

13 ZAŁĄCZNIK

Załącznik A Operatory IEC i dodatkowe funkcje poszerzające normę

Przegląd

CoDeSys obsługuje wszystkie operatory IEC. Są one w przeciwieństwie do funkcji standardowych (biblioteka standardowa) znane pośrednio w całym projekcie. Oprócz operatorów IEC **CoDeSys** obsługuje ponadto następujące nie wymagane przez normę operatory: INDEXOF i SIZEOF (patrz operatory arytmetyczne), ADR i BITADR (patrz operatory adresu).

Uwaga: W operacjach z typem danych zmiennoprzecinkowych wynik obliczeń zależy od zastosowanego sprzętu w systemie docelowym.

W implementacji modułu operatory są stosowane jak funkcje.

- > Operatory arytmetyczne
- > Operatory bitstring
- > Operatory bit-shift
- > Operatory wyboru
- > Operatory porównawcze
- > Operatory adresu
- > Operator wywołania
- > Konwersje typów
- > Operatory numeryczne

13.1 Operatory arytmetyczne...

ADD

Dodawanie zmiennych typu BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL i LREAL.

Można dodawać również dwie zmienne TIME, sumą jest wtedy również czas (np. działa $t\#45s + t\#50s = t\#1m35s$)

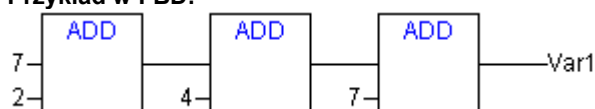
Przykład w IL:

```
LD 7
ADD 2,4,7
ST Var1
```

Przykład w ST:

```
var1 := 7+2+4+7;
```

Przykład w FBD:



MUL

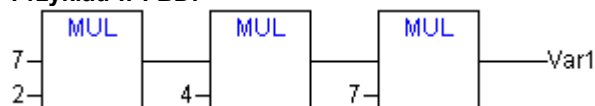
Mnożenie zmiennych typu BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL i LREAL.

Przykład w IL:

```
LD 7
MUL 2,4,7
ST Var1
```

Przykład w ST:

```
var1 := 7*2*4*7;
```

Przykład w FBD:**SUB**

Odejmowanie zmiennej BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL i LREAL od innej zmiennej jednego z podanych typów.

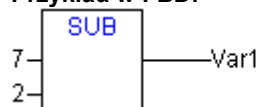
Zmienną TIME można odjąć od innej zmiennej TIME, wynik jest wtedy typu TIME. Nie są zdefiniowane ujemne wartości TIME.

Przykład w IL:

```
LD 7
SUB 2
ST Var1
```

Przykład w ST:

```
var1 := 7-2;
```

Przykład w FBD:**DIV**

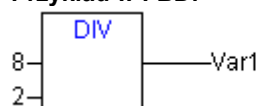
Dzielenie zmiennej BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL i LREAL przez inną zmienną jednego z podanych typów.

Przykład w IL:

```
LD 8
DIV 2
ST Var1 (* wynik to 4 *)
```

Przykład w ST:

```
var1 := 8/2;
```

Przykład w FBD:

Wskazówka: Po zdefiniowaniu w projekcie funkcji o nazwie **CheckDivByte**, **CheckDivWord**, **CheckDivDWord** i **CheckDivReal**, można z ich pomocą w przypadku stosowania operatora DIV sprawdzić wartość dzielnika, by przykładowo zapobiec dzieleniu przez 0. Nazwa funkcji jest ustalona i może to być tylko ona.

Uwaga: Zachowanie w przypadku dzielenia przez 0 zależy od zastosowanego systemu operacyjnego i docelowego.

Przykład implementacji funkcji CheckDivReal:

```
FUNCTION CheckDivReal : REAL
VAR_INPUT
  divisor:REAL;
END_VAR

IF divisor = 0 THEN
  CheckDivReal:=1;
ELSE
  CheckDivReal:=divisor;
END_IF;
```

Wynik funkcji CheckDivReal wird jest stosowany przez operator DIV jako dzielnik. W poniżej przedstawionym przykładzie programu zapobiega się dzieleniu przez 0, dzielnik (d) zostaje ustawiony z 0 na 1. Wynik dzielenia erg to odpowiednio do tego 799.

```
PROGRAM PLC_PRG
VAR
  erg:REAL;
  v1:REAL:=799;
  d:REAL;
END_VAR

erg:= v1/d;
```

Uwaga: Zawarte w CheckLib funkcje CheckDiv stanowią rozwiązania przykładowe. Przed zastosowaniem biblioteki należy sprawdzić, czy funkcje pracują zgodnie z założeniami lub zaimplementować bezpośrednio do projektu jako moduł odpowiednią funkcję CheckDiv.

MOD

Dzielenie modulo zmiennej BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT przez inną zmienną jednego z podanych typów. Wynik tej funkcji to całkowita reszta z dzielenia.

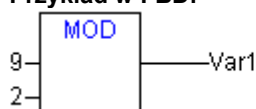
Przykład w IL:

```
LD 9
MOD 2
ST Var1 (* wynik to 1 *)
```

Przykład w ST:

```
var1 := 9 MOD 2;
```

Przykład w FBD:

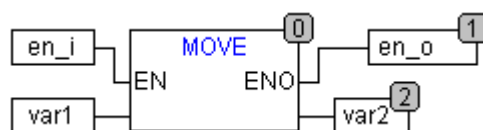


MOVE

Przyporządkowanie zmiennej do innej zmiennej odpowiedniego typu. Przez to, że MOVE jest dostępne jako moduł w edytorach CFC i LD, można zastosować funkcjonalność EN/ENO również w stosunku do przyporządkowania zmiennej. W edytorze FBD nie jest to niestety możliwe.

Przykład w CFC w połączeniu z funkcją EN/ENO:

Tylko jeśli en_i jest TRUE, wartość zmiennej var1 zostaje przydzielona zmiennej var2.



Przykład w IL:

```
LD ivar1
MOVE
ST ivar2 (* wynik: var2 uzyskuje wartość var1 *)
```

(odpowiada: LD ivar1
ST ivar2)

Przykład w ST:

```
ivar2 := MOVE(ivar1);
```

(odpowiada:

```
ivar2 := ivar1;)
```

INDEXOF

Funkcja ta nie jest zalecana przez normę IEC61131-3.

INDEXOF jako wynik daje wewnętrzny indeks modułu.

Przykład w ST:

```
var1 := INDEXOF(moduł2);
```

SIZEOF

Funkcja ta nie jest zalecana przez normę IEC61131-3.

SIZEOF jako wynik daje ilość bajtów potrzebnych dla podanej zmiennej.

Przykład w IL:

```
arr1:ARRAY[0..4] OF INT;
Var1 INT
LD arr1
SIZEOF
ST Var1 (* wynik to 10 *)
```

Przykład w ST:

```
var1 := SIZEOF(arr1);
```

13.2 Operatory bitstring...

AND

Bitowe AND w argumentach bitowych. Argumenty powinny być typu BOOL, BYTE, WORD lub DWORD.

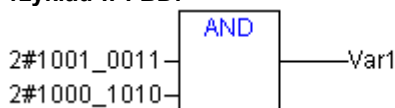
Przykład w IL:

```
Var1 BYTE
LD 2#1001_0011
AND 2#1000_1010
ST Var1 (* wynik to 2#1000_0010 *)
```

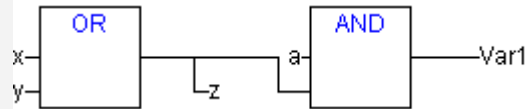
Przykład w ST:

```
var1 := 2#1001_0011 AND 2#1000_1010
```

Przykład w FBD:



Wskazówka: W przypadku wpisania przy stosowaniu generatorów kodu 68xxx lub C w FBD jednego z przedstawionych w tym miejscu przebiegów programu, należy uwzględnić:



Przyporządkowanie wartości drugiej zmiennej wejściowej w module operatora AND do zmiennej „z” nie jest przeprowadzany w związku ze zoptymalizowaną procedurą przetwarzania w FBD, jeśli zmienna wejściowa ma wartość FALSE.

OR

Bitowe OR w argumentach bitowych. Argumenty powinny być typu BOOL, BYTE, WORD lub DWORD.

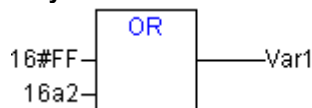
Przykład w IL:

```
Var1 BYTE
LD 2#1001_0011
OR 2#1000_1010
ST Var1 (* wynik to 2#1001_1011 *)
```

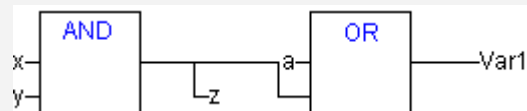
Przykład w ST:

```
Var1 := 2#1001_0011 OR 2#1000_1010
```

Przykład w FBD:



Wskazówka: W przypadku wpisania przy stosowaniu generatorów kodu 68xxx lub C w FBD jednego z przedstawionych w tym miejscu przebiegów programu, należy uwzględnić:



Przyporządkowanie wartości drugiej zmiennej wejściowej w module operatora OR do zmiennej „z” nie jest przeprowadzany w związku ze zoptymalizowaną procedurą przetwarzania w FBD, jeśli zmienna wejściowa ma wartość TRUE!

XOR

Bitowe XOR w argumentach bitowych. Argumenty powinny być typu BOOL, BYTE, WORD lub DWORD.

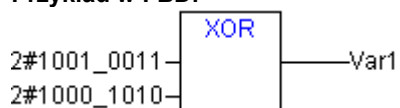
Przykład w IL:

```
Var1 BYTE
LD 2#1001_0011
XOR 2#1000_1010
ST Var1 (* wynik to 2#0001_1001 *)
```

Przykład w ST:

```
Var1 := 2#1001_0011 XOR 2#1000_1010
```

Przykład w FBD:



Wskazówka: Należy uwzględnić zachowanie modułu XOR w rozszerzonej formie, czyli w przypadku ponad 2 wejść: Wejścia są sprawdzane parami i odpowiednie wyniki są ze sobą porównywane (odpowiada normie, lecz niekoniecznie oczekiwaniom).

NOT

Bitowe NOT argumentu bitowego. Argument powinien być typu BOOL, BYTE, WORD lub DWORD.

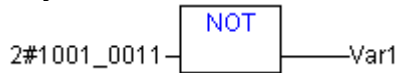
Przykład w IL:

```
Var1 BYTE
LD 2#1001_0011
NOT
ST Var1 (* wynik to 2#0110_1100 *)
```

Przykład w ST:

```
Var1 := NOT 2#1001_0011
```

Przykład w FBD:



13.3 Operatory bit-shift...

Wskazówka: Dla wersji włącznie do Service Pack 5 wersji 2.2: Generator kodu dla systemów docelowych Infineon C16x przeprowadza operacje obliczeniowe bit-shift za pomocą Modulo 16.

SHL

Bitowe lewe shift argumentu: erg:= SHL (in, n)

in zostaje przesunięte o n bitów w lewo i wypełnione od prawej zerami.

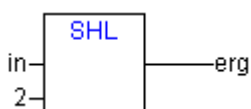
Wskazówka: Należy zwrócić uwagę na to, że ilość bitów uwzględnianych w operacji obliczeniowej jest zadana przez typ danych zmiennej wejściowej. Jeśli chodzi tutaj o stałą, uwzględniany jest możliwie najmniejszy typ danych. Typ danych zmiennej wyjściowej nie ma wpływu na operację obliczeniową.

Poniższy przykład w systemie szesnastkowym ilustruje przy takiej samej wartości zmiennych wejściowych erg_byte i erg_word różnice w operacji, zależnie od tego, czy in jest typu BYTE czy WORD.

Przykład w ST:

```
PROGRAM shl_st
VAR
  in_byte : BYTE:=16#45;
  in_word : WORD:=16#45;
  erg_byte : BYTE;
  erg_word : WORD;
  n: BYTE :=2;
END_VAR
erg_byte:=SHL(in_byte,n); (* wynik to 16#14 *)
erg_word:=SHL(in_word,n); (* wynik to 16#0114 *)
```

Przykład w FBD:



Przykład w IL:

```
LD 16#45
SHL 2
ST erg_byte
```


SHR

Bitowe prawe shift argumentu: $erg := SHR(in, n)$

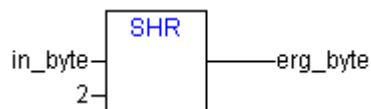
in jest przesuwane o n bitów w prawo. W przypadku stosowania typu danych bez znaku liczby (BYTE, WORD, DWORD), następuje uzupełnienie zerami od lewej. W typach danych ze znakiem liczby, jak np. INT, przeprowadzane jest natomiast przesunięcie arytmetyczne, tzn. wypełnianie wartością najwyższego bitu.

Wskazówka: Należy zwrócić uwagę na to, że ilość bitów uwzględnianych w operacji obliczeniowej jest zadana przez typ danych zmiennej wejściowej. Jeśli chodzi tutaj o stałą, uwzględniany jest możliwie najmniejszy typ danych. Typ danych zmiennej wyjściowej nie ma wpływu na operację obliczeniową.

Poniższy przykład w systemie szesnastkowym ilustruje wynik operacji, przy czym jako zmienne wejściowe służą raz `erg_byte` typu BYTE, raz `erg_word` typu WORD.

Przykład w ST:

```
PROGRAM shr_st
VAR
    in_byte : BYTE:=16#45;
    in_word : WORD:=16#45;
    erg_byte : BYTE;
    erg_word : WORD;
    n: BYTE :=2;
END_VAR
erg_byte:=SHR(in_byte,n); (* wynik to 11 *)
erg_word:=SHR(in_word,n); (* wynik to 0011 *)
```

Przykład w FBD:**Przykład w IL:**

```
LD 16#45
SHR 2
ST erg_byte
```

ROL

Bitowa rotacja argumentu w lewo: $erg := ROL(in, n)$

erg, in oraz n powinny być typu BYTE, WORD lub DWORD. in jest przesuwane n razy o jedno miejsce w lewo, przy czym bit najbardziej wysunięty w prawo zostaje wsunięty ponownie.

Wskazówka: Należy zwrócić uwagę na to, że ilość bitów uwzględnianych w operacji obliczeniowej jest zadana przez typ danych zmiennej wejściowej. Jeśli chodzi tutaj o stałą, uwzględniany jest możliwie najmniejszy typ danych. Typ danych zmiennej wyjściowej nie ma wpływu na operację obliczeniową.

Poniższy przykład w systemie szesnastkowym ilustruje przy takiej samej wartości zmiennych wejściowych `erg_byte` i `erg_word` różnice w operacji, zależnie od tego, czy in jest typu BYTE czy WORD.

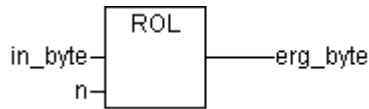
Przykład w ST:

```
PROGRAM rol_st
VAR
    in_byte : BYTE:=16#45;
    in_word : WORD:=16#45;
    erg_byte : BYTE;
    erg_word : WORD;
    n: BYTE :=2;
END_VAR
```

Operatory wyboru...

```
END_VAR
erg_byte:=ROL(in_byte,n); (* wynik to 16#15 *)
erg_word:=ROL(in_word,n); (* wynik to 16#0114 *)
```

Przykład w FBD:



Przykład w IL:

```
LD 16#45
ROL 2
ST erg_byte
```

ROR

Bitowa rotacja argumentu w prawo: `erg:= ROR (IN, N)`

`erg`, `in` oraz `n` powinny być typu `BYTE`, `WORD` lub `DWORD`. `in` jest przesuwane `n` razy o jedno miejsce w prawo, przy czym bit najbardziej wysunięty w prawo zostaje wsunięty ponownie w lewo.

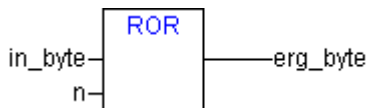
Wskazówka: Należy zwrócić uwagę na to, że ilość bitów uwzględnianych w operacji obliczeniowej jest zadana przez typ danych zmiennej wejściowej. Jeśli chodzi tutaj o stałą, uwzględniany jest możliwie najmniejszy typ danych. Typ danych zmiennej wyjściowej nie ma wpływu na operację obliczeniową.

Poniższy przykład w systemie szesnastkowym ilustruje przy takiej samej wartości zmiennych wejściowych `erg_byte` i `erg_word` różnice w operacji, zależnie od tego, czy `in` jest typu `BYTE` czy `WORD`.

Przykład w ST:

```
PROGRAM ror_st
VAR
    in_byte : BYTE:=16#45;
    in_word : WORD:=16#45;
    erg_byte : BYTE;
    erg_word : WORD;
    n: BYTE :=2;
END_VAR
erg_byte:=ROR(in_byte,n); (* wynik to 16#51 *)
erg_word:=ROR(in_word,n); (* wynik to 16#4011 *)
```

Przykład w FBD:



Przykład w IL:

```
LD 16#45
ROR 2
ST erg_byte
```

13.4 Operatory wyboru...

Wszystkie operacje wyboru można przeprowadzić na zmiennych. Dla większej przejrzystości ograniczamy się w poniższych przykładach na stałe jako operatory.

SEL

Wybór binarny.

OUT := SEL(G, IN0, IN1) oznacza:

OUT := IN0 if G=FALSE;

OUT := IN1 if G=TRUE.

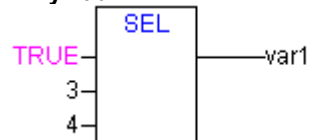
IN0, IN1 oraz OUT mogą być każdego typu, G musi być typu BOOL. Wynik wyboru to IN0, jeśli G jest FALSE, lub IN1, jeśli G jest TRUE.

Przykład w IL:

```
LD TRUE
SEL 3,4 (* IN0 = 3, IN1 =4 *)
ST Var1 (* wynik to 4 *)
LD FALSE
SEL 3,4
ST Var1 (* wynik to 3 *)
```

Przykład w ST:

```
Var1:=SEL(TRUE,3,4); (* wynik dla Var1 to 4 *)
```

Przykład w FBD:

Wskazówka: W celu optymalizacji czasu przebiegu przetwarzanie jest wykonywane w następujący sposób: Wyrażenie przed IN0 jest obliczane tylko, jeśli G jest FALSE. Wyrażenie przed IN1 jest obliczane tylko, jeśli G jest TRUE.

Natomiast w symulacji są obliczane wszystkie odgałęzienia.

MAX

Funkcja maksimum. Daje większą z dwóch wartości.

OUT := MAX(IN0, IN1)

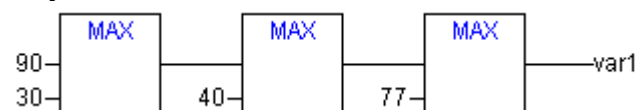
IN0, IN1 oraz OUT mogą być dowolnego typu.

Przykład w IL:

```
LD 90
MAX 30
MAY 40
MAX 77
ST Var1 (* wynik to 90 *)
```

Przykład w ST:

```
Var1:=MAX(30,40); (* wynik to 40 *)
Var1:=MAX(40,MAX(90,30)); (* wynik to 90 *)
```

Przykład w FBD:**MIN**

Funkcja minimum. Daje mniejszą z dwóch wartości.

OUT := MIN(IN0, IN1)

IN0, IN1 oraz OUT mogą być dowolnego typu.

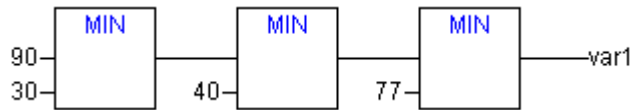
Przykład w IL:

```
LD 90
MIN 30
MIN 40
MIN 77
ST Var1 (* wynik to 30 *)
```

Przykład w ST:

```
Var1:=MIN(90,30); (* wynik to 30 *);
Var1:=MIN(MIN(90,30),40); (* wynik to 30 *);
```

Przykład w FBD:



LIMIT

Ograniczenie

```
OUT := LIMIT(Min, IN, Max) oznacza:
OUT := MIN (MAX (IN, Min), Max)
```

Max to górna, Min zaś to dolna granica wyniku. W przypadku przekroczenia przez wartość IN górnej granicy Max, LIMIT daje Max. Jeśli zaś IN przekroczy w dół Min, wynik to Min.

IN oraz OUT mogą być dowolnego typu.

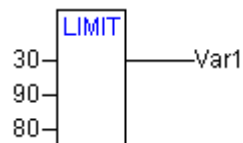
Przykład w IL:

```
LD 90
LIMIT 30,80
ST Var1 (* wynik to 80 *)
```

Przykład w ST:

```
Var1:=LIMIT(30,90,80); (* wynik to 80 *);
```

Przykład w FBD:



MUX

Multiplekser

```
OUT := MUX(K, IN0, ..., INn) oznacza:
OUT := INK.
```

IN0, ..., INn oraz OUT mogą być dowolnego typu. K musi być typu BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT lub UDINT. MUX wybiera spośród wartości wartości k-tą. Pierwsza wartość odpowiada K=0. Jeśli K jest większe od ilości spodziewanych dalszych wejść (n), przekazana zostaje ostatnia wartość (INn).

Przykład w IL:

```
LD 0
MUX 30,40,50,60,70,80
ST Var1 (* wynik to 30 *)
```

Przykład w ST:

```
Var1:=MUX(0,30,40,50,60,70,80); (* wynik to 30 *);
```

Wskazówka: W celu optymalizacji czasu przebiegu obliczane jest tylko wyrażenie poprzedzające INK. Natomiast w symulacji są obliczane wszystkie odgałęzienia.

13.5 Operatory porównawcze...

GT

Większy od.

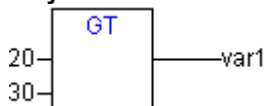
Operator boolean z wynikiem TRUE, jeśli pierwszy argument jest większy od drugiego. Argumenty mogą być typu BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME i STRING.

Przykład w IL:

```
LD 20
GT 30
ST Var1 (* wynik to FALSE *)
```

Przykład w ST:

```
VAR1 := 20 > 30 > 40 > 50 > 60 > 70;
```

Przykład w FBD:

LT

Mniejszy od.

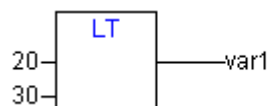
Operator boolean z wynikiem TRUE, jeśli pierwszy argument jest mniejszy od drugiego. Argumenty mogą być typu BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME i STRING.

Przykład w IL:

```
LD 20
LT 30
ST Var1 (* wynik to TRUE *)
```

Przykład w ST:

```
VAR1 := 20 < 30;
```

Przykład w FBD:

LE

Mniejszy lub równy.

Operator boolean z wynikiem TRUE, jeśli pierwszy argument jest mniejszy lub równy drugiemu. Argumenty mogą być typu BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME i STRING.

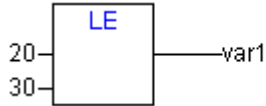
Przykład w IL:

```
LD 20
LE 30
ST Var1 (* wynik to TRUE *)
```

Przykład w ST:

```
VAR1 := 20 <= 30;
```

Przykład w FBD:



GE

Większy lub równy

Operator boolean z wynikiem TRUE, jeśli pierwszy argument jest większy lub równy drugiemu. Argumenty mogą być typu BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME i STRING.

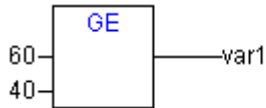
Przykład w IL:

```
LD 60
GE 40
ST Var1 (* wynik to TRUE *)
```

Przykład w ST:

```
VAR1 := 60 >= 40;
```

Przykład w FBD:



EQ

Równość

Jeśli argumenty są równe, operator boolean z wynikiem TRUE. Argumenty mogą być typu BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME i STRING.

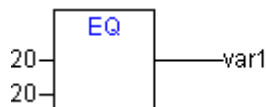
Przykład w IL:

```
LD 40
EQ 40
ST Var1 (* wynik to TRUE *)
```

Przykład w ST:

```
VAR1 := 40 = 40;
```

Przykład w FBD:



NE**Nierówność**

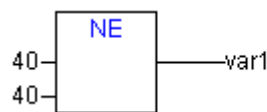
Jeśli argumenty są nierówne, operator boolean z wynikiem TRUE. Argumenty mogą być typu BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME i STRING.

Przykład w IL:

```
LD 40
NE 40
ST Var1 (* wynik to FALSE *)
```

Przykład w ST:

```
VAR1 := 40 <> 40;
```

Przykład w FBD:**13.6 Operatory adresu...**

Uwaga: W przypadku zastosowania zmiany online, zawartość adresów może ulec przemieszczeniu. Przy stosowaniu wskaźników należy zwracać uwagę na adresy.

ADR

Funkcja adresu nie jest zalecana przez normę IEC61131-3.

ADR daje adres argumentu w DWORD. Adres ten może być przesłany do funkcji producenta i tam potraktowany jako wskaźnik lub przydzielony w obrębie projektu do wskaźnika.

Przykład w ST:

```
dwVar:=ADR(bVAR);
```

Przykład w IL:

```
LD bVar
ADR
ST dwVar
man_fun1
```

ADRINST

Funkcja adresu nie jest zalecana przez normę IEC61131-3.

ADRINST daje w obrębie instancji bloku funkcji adres tej instancji w DWORD. Adres ten może być przesłany do funkcji i tam potraktowany jako wskaźnik lub przydzielony w obrębie projektu do wskaźnika.

Przykłady w ST (w obrębie instancji bloku funkcji):

```
dvar:=ADRINST(); (* adres instancji zapisany w zmiennej dvar *)
fun(a:=ADRINST()); (* przekazanie adresu instancji do parametru wejściowego a w
funkcji fun *)
```

Przykłady w IL:

```
ADRINST
ST dvar

ADRINST
fun
```

BITADR

Funkcja adresu nie jest zalecana przez normę IEC61131-3.

BITADR daje offset bitowy w obrębie segmentu w DWORD. Offset zależy od tego, czy w ustawieniach systemu docelowego jest aktywna opcja adresowania bajtów.

```
VAR
  var1 AT %IX2.3:BOOL;
  bitoffset: DWORD;
END_VAR
```

Przykład w ST:

```
bitoffset:=BITADR(var1); (* wynik adresowania bajtów=TRUE: 19, w adresowaniu
bajtów=FALSE: 35 *)
```

Przykład w IL:

```
LD Var1
BITADR
ST Var2
```

Operator zawartości

Zmiana referencji wskaźnika następuje za pomocą operatora zawartości „^” po identyfikatorze wskaźnika.

Przykład w ST:

```
pt:POINTER TO INT;
var_int1:INT;
var_int2:INT;
pt := ADR(var_int1);
var_int2:=pt^;
```

13.7 Operator wywołania...

CAL

wywołanie bloku funkcji

Za pomocą CAL w IL zostaje wywołana instancja bloku funkcji. Po nazwie instancji bloku funkcji (w nawiasie okrągłym) jest umieszczone przypisanie zmiennych wprowadzania bloku funkcji.

Przykład:

Wywołanie instancji *Inst* bloku funkcji z obciążeniem zmiennych wejściowych *Par1*, *Par2* na 0 lub TRUE.

```
CAL INST(PAR1 := 0, PAR2 := TRUE)
```

13.8 Konwersje typów...

Nie jest dozwolona pośrednia konwersja z „większego” typu na „mniejszy” (przykładowo z INT w BYTE lub z DINT w WORD). Jeśli chce się tak zrobić, należy zastosować specjalne konwersje typów. Zasadniczo można dokonać konwersji każdego typu podstawowego w inny typ podstawowy.

Składnia:

```
<elem.Typ1>_TO_<elem.Typ2>
```

W przypadku konwersji ...TO_STRING należy zwrócić uwagę, że string jest generowany jako „wyrównany do lewej”. Jeśli jest zdefiniowany jako zbyt krótki, następuje odpowiednie odcięcie od prawej strony.

Konwersje BOOL_TO

Konwersja typu BOOL na każdy inny typ:

W typach liczb wynik wynosi 1 przy argumencie TRUE, zaś 0 przy argumencie FALSE.

W typie STRING wynik „TRUE” lub „FALSE”.

Przykłady w IL:

```
LD TRUE (* wynik to 1 *)
BOOL_TO_INT
ST i

LD TRUE (* wynik to 'TRUE' *)
BOOL_TO_STRING
ST str

LD TRUE (* wynik to T#1ms *)
BOOL_TO_TIME
ST t

LD TRUE (* wynik to TOD#00:00:00.001 *)
BOOL_TO_TOD
ST

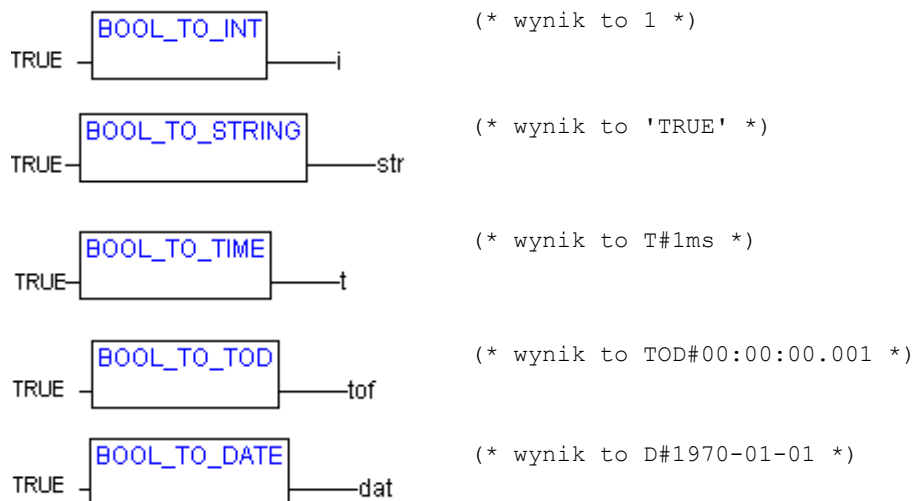
LD FALSE (* wynik to D#1970-01-01 *)
BOOL_TO_DATE
ST dat

LD TRUE (* wynik to DT#1970-01-01-00:00:01 *)
BOOL_TO_DT
ST dandT
```

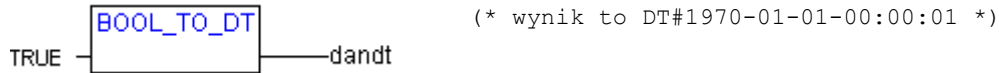
Przykłady w ST:

```
i:=BOOL_TO_INT(TRUE); (* wynik to 1 *)
str:=BOOL_TO_STRING(TRUE); (* wynik to 'TRUE' *)
t:=BOOL_TO_TIME(TRUE); (* wynik to T#1ms *)
tof:=BOOL_TO_TOD(TRUE); (* wynik to TOD#00:00:00.001 *)
dat:=BOOL_TO_DATE(FALSE); (* wynik to D#1970-01-01 *)
dandt:=BOOL_TO_DT(TRUE); (* wynik to DT#1970-01-01-00:00:01 *)
```

Przykłady w FBP:



Konwersje typów...



Konwersje BOOL_TO

Konwersja z innego typu na typ BOOL:

Wynik jest TRUE, jeśli argument nie jest równy 0. Wynik jest FALSE, jeśli argument jest równy 0.

W typie STRING wynik jest TRUE, jeśli argument jest „TRUE”, w innym przypadku wynik jest FALSE.

Przykłady w IL:

```
LD 213 (* wynik to TRUE *)  
BYTE_TO_BOOL  
ST b
```

```
LD 0 (* wynik to FALSE *)  
INT_TO_BOOL  
ST b
```

```
LD T#5ms (* wynik to TRUE *)  
TIME_TO_BOOL  
ST b
```

```
LD 'TRUE' (* wynik to TRUE *)  
STRING_TO_BOOL  
ST b
```

Przykłady w ST:

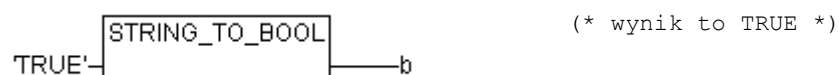
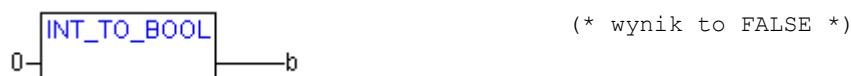
```
b := BYTE_TO_BOOL(2#11010101); (* wynik to TRUE *)
```

```
b := INT_TO_BOOL(0); (* wynik to FALSE *)
```

```
b := TIME_TO_BOOL(T#5ms); (* wynik to TRUE *)
```

```
b := STRING_TO_BOOL('TRUE'); (* wynik to TRUE *)
```

Przykłady w FBP:



Konwersje w zakresie typów liczb całkowitych

Konwersje z typu liczby całkowitej na inny typ liczby:

Może się zdarzyć, że w trakcie konwersji z większych na mniejsze typy zostaną utracone informacje. Jeśli liczba do konwersji przekracza granicę zakresu, nie są uwzględniane pierwsze jej bajty.

Przykład w ST:

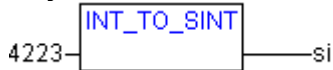
```
si := INT_TO_SINT(4223); (* wynik to 127 *)
```

Jeśli liczba całkowita 4223 (16#107f w systemie szesnastkowym) jest zapisywana w zmiennej SINT, zawiera ona liczbę 127 (16#7f w systemie szesnastkowym).

Przykład w IL:

```
LD 2
INT_TO_REAL
MUL
```

Przykład w FBD:



Konwersje REAL_TO- / LREAL_TO

Konwersja typu REAL lub LREAL na inny typ:

Zaokrąglenie w górę lub w dół do wartości całkowitej i zamiana na odpowiedni typ. Wyjątek stanowią typy STRING, BOOL, REAL i LREAL.

Może się zdarzyć, że w trakcie konwersji z większych na mniejsze typy zostaną utracone informacje.

W konwersji na typ STRING, łączna liczba miejsc po przecinku jest ograniczona do 16. Jeśli liczba (L)REAL zawiera więcej miejsc, szesnaste miejsce zostaje zaokrąglone i w taki sposób przedstawione w stringu. Jeśli STRING dla liczby jest zdefiniowany za krótko, następuje odpowiednie odcięcie od prawej strony.

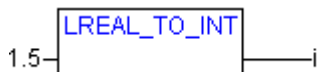
Przykład w ST:

```
i := REAL_TO_INT(1.5); (* wynik to 2 *)
j := REAL_TO_INT(1.4); (* wynik to 1 *)
i := REAL_TO_INT(-1.5); (* wynik to -2 *)
j := REAL_TO_INT(-1.4); (* wynik to -1 *)
```

Przykład w IL:

```
LD 2.7
REAL_TO_INT
GE %MW8
```

Przykład w FBD:



Konwersje TIME_TO- / TIME_OF_DAY

Konwersja typu TIME lub TIME_OF_DAY na inny typ:

Wewnętrznie czas jest zapisywany w milisekundach w DWORD (w TIME_OF_DAY od godziny 00:00). Wartość ta zostaje poddana konwersji.

Może się zdarzyć, że w trakcie konwersji z większych na mniejsze typy zostaną utracone informacje.

W typie STRING wynik to stała czasu.

Przykłady w IL:

```
LD T#12ms (* wynik to 'T#12ms' *)
TIME_TO_STRING
ST st̄

LD T#300000ms (* wynik to 300000 *)
TIME_TO_DWORD
ST d̄w
```

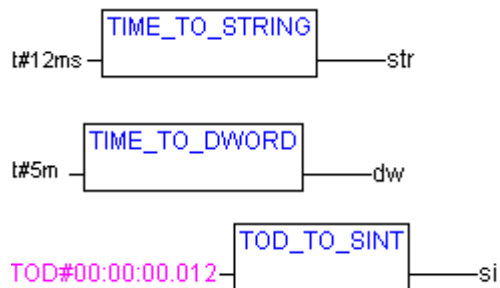
Konwersje typów...

```
LD TOD#00:00:00.012 (* wynik to 12 *)
TOD_TO_SINT
ST si
```

Przykłady w ST:

```
str :=TIME_TO_STRING(T#12ms);
dw:=TIME_TO_DWORD(T#5m);
si:=TOD_TO_SINT(TOD#00:00:00.012);
```

Przykłady w FBP:



Konwersje DATE_TO / DT_TO

Konwersja typu DATE lub DATE_AND_TIME na inny typ:

Data jest zachowywana wewnątrznie w DWORD w sekundach od 01.01.1970. Wartość ta zostaje poddana konwersji.

Może się zdarzyć, że w trakcie konwersji z większych na mniejsze typy zostaną utracone informacje.

W przypadku typu STRING wynik to stała daty.

Przykłady w IL:

```
LD D#1970-01-01 (* wynik to FALSE *)
DATE_TO_BOOL
ST b

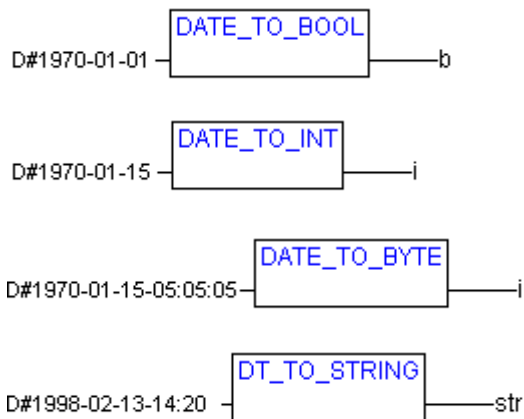
LD D#1970-01-15 (* wynik to 29952 *)
DATE_TO_INT
ST i

LD DT#1970-01-15-05:05:05 (* wynik to 129 *)
DT_TO_BYTE
ST byt

LD DT#1998-02-13-14:20 (* wynik to
DT_TO_STRING 'DT#1998-02-13-14:20' *)
ST str
```

Przykłady w ST:

```
b :=DATE_TO_BOOL(D#1970-01-01);
i :=DATE_TO_INT(D#1970-01-15);
byt :=DT_TO_BYTE(DT#1970-01-15-05:05:05);
str:=DT_TO_STRING(DT#1998-02-13-14:20);
```

Przykłady w FBP:**Konwersje STRING_TO**

Konwersja typu STRING na inny typ: Argument typu STRING musi mieć prawidłową wartość typu docelowego, w innym razie wynik wynosi 0.

Przykłady w IL:

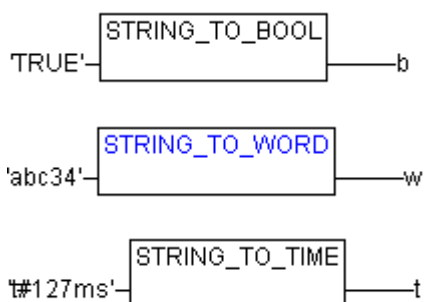
```
LD 'TRUE' (* wynik to TRUE *)
STRING_TO_BOOL
ST b
```

```
LD 'abc34' (* wynik to 0 *)
STRING_TO_WORD
ST w
```

```
LD 't#127ms' (* wynik to T#127ms *)
STRING_TO_TIME
ST t
```

Przykłady w ST:

```
b :=STRING_TO_BOOL('TRUE');
w :=STRING_TO_WORD('abc34');
t :=STRING_TO_TIME('T#127ms');
```

Przykłady w FBP:**TRUNC**

Konwersja typu REAL na typ INT. Pobierana jest tylko wartość części liczby, będąca liczbą całkowitą. Może się zdarzyć, że w trakcie konwersji z większych na mniejsze typy zostaną utracone informacje.

Przykład w IL:

```
LD 2.7
TRUNC
GE %MW8
```

Przykłady w ST:

```
i:=TRUNC(1.9); (* wynik to 1 *)
i:=TRUNC(-1.4); (* wynik to -1 *)
```

13.9 Operatory numeryczne...

ABS

Daje wartość bezwzględną liczby. ABS(-2). Możliwe są następujące kombinacje typów IN i OUT:

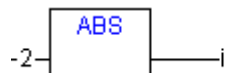
IN	OUT
INT	INT, REAL, WORD, DWORD, DINT
REAL	REAL
BYTE	INT, REAL, BYTE, WORD, DWORD, DINT
WORD	INT, REAL, WORD, DWORD, DINT
DWORD	REAL, DWORD, DINT
SINT	REAL
USINT	REAL
UINT	INT, REAL, WORD, DWORD, DINT, UDINT, UINT
DINT	REAL, DWORD, DINT
UDINT	REAL, DWORD, DINT, UDINT

Przykład w IL:

```
LD -2
ABS
ST i (* wynik to 2 *)
```

Przykład w ST:

```
i:=ABS(-2);
```

Przykład w FBD:**SQRT**

Daje pierwiastek kwadratowy danej liczby.

IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 16
SQRT
ST q (* wynik to 4 *)
```

Przykład w ST:

```
q:=SQRT(16);
```

Przykład w FBD:**LN**

Daje logarytm naturalny danej liczby.

IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 45
LN
ST q (* wynik to 3.80666 *)
```

Przykład w ST:

```
q:=LN(45);
```

Przykład w FBD:**LOG**

Daje logarytm dziesiętny danej liczby.

IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 314,5
LOG
ST q (* wynik to 2.49762 *)
```

Przykład w ST:

```
q:=LOG(314.5);
```

Przykład w FBD:**EXP**

Daje funkcję wykładniczą.

IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 2
EXP
ST q (* wynik to 7.389056099 *)
```

Przykład w ST:

```
q:=EXP(2);
```

Przykład w FBD:**SIN**

Daje sinus danej liczby. Wartość jest obliczana w miarze łukowej.

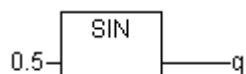
IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 0.5
SIN
ST q (* wynik to 0.479426 *)
```

Przykład w ST:

```
q:=SIN(0.5);
```

Przykład w FBD:**COS**

Daje cosinus danej liczby. Wartość jest obliczana w miarze łukowej.

IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 0.5
COS
ST q (* wynik to 0.877583 *)
```

Przykład w ST:

```
q:=COS(0.5);
```

Przykład w FBD:**TAN**

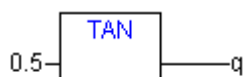
Daje tangens danej liczby. Wartość jest obliczana w miarze łukowej. IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 0.5
TAN
ST q (* wynik to 0.546302 *)
```

Przykład w ST:

```
q:=TAN(0.5);
```


Przykład w FBD:**ASIN**

Daje arcus sinus (funkcję odwrotną do funkcji sinus) danej liczby. Wartość jest obliczana w miarze łukowej.

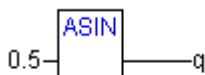
IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 0.5
ASIN
ST q (* wynik to 0.523599 *)
```

Przykład w ST:

```
q:=ASIN(0.5);
```

Przykład w FBD:**ACOS**

Daje arcus cosinus (funkcję odwrotną do funkcji cosinus) danej liczby. Wartość jest obliczana w miarze łukowej.

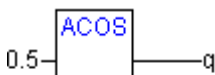
IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 0.5
ABS
ST q (* wynik to 1.0472 *)
```

Przykład w ST:

```
q:=ACOS(0.5);
```

Przykład w FBD:**ATAN**

Daje arcus tangens (funkcję odwrotną do funkcji tangens) danej liczby. Wartość jest obliczana w miarze łukowej.

IN może być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 0.5
ABS
ST q (* wynik to 0.463648 *)
```

Przykład w ST:

```
q:=ATAN(0.5);
```

Przykład w FBD:



EXPT

Potęgowanie zmiennej z inną zmienną w wykładniku:

```
OUT = IN1IN2.
```

IN1 i IN2 mogą być typu BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, natomiast OUT musi być typu REAL.

Przykład w IL:

```
LD 7  
EXPT 2  
ST var1 (* wynik to 49 *)
```

Przykład w ST:

```
var1 := EXPT(7,2);
```

Przykład w FBD:



13.10 Operator inicjalizacji

Operator INI

Za pomocą operatora INI można uruchomić zmienne Retain bloku funkcji stosowanej w module instancji.

Operator musi być przydzielony do zmiennej boolean.

Składnia: <zmienna boolean> := INI (<instancja bloku funkcji, TRUE|FALSE)

Jeśli drugi parametr operatora jest ustawiony na TRUE, inicjalizowane są wszystkie zmienne Retain, zdefiniowane w bloku funkcji FB.

Przykład w ST: fbinst jest instancją bloku funkcji fb, w którym jest zdefiniowana zmienna retain.

Deklaracja w module:

```
fbinst:fb;  
b:bool;
```

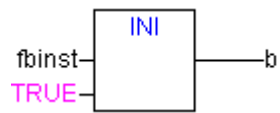
Część programu:

```
b := INI(fbinst, TRUE);  
ivar:=fbinst.retvar (* => inicjalizacja retvar *)
```

Przykład wywołania operatora w IL:

```
LD fbinst  
INI TRUE  
ST b
```

Przykład wywołania operatora w FBD:



Operator inicjalizacji

Załącznik B Argumenty w CoDeSys

W **CoDeSys** mogą być stosowane stałe, zmienne, adresy i w razie potrzeby wywołania funkcji jako argumenty.

13.11 Stałe...

Stałe typu BOOL

Stałe typu BOOL są wartościami prawdziwości TRUE i FALSE.

Stałe TIME

W **CoDeSys** można zadeklarować stałe TIME. Są one w szczególności używane do obsługi timera z biblioteki standardowej. Stała TIME zawsze składa się w „t” lub „T” umieszczonego na początku (lub „time” lub „TIME” w pełnej formie) i znaku „#”.

Następnie pojawia się właściwa deklaracja. Może się ona składać z dni (określane za pomocą „d”), godzin (określane za pomocą „h”), minut (określane za pomocą „m”), sekund (określane za pomocą „s”) i milisekund (określane za pomocą „ms”). Należy zwrócić uwagę, iż dane czasu muszą być uporządkowane według wielkości (d przed h przed m przed s przed ms), przy czym nie muszą występować wszystkie dane czasu.

Przykłady poprawnej stałej TIME w przydziale ST:

```
TIME1 := T#14ms;
```

```
TIME1 := T#100S12ms;      (* Dozwolone jest przepelnienie w zakresie najwyzszego skladnika *)
```

```
TIME1 := t#12h34m15s;
```

niepoprawne byłoby:

```
TIME1 := t#5m68s;        (* Przepelnienie w nizszym miejscu *)
```

```
TIME1 := 15ms;          (* Brakuje T# *)
```

```
TIME1 := t#4ms13d;      (* niepoprawna kolejnosc danych czasu *)
```

Stałe DATE

Typ ten służy do podawania czasu. Stała DATE jest deklarowana za pomocą „d”, „D”, „DATE” lub „date” na początku a następnie znaku „#”. Następnie można wpisać dowolną datę w kolejności rok-miesiąc-dzień.

Przykłady:

```
DATE#1996-05-06
```

```
d#1972-03-29
```

Wartości DATE (skrót D) wewnątrznie są traktowane jak DWORD. Czas jest podawany w **sekundach**, przy czym jest liczony od 01.01.1970 od godziny 00:00.

Stałe TIME_OF_DAY

Za pomocą tego typu można zapisywać godziny. Deklaracja TIME_OF_DAY rozpoczyna się od „tod#”, „TOD#”, „TIME_OF_DAY#” lub „time_of_day#”, następnie można podać godzinę w formacie: godzina:minuta:sekunda. Sekundy można przy tym podać jako liczby rzeczywiste, można też podawać ułamki sekund.

Przykłady:

```
TIME_OF_DAY#15:36:30.123
```

```
tod#00:00:00
```

Wartości TIME_OF_DAY (skrót TOD) wewnątrznie są traktowane jak DWORD. Czas jest podawany w **milisekundach**, przy czym jest on liczony od godziny 00:00.

Stałe DATE_AND_TIME

Stałe daty i czasy można również ze sobą łączyć w tzw. stałe DATE_AND_TIME. Stałe DATE_AND_TIME rozpoczynają się od „dt#”, „DT#”, „DATE_AND_TIME#” lub „date_and_time#”. Po danych daty następuje myślnik, a potem godzina.

Przykłady:

```
DATE_AND_TIME#1996-05-06-15:36:30
dt#1972-03-29-00:00:00
```

Wartości DATE_AND_TIME (skrót DT) wewnątrznie są traktowane jak DWORD. Czas jest podawany w **sekundach**, przy czym jest liczony od 01.01.1970 od godziny 00:00.

Stałe liczbowe

Wartości liczbowe mogą występować jako liczby binarne, liczby systemu ósemkowego, liczby dziesiętne i liczby systemu szesnastkowego. Jeśli wartość całkowita nie jest liczbą dziesiętną, przed stałą całkowitą po jej podstawie należy zapisać krzyżyk (#). Wartości liczbowe 10 do 15 w liczbach systemu szesnastkowego należy podawać w typowy sposób za pomocą liter A–F.

W obrębie wartości liczbowej dozwolone są podkreślenia.

Przykłady:

```
14          (liczba dziesiętna)
2#1001_0011 (liczba binarna)
8#67       (liczba systemu ósemkowego)
16#A      (liczba systemu szesnastkowego)
```

Możliwy typ tych wartości liczbowych to BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL lub LREAL.

Nie jest dozwolona wewnętrzna konwersja z „większych” na „mniejsze” typy. Tzn. zmiennej DINT nie można stosować bez problemów jako zmiennej INT. W tym celu należy stosować konwersję typu.

Stałe REAL- / LREAL

Stałe REAL i LREAL można podawać jako ułamki dziesiętne i w prezentacji wykładniczej. Stosuje się przy tym pisownię amerykańską z kropką.

Przykład:

```
7.4 zamiast 7,4
1.64e+009 zamiast 1,64e+009
```

Stałe STRING

String to dowolny ciąg znaków. Stałe STRING są ograniczone od przodu i tyłu prostymi apostrofami. Można również wpisywać spacje i znaki diakrytyczne. Są one traktowane dokładnie tak samo jak wszystkie inne znaki.

W ciągach znaków kombinacja znaku dolara (\$) z następującymi po nim dwoma cyframi w systemie szesnastkowym jest interpretowana jako szesnastkowe wyświetlenie kodu znaku ośmiobitowego. Ponadto kombinacje dwóch znaków rozpoczynające się od symbolu dolara występujące w ciągu znaków są interpretowane w następujący sposób:

```
$$          Symbol dolara
$'          Apostrof
$L lub $I   Zmiana wiersza
```

\$N lub \$n	Nowy wiersz
\$P lub \$p	Zmiana strony
\$R lub \$r	Łamanie linii
\$T lub \$t	Tabulator

Przykłady:

```
'w1Wüß?'
'Zuza i Marek'
':-)'
```

Stałe typowane (Typed Literals)

Za wyjątkiem stałych REAL/LREAL (w tym przypadku zawsze jest stosowane LREAL) przy obliczaniu za pomocą stałych IEC stosowany jest możliwie najmniejszy typ danych. Jeśli ma być zastosowany inny typ danych, można to osiągnąć za pomocą stałych w typach (typed literals) bez potrzeby wyraźnego deklarowania stałej. Stała zostaje w tym celu opatrzona przedrostkiem określającym typ:

Pisownia to: <Type>#<Literal>

za pomocą <Type> jest podany wymagany typ danych, możliwe wpisy to: BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, LREAL. Typ należy wpisać wielkimi literami.

Za pomocą <Literal> jest podana stała. Wpis musi być odpowiedni w stosunku do typów danych określonych w <Type>.

Przykład:

```
var1:=DINT#34;
```

Jeśli stałej nie można przekształcić bez utraty danych w typ docelowy, podawany jest następujący komunikat błędu:

Stałe typowane można stosować wszędzie tam gdzie zwykle stałe.

13.12 Zmienne...

Zmienne są deklarowane w części deklaracji modułu lub w globalnych listach zmiennych.

Wskazówka: Można definiować zmienną lokalną z taką samą nazwą jak zmienna globalna. W obrębie modułu zmienna lokalna ma zawsze pierwszeństwo. Nie jest możliwe nadanie tej samej nazwy dwóm zdefiniowanym globalnie zmiennym; przykładowo błąd kompilacji pojawia się, jeśli zarówno w liście zmiennych globalnych, jak i w konfiguracji sterownika jest zdefiniowana zmienna „var1”.

Nazwa zmiennych nie może zawierać spacji i przegłosów, nie może ona być deklarowana podwójnie i być identyczna ze słowami kluczowymi. Nie jest uwzględniana pisownia wielkimi/małymi literami, nazwa VAR1, Var1 i var1 oznacza tę samą zmienną. Podkreślenia są dla identyfikatorów znaczące, np. „A_BCD” oraz „AB_CD” to różne identyfikatory. Niedozwolone jest stosowanie kilku podkreśleń kolejno na początku lub wewnątrz identyfikatora. Długość nazwy oraz istotny zakres. Zmienne mogą być stosowane wszędzie tam, gdzie pozwala na to zadeklarowany typ.

Dostępne zmienne można wywołać z poziomu asystenta deklaracji.

Flagi systemowe

Flagi systemowe to pośrednio zadeklarowane zmienne zależne od swojego specjalnego sterownika. W celu znalezienia flag systemowych posiadanych przez system należy skorzystać w poleceniu „Wstaw” „Argument”, po czym pojawia się okno pomocy przy wpisywaniu, w którym należy wybrać kategorię **Zmienna systemowa**.

Dostęp do zmiennych tablic, struktur i modułów

Dostęp do składników tablic dwuwymiarowych odbywa się przy użyciu następującej składni:

```
<nazwa pola>[Index1, Index2]
```

Dostęp do zmiennych ze struktur odbywa się przy użyciu następującej składni:

```
<nazwa struktury>.<nazwa zmiennej>
```

Dostęp do zmiennych z bloków funkcji odbywa się przy użyciu następującej składni:

```
<nazwa modułu>.<nazwa zmiennej>
```

Adresowanie bitów w zmiennych

W zmiennych całkowitych można wywołać poszczególne bity. W tym celu do zmiennej jest dołączany oddzielony za pomocą kropki indeks bitu do zaadresowania. Indeks bitu można podać za pomocą dowolnej stałej. Indeksowanie opiera się na 0.

Wskazówka: Dostęp bitowy w zmiennych bezpośrednich nie jest dozwolony (ma to znaczenie w przypadku stosowania biblioteki SysLibDirect.lib.)

Przykład:

```
a : INT;
b : BOOL;
...
a.2 := b;
```

Trzeci bit zmiennej a jest ustawiany na wartość zmiennej b.

Jeśli indeks jest większy niż szerokość bitu, pojawia się następujący komunikat o błędzie: „Indeks '<n>' poza poprawnym zakresem zmiennej '<var>'!”.

Adresowanie bitów jest możliwe w przypadku następujących typów zmiennych: SINT, INT, DINT, USINT, UINT, UDINT, BYTE, WORD, DWORD.

Jeśli dany typ zmiennej nie jest dopuszczalny, pojawia się następujący komunikat o błędzie: „Niedopuszczalny typ danych '<Typ>' do indeksowania bezpośredniego”.

Dostępu bitowego nie można przydzielić zmiennej VAR_IN_OUT.

Dostęp bitowy za pomocą stałej globalnej:

W przypadku zadeklarowania stałej globalnej definiującej numer bitu można zastosować tę stałą do dostępu bitowego.

Wskazówka: Opcja projektu „Zastąp stałe” (kategoria Opcje kompilacji) musi być włączona!

Poniższe przykłady ilustrują dostęp bitowy do zwykłej zmiennej lub zmiennej strukturalnej:

Deklaracja dla obu przykładów w liście zmiennych globalnych:

Za pomocą zmiennej enable podaje się, do którego z kolei bitu ma być przeprowadzony dostęp:

```
VAR_GLOBAL CONSTANT
    enable:int:=2;
END_VAR
```

Przykład 1, dostęp bitowy do zmiennej całkowitej:

Deklaracja w module:

```
VAR
    xxx:int;
END_VAR
```

Dostęp bitowy:

```
xxx.enable:=true; -> trzeci bit w zmiennej xxx zostaje ustawiony na TRUE
```

Przykład 2, dostęp bitowy do całkowitego składnika struktury:

Deklaracja struktury stru1:


```

TYPE stru1 :
STRUCT
    bvar:BOOL;
    rvar:REAL;
    wvar:WORD;
    {bitaccess enable 42 'Udostępnić napęd'}
END_STRUCT
END_TYPE

```

Deklaracja w module:

```

VAR
    x:stru1;
END_VAR

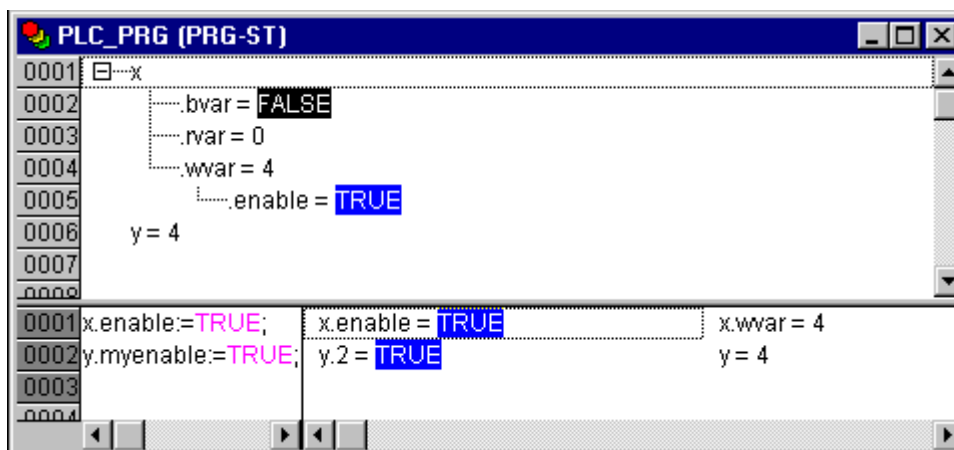
```

Dostęp bitowy:

```
x.enable:=true;
```

Na podstawie tego 42 w zmiennej x zostaje ustawiony na TRUE. Ponieważ bvar zawiera 8 bitów, zaś rvar 32 bity, dostęp ten następuje do 2. bitu w zmiennej wvar, który przez to uzyskuje wartość 4.

Uwaga: W celu poprawnej ilustracji zmiennej przeprowadzającej dostęp bitowy do zmiennej strukturalnej, przy monitorowaniu, w pomocy przy wpisywaniu i w funkcji Intellisense należy zastosować ukazaną w przykładzie **Pragmę {bitaccess}** (patrz rozdział 8.2.3). Następnie w oknie deklaracji poniżej zmiennej strukturalnej w trakcie monitoringu zostaje wyświetlona stała globalna:



13.13 Adresy...

Uwaga: W przypadku zastosowania zmiany online, zawartość adresów może ulec przemieszczeniu. Przy stosowaniu wskaźników należy zwracać uwagę na adresy.

Adres

Bezpośrednie wyświetlanie miejsc w pamięci następuje za pomocą specjalnych ciągów znaków. Powstają one ze złączenia znaku procent „%”, przedrostka zakresu, przedrostka rozmiaru i jednej bądź wielu liczb naturalnych oddzielonych od siebie za pomocą spacji.

Obsługiwane są następujące przedrostki zakresu:

- I Wejście
- Q Wyjście
- M Znacznik

Obsługiwane są następujące przedrostki rozmiarów:

Funkcje...

X	Pojedynczy bit
None	Pojedynczy bit
B	Bajt (8 bitów)
W	Słowo (16 bitów)
D	Podwójne słowo (32 bity)

Przykłady:

%QX7.5 i %Q7.5	Bit wyjściowy 7.5
%IW215	Słowo wejściowe 215
%QB7	Bajt wyjściowy 7
%MD48	Słowo podwójne na miejscu pamięci 48 w znaczniku

Poprawność adresu zależy od aktualnej konfiguracji sterownika programu.

Wskazówka: Wartości boolean są alokowane w bajtach, jeśli nie jest podany jednoznacznie adres pojedynczego bitu. Przykład: Zmiana wartości z varbool1 AT %QW0 dotyczy zakresu od QX0.0 do QX0.7.

Patrz również rozdział Załącznik A, operatory IEC i dodatkowe funkcje poszerzające normę, operatory adresu.

Znacznik

Do dostępu do znacznika można używać wszystkich obsługiwanych rozmiarów.

Przykładowo adres %MD48 adresowałby bajty nr 192, 193, 194 i 195 w zakresie znacznika ($48 * 4 = 192$). Pierwszy bajt to bajt nr 0.

W taki sam sposób można mieć dostęp do słów i bajtów, a nawet do bitów: Za pomocą %MX5.0 korzysta się z pierwszego bitu w piątym słowie (bity z reguły są zachowywane w odniesieniu do słów).

Patrz również rozdział Załącznik A, operatory IEC i dodatkowe funkcje poszerzające normę, operatory adresu.

13.14 Funkcje...

W ST jako argument może również wystąpić wywołanie funkcji.

Przykład:

```
Wynik:= Fct(7) + 3;
```

Funkcja TIME()

Funkcja ta daje czas w milisekundach, które upłynęły od chwili uruchomienia systemu.

Typ danych to TIME.

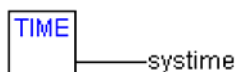
Przykład w IL:

```
TIME  
ST systime (* wynik np.: T#35m11s342ms *)
```

Przykład w ST:

```
systime:=TIME();
```

Przykład w FBD:



Załącznik C

Załącznik D Typy danych w CoDeSys

Użytkownik w programowaniu może stosować standardowe i zdefiniowane przez siebie typy danych. Do każdego identyfikatora jest przyporządkowany typ danych ustalający, ile pamięci jest rezerwowane i które wartości odpowiadają zawartości pamięci.

13.15 Standardowe typy danych

BOOL

Zmienne typu danych **BOOL** mogą przyjmować wartości prawdziwe TRUE i FALSE. Rezerwowanych jest 8 bitów pamięci.

Typy danych w liczbach całkowitych

Do całkowitych typów danych zaliczają się **BYTE**, **WORD**, **DWORD**, **SINT**, **USINT**, **INT**, **UINT**, **DINT**, **UDINT**.

Różne typy liczb pokrywają różne zakresy liczbowe. Dla typów danych w liczbach całkowitych odnoszą się następujące granice zakresów:

Typ	Dolna granica	Górna granica	Ilość pamięci
BYTE	0	255	8 bitów
WORD	0	65535	16 bitów
DWORD	0	4294967295	32 bity
SINT:	-128	127	8 bitów
USINT:	0	255	8 bitów
INT:	-32768	32767	16 bitów
UINT:	0	65535	16 bitów
DINT:	-2147483648	2147483647	32 bity
UDINT:	0	4294967295	32 bity

W związku z tym może się zdarzyć, że w trakcie konwersji z większych na mniejsze typy zostanie utracona jakaś informacja.

REAL / LREAL

Typy danych **REAL** i **LREAL** to tzw. typy zmiennoprzecinkowe. Są one konieczne przy stosowaniu liczb wymiernych. Zarezerwowana pamięć wynosi 32 bity w przypadku REAL i 64 bity w przypadku LREAL.

Wartości dopuszczalne dla REAL: 1.175494351e-38F do 3.402823466e+38F

Wartości dopuszczalne dla LREAL: 2.2250738585072014e-308 do 1.7976931348623158e+308

STRING

Zmienna typu danych **STRING** może stanowić dowolny ciąg znaków. Informacja o rozmiarze zarezerwowanej pamięci w deklaracji odnosi się do znaków i może być umieszczona w nawiasach okrągłych lub kwadratowych. Jeśli nie podano rozmiaru, standardowo przyjmuje się 80 znaków.

Długość stringu zasadniczo nie jest ograniczona, jednak funkcje stringów mogą przetwarzać tylko długości od 1 do 255!

Przykład deklaracji stringu z 35 znakami:

```
str:STRING(35):='To jest string';
```

Typy danych czasu

Typy danych **TIME**, **TIME_OF_DAY** (skrót **TOD**), **DATE** i **DATE_AND_TIME** (skrót **DT**) są wewnątrznie traktowane jak **DWORD**.

W **TIME** i **TOD** czas jest podawany w milisekundach, przy czym **TOD** jest liczone od godziny 00:00.

W **DATE** i **DT** czas jest podawany w sekundach, przy czym jest liczony od 01.01.1970, godzina 00:00.

Poniżej podano formaty danych czasu do przydziału (stałe czasu i daty):

Stałe czasowe **TIME**

Stała **TIME** zawsze składa się w „t” lub „T” umieszczonego na początku (lub „time” lub „TIME” w pełnej formie) i znaku „#”.

Następnie pojawia się właściwa deklaracja. Może się ona składać z dni (określane za pomocą „d”), godzin (określane za pomocą „h”), minut (określane za pomocą „m”), sekund (określane za pomocą „s”) i milisekund (określane za pomocą „ms”). Należy zwrócić uwagę, iż dane czasu muszą być uporządkowane według wielkości (d przed h przed m przed s przed ms), przy czym nie muszą występować wszystkie dane czasu.

Wartość maksymalna: 49d17h2m47s295ms (4194967295 ms)

Przykłady poprawnej stałej **TIME** w przydziale **ST**:

`TIME1 := T#14ms;`

`TIME1 := T#100S12ms;` (* Dozwolone jest przepelnienie w zakresie najwyższego składnika *)

`TIME1 := t#12h34m15s;`

niepoprawne byłoby:

`TIME1 := t#5m68s;` (* Przepelnienie w niższym miejscu *)

`TIME1 := 15ms;` (* Brakuje T# *)

`TIME1 := t#4ms13d;` (* niepoprawna kolejność danych czasu *)

Stałe **DATE**, do danych daty:

Stała **DATE** jest deklarowana za pomocą „d”, „D”, „DATE” lub „date” na początku a następnie znaku „#”. Następnie można wpisać dowolną datę w kolejności rok-miesiąc-dzień. Możliwe wartości: 1970-00-00 do 2106-02-06.

Przykłady:

`DATE#1996-05-06`

`d#1972-03-29`

Stałe **TIME_OF_DAY**, do zachowania godzin:

Deklaracja **TIME_OF_DAY** rozpoczyna się od „tod#”, „TOD#”, „TIME_OF_DAY#” lub „time_of_day#”, następnie można podać godzinę w formacie: godzina:minuta:sekunda. Sekundy można przy tym podać jako liczby rzeczywiste, można też podawać ułamki sekund. Możliwe wartości: 00:00:00 do 23:59:59.999.

Przykłady:

`TIME_OF_DAY#15:36:30.123`

`tod#00:00:00`

Stałe DATE_AND_TIME, kombinacja daty i czasu:

Stałe DATE_AND_TIME rozpoczynają się od „dt#”, „DT#”, „DATE_AND_TIME#” lub „date_and_time#”. Po danych daty następuje myślnik i potem godzina. Możliwe wartości: 1970-00-00-00:00:00 do 2106-02-06-06:28:15.

Przykłady:

```
DATE_AND_TIME#1996-05-06-15:36:30
dt#1972-03-29-00:00:00
```

13.16 Zdefiniowane typy danych

Tablica

Z podstawowych typów danych obsługiwane są pola (tablice) jedno-, dwu- i trójwymiarowe. Tablice mogą być definiowane w deklaracji modułu i w listach zmiennych globalnych. Wskutek zagnieżdżenia tablic (ARRAY[0..2] OF ARRAY[0..3] OF ...) może powstać maksymalnie 9 wymiarów.

Składnia:

<nazwa_pola>:ARRAY [<ug1>..

ug1, ug2, ug3 podają dolną granicę obszaru pola, og1, og2, og3 górną granicę. Wartości graniczne muszą być liczbami całkowitymi i odpowiadać zakresowi wartości DINT.

Przykład:

```
Talia kart: ARRAY [1..13, 1..4] OF INT;
```

Inicjalizacja tablic:

Przykłady pełnej inicjalizacji tablicy: arr1 : ARRAY [1..5] OF INT := 1,2,3,4,5;

```
arr2 : ARRAY [1..2,3..4] OF INT := 1,3(7);
(* skrót na 1,7,7,7 *)
```

```
arr3 : ARRAY [1..2,2..3,3..4] OF INT := 2(0),4(4),2,3;
(* skrót na 0,0,4,4,4,4,2,3 *)
```

Przykład inicjalizacji tablicy struktury:

```
TYPE STRUCT1
STRUCT
    p1:int;
    p2:int;
    p3:dword;
END_STRUCT
ARRAY[1..3] OF STRUCT1:= (p1:=1,p2:=10,p3:=4723), (p1:=2,p2:=0,p3:=299),
(p1:=14,p2:=5,p3:=112);
```

Przykład częściowej inicjalizacji tablicy:

```
arr1 : ARRAY [1..10] OF INT := 1,2;
```

Elementy, dla których nie jest zadana wartość, są inicjalizowane na podstawie domyślnej wartości początkowej podstawowego typu. W powyższym przykładzie elementy anarray[6] do anarray[10] są inicjalizowane za pomocą 0.

Dostęp do składników tablicy:

Do składników tablic dostęp w polu dwuwymiarowym odbywa się za pomocą następującej składni:

```
<Feld_Name>[Index1,Index2]
```

Przykład:

```
Talia kart[9,2]
```

Wskazówka: Jeśli w projekcie zostanie zdefiniowana funkcja o nazwie `CheckBounds` można z jej pomocą automatycznie sprawdzić przekroczenie obszaru tablicy.

Funkcja Checkbounds

Jeśli w projekcie zostanie zdefiniowana funkcja o nazwie **CheckBounds** można z jej pomocą automatycznie sprawdzić przekroczenie obszaru tablicy. Nazwa funkcji jest ustalona i może to być tylko ona.

Przykład funkcji CheckBounds:

```
FUNCTION CheckBounds : DINT
VAR_INPUT
    index, lower, upper: DINT;
END_VAR
IF index < lower THEN
    CheckBounds := lower;
ELSIF index > upper THEN
    CheckBounds := upper;
ELSE CheckBounds := index;
END_IF
```

Następujący program przykładowy do testowania funkcji `CheckBounds` sięga poza granice zdefiniowanej tablicy. Funkcja `CheckBounds` gwarantuje, że wartość `TRUE` jest przydzielana nie na miejscu `A[10]`, lecz w górnej jeszcze poprawnej granicy zakresu `A[7]`. Za pomocą funkcji `CheckBounds` można korygować dostęp poza granicami tablicy.

Program testowy do funkcji CheckBounds:

```
PROGRAM PLC_PRG
VAR
    a: ARRAY[0..7] OF BOOL;
    b: INT:=10;
END_VAR
a[b]:=TRUE;
```

Uwaga: Zawarta w `CheckLib` funkcja `CheckBounds` to rozwiązanie przykładowe! Przed zastosowaniem biblioteki należy sprawdzić, czy funkcja pracuje zgodnie z założeniami lub zaimplementować bezpośrednio do projektu odpowiednią funkcję `CheckBounds` jako moduł.

Wskaźniki

We wskaźnikach zapisuje się adresy zmiennych lub bloków funkcji w czasie przebiegu programu.

Składnia deklaracji znaczników jest następująca:

```
<identyfikator>: POINTER TO <typ danych/blok funkcji>;
```

Znacznik może wskazywać każdy dowolny typ danych i blok funkcji, również zdefiniowane przez użytkownika.

Za pomocą operatora adresu `ADR` do wskaźnika jest przydzielany adres zmiennej lub bloku funkcji.

Zmiana referencji wskaźnika następuje za pomocą operatora zawartości „`^`” po identyfikatorze wskaźnika.

Należy pamiętać: Wskaźnik jest obliczany za pomocą bajtów. Za pomocą instrukcji `p=p+SIZEOF(p^)`; można przeprowadzić obliczanie jak w kompilatorze C.

Przykład:

```
pt:POINTER TO INT;
var_int1:INT := 5;
var_int2:INT;
pt := ADR(var_int1);
var_int2:= pt^; (* var_int2 ist nun 5 *)
```


Uwaga: W przypadku zastosowania zmiany online, zawartość adresów może ulec przemieszczeniu. Przy stosowaniu wskaźników należy zwracać uwagę na adresy.

Funkcje CheckPointer i CheckPointerAligned

Aby sprawdzić dostęp do wskaźników w czasie przebiegu można utworzyć funkcje Check z niżej podanymi nazwami, które przed każdym dostępem automatycznie wywołują zawartość wskaźnika, jeśli są one dostępne w projekcie (bezpośrednio w projekcie lub przez bibliotekę).

- Funkcja **CheckPointer** sprawdza, czy wywoływany przez wskaźnik adres znajduje się w poprawnym obszarze pamięci.
- Funkcja **CheckPointerAligned**, zawiera funkcjonalność CheckPointer i dodatkowo sprawdza alokację pamięci.

Funkcje muszą mieć dokładnie podane nazwy. Zwracają one adres stosowany do zmiany referencji wskaźnika, we właściwym przypadku więc ten przekazany jako pierwszy parametr wejściowy (w poniższym przykładzie dwAddress).

W poniższym przykładzie funkcji CheckPointerAligned widać, które parametry wejściowe są oczekiwane. Nazwy parametrów stanowią również przykład. Funkcja CheckPointer musi wyglądać dokładnie tak samo, jedynie odpada parametr ziarnistości dostępu.

```

FUNCTION
CheckPointerAligned : DWORD
    (* Typ danych funkcji (wartość zwracana) musi
    odpowiadać typowi danych dla wskaźnika w aktualnie
    ustawionym systemie docelowym, tzn. DWORD dla
    systemów stosujących wskaźniki 32 bitowe, WORD
    dla systemów stosujących wskaźniki 16 bitowe *)

VAR_INPUT
    dwAddress : DWORD;
    (* Adres docelowy wskaźnika; typ danych musi
    odpowiadać typowi danych dla wskaźnika w aktualnie
    ustawionym systemie docelowym, patrz wyżej;
    wartość zwracana funkcji *)

    iSize : DINT;
    (* Rozmiar dostępu; typ danych musi być
    kompatybilny z liczbami całkowitymi i pokrywać
    maksymalny możliwy rozmiar danych w adresie
    dostępu *)

    iGran : DINT;
    (* ! Odpada w funkcjach CheckPointer !: Ziarnistość
    dostępu, np. „2”, jeśli INT jest najmniejszym
    stosowanym typem danych bez struktury w adresie;
    typ danych musi być kompatybilny z liczbami
    całkowitymi *)

    bWrite: BOOL;
    (* Dostęp z odczytem lub zapisem; TRUE = zapis;
    typem danych musi być BOOL *)


END_VAR

```

Jeśli w projekcie istnieje zarówno funkcja CheckPointer, jak i CheckPointerAligned, zostaje wywołana funkcja CheckPointerAligned.

Typ wyliczania, enumeracja

Typ wyliczania to zdefiniowany przez siebie typ danych składający się z pewnej ilości stałych stringów. Stałe te są określane jako wartości enumeracji.

Wartości enumeracji są znane w całym projekcie, również jeśli zostały one lokalnie zadeklarowane w module. Najlepiej jest utworzyć typy wyliczania jako obiekty w Organizерze obiektów w zakładce  **Typy danych**. Rozpoczynają się one od słowa kluczowego **TYPE** zaś kończą **END_TYPE**.

Składnia:

```

TYPE <identyfikator>:(<Enum_0> ,<Enum_1>, ..., <Enum_n>);
END_TYPE

```

Zmienna typu <Bezeichner> (identyfikator) może przyjąć jedną z wartości enumeracji i jest inicjalizowana za pomocą pierwszej z nich. Wartości są kompatybilne z liczbami całkowitymi, tzn. można z ich pomocą przeprowadzać operacje jak z INT. Do zmiennej może być przydzielona liczba x. Jeśli wartości enumeracji nie są zainicjalizowane, liczenie rozpoczyna się od 0. Przy inicjalizacji należy uważać na to, by wartości początkowe były w porządku rosnącym. Poprawność liczb jest sprawdzana w czasie przebiegu.

Przykład:


```
TYPE AMPEL: (Czerwone, żółte, zielone:=10); (* Czerwone ma wartość początkowa
0, żółte 1, zielone 10 *)
END_TYPE
AMPEL1 : AMPEL ;
AMPEL1:=0; (* Sygnalizacja uliczna ma wartość Czerwone *)
FOR i:= Czerwone TO Zielone DO
  i := i + 1;
END_FOR;
```

Ta sama wartość enumeracji nie może być stosowana dwukrotnie w obrębie samej enumeracji, jak i w przypadku stosowania różnych enumeracji w obrębie tego samego modułu.

Przykład:

```
AMPEL: (czerwone, żółte, zielone);
KOLOR: (niebieski, biały, czerwony);
Błąd: czerwone nie może być stosowane dla AMPEL (sygnalizatora) i FARBE
(koloru), jeśli są one stosowane w tym samym module.
```

Struktury

Struktury są tworzone jako obiekty (typy danych) w Organizeryze obiektów w zakładce  **Typy danych**. Rozpoczynają się one od słów kluczowych TYPE i STRUCT zaś kończą END_STRUCT i END_TYPE.

Składnia deklaracji struktury jest następująca:

```
TYPE <nazwa struktury>:
STRUCT
  <deklaracja zmiennej 1>
  .
  .
  <deklaracja zmiennej n>
END_STRUCT
END_TYPE
```

<nazwa struktury> to typ znany w całym projekcie i może być stosowany jak standardowy typ danych.

Dozwolone są struktury zagnieżdżone. Jedynym ograniczeniem jest, iż zmienne nie mogą być wstawiane w adresy (deklaracja AT nie jest dozwolona!).

Przykład definicji struktury ciąg poligonowy:

```
TYPE Ciąg poligonowy:
STRUCT
  Start:ARRAY [1..2] OF INT;
  Punkt1:ARRAY [1..2] OF INT;
  Punkt2:ARRAY [1..2] OF INT;
  Punkt3:ARRAY [1..2] OF INT;
  Punkt4:ARRAY [1..2] OF INT;
  Koniec:ARRAY [1..2] OF INT;
END_STRUCT
END_TYPE
```

Przykład inicjalizacji struktury typu Ciąg poligonowy:

```
Poly_1: Ciąg poligonowy := ( Start:=3,3, Punkt1:=5,2, Punkt2:=7,3, Punkt3:=8,5,
Punkt4:=5,7, End := 3,5);
```

Inicjalizacja za pomocą zmiennych nie jest możliwa. Przykład inicjalizacji tablicy struktury patrz w „Tablice”.

Dostęp do struktur:


Dostęp do składników struktury jest dokonywany za pomocą następującej składni:

```
<nazwa_struktury>.<nazwa_składnika>
```

Dla wyżej podanego przykładu struktury ciągu poligonowego dostęp do składnika Start jest wykonywany odpowiednio za pomocą Poly_1.Start

Referencje

Zdefiniowany przez siebie typ danych Referencja służy do utworzenia alternatywnej nazwy (alias) dla typu danych lub bloku funkcji.

Referencje należy utworzyć jako obiekty w Organizercie obiektów w zakładce  **Typy danych**. Rozpoczynają się one od słowa kluczowego TYPE zaś kończą END_TYPE.

Składnia:

```
TYPE <identyfikator>: <wyrażenie przydziału>;  
END_TYPE
```

Przykład:

```
TYPE message:STRING[50];  
END_TYPE;
```

Typy podzakresów

Typ podzakresu to typ danych, którego zakres wartości obejmuje tylko podzbiór typów podstawowych.

Deklaracja może nastąpić w zakładce  **Typy danych**, zmienna może jednak zostać zadeklarowana bezpośrednio za pomocą typu podzakresu:

Składnia deklaracji w zakładce „Typy danych”:

```
TYPE <nazwa> : <Inttype> (<ug>..<og>) END_TYPE;
```

<nazwa> musi być poprawnym identyfikatorem IEC.

<Inttype> jest jednym z typów danych SINT, USINT, INT, UINT, DINT, UDINT, BYTE, WORD, DWORD (LINT, ULINT, LWORD).

<ug> jest stałą, która musi być kompatybilna z typami podstawowymi ustalającymi dolną granicę typu zakresu. Sama dolna granica należy do tego zakresu.

<og> jest stałą, która musi być kompatybilna z typami podstawowymi ustalającymi górną granicę typu zakresu. Sama górna granica należy do tego zakresu.

Przykład:

```
TYPE  
  SubInt : INT (-4095..4095);  
END_TYPE
```

Bezpośrednia deklaracja zmiennej za pomocą typu podzakresu

Jeśli zakres dolny nie zawiera „0”, należy w poprawny sposób podać wartość początkową:

```
VAR  
  i1 : INT (-4095..4095);  
  i2: INT (5..10):=5;  
  ui : UINT (0..10000);  
END_VAR
```

Jeśli do typu podzakresu zostanie przydzielona stała (w deklaracji lub implementacji) nienależąca do tego zakresu (np. $i:=5000$) pojawia się komunikat błędu.

Aby sprawdzić zachowanie granic zakresów w czasie przebiegu, należy wstawić funkcje **CheckRangeSigned** lub **CheckRangeUnsigned**. Można w nich w odpowiedni sposób wychwycić naruszenia zakresu (np. może zostać odcięta wartość lub ustawiona flaga błędu). Są one wywoływane **pośrednio**, gdy tylko zostanie zapisana zmienna typu dolnego zakresu utworzonego z typu ze znakiem liczby.

Przykład:

W przypadku zmiennej typu zakresu ze znakiem liczby (więc jak i od góry) jest wywoływana funkcja **CheckRangeSigned**, która mogłaby zostać zaprogramowana w następujący sposób, by przyciąć wartość do dozwolonego zakresu:

```
FUNCTION CheckRangeSigned : DINT
VAR_INPUT
    value, lower, upper: DINT;
END_VAR
IF (value < lower) THEN
    CheckRangeSigned := lower;
ELSIF(value > upper) THEN
    CheckRangeSigned := upper;
ELSE
    CheckRangeSigned := value;
END_IF
```

Obligatoryjna dla wywołania automatycznego jest nazwa funkcji **CheckRangeSigned** oraz utworzenie **interfejsu**: wartość zwracana i trzy parametry typu **DINT**.

Przy wywołaniu funkcja ma następujące parametry:

- value: otrzymuje wartość, która ma być przydzielona do zakresu typu
- lower: dolna granica zakresu
- upper: górna granica zakresu
- Return value: wartość rzeczywiście przydzielana do zakresu typu

Z przydziału $i := 10*y$; w tym przykładzie powstaje wewnątrz:

```
i := CheckRangeSigned(10*y, -4095, 4095);
```

Jeśli y ma przykładowo wartość 1000, to i pomimo tego przydziału ma tylko wartość 4095.

Odpowiednio dla funkcji **CheckRangeUnsigned**: Nazwa funkcji i interfejs muszą być poprawne:

```
FUNCTION CheckRangeUnsigned : UDINT
VAR_INPUT
    value, lower, upper: UDINT;
END_VAR
```

Uwaga: Jeśli brak jest obu funkcji **CheckRangeSigned** i **CheckRangeUnsigned**, w czasie przebiegu nie jest przeprowadzane sprawdzanie typów podzakresów. Zmienna i mogłaby mieć więc dowolną wartość z przedziału od -32768 do 32767 !

Jeśli jak wykazano powyżej funkcja **CheckRangeSigned** lub **CheckRangeUnsigned** jest zaimplementowana, przy stosowaniu podzakresów w **pętli FOR** mogłaby powstać pętla nieskończona. Dzieje się tak dokładnie wtedy, gdy podany zakres dla pętli **FOR** jest dokładnie tak samo duży lub większy od zakresu dla typu podzakresu.

Zawarta w bibliotece **CheckLib** funkcja **CheckRangeSigned** to rozwiązanie przykładowe! Przed zastosowaniem biblioteki należy sprawdzić, czy funkcja pracuje zgodnie z założeniami lub zaimplementować bezpośrednio do projektu jako moduł odpowiednią funkcję **CheckRange**.

Przykład:

```
VAR  
  ui : UINT (0..10000);  
END_VAR  
FOR ui:=0 TO 10000 DO  
  ...  
END_FOR
```

Nie należy wychodzić z pętli FOR, ponieważ ui nie może być większe niż 10000.

Przy stosowaniu wartości przyrostowych w pętli FOR należy również zwrócić uwagę na zawartość funkcji CheckRange.

Zdefiniowane typy danych

Załącznik E Biblioteki CoDeSys

13.17 Biblioteka Standard.lib

Biblioteka standardowa

Biblioteka „standard.lib” jest dostępna w standardowej wersji programu. Zawiera ona wszystkie funkcje i moduły funkcji, które są wymagane dla systemu programowania IEC jako standardowe moduły IEC61131-3. Różnica między funkcją standardową i operatorem polega na tym, że operator jest pośrednio rozpoznawalny w systemie programowania, podczas gdy moduły standardowe muszą być podłączone do projektu jako biblioteka (standard.lib).

Kod źródłowy tych modułów jest dostępny w bibliotece C i stanowi składnik systemu CoDeSys.

13.17.1 Funkcje String...

Należy pamiętać: Funkcje string nie są „thread-safe”! Przy stosowaniu zadań można stosować funkcje String wyłącznie w jednym zadaniu. W przypadku stosowania tej samej funkcji w różnych zadaniach istnieje ryzyko nadpisania. Dozwolona długość stringu przy stosowaniu tych funkcji jest 1–255.

LEN

Funkcja LEN daje długość stringu.

Wejście STR jest typu STRING, wartość zwracana funkcji typu INT.

Przykład w IL:

```
LD 'SUSI'
LEN
ST VarINT1 (* wynik to 4 *)
```

Przykład w FBD:



Przykład w ST:

```
VarSTRING1 := LEN ('SUSI');
```

LEFT

Funkcja LEFT daje lewy string początkowy stringu.

Wejście STR jest typu STRING, SIZE typu INT, wartość zwracana funkcji jest typu STRING.

LEFT (STR, SIZE) oznacza: Pobierz pierwsze znaki SIZE od lewej strony w stringu STR.

Przykład w IL:

```
LD 'SUSI'
LEFT 3
ST VarSTRING1 (* wynik to 'SUS' *)
```

Przykład w FBD:



Przykład w ST:

```
VarSTRING1 := LEFT ('SUSI',3);
```

RIGHT

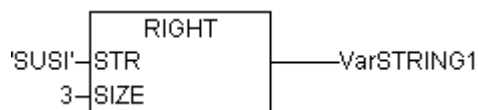
Funkcja RIGHT daje prawy string początkowy stringu.

Wejście STR jest typu STRING, SIZE typu INT, wartość zwracana funkcji jest typu STRING.

RIGHT (STR, SIZE) oznacza: Pobierz pierwsze znaki SIZE od prawej strony w stringu STR.

Przykład w IL:

```
LD 'SUSI'
RIGHT 3
ST VarSTRING1 (* wynik to 'USI' *)
```

Przykład w FBD:**Przykład w ST:**

```
VarSTRING1 := RIGHT ('SUSI',3);
```

MID

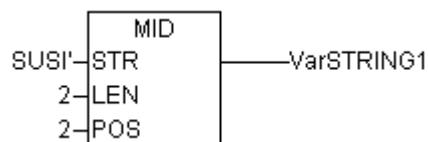
Funkcja MID daje string częściowy danego stringu.

Wejście STR jest typu STRING, LEN i POS typu INT, wartość zwracana funkcji typu STRING.

MID (STR, LEN, POS) oznacza: Pobierz LEN znaków ze stringu STR, począwszy od znaku na miejscu POS.

Przykład w IL:

```
LD 'SUSI'
MID 2,2
ST VarSTRING1 (* wynik to 'US' *)
```

Przykład w FBD:**Przykład w ST:**

```
VarSTRING1 := MID ('SUSI',2,2);
```

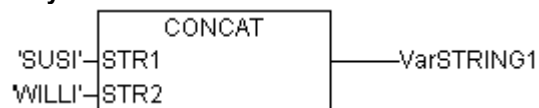
CONCAT

Funkcja CONCAT daje konkatencję (złączenie) dwóch stringów.

Wejścia STR1 i STR2 i wartość zwracana funkcji są typu STRING.

Przykład w IL:

```
LD 'SUSI'
CONCAT 'WILLI'
ST VarSTRING1 (* wynik to 'SUSIWILLI' *)
```


Przykład w FBD:**Przykład w ST:**

```
VarSTRING1 := CONCAT ('SUSI', 'WILLI');
```

INSERT

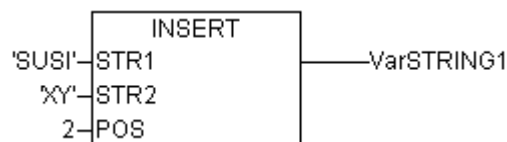
Funkcja INSERT wstawia od określonego miejsca w stringu inny string.

Wejścia STR1 i STR2 są typu STRING, POS typu INT, wartość zwracana funkcji jest typu STRING.

INSERT(STR1, STR2, POS) oznacza: wstaw STR2 w STR1 po miejscu POS-tym.

Przykład w IL:

```
LD    'SUSI'
INSERT 'XY',2
ST    VarSTRING1 (* wynik to 'SUXYSI' *)
```

Przykład w FBD:**Przykład w ST:**

```
VarSTRING1 := INSERT ('SUSI', 'XY', 2);
```

DELETE

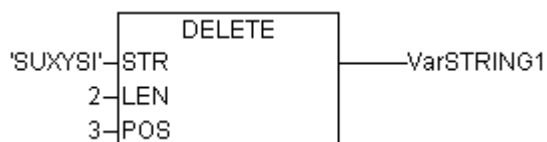
Funkcja DELETE usuwa od określonego miejsca string częściowy z danego stringu.

Wejście STR jest typu STRING, LEN i POS typu INT, wartość zwracana funkcji typu STRING.

DELETE(STR, LEN, POS) oznacza: Usuń LEN znaków z STR, począwszy od POS.

Przykład w IL:

```
LD    'SUXYSI'
DELETE 2,3
ST    Var1 (* wynik to 'SUSI' *)
```

Przykład w FBD:**Przykład w ST:**

```
Var1 := DELETE ('SUXYSI', 2, 3);
```

REPLACE

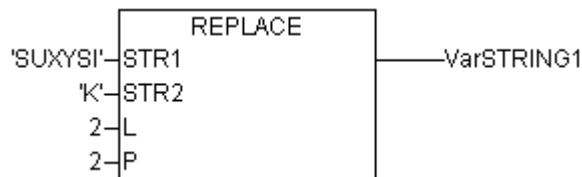
Funkcja REPLACE zastępuje string częściowy danego stringu innym.

Wejścia STR1 i STR2 są typu STRING, LEN i POS typu INT, wartość zwracana funkcji jest typu STRING.

REPLACE(STR1, STR2, L, P) oznacza: Zastąp L znaków z STR1 przez STR2 poczynając od znaku na miejscu P.

Przykład w IL:

```
LD      'SUXYSI'
REPLACE 'K',2,2
ST      VarSTRING1 (* wynik to 'SKYSI' *)
```

Przykład w FBD:**Przykład w ST:**

```
VarSTRING1 := REPLACE ('SUXYSI','K',2,2);
```

FIND

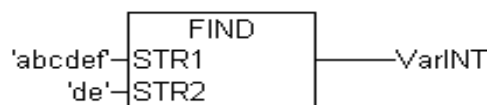
Funkcja FIND wyszukuje w stringu string częściowy.

Wejścia STR1 i STR2 są typu STRING, wartość zwracana funkcji jest typu INT.

FIND(STR1, STR2) oznacza: znajdź pozycję pierwszego znaku pierwszego wystąpienia STR2 w STR1. Jeśli STR2 nie występuje w STR1, obowiązuje OUT := 0.

Przykład w IL:

```
LD      'abcdef'
FIND   'de'
ST      VarINT1 (* wynik to '4' *)
```

Przykład w FBD:**Przykład w ST:**

```
VarINT1 := FIND ('abcdef','de');
```

13.17.2 Bistabilne bloki funkcji...**SR**

Ustawianie bistabilnego bloku funkcji jako dominującego:

$Q1 = SR (SET1, RESET)$ oznacza:

$$Q1 = (NOT\ RESET\ AND\ Q1)\ OR\ SET1$$

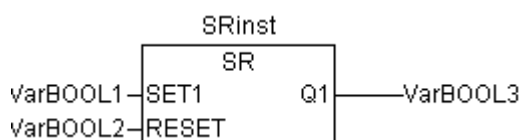
Wejścia SET1 i RESET oraz wyjście Q1 są typu BOOL.

Przykład deklaracji:

```
SRInst : SR;
```

Przykład w IL:

```
CAL SRInst (SET1 := VarBOOL1, RESET := VarBOOL2)
LD SRInst.Q1
ST VarBOOL3
```

Przykład w FBD:**Przykład w ST:**

```
SRInst(SET1:= VarBOOL1 , RESET:=VarBOOL2 );
VarBOOL3 := SRInst.Q1 ;
```

RS

Zerowanie bistabilnego bloku funkcji:

Q1 = RS (SET, RESET1) oznacza:

$$Q1 = \text{NOT RESET1 AND } (Q1 \text{ OR SET})$$

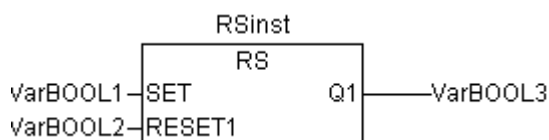
Wejścia SET i RESET1 oraz wyjście Q1 są typu BOOL.

Przykład deklaracji:

```
RSInst : RS ;
```

Przykład w IL:

```
CAL RSInst(SET:= VarBOOL1,RESET1:=VarBOOL2)
LD RSInst.Q1
ST VarBOOL3
```

Przykład w FBD:**Przykład w ST:**

```
RSInst(SET:= VarBOOL1 , RESET1:=VarBOOL2 );
VarBOOL3 := RSInst.Q1 ;
```

SEMA

Semafor oprogramowania (z możliwością przerwania).

BUSY = SEMA(CLAIM, RELEASE) oznacza:

```
BUSY := X;
IF CLAIM THEN X:=TRUE;
ELSIF RELEASE THEN BUSY := FALSE; X:= FALSE;
END_IF
```

X to wewnętrzna zmienna BOOL, zainicjalizowana za pomocą FALSE. Wejścia CLAIM i RELEASE oraz wyjście BUSY są typu BOOL.

Jeśli wywołano SEMA i BUSY jest TRUE, oznacza to, że SEMA zostało już wcześniej zajęte (SEMA zostało wywołane za pomocą CLAIM = TRUE). Jeśli BUSY jest FALSE, SEMA nie zostało jeszcze wywołane, lub zostało zwolnione (wywołanie za pomocą RELEASE = TRUE).

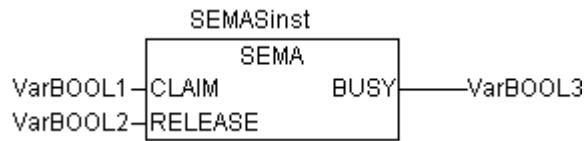
Przykład deklaracji:

```
SEMAInst : SEMA;
```

Przykład w IL:

```
CAL SEMAInst(CLAIM:=VarBOOL1,RELEASE:=VarBOOL2)
LD SEMAInst.BUSY
```

```
ST VarBOOL3
```

Przykład w FBD:**Przykład w ST:**

```
SEMAInst (CLAIM:= VarBOOL1 , RELEASE:=VarBOOL2 );
VarBOOL3 := SEMAInst.BUSY;
```

13.17.3 Rozpoznawanie krawędzi...**R_TRIG**

Blok funkcji R_TRIG wykrywa krawędź wzrastającą.

```
FUNCTION_BLOCK R_TRIG
VAR_INPUT
  CLK : BOOL;
END_VAR
VAR_OUTPUT
  Q : BOOL;
END_VAR
VAR
  M : BOOL := FALSE;
END_VAR
  Q := CLK AND NOT M;
  M := CLK;
```

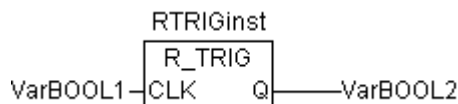
Tak długo, jak zmienna wejściowa CLK daje FALSE, komunikat Q i zmienna pomocnicza M będą FALSE. Jeśli CLK daje TRUE, Q najpierw będzie dawać TRUE, a następnie M będzie ustawione na TRUE. Tzn.: przy każdym kolejnym wywołaniu instancji bloku funkcji Q będzie znowu dawać FALSE, aż CLK będzie miało opadającą i znowu wzrastającą krawędź.

Przykład deklaracji:

```
RTRIGInst : R_TRIG ;
```

Przykład w IL:

```
CAL RTRIGInst (CLK := VarBOOL1)
LD RTRIGInst.Q
ST VarBOOL2
```

Przykład w FBD:**Przykład w ST:**

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

F_TRIG

Blok funkcji F_TRIG wykrywa krawędź opadającą.

```
FUNCTION_BLOCK F_TRIG
VAR_INPUT
  CLK: BOOL;
END_VAR
```

```

VAR OUTPUT
  Q: BOOL;
END_VAR

VAR
  M: BOOL := FALSE;
END_VAR

  Q := NOT CLK AND NOT M;
  M := NOT CLK;

```

Tak długo jak zmienna wejściowa CLK daje TRUE, dane wyjściowe będą Q zaś zmienna pomocnicza M będzie FALSE. Jeśli CLK daje FALSE, Q będzie dawać TRUE, a następnie M będzie ustawione na TRUE. Tzn.: przy każdym kolejnym wywołaniu instancji bloku funkcji Q będzie znowu dawać FALSE, aż CLK będzie miało wzrastającą i znowu opadającą krawędź.

Przykład deklaracji:

```
FTRIGInst : F_TRIG ;
```

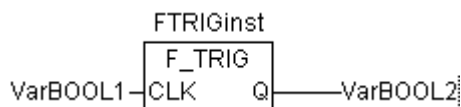
Przykład w IL:

```

CAL FTRIGInst (CLK := VarBOOL1)
LD FTRIGInst.Q
ST VarBOOL2

```

Przykład w FBD:



Przykład w ST:

```

FTRIGInst (CLK:= VarBOOL1);
VarBOOL2 := FTRIGInst.Q;

```

13.17.4 Licznik...

CTU

Licznik w górę bloku funkcji:

Wejścia CU i RESET oraz wyjście Q są typu BOOL, wejście PV i wyjście CV są typu WORD.

Jeśli RESET jest TRUE, zmienna licznika CV jest inicjalizowana za pomocą 0. Jeśli CU ma wzrastającą krawędź FALSE na TRUE, CV zostaje zwiększone o 1.

Q daje TRUE, jeśli CV jest większe lub równe górnej granicy PV.

Przykład deklaracji:

```
CTUInst : CTU ;
```

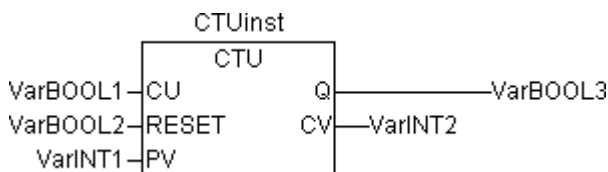
Przykład w IL:

```

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

```

Przykład w FBD:



Przykład w ST:

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

CTD

Licznik w dół bloku funkcji:

Wejścia CD i LOAD oraz wyjście Q są typu BOOL, wejście PV i wyjście CV są typu WORD.

Jeśli LOAD jest TRUE, zmienna licznika CV jest inicjalizowana za pomocą górnej granicy PV. Jeśli CD ma wzrastającą krawędź FALSE na TRUE, CV zostaje każdorazowo zmniejszone o 1, tak długo jak CV jest większe od 0 (jeśli nie jest powodowany niedomiar).

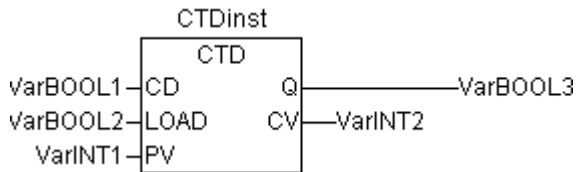
Q daje TRUE, jeśli CV jest równe 0.

Przykład deklaracji:

```
CTDInst : CTD ;
```

Przykład w IL:

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

Przykład w FBD:**Przykład w ST:**

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

CTUD

Licznik w górę/w dół bloku funkcji:

Wejścia CU, CD, RESET, LOAD oraz wyjścia QU i QD są typu BOOL, PV i CV są typu WORD.

Jeśli działa RESET, zmienna licznika CV jest inicjalizowana za pomocą 0. Jeśli działa LOAD, CV jest inicjalizowane za pomocą PV.

Jeśli CU ma wzrastającą krawędź FALSE na TRUE, CV zostaje zwiększone o 1. Jeśli CD ma wzrastającą krawędź FALSE na TRUE, CV zostaje każdorazowo zmniejszone o 1, tak długo jak CV nie powoduje niedomiaru.

QU daje TRUE, jeśli CV jest większe lub równe PV.

QD daje TRUE, jeśli CV jest równe 0.

Przykład deklaracji:

```
CTUDInst : CUTD ;
```

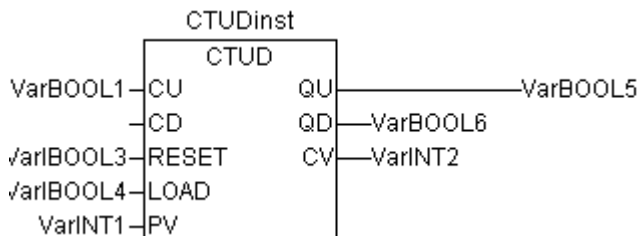
Przykład w IL:

```
CAL CTUDInst(CU:=VarBOOL2, RESET:=VarBOOL3, LOAD:=VarBOOL4, PV:=VarINT1)
LD CTUDInst.Q
ST VarBOOL5
```

```

LD   CTUDInst.QD
ST   VarBOOL5
LD   CTUInst.CV
ST   VarINT2

```

Przykład w FBD:**Przykład w ST:**

```

CTUDInst(CU := VarBOOL1, CU:= VarBOOL2, RESET := VarBOOL3, LOAD:=VarBOOL4 , PV:=
VarINT1);
VarBOOL5 := CTUDInst.QU ;
VarBOOL6 := CTUDInst.QD ;
VarINT2 := CTUDInst.CV;

```

13.17.5 Timer...**TP**

Blok funkcji TP to pulsator.

TP(IN, PT, Q, ET) oznacza:

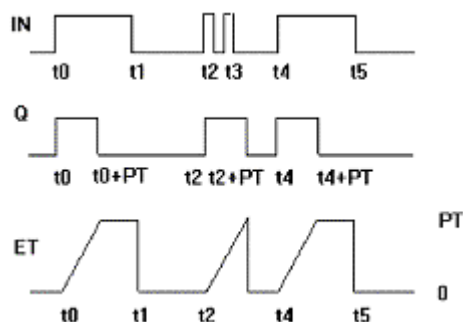
IN i PT są zmiennymi wejściowymi typu BOOL lub TIME. Q i ET są zmiennymi wyjściowymi typu BOOL lub TIME. Jeśli IN jest FALSE, wydawane jest FALSE lub 0.

Jeśli IN jest TRUE, w ET czas jest naliczany w milisekundach do chwili, gdy wartość będzie równa wartości w PT, a następnie pozostaje stała.

Q jest TRUE po zmianie IN na TRUE i ET jest jeszcze mniejsze lub równe PT. W innym przypadku jest FALSE.

Q daje tym samym sygnał dla okresu podanego w PT.

Ilustracja graficzna przebiegu czasu TP:



Przykład deklaracji:

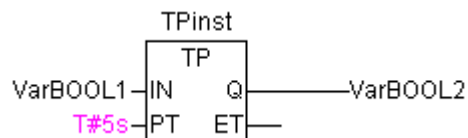
```
TPInst : TP ;
```

Przykład w IL:

```

CAL TPInst(IN := VarBOOL1, PT := T#5s)
LD   TPInst.Q
ST   VarBOOL2

```

Przykład w FBD:**Przykład w ST:**

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

TON

Blok funkcji Timer on-delay przeprowadza opóźnienie załączania.

TON(IN, PT, Q, ET) oznacza:

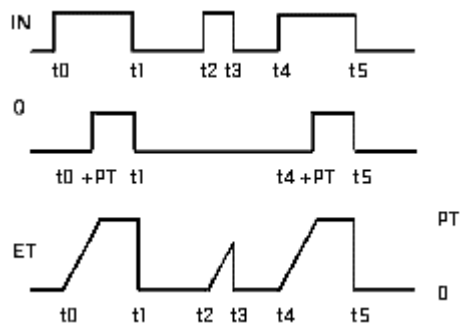
IN i PT są zmiennymi wejściowymi typu BOOL lub TIME. Q i ET są zmiennymi wyjściowymi typu BOOL lub TIME. Jeśli IN jest FALSE, wydawane jest FALSE lub 0.

Jeśli IN jest TRUE, w ET czas jest naliczany w milisekundach do chwili, gdy wartość będzie równa wartości w PT, a następnie pozostaje równa.

Q jest TRUE, jeśli IN TRUE i ET jest równe PT. W innym przypadku jest FALSE.

Q ma tym samym wzrastającą krawędź, jeśli czas w PT podany w milisekundach upłynął.

Ilustracja graficzna zachowania czasu TON:

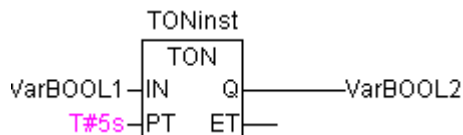


Przykład deklaracji:

```
TONInst : TON ;
```

Przykład w IL:

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

Przykład w FBD:**Przykład w ST:**

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

TOF

Blok funkcji Timer off-delay przeprowadza opóźnienie wyłączenia.

TOF(IN, PT, Q, ET) oznacza:

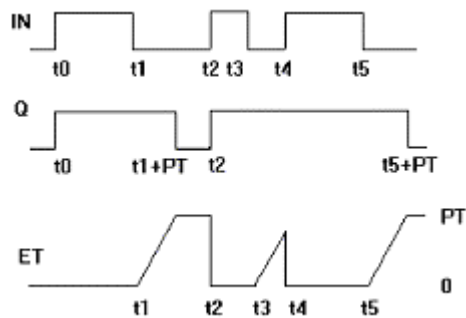
IN i PT są zmiennymi wejściowymi typu BOOL lub TIME. Q i ET są zmiennymi wyjściowymi typu BOOL lub TIME. Jeśli IN jest TRUE, wydawane jest TRUE lub 0.

Jeśli IN jest FALSE, w ET czas jest naliczany w milisekundach do chwili, gdy wartość będzie równa wartości w PT, a następnie pozostaje równa.

Q jest FALSE jeśli IN jest FALSE i ET jest równe PT. W innym przypadku jest TRUE.

Q ma przez to krawędź opadającą, po upływie czasu podanego w PT w milisekundach.

Graficzna ilustracja zachowania czasowego TOF:



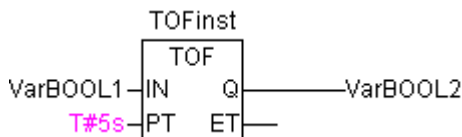
Przykład deklaracji:

```
TOFInst : TOF ;
```

Przykład w IL:

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

Przykład w FBD:

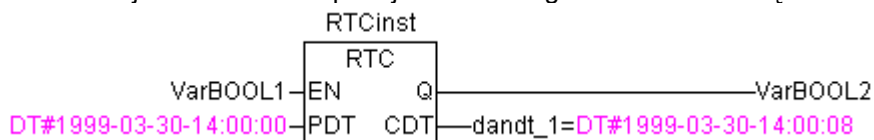


Przykład w ST:

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

RTC

Blok funkcji Runtime Clock podaje od zadanego czasu startu datę i czas w sposób ciągły.



RTC(EN, PDT, Q, CDT) oznacza:

EN i PDT są zmiennymi wejściowymi typu BOOL lub DATE_AND_TIME. Q i CDT to zmienne wyjściowe typu BOOL lub DATE_AND_TIME. Jeśli EN jest FALSE, dane wyjściowe Q i CDT są FALSE lub DT#1970-01-01-00:00:00.

Jeśli EN zmienia się na TRUE, ustawiony zostaje czas PDT, jest liczony w sekundach i wydawany za pomocą CDT (patrz przykład na ilustracji powyżej) jeśli EN jest TRUE. Jeśli EN zostaje ustawione ponownie na FALSE, CDT przeskakuje na wartość wyjściową DT#1970-01-01-00:00:00. Należy zwrócić uwagę, że czas z PDT jest ustawiany za pomocą krawędzi wzrastającej w EN.

13.18 Biblioteka Util.lib

Biblioteka ta zawiera dodatkowy zbiór różnych modułów, które mogą być stosowane do konwersji BCD, funkcji bit/bajt, matematycznych funkcji pomocniczych, jako regulatory, generatory sygnału, manipulatory funkcji i do przetwarzania wartości analogowych.

Ponieważ niektóre z funkcji i modułów funkcji zawierają zmienne REAL, które nie są obsługiwane przez niektóre systemy czasu przebiegu, istnieje dodatkowa biblioteka UTIL_NO_REAL, w której brak jest tych modułów.

13.18.1 Konwersja BCD...

Bajt w formacie BCD zawiera wartości całkowite z zakresu 0–99. Przy tym dla każdego miejsca dziesiętnego stosowane są cztery bity. Miejsce dziesiętne jest przy tym zapisywane w bitach 4–7. Tym samym format BCD jest podobny do ilustracji szesnastkowej, tylko w bajcie BCD mogą być zapisywane wartości z zakresu 0–99, jednak bajt szesnastkowy wystarcza od 0 do FF.

Przykład: przekształcanie liczby całkowitej 51 w format BCD. 5 to binarnie 0101, 1 to binarnie 0001, przez to bajt BCD to 01010001, co odpowiada wartości \$51=81.

BCD_TO_INT

Funkcja ta przekształca bajt w formacie BCD na wartość INT:

Wartość wejściowa funkcji jest typu BYTE zaś dane wyjściowe typu INT.

Jeśli ma być przesłany bajt nie odpowiadający formatowi BCD, informacja wyjściowa wynosi -1.

Przykłady w ST:

```
i:=BCD_TO_INT(73); (* wynik to 49 *)
k:=BCD_TO_INT(151); (* wynik to 97 *)
l:=BCD_TO_INT(15); (* dane wyjściowe -1, ponieważ nie format BCD *)
```

INT_TO_BCD

Funkcja ta przekształca wartość INTEGER na bajt w formacie BCD:

Wartość wejściowa funkcji jest typu INT zaś dane wyjściowe typu BYTE.

Jeśli ma być przesłana wartość INTEGER, której nie można przekształcić na bajt BCD, na wyjściu pojawia się 255.

Przykłady w ST:

```
i:=INT_TO_BCD(49); (* wynik to 73 *)
k:=BCD_TO_INT(97); (* wynik to 151 *)
l:=BCD_TO_INT(100); (* błąd! Dane wyjściowe: 255 *)
```

13.18.2 Funkcje bit/bajt...

EXTRACT

Wejścia tej funkcji to DWORD X, oraz BYTE N. Wydawana jest wartość BOOL z zawartością N-tego bitu wpisu X, przy czym liczenie rozpoczyna się od bitu zerowego.

Przykłady w ST:

```
FLAG:=EXTRACT(X:=81, N:=4);
(* wynik : TRUE, ponieważ 81 to binarnie 1010001, 4 bit to 1 *)
FLAG:=EXTRACT(X:=33, N:=0);
(* wynik : TRUE, ponieważ 33 to binarnie 100001, zerowy bit to 1 *)
```

PACK

Funkcja ta wymaga osiem bitów wejściowych B0, B1, ..., B7 typu BOOL, czyli zwrócenia BYTE. Z tę funkcją jest ściśle powiązany blok funkcji UNPACK.

PUTBIT

Wpis tej funkcji składa się z DWORD X, BYTE N i wartości Boolean B.

PUTBIT ustawia N-ty bit X na wartość B, przy czym liczenie rozpoczyna się od bitu zerowego.

Przykład w ST:

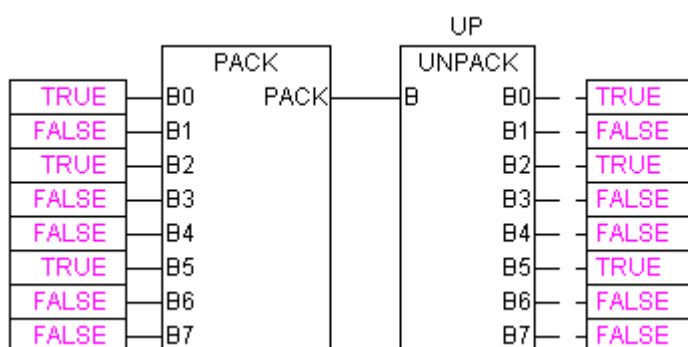
```
var1:=38; (* binarnie 100110 *)
var2:=PUTBIT(A,4,TRUE); (* wynik: 54 = 2#110110 *)
var3:=PUTBIT(A,1,FALSE); (* wynik: 36 = 2#100100 *)
```

UNPACK

Blok funkcji UNPACK przekształca wpis B typu BYTE na 8 zmiennych wyjściowych B0, ..., B7 typu BOOL i stanowi przez to przeciwieństwo funkcji PACK.

Przykład w CFC:

Dane wyjściowe:

**13.18.3 Matematyczne funkcje pomocnicze...****DERIVATIVE**

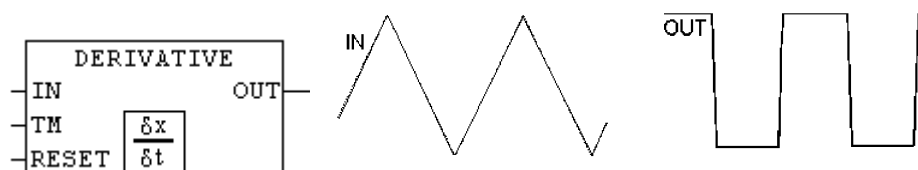
Ten blok funkcji określa w przybliżeniu lokalną pochodną czasu.

Za pomocą IN wartość funkcji jest przesyłana jako zmienna REAL. TM zawiera miniony czas w ms w DWORD i przez wejście RESET typu BOOL można ponownie uruchomić blok funkcji za pomocą przekazania wartości TRUE.

Wyjście OUT jest typu REAL.

W celu osiągnięcia możliwie jak najlepszego wyniku, DERIVATIVE przybliża się za pomocą ostatnich czterech wartości, by na możliwie jak najniższym poziomie utrzymać błędy spowodowane przez niedokładności (np. pomiaru) w zmiennych wejściowych.

Moduł w CFC:



INTEGRAL

Ten blok funkcji określa w przybliżeniu całkę funkcji na podstawie czasu.

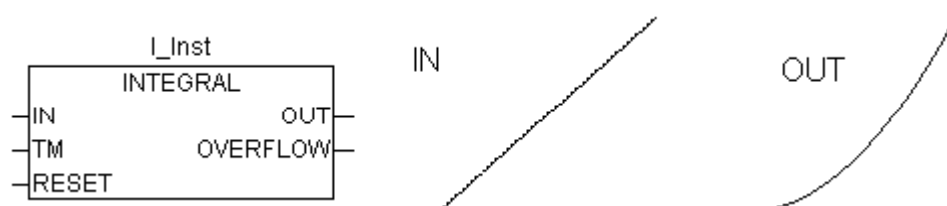
Analogicznie do DERIVATIVE za pomocą IN przesyłana jest wartość funkcji jako zmienna REAL. TM zawiera miniony czas w ms w DWORD i przez wejście RESET typu BOOL można ponownie uruchomić blok funkcji za pomocą przekazania wartości TRUE.

Wyjście OUT jest typu REAL.

Całka przybliżana jest za pomocą funkcji schodkowej.

Jeśli wartość całki osiąga koniec zakresu wartości zmiennej REAL (ok. $\pm 10^{38}$), zmienna wyjściowa OVERFLOW typu BOOL jest ustawiana na TRUE i moduł jest blokowany do chwili ponownej inicjalizacji za pomocą RESET.

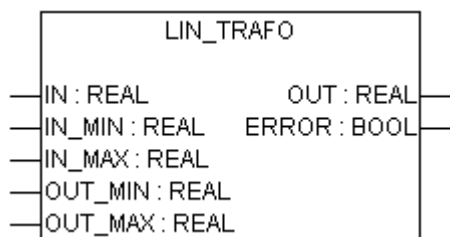
Moduł w CFC: Przykład: Całka na podstawie funkcji linearnej:



LIN_TRAFO

Ten blok funkcji przekształca wartość real z przedziału zdefiniowanego za pomocą dolnej i górnej granicy w odpowiednią wartość real będącą w odpowiedniej proporcji również znajdującą się w przedziale zdefiniowanym za pomocą dolnej i górnej granicy. Podstawę stanowi następujące równanie:

$$(IN - IN_MIN) : (IN_MAX - IN_MIN) = (OUT - OUT_MIN) : (OUT_MAX - OUT_MIN)$$



Zmienne wejściowe:

Zmienna	Typ danych	Opis
IN	REAL	Wartość wejściowa
IN_MIN	REAL	Dolna granica wartości wejściowych
IN_MAX	REAL	Górna granica wartości wejściowych
OUT_MIN	REAL	Dolna granica wartości wyjściowych
OUT_MAX	REAL	Górna granica wartości wyjściowych

Zmienne wyjściowe

Zmienna	Typ danych	Opis
OUT	REAL	Wartość wyjściowa
ERROR	BOOL	Wydawanie błędu: TRUE, jeśli $IN_MIN = IN_MAX$, lub jeśli IN znajduje się poza podanym zakresem wejściowym

Przykład zastosowania:

Czujnik temperatury daje na wejściu IN wartości w woltach. W danych temperatury należy je przetransformować na stopnie Celsjusza i wydać w OUT. Wejściowy zakres (w woltach) jest definiowany za pomocą IN_MIN=0 i IN_MAX=10. Wyjściowy zakres (stopni Celsjusza) jest definiowany za pomocą OUT_MIN=-20 i OUT_MAX=40. Odpowiednio do tego dla wejścia 5 Volt jest wydawana temperatura 10 stopni Celsjusza.

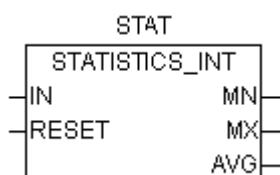
STATISTICS_INT

Ten blok funkcji oblicza niektóre statystyczne wartości standardowe:

Wpis IN jest typu INT. Jeśli wpis Boolean RESET staje się TRUE, wszystkie wartości są inicjalizowane ponownie.

Dane wyjściowe MN zawierają wartość minimalną, zaś MX wartość maksymalną IN. AVG opisuje przeciętną, czyli spodziewaną wartość IN. Wszystkie trzy dane wyjściowe są typu INT.

Moduł w CFC:



STATISTICS_REAL

Ten blok funkcji odpowiada STATISTICS_INT, lecz wpis IN tak samo jak dane wyjściowe MN, MX, AVG jest typu REAL.

VARIANCE

VARIANCE oblicza wariancję wpisanych wartości.

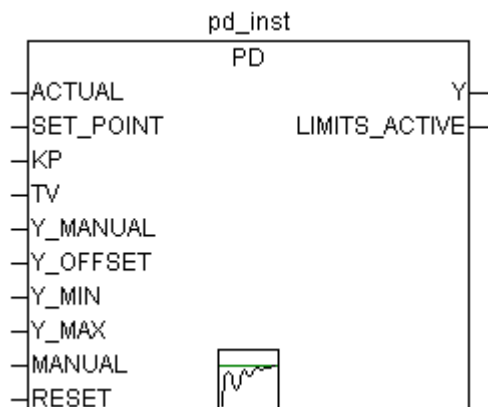
Wpis IN jest typu REAL, RESET typu BOOL, zaś informacja wyjściowa OUT jest typu REAL.

Na podstawie wpisanych wartości ten blok funkcji oblicza wariancję statystyczną. Za pomocą RESET = TRUE można ponownie inicjalizować VARIANCE. Standardowe odchylenie można w prosty sposób obliczyć z pierwiastka kwadratowego VARIANCE.

13.18.4 Regulator...

PD

Biblioteka util.lib oddaje do dyspozycji niżej przedstawiony blok funkcji regulatora PD:



Wejścia modułu:

Zmienna	Typ danych	Opis
ACTUAL	REAL	Wartość rzeczywista wielkości regulowanej
SET_POINT	REAL	Wartość zadana, wielkość przewodnia
KP	REAL	Współczynnik proporcjonalności, współczynnik wzmocnienia części P
TV	REAL	Czas różniczkowania, współczynnik wzmocnienia części D; dane w sekundach, np. „0.5” dla 500 ms
Y_MANUAL	REAL	Ręczna wartość wydawana; jest wydawana, jeśli MANUAL = TRUE
Y_OFFSET	REAL	Offset wielkości nastawczej Y
Y_MIN, Y_MAX	REAL	Dolna lub górna granica wielkości nastawczej Y. W przypadku przekroczenia tej granicy w dół lub w górę, wyjście LIMITS_ACTIVE jest ustawiane na TRUE i Y jest utrzymywane w zakresie granic. Ta kontrola jest włączona, tylko jeśli Y_MIN < Y_MAX.
MANUAL	BOOL	Włącza ręczne zadawanie wielkości nastawczej za pomocą Y_MANUAL (TRUE) lub wyłącza (FALSE)
RESET	BOOL	TRUE ponownie inicjalizuje regulator. W trakcie inicjalizacji Y = Y_OFFSET

Wyjścia modułu:

Zmienna	Typ danych	Opis
Y	REAL	Wielkość nastawcza obliczana przez moduł (patrz poniżej)
LIMITS_ACTIVE	BOOL	Wskazuje za pomocą TRUE przekroczenie przez Y dozwolonego przedziału (Y_MIN, Y_MAX).

Y_OFFSET, Y_MIN i Y_MAX służą do transformacji wielkości nastawczej do zadanego zakresu.

Za pomocą MANUAL można przełączyć na tryb ręczny zaś za pomocą RESET można ponownie zainicjalizować regulator.

W trybie normalnym (MANUAL = RESET = LIMITS_ACTIVE = FALSE) regulator oblicza błąd regulacji e jako różnicę SET_POINT – ACTUAL, tworzy jego pochodną po czasie $\delta e / \delta t$ i wewnątrz zapisuje te wartości.

Dane wyjściowe, wielkość nastawcza (Y) jest obliczana w następujący sposób:

$$Y = KP \cdot (\Delta + TV \delta/\delta t) + Y_OFFSET \text{ przy czym } \Delta = SET_POINT - ACTUAL$$

Oprócz części P również aktualna zmiana błędu regulacji (część D) ma wpływ na wielkość nastawczą.

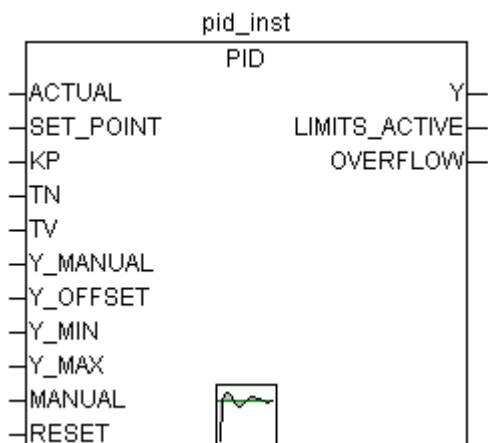
Dodatkowo Y jest ograniczane do dozwolonego zakresu od Y_MIN do Y_MAX. W przypadku przekroczenia przez Y tego zakresu LIMITS_ACTIVE staje się TRUE. Jeśli wymagane jest ograniczenie wielkości nastawczej, Y_MIN i Y_MAX ustawia się na 0.

Jeśli MANUAL jest TRUE, wydawane jest Y_MANUAL.

Regulator P można wytworzyć w prosty sposób przez ustawienie TV na stałe na 0.

PID

Biblioteka util.lib oddaje do dyspozycji niżej przedstawiony blok funkcji regulatora PID:



W odróżnieniu od regulatora PD ten moduł funkcji zawiera kolejne wejście REAL TN dla czasu cofania w sekundach (np. „0.5” dla 500 ms).

Wejścia modułu:

Zmienna	Typ danych	Opis
ACTUAL	REAL	Wartość rzeczywista wielkości regulowanej
SET_POINT	REAL	Wartość zadana, wielkość przewodnia
KP	REAL	Współczynnik proporcjonalności, współczynnik wzmocnienia części P
TN	REAL	Czas cofania, odwrotny współczynnik wzmocnienia części I; dane w sekundach, np. „0.5” dla 500 ms
TV	REAL	Czas różniczkowania, współczynnik wzmocnienia części D; dane w sekundach
Y_MANUAL	REAL	Ręczna wartość wydawana; wydawana, jeśli MANUAL = TRUE
Y_OFFSET	REAL	Offset wielkości nastawczej Y
Y_MIN, Y_MAX	REAL	Dolna lub górna granica wielkości nastawczej Y. W przypadku przekroczenia tej granicy w dół lub w górę, wyjście LIMITS_ACTIVE jest ustawiane na TRUE i Y jest utrzymywane w zakresie granic. Ta kontrola jest włączona, tylko jeśli $Y_MIN < Y_MAX$.
MANUAL	BOOL	Włącza ręczne zadawanie wielkości nastawczej za pomocą Y_MANUAL (TRUE) lub wyłącza (FALSE)
RESET	BOOL	TRUE ponownie inicjalizuje regulator. W trakcie inicjalizacji $Y = Y_OFFSET$.

Wyjścia modułu:

Zmienna	Typ danych	Opis
Y	REAL	Wielkość nastawcza obliczana przez moduł (patrz poniżej)
LIMITS_ACTIVE	BOOL	Wskazuje za pomocą TRUE przekroczenie przez Y dozwolonego przedziału (Y_MIN, Y_MAX).
OVERFLOW	BOOL	Wskazuje za pomocą TRUE przepełnienie (patrz poniżej)

Y_OFFSET, Y_MIN i Y_MAX służą do przekształcenia wielkości nastawczej w zadany przedział.

Za pomocą MANUAL można dokonać przełączenia na tryb ręczny, zaś za pomocą RESET można ponownie zainicjalizować regulator.

W normalnym trybie (MANUAL = RESET = LIMITS_ACTIVE = FALSE) regulator oblicza błąd regulacji e jako różnicę SET_POINT – ACTUAL, tworzy jego pochodną po czasie $\delta e / \delta t$ i wewnętrznie zapisuje te wartości.

Dane wyjściowe, wielkość nastawcza (Y) zawiera w przeciwieństwie do regulatora PD dodatkową całkę i oblicza się to w następujący sposób:

$$Y = KP \cdot (\Delta + 1/TN \int e dt + TV \delta e / \delta t) + Y_OFFSET$$

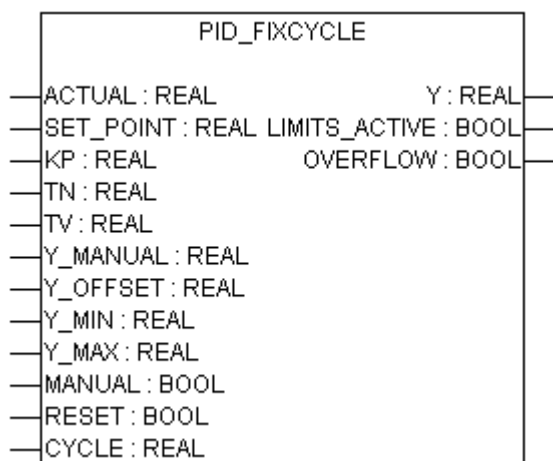
Oprócz części P również aktualna zmiana błędu regulacji (część D) i historia błędu regulacji (część I) mają wpływ na wielkość nastawczą.

Regulator PID można w łatwy sposób zamienić na **regulator PI** ustawiając TV=0.

Za pomocą dodatkowej części I może – w przypadku błędnej parametryzacji regulatora, jeśli całka błędu Δ jest zbyt duża – dojść do **przepięcenia**. Dla bezpieczeństwa istnieje wyjście Boolean OVERFLOW, które w tym przypadku przyjmuje wartość TRUE. Dzieje się tak tylko wtedy, gdy system regulacji jest niestabilny wskutek niewłaściwej parametryzacji. Równocześnie regulator wyłącza się i zostaje uruchomiony po ponownej inicjalizacji.

PID_FIXCYCLE

Biblioteka util.lib oddaje do dyspozycji niżej przedstawiony blok funkcji regulatora PID_FIXCYCLE:



Funkcje tego modułu odpowiadają regulatorowi PID, z tą różnicą, że nie jest automatycznie wewnętrznie mierzony czas cyklu, lecz jest on na stałe zadany w sekundach za pomocą wejścia CYCLE.

13.18.5 Generatory sygnału...

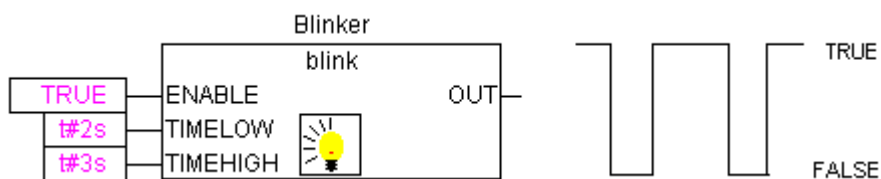
BLINK

Blok funkcji BLINK generuje sygnał pulsujący. Wpis składa się z ENABLE typu BOOL, oraz TIMELOW i TIMEHIGH typu TIME. Dane wyjściowe OUT są typu BOOL.

Jeśli ENABLE jest ustawiane na TRUE, BLINK rozpoczyna na zmianę ustawianie wydawania informacji dla okresu TIMEHIGH z TRUE, następnie dla okresu TIMELOW z FALSE.

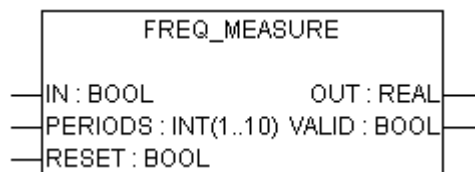
Po ponownym ustawieniu ENABLE na FALSE, informacja wyjściowa OUT nie jest już zmieniana, tzn. nie są już wytwarzane kolejne impulsy. Jeśli przez zerowanie ENABLE również OUT ma być FALSE, można to osiągnąć przez zastosowanie „OUT AND ENABLE” (wstawienie modułu AND z parametrem ENABLE) na wyjściu.

Przykład w CFC:



FREQ_MEASURE

Ten blok funkcji mierzy (przeciętną) częstotliwość (Hz) boolean sygnału wejściowego (częstotliwość = 1/okres pomiaru). Można przy tym podać, na przestrzeni ilu okresów pomiaru ma nastąpić uśrednienie. Okres pomiaru to okres między dwoma wzrastającymi krawędziami sygnału wejściowego.



Zmienne wejściowe:

Zmienna	Typ danych	Opis
IN	BOOL	Sygnał wejściowy
PERIODS	INT	Ilość okresów pomiaru, czyli okresów między krawędziami wzrastającymi, na podstawie których należy ustalić przeciętną częstotliwość sygnału wejściowego. Możliwe wartości: 1 do 10
RESET	BOOL	Reset wszystkich parametrów na 0

Zmienne wyjściowe:

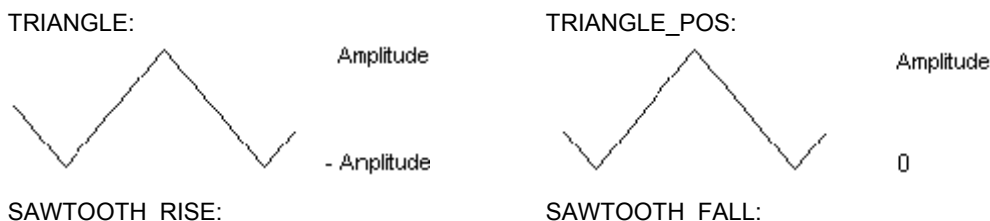
Zmienna	Typ danych	Opis
OUT	REAL	ustalona częstotliwość w [Hz]
VALID	BOOL	FALSE tak długo, jak nie został zakończony pomiar, lub jeśli okres > 3*OUT, co wskazuje na błąd odnośnie sygnału wejściowego.

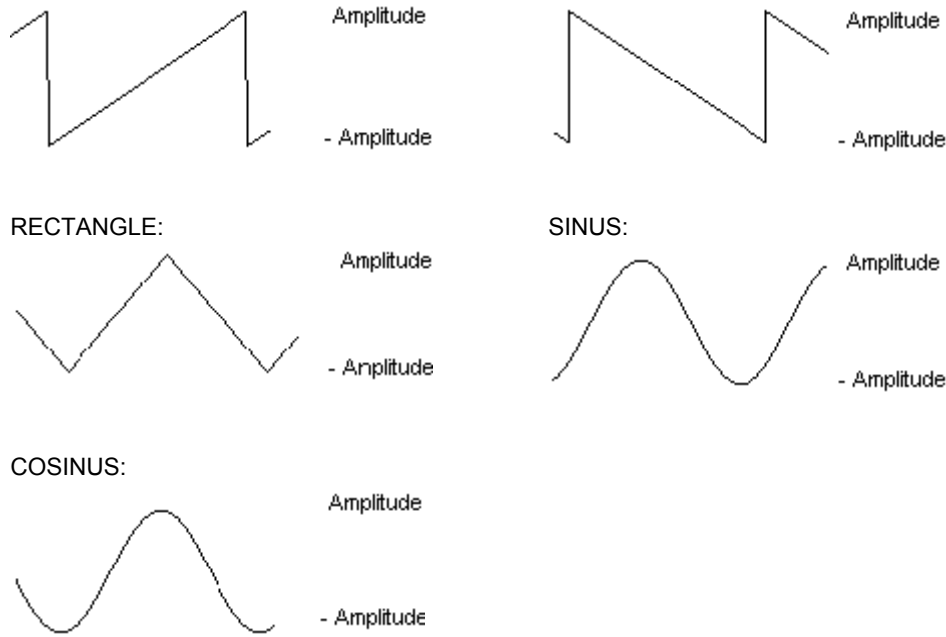
GEN

Generator funkcji generuje typowe funkcje okresowe:

Wpisy składają się z MODE w predefiniowanym typie wyliczenia GEN_MODE, BASE typu BOOL, PERIOD typu TIME, dwóch wartości INT: CYCLES i AMPLITUDE oraz wejścia RESET typu boolean. Wydawany jest INT z oznaczeniem OUT.

MODE opisuje funkcję, która ma zostać wytworzona, przy czym wartości enumeracji TRIANGLE i TRIANGLE_POS dają dwie funkcje trójkąta, SAWTOOTH_RISE wzrastające, SAWTOOTH_FALL opadający sygnał w kształcie zęba piły, RECTANGLE sygnał prostokątny zaś SINUS i COSINUS funkcje sinus i cosinus:





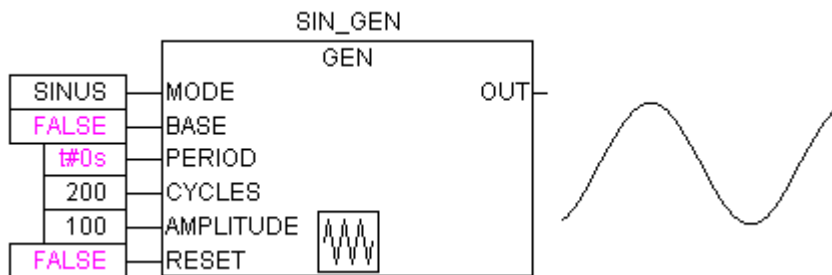
BASE podaje, czy czas trwania okresu rzeczywiście odnosi się do zadanego czasu (wtedy BASE=TRUE) lub do pewnej liczby cykli, czyli wywołań bloku funkcji (BASE=FALSE).

PERIOD lub CYCLES ustalają odpowiedni czas trwania okresu.

AMPLITUDE definiuje w prosty sposób amplitudę funkcji do wytworzenia.

Generator funkcji jest ponownie ustawiany na 0, gdy tylko RESET = TRUE.

Przykład w CFC:



13.18.6 Manipulatory funkcji...

CHARCURVE

Ten blok funkcji służy do odwzorowania wartości za pomocą funkcji częściowo liniowej:



Do IN typu INT są podawane wartości do manipulacji. BYTE N określa liczbę punktów ustalonych przez funkcję odwzorowania. Ta charakterystyka jest definiowana w ARRAY P[0..10] z P typu POINT, strukturze składającej się z dwóch wartości INT (X i Y).

Dane wyjściowe składają się z OUT typu INT, manipulowanej wartości i BYTE ERR wskazującego w razie potrzeby błędy.

Punkty P[0]..P[N-1] w ARRAY muszą być posortowane na podstawie swoich wartości X, w przeciwnym razie ERR uzyskuje wartość 1. Jeśli wpis IN nie znajduje się między P[0].X i P[N-1].X, ERR=2 i OUT otrzymuje odpowiednią wartość graniczną P[0].Y lub P[N-1].Y.

Jeśli N znajduje się poza wartościami dopuszczalnymi 2 i 11, ERR=4.

Przykład w ST:

Następnie należy w nagłówku zdefiniować ARRAY P:

```
VAR
```

```
...
```

```
KENNLINIE:CHARCURVE;
```

```
KL:ARRAY[0..10] OF POINT:=(X:=0,Y:=0),(X:=250,Y:=50),  
(X:=500,Y:=150),(X:=750,Y:=400),7((X:=1000,Y:=1000));
```

```
ZAEHLER:INT;
```

```
...
```

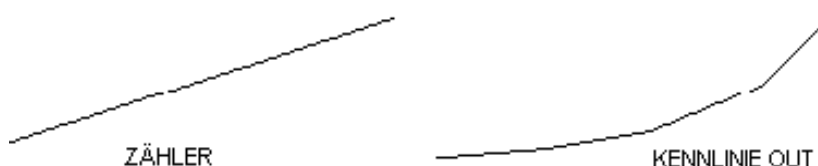
```
END_VAR
```

Następnie do CHARCURVE należy podać wartość rosnącą:

```
ZAEHLER:=ZAEHLER+10;
```

```
KENNLINIE(IN:=ZAEHLER,N:=5,P:=KL);
```

Ilustracja późniejszego śledzenia:



RAMP_INT

RAMP_INT służy do ograniczenia wzrostu lub spadku zasilanej funkcji:

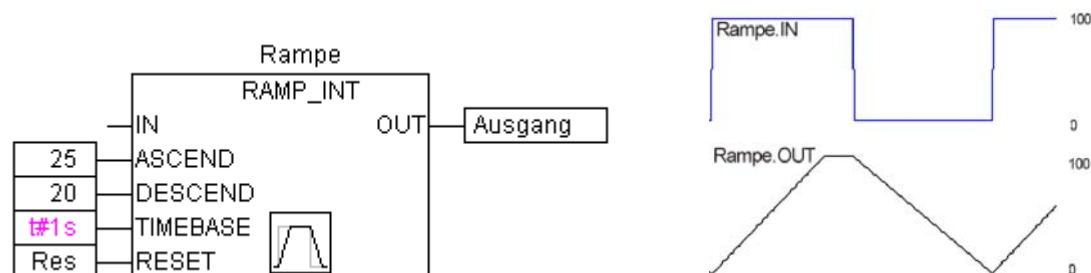
Wpis składa się po pierwsze z trzech wartości INT: IN, wpisu funkcji oraz ASCEND i DESCEND, maksymalnego wzrostu lub spadku w interwale czasu, ustalonego przez TIMEBASE typu TIME. Po ustawieniu RESET na TRUE, następuje ponowna inicjalizacja RAMP_INT.

Dane wyjściowe OUT typu INT zawierają ograniczoną wzrostem i spadkiem wartość funkcji.

Należy zwrócić uwagę, że problemy mogą wynikać, jeśli ustawione TIMEBASE jest mniejsze od czasu trwania cyklu przebiegu.

Po ustawieniu TIMEBASE na t#0s ograniczenie wzrostu lub spadku odnosi się do cyklu, czyli wywołania bloku funkcji.

Przykład w CFC:



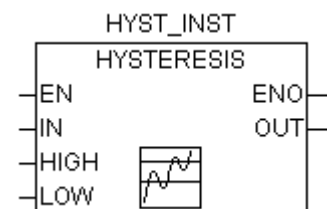
RAMP_REAL

RAMP_REAL działa dokładnie jak RAMP_INT, jedynie wejścia IN, ASCEND, DESCEND oraz wyjście OUT są typu REAL.

13.18.7 Przetwarzanie wartości analogowych...

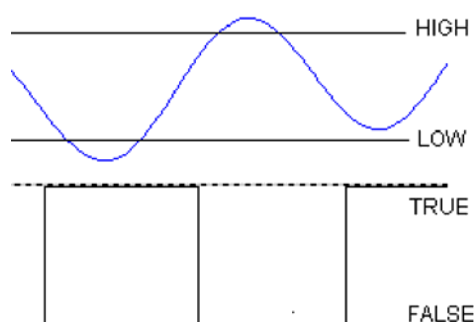
HYSTERESIS

Wejście tego bloku funkcji składa się z trzech wartości INT – IN, HIGH i LOW. Dane wyjściowe OUT są typu BOOL.



Jeśli IN przekracza wartość graniczną LOW, OUT staje się TRUE. Jeśli IN przekracza górną granicę HIGH, wydawane jest FALSE.

Ilustracja na przykładzie:



LIMITALARM

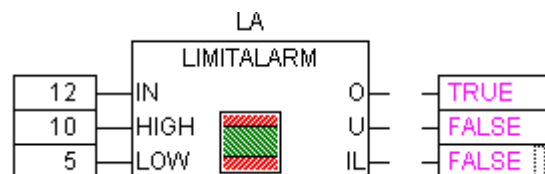
Ten blok funkcji podaje, czy wartość wejściowa znajduje się w zadanym interwale i w razie potrzeby, która granica została przekroczona.

Wartości wejściowe IN, HIGH i LOW są odpowiednio typu INT, informacja wyjściowa O, U i IL typu BOOL.

Jeśli IN przekracza górną granicę HIGH, O staje się TRUE, w przypadku przekroczenia w dół LOW staje się U TRUE. Jeśli natomiast IN leży między LOW i HIGH, IL staje się TRUE.

Przykład w CFC:

Wynik:



13.19 Biblioteka AnalyzationNew.lib

Biblioteka ta zawiera moduły analizy wyrażeń. Jeśli wyrażenie złożone ma wartość łączną FALSE, można ustalić te z jego składników, które przyczyniły się do tego wyniku. W edytorze AS flaga SFCErrAnalysisTable stosuje te funkcje pośrednio do analizy wyrażeń przejścia.

Przykład wyrażenia:

b OR NOT(y < x) OR NOT (NOT d AND e)

Funkcje:

Dla wszystkich modułów wspólne są następujące zmienne:

InputExpr: BOOL, wyrażenie do analizy

DoAnalyze: BOOL, TRUE uruchamia analizę

ExpResult: BOOL, aktualna wartość wyrażenia

Różny jest wynik analizy:

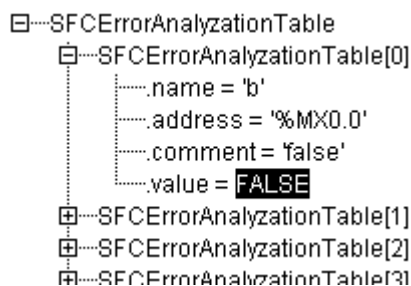
AnalyseExpression podaje w stringu składniki wyrażenia, które przyczyniły się do łącznej wartości FALSE. Stosowana przy tym jest funkcja **AppendErrorString**, przy czym w stringu wyjściowym poszczególne składniki są rozdzielone za pomocą znaku „|”.

OutString: STRING, wynik analizy, uszeregowanie wykorzystywanych składników wyrażenia (np.: $y < x \mid d$)

AnalyseExpressionTable zapisuje w tablicę składniki wyrażenia, które przyczyniły się do łącznej wartości FALSE, przy czym dla każdego składnika w **Struktur ExpressionResult** jest podawana następująca informacja: Nazwa (name), adres (address), komentarz (comment), aktualna wartość (value).

OutTable: ARRAY [0..15] OF ExpressionResult;

np.:



AnalyseExpressionCombined zawiera funkcjonalności **AnalyseExpression** oraz **AnalyseExpressionTable**

13.20 Biblioteki systemowe CoDeSys

Wskazówka

Które biblioteki systemowe mogą być stosowane programie sterującym, zależy od stosowanego systemu docelowego. Patrz dokument SysLibs_Ueberblick.pdf.

Załącznik F Przegląd: Operatory i moduły biblioteki

Poniższa tabela ukazuje przegląd **operatorów i modułów biblioteki**, dostępnych w CoDeSys lub bibliotekach Standard.lib i Util.lib. Przedstawiona została pisownia dla edytorów tekstu ST i IL. Do IL odnosi się ponadto lista dostępnych modyfikatorów.

Należy zwrócić uwagę na kolumnę „Operator IL”: wyświetlany jest tylko wiersz, w którym jest stosowany operator. Warunek stanowi wczytanie (pierwszego) potrzebnego argumentu (np. LD in) w poprzedzającym wierszu.

Kolumna „**Mod.AWL**” ukazuje możliwe modyfikatory w IL:

- C Instrukcja zostanie wykonana tylko wtedy, gdy wynik poprzedniego wyrażenia jest TRUE.
- N z JMPC, CALC, RETC: Instrukcja zostanie wykonana tylko wtedy, gdy wynik poprzedniego wyrażenia jest FALSE.
- N inaczej: Negacja argumentu (nie akumulatora).
- (Nawiasy ograniczają operator, dopiero po osiągnięciu nawiasu zamykającego jest wykonywana operacja poprzedzająca nawias.

Szczegółowy opis stosowania zawierają odpowiednie załączniki do operatorów w CoDeSys lub bibliotekach.

13.21 Operatory w CoDeSys

w ST	w IL	Mod.AWL	Znaczenie
'			Ograniczenie stringu (np. „string1”)
.. []			Tablica: wyświetlanie zakresu tablicy (np. ARRAY[0..3] OF INT)
:			Separator między argumentem i typem w deklaracji (np. var1 : INT;)
;			Zakończenie polecenia (np. a:=var1;)
^			Usuń referencje wskaźników (np. pointer1^)
	LD var1	N	Wczytaj wartość var1 do akumulatora
:=	ST var1	N	Zapisz aktualny wynik w miejscu argumentu var1
	S boolvar		Ustaw argument boolean boolvar dokładnie na TRUE, jeśli aktualny wynik jest TRUE.
	R boolvar		Ustawia argument boolean na FALSE, jeśli bieżący wynik jest FALSE
	JMP marke	CN	Przeskocz do znacznika
<Nazwa programu>	CAL prog1	CN	Wywołaj program prog1

Operatory w CoDeSys

<i>w ST</i>	<i>w IL</i>	<i>Mod.AWL</i>	<i>Znaczenie</i>
<Nazwa instancji>	CAL inst1	CN	Wywołaj instancję bloku funkcji inst1
<Nazwa funkcji>(vx, vy, ...)	<Nazwa funkcji> vx, vy	CN	Wywołaj funkcję i prześlij zmienne vx, vy
RETURN	RET	CN	Opuszcza moduł i powraca do elementu, który spowodował wywołanie.
	(Wartość po nawiasie jest uznawana za argument, poprzednia operacja zostaje zatrzymana do zamknięcia nawiasu.
)		Analizuje zawieszoną operację
AND	AND	N,(Bitowe AND
OR	OR	N,(Operator bitowy OR
XOR	XOR	N,(Operator bitowy specjalny OR
NOT	NOT		Operator bitowy NOT
+	ADD	(Dodawanie
-	SUB	(Odejmowanie
*	MUL	(Mnożenie
/	DIV	(Dzielenie
>	GT	(większe
>=	GE	(większe/równe
=	EQ	(równe
<>	NE	(nierówne
<=	LE	(mniejsze/równe
<	LT	(mniejsze
MOD (in)	MOD		Dzielenie modulo
INDEXOF(in)	INDEXOF		wewnętrzny indeks modułu in1; [INT]
SIZEOF(in)	SIZEOF		potrzebna liczba bajtów do danego typu danych w in
SHL (K,in)	SHL		bitowe przesunięcie argumentu w lewo o K
SHR (K,in)	SHR		bitowe przesunięcie argumentu w prawo o K
ROL (K,in)	ROL		bitowa rotacja argumentu w lewo o K
ROR (K,in)	ROR		bitowa rotacja argumentu w prawo o K
SEL (G,in0,in1)	SEL		wybór binarny pomiędzy 2 argumentami in0 (G jest FALSE) a in1 (G jest TRUE)

<i>w ST</i>	<i>w IL</i>	<i>Mod.AWL</i>	<i>Znaczenie</i>
MAX (in0,in1)	MAX		daje z dwóch wartości tę większą
MIN (in0,in1)	MIN		daje z dwóch wartości tę mniejszą
LIMIT (MIN,in,Max)	LIMIT		Ograniczenie zakresu wartości (in w przypadku przekroczenia jest ustawiane na Min lub Max)
MUX (K,in0,...in_n)	MUX		Wybór K-tej wartości z określonej liczby wartości (in0 do In_n)
ADR (in)	ADR		Adres argumentu w [DWORD]
ADRINST()	ADRINST()		Adres aktualnej instancji bloku funkcji.
BITADR(in)	BITADR		Offset bitowy argumentu w [DWORD]
BOOL_TO_<type>(in)	BOOL_TO_<type>		Konwersja typu argumentu boolean na inny typ podstawowy
<type>_TO_BOOL(in)	<type>_TO_BOOL		Konwersja typu argumentu na BOOL
INT_TO_<type>(in)	INT_TO_<type>		Konwersja typu argumentu INT na inny typ podstawowy
REAL_TO_<type>(in)	REAL_TO_<type>		Konwersja typu argumentu REAL na inny typ podstawowy
LREAL_TO_<type>(in)	LREAL_TO_<type>		Konwersja typu argumentu LREAL na inny typ podstawowy
TIME_TO_<type>(in)	TIME_TO_<type>		Konwersja typu argumentu TIME na inny typ podstawowy
TOD_TO_<type>(in)	TOD_TO_<type>		Konwersja typu argumentu TOD na inny typ podstawowy
DATE_TO_<type>(in)	DATE_TO_<type>		Konwersja typu argumentu na inny typ podstawowy
DT_TO_<type>(in)	DT_TO_<type>		Konwersja typu argumentu na inny typ podstawowy
STRING_TO_<type>(in)	STRING_TO_<type>		Konwersja typu argumentu na inny typ podstawowy, in musi zawierać prawidłową wartość typu docelowego
TRUNC(in)	TRUNC		Konwersja z REAL na INT
ABS (in)	ABS		Wartość bezwzględna argumentu
SQRT (in)	SQRT		Pierwiastek kwadratowy argumentu
LN (in)	LN		logarytm naturalny argumentu
LOG (in)	LOG		Logarytm argumentu o podstawie 10
EXP (in)	EXP		Funkcja wykładnicza argumentu
SIN (in)	SIN		Sinus argumentu
COS (in)	COS		Cosinus argumentu

Moduły biblioteki Standard.lib

<i>w ST</i>	<i>w IL</i>	<i>Mod.AWL</i>	<i>Znaczenie</i>
TAN (in)	TAN		Tangens argumentu
ASIN(in)	ASIN		Arcus sinus argumentu
ACOS(in)	ACOS		Arcus cosinus argumentu
ATAN(in)	ATAN		Arcus tangens argumentu
EXPT (in,expt)	EXPT expt		Potęgowanie argumentu do potęgi expt

13.22 Moduły biblioteki Standard.lib

<i>w ST</i>	<i>w IL</i>	<i>Znaczenie</i>
LEN (in)	LEN	Długość stringu argumentu
LEFT (str,size)	LEFT	Lewy string początkowy (wielkość size) stringu str
RIGHT (str,size)	RIGHT	Prawy string początkowy (wielkość size) stringu str
MID (str,size,pos)	MID	String częściowy wielkości size ze stringu str na pozycji pos
CONCAT ('str1','str2')	CONCAT 'str2'	Połączenie dwóch stringów
INSERT ('str1','str2',pos)	INSERT 'str2',p	Wstawianie stringu str1 w string str2 w pozycji pos
DELETE ('str1',len,pos)	DELETE len,pos	Usuwanie stringu częściowego długości len, począwszy od pozycji pos w str1
REPLACE ('str1','str2',len,pos)	REPLACE 'str2',len,pos	Zamiana stringu częściowego długości len, począwszy od pozycji pos w str1 przez str2
FIND ('str1','str2')	FIND 'str2'	Szukanie stringu częściowego str2 w str1
SR	SR	Ustawianie bistabilnego bloku funkcji jako dominującego
RS	RS	Zerowanie bistabilnego bloku funkcji
SEMA	SEMA	FB: Semafor oprogramowania (z możliwością przerwania)
R_TRIG	R_TRIG	FB: Rozpoznawanie krawędzi wzrastającej
F_TRIG	F_TRIG	FB: Rozpoznawanie krawędzi opadającej
CTU	CTU	FB: Licznik w górę
CTD	CTD	FB: Licznik w dół
CTUD	CTUD	FB: Licznik w górę i w dół
TP	TP	FB: Pulsator
TON	TON	FB: Opóźnienie załączenia
TOF	TOF	FB: Opóźnienie wyłączenia
RTC	RTC	FB: Zegar czasu przebiegu

13.23 Moduły biblioteki Util.lib

<i>Moduł</i>	<i>Znaczenie</i>
BCD_TO_INT	Konwersja bajtu z formatu BCD na INT
INT_TO_BCD	Konwersja bajtu z formatu INT na BCD
EXTRACT(in,n)	n-ty bit z DWORD jest wydawany w typie BOOL
PACK	do 8 bitów jest pakowane w bajcie
PUTBIT	bit w DWORD jest ustawiany na określoną wartość
UNPACK	bajt jest przekształcany na pojedyncze bity
DERIVATIVE	Pochodna
INTEGRAL	Całka
LIN_TRAFO	Przekształcenie wartości REAL
STATISTICS_INT	Min., maks., przeciętna wartość w INT
STATISTICS_REAL	Min., maks., przeciętna wartość w REAL
VARIANCE	Obliczanie wariancji
PD	Regulator PD
PID	Regulator PID
BLINK	Sygnal pulsujący
FREQ_MEASURE	Częstotliwość sygnału wejściowego typu boolean
GEN	Funkcje okresowe
CHARCURVE	Funkcja liniowa
RAMP_INT	Ograniczenie wzrostu/spadku funkcji, (INT)
RAMP_REAL	Ograniczenie wzrostu/spadku funkcji, (REAL)
HYSTERESIS	Histereza
LIMITALARM	Sprawdzone jest przekroczenie granicy wartości do wpisania.

Załącznik G Linie poleceń/plik poleceń

13.24 Linie poleceń

Istnieje możliwość podania do CoDeSys przy uruchomieniu określonych poleceń uwzględnianych następnie w trakcie wykonywania. Te polecenia wiersza poleceń rozpoczynają się od „/”. Nie jest uwzględniana pisownia wielką i małą literą. Przetwarzanie następuje w sekwencjach od strony lewej do prawej.

/online	CoDeSys próbuje po uruchomieniu aktualnego projektu przejść do trybu online.
/run	CoDeSys rozpoczyna aplikację po zalogowaniu. Obowiązuje wyłącznie w powiązaniu z /online.
/batch	CoDeSys jest uruchamiany bez uruchomienia interfejsu użytkownika i daje jako wartość zwracaną kod błędu pierwszego polecenia przetworzonego z błędem lub wartość zwracaną pierwszego polecenia przetworzonego z błędem. CoDeSys kończy się natychmiast po przetworzeniu pliku poleceń. Przetwarzanie pliku poleceń zostaje przerwane po pierwszym błędnie przetworzonym poleceniu. Ostrzeżenia nie powodują zakończenia przetwarzania pliku poleceń. Jeśli nie występują błędy ani ostrzeżenia, wartość zwracana to S_OK. Wartość zwracana jest zawsze zakodowana jako HRESULT (patrz CoDeSys Automation Interface – interfejs automatyki).
/show ... /show hide /show icon /show max /show normal	Istnieje możliwość ustawienia okna CoDeSys-Frame. Okno nie jest wyświetlane i nie ukazuje się również w pasku zadań. Okno jest zminimalizowane. Okno jest zmaksymalizowane. Okno jest wyświetlane w ostatnim zachowanym stanie, który nie był „zminimalizowany” lub „zmaksymalizowany”.
/out <outfile>	Wszystkie komunikaty oprócz okna komunikatów są wyświetlane również w pliku <outfile>.
/noinfo	Przy starcie CoDeSys nie pojawia się ekran startowy.
/userlevel <group>	Istnieje możliwość zdefiniowania grupy roboczej (np. „/userlevel 0” dla grupy roboczej 0)
/password <password>	Hasło dla grupy roboczej można wpisać w sposób bezpośredni (np. „/password abc”)
/openfromplc	Wczytywanie projektu z aktualnie podpiętego sterownika.
/visudownload	W przypadku uruchomienia CoDeSys HMI za pomocą projektu niezgodnego z projektem w sterowniku można przeprowadzić pobieranie. (Okno dialogowe z zapytaniem, na które należy odpowiedzieć TAK lub NIE)
/notargetchange	Zmianę systemu docelowego można przeprowadzić wyłącznie za pomocą pliku poleceń. Patrz polecenie „target...”.
/cmd <cmdfile>	Po uruchomieniu są wykonywane polecenia zawarte w pliku poleceń <cmdfile>.

Wpisywanie do wiersza poleceń odbywa się w następujący sposób:

```
"<ścieżka pliku CoDeSys-Exe>" "<ścieżka projektu>" /<Polecenie1> /<Polecenie2> ...
```

Przykład linii poleceń:

```
"D:\dir1\codesys" "C:\projects\ampel.pro" /show hide  
/cmd command.cmd
```

Plik poleceń (Cmdfile)

Otwarty zostaje plik `ampel.pro`, jednak okno nie jest wyświetlane. Następuje przetwarzanie pliku poleceń (cmdfile) `command.cmd`.

13.25 Plik poleceń (Cmdfile)

Poniżej można znaleźć listę poleceń, które można zastosować w pliku poleceń (<cmdfile>) i które można znowu wywołać za pomocą linii poleceń. Nie jest uwzględniana pisownia wielką i małą literą. Linia poleceń pojawia się jako komunikat w oknie komunikatów w pliku komunikatów (patrz poniżej), chyba że polecenie poprzedza znak „@”.

Ignorowane są znaki występujące po średniku (;) (komentarze). Parametry zawierające spacje należy umieścić w cudzysłowach. Znaki specjalne można stosować wyłącznie wtedy, gdy plik poleceń jest utworzony w kodzie ANSI. Przy podawaniu parametrów poleceń można stosować słowa kluczowe. Odpowiednią listę można znaleźć na końcu opisu poleceń.

Polecenia sterujące:

<code>onerror continue</code>	Poniższe polecenia są przetwarzane dalej, również w przypadku wystąpienia błędu
<code>onerror break</code>	Poniższe polecenia nie są przetwarzane po wystąpieniu błędu

Polecenia menu Online:

<code>online login</code>	Logowanie z wczytanym projektem („Zaloguj online”)
<code>online logout</code>	Wylogowanie („Online” „Wyloguj”)
<code>online run</code>	Uruchamia program użytkownika („Online” „Start”)
<code>online sim</code>	Włącza symulację („Online” „Symulacja”)
<code>online sim off</code>	Wyłączenie symulacji („Online” „Symulacja”)
<code>online bootproject</code>	Utworzenie projektu bootowalnego. Polecenie to można stosować w trybach offline i online. (Patrz również rozdział 6.6, „Online” „Utwórz projekt bootowalny”!)
<code>online sourcecodedownload</code>	Przesyła kod źródłowy aktualnego projektu do sterownika. („Online” „Wczytaj kod źródłowy”)
<code>online stop</code>	Zatrzymuje aktualny program na systemie docelowym („Online” „Stop”)
<code>online sim off</code>	Wyłączenie symulacji („Online” „Symulacja”)

Polecenia menu Plik:

<code>file new</code>	Tworzy nowy projekt („Plik” „Nowy”)
<code>file open <projectfile></code> możliwe polecenia dodatkowe:	Wczytywanie podanego projektu („Plik” „Otwórz”)
<code>/readpwd:<readpassword></code>	Równocześnie jest podawane hasło do odczytu, przez co w przypadku projektu zabezpieczonego hasłem nie pojawia się już okno dialogowe do wpisania hasła.
<code>/writepwd:<writepassword></code>	Równocześnie jest podawane hasło pełnego dostępu, przez co nie pojawia się już okno dialogowe do wpisania hasła.
<code>file close</code>	Zamknięcie wczytanego projektu („Plik” „Zamknij”)
<code>file save</code>	Zapisuje wczytany projekt („Plik” „Zapisz”).
<code>file saveas <projectfile></code> w opcji do uzupełnienia za pomocą: <type><version>	Zapisuje wczytany projekt pod podaną nazwą („Plik” „Zapisz jako”) Domyślnie: Projekt zostaje zapisany jako plik *.pro w aktualnej wersji produktu (wersji utworzenia pliku). Dodatkowo można określić jego zapis jako wewnętrznej lub zewnętrznej biblioteki lub jako projektu starszej wersji:

	<p>wewnętrznej lub zewnętrznej biblioteki lub jako projektu starszej wersji: możliwe wpisy dla <type>: "internallib" zapis jako biblioteka wewnętrzna "externallib" zapis jako biblioteka zewnętrzna "pro" zapis jako projekt</p> <p>możliwe wpisy dla <version>: 15, 20, 21, 22 (wersje produktu 1.5, 2.0, 2.1, 2.2)</p> <p>Przykład: "file save as lib_xy internallib22" -> Projekt xy.pro utworzony w aktualnej wersji CoDeSys zostaje zapisany jako lib_xy.lib w wersji 2.2.</p>
file archive <filename>	Cały aktualny projekt zostaje zarchiwizowany w pliku ZIP pod nazwą <filename>. („Plik” „Zapisz/wyślij archiwum”)
file printersetup <filename>.dfr opcjonalnie do uzupełnienia przez: pageperobject lub pagepersubobject	Ustawienia dokumentacji projektu: Plik ramki *.dfr i dodatkowe dane podziału wydruku na strony: „pageperobject” (nowa strona na obiekcie) lub „pagepersubobject” (nowa strona na obiekcie podrzędny); patrz niżej „project documentation”
file quit	Zamyka CoDeSys („Plik” „Zakończ”)

Polecenia menu Projekt:

project build	Kompilacja przyrostowa projektu („Projekt” „Kompiluj”)
project rebuild lub project compile	Kompilacja całego wczytanego projektu („Projekt” „Kompiluj wszystko”)
project clean	Usuwanie informacji o kompilacji oraz o zmianie online z aktualnego projektu („Projekt” „Wyczyść wszystko”).
project import <file1> ... <fileN>	Podane pliki <file1> ... <fileN> są importowane do wczytanego projektu („Projekt” „Importuj”). Można stosować symbole zastępcze, np. „project import C:\projects*.exp” importuje wszystkie pliki z rozszerzeniem *.exp z katalogu C:\projects.
project export <expfile>	Eksport wczytanego projektu do podanego pliku <expfile> („Projekt” „Eksportuj”)
project expmul <dir>	Eksport każdego obiektu wczytanego projektu do oddzielnego pliku o nazwie danego obiektu umieszczonego w katalogu <dir>.
project documentation	Wydruk projektu na podstawie aktualnych ustawień („Plik” „Ustawienia drukowania” („Projekt” „Dokumentowanie”); patrz również wyżej „file printersetup”)

Sterowanie plikiem komunikatów:

out open <msgfile>	Otwieranie podanego pliku jako komunikatu. Dołączanie nowych komunikatów
out close	Zamykanie aktualnie otwartego pliku komunikatów
out clear	Usuwanie wszystkich komunikatów z aktualnie otwartego pliku komunikatów.

Sterowanie wydawaniem komunikatów:

echo on	Wiersze poleceń mają również formę komunikatu.
echo off	Wiersze poleceń nie mają formy komunikatu.
echo <text>	<text> jest komunikatem

Plik poleceń (Cmdfile)

Sterowanie i zamiana obiektów lub plików przy imporcie, eksporcie, kopiowaniu:

replace yesall	Zamień wszystkie (ewentualnie wstawione „query on” nie jest uwzględniane, nie pojawia się okno dialogowe z dodatkowym zapytaniem).
replace noall	Nie zamieniaj niczego (ewentualnie wstawione „query on” nie jest uwzględniane, nie pojawia się dialog z dodatkowym zapytaniem).
replace query	Jeśli jest ustawione „query on”, pojawia się okno dialogowe z zapytaniem o zamianę obiektów, również w przypadku, gdy jest ustawione „replace yesall” lub „replace noall”.

Sterowanie domyślnym zachowaniem okien dialogowych CoDeSys:

query on	Okna dialogowe są wyświetlane i wymagają wpisania przez użytkownika danych.
query off ok	Wszystkie okna dialogowe zachowują się, jak gdyby użytkownik kliknął OK.
query off no	Wszystkie okna dialogowe zachowują się, jak gdyby użytkownik kliknął Nie.
query off cancel	Wszystkie okna dialogowe zachowują się, jak gdyby użytkownik kliknął Anuluj.

Polecenie wywołania plików poleceń jako podprogramów:

call <parameter1> ... <parameter10>	Wywoływanie plików poleceń jako podprogramów. Można przesłać do 10 parametrów. Dostęp do parametrów w wywołanym pliku za pomocą \$0 – \$9.
--	--

Ustawianie katalogów stosowanych przez CoDeSys: (-> Kategoria Ogólne w oknie dialogowym opcji projektu w „Katalogi”: Jeśli w jednej komendzie jest podane więcej katalogów, należy je rozdzielić za pomocą średnika i spacji i wstawić całość w cudzysłowy.

Przykład, dwa katalogi:

```
dir lib lib "D:\codesys\Libraries\Standard;  
D:\codesys\Libraries\NetVar"
```

dir lib <libdir>	Ustawia <libdir> jako katalog biblioteki
dir compile <compiledir>	Ustawia <compiledir> jako katalog na pliki kompilacji
dir config <configdir>	Ustawia <configdir> jako katalog na pliki konfiguracyjne
dir upload <uploaddir>	Ustawia <uploaddir> jako katalog na pliki upload

Opóźnienie przetwarzania CMDFILE:

delay 5000	Odczekiwanie 5 sekund (dokładność wykonania w krokach co 100 ms)
------------	--

Sterowanie Menadżerem podglądu zmiennych i receptur: .

watchlist load <file>	Wczytywanie zachowanej w <file> listy kontrolnej i otwieranie odpowiedniego okna („Dodatki” „Wczytaj listę kontrolną”)
watchlist save <file>	Zapisywanie aktualnej listy kontrolnej w <file> („Dodatki” „Zapisz listę kontrolną”)
watchlist set <text>	Lista kontrolna zostaje ustawiona jako aktywna lista kontrolna (odpowiada wybraniu listy w lewej części menadżera podglądu zmiennych i receptur).
watchlist read	Aktualizacja wartości zmiennych kontrolnych („Dodatki” „Odczytaj instrukcję”)
watchlist write	Wstawia w zmienne kontrolne wartości z listy kontrolnej > („Dodatki” „Zapisz instrukcję”)

Podłączanie bibliotek:

library add <Bibliotheksdatei1> <Bibliotheksdatei2> .. <BibliotheksdateiN>	Załącza podane pliki bibliotek do listy bibliotek aktualnie otwartego projektu. Jeśli w ścieżce pliku chodzi o ścieżkę względną, ustawiony w projekcie katalog biblioteki jest stosowany jako katalog główny ścieżki.
library delete [<Bibliothek1> <Bibliothek2> .. <BibliothekN>]	Usuwanie podanych bibliotek z listy bibliotek w aktualnie otwartym projekcie.

Kopiowanie obiektów:

object copy <Quellprojektdatei> <Quellpfad> <Zielpfad> (plik projektu docelowego, ścieżka źródłowa, ścieżka docelowa)	Kopiowanie obiektów z podanej ścieżki pliku projektu źródłowego do ścieżki docelowej właśnie otwartego projektu. Jeśli ścieżka źródłowa jest nazwą obiektu, zostaje on skopiowany. W przypadku folderu są kopiowane wszystkie obiekty znajdujące się poniżej. W tym przypadku przejmowana jest struktura folderów poniżej folderu źródłowego. W przypadku braku ścieżki docelowej zostaje ona utworzona.
setreadonly <TRUE FALSE> <Objekttyp> <Objektname> (typ obiektu, nazwa obiektu)	Za pomocą TRUE obiekt jest ustawiany jako tylko do odczytu. W typach obiektów pou, dut, gvl, vis należy dodatkowo podać nazwę obiektu. Możliwe typy obiektów: pou (moduł), dut (typ danych), gvl (lista zmiennych globalnych), vis (wizualizacja), cnc (obiekt CNC), liblist (biblioteki), targetsettings (ustawienia systemu docelowego), toolinstanceobject (instancja w narzędziach), toolmanagerobject (wszystkie instancje w narzędziach), customplconfig (konfiguracja sterownika), projectinfo (informacja o projekcie), taskconfig (konfiguracja zadań), trace (ślady), watchentrylist (menedżer podglądu zmiennych i receptur), alarmconfig (konfiguracja alarmów) np. po „object setreadonly TRUE pou plc_prg” do PLC_PRG jest możliwy tylko dostęp do odczytu

Ustawianie parametrów komunikacji (bramka, urządzenie):

gateway local	Ustawia bramkę komputera lokalnego jako aktualną bramkę.
gateway tcpip <Adresse> <Port>	Ustawia bramkę ustawioną w podanym komputerze zdalnym jako aktualną bramkę. <Adresse>: Adres TCP/IP lub nazwa hosta komputera zdalnego <Port>: Port TCP/IP bramki zdalnej Uwaga: można dojść wyłącznie do bramek bez ustawionego hasła.
device guid <guid>	Ustawia urządzenie z podanym GUID jako urządzenie aktualne. GUID musi odpowiadać następującemu formatowi: {01234567-0123-0123-0123-0123456789ABC} Nawiasy klamrowe oraz myślniki muszą być umieszczone w podanych miejscach.
device name<devicename>	Ustawia urządzenie o podanej nazwie jako urządzenie aktualne.
device instance <Instanzname>	Ustawia podaną nazwę instancji dla aktualnego urządzenia.
device parameter <Id> <parametername> <Wert> (nazwa parametru, wartość)	Przydziela do parametru o podanym Id lub podanej w opcji nazwie podaną wartość, która następnie jest interpretowana przez urządzenie.

Wywołanie systemowe:

system <Befehl> (polecenie)	Wykonuje podane polecenie systemu operacyjnego.
-----------------------------	---

Plik poleceń (Cmdfile)

Wybór systemu docelowego:

target <Id> <name>	Ustawia dla aktualnego projektu platformę docelową. Podawanie ID lub nazwy zdefiniowanej w pliku docelowym. W przypadku uruchomienia CoDeSys za pomocą opcji linii poleceń "/notargetchange" system docelowy można ustawić wyłącznie za pomocą tego polecenia.
--------------------	--

Zapytanie o status systemu:

state offline	Daje „S_OK”, jeśli chwilowo brak jest połączenia pomiędzy systemem programowania i systemem docelowym (tryb offline), w innym przypadku „HRESULT[0x800441f0)” (tryb online).
state online	Daje „S_OK”, jeśli chwilowo istnieje połączenie pomiędzy systemem programowania i systemem docelowym (tryb online), w innym przypadku „HRESULT[0x800441f0)” (tryb offline).

Polecenia odnośnie zarządzania projektem w bazie danych projektu ENI:

W poniższym opisie poleceń są stosowane symbole zastępcze:

<Kategorie> (kategoria): Do zastąpienia przez „project”, lub „shared” lub „compile” dla wymaganych kategorii baz danych projektu, wspólnych obiektów, plików do kompilacji.

<Bausteinname> (nazwa modułu): Nazwa obiektu, odpowiada nazwie obiektu zastosowanej w CoDeSys.

<Objekttyp> (typ obiektu): Do zastąpienia skrótem, który obiekt otrzymuje jako rozszerzenie nazwy modułu w bazie danych i który oddaje typ obiektu (definiowane przez listę typów obiektów, patrz Administracja ENI, „Typy obiektów”).
Przykład: Obiekt „GLOBAL_1.GVL” -> nazwa modułu „GLOBAL_1”, typ obiektu „GVL” (lista zmiennych globalnych)

<Kommentar> (komentarz): Zamienić na tekst komentarza w cudzysłowach (”), zapisany w historii wersji odpowiedniego procesu.

Polecenia konfiguracji połączenia z bazą danych projektu za pomocą serwera ENI:

eni on eni off	Opcja „Zastosuj bazę danych projektu ENI” jest włączona lub wyłączona (Okno dialogowe „Projekt” „Opcje” „Baza danych projektu”)
eni project readonly on eni project readonly off	Opcja „Dostęp tylko do odczytu” dla kategorii bazy danych projektu jest włączona lub wyłączona (Okno dialogowe „Projekt” „Opcje” „Baza danych projektu” „Obiekty projektu”)
eni shared readonly on eni shared readonly off	Opcja „Dostęp tylko do odczytu” dla kategorii bazy danych Obiekty wspólne jest włączona lub wyłączona (Okno dialogowe „Projekt” „Opcje” „Baza danych projektu” „Obiekty wspólne”)
eni set local <Bausteinname> (nazwa modułu)	Przyporządkowuje moduł do kategorii „Lokalne”, tzn. nie jest on zarządzany w bazie danych projektu (Okno dialogowe „Projekt” „Obiekt” „Właściwości” „Połączenie z bazą danych”).
eni set shared <Bausteinname> (nazwa modułu)	Przyporządkowuje moduł do kategorii baz danych „Obiekty wspólne” (okno dialogowe „Projekt” „Obiekt” „Właściwości” „Połączenie z bazą danych”)
eni set project <Bausteinname> (nazwa modułu)	Przyporządkowuje moduł do kategorii bazy danych „Projekt” (Okno dialogowe „Projekt” „Obiekt” „Właściwości” „Baza danych projektu”)

eni <Kategorie> server <TCP/IP_Adresse> <Port> <Projektname> <Benutzername> <Passwort> (kategoria, adres TCP/IP, port, nazwa projektu, nazwa użytkownika, hasło)	Konfiguruje połączenie z serwerem ENI dla kategorii „Obiekty projektu” (Okno dialogowe „Projekt” „Opcje” „Baza danych projektu”); Przykład: eni project server localhost 80 batchtest\project EniBatch Batch (TCP/IP-Adresse = localhost, Port = 80, Projektname = batchtest\project, Benutzername = EniBatch, Passwort = Batch) (adres tcp/ip/ port, nazwa projektu, nazwa użytkownika, hasło)
eni compile sym on eni compile sym off	Włącza/wyłącza opcję „Utwórz informacje o symbolu ASCII (.sym)” dla obiektów kategorii pliki kompilacji (Okno dialogowe „Projekt” „Opcje” „Baza danych projektu” „Ustawienia ENI” dla „Plików kompilacji”)
eni compile sdb on eni compile sdb off	Włącza/wyłącza opcję „Utwórz binarne informacje o symbolu (.sym)” dla obiektów kategorii pliki kompilacji (Okno dialogowe „Projekt” „Opcje” „Baza danych projektu” „Ustawienia ENI” dla „Plików kompilacji”)
eni compile prg on eni compile prg off	Włącza/wyłącza opcję „Utwórz projekt bootowalny” dla obiektów kategorii pliki kompilacji (Okno dialogowe „Projekt” „Opcje” „Baza danych projektu” „Ustawienia ENI” dla „Plików kompilacji”)

Polecenia menu „Projekt” „Baza danych projektu” do pracy z bazą danych:

eni set <Kategorie> (kategoria)	Przyporządkowuje obiekt do kategorii bazy danych („Ustal”)
eni set <Kategorie>set <Objekttyp>:<Bausteinname> <Objekttyp>:<Bausteinname> (kategoria, typ obiektu, nazwa modułu, typ obiektu, nazwa modułu)	Obiekty podane na liście rozdzielonej za pomocą spacji zostają przyporządkowane do kategorii bazy danych. („Ustal wielokrotnie”) Przykład: "eni set project pou:as_fub pou:st_prg" (moduły (pou) as_fub oraz st_prg zostają przyporządkowane do kategorii bazy danych)
eni <Kategorie> getall	Przywołuje wszystkie obiekty tej kategorii z bazy danych projektu („Przywołaj wszystko”)
eni <Kategorie>get <Objekttyp>:<Bausteinname> <Objekttyp>:<Bausteinname> (kategoria, typ obiektu, nazwa modułu, typ obiektu, nazwa modułu)	Obiekty podanej kategorii zawarte na liście rozdzielonej za pomocą spacji są wywoływane z bazy danych. („Przywołaj”) Przykład: "eni project get pou:as_fub gvl:global_1" (przywołanie modułu as_fub.pou oraz globalnej listy zmiennych global_1.gvl)
eni <Kategorie> checkoutall "<Kommentar>" (kategoria, komentarz)	Wyrejestrowuje wszystkie obiekty z bazy danych projektu. Proces wyrejestrowania zostaje opatrzony komentarzem <Kommentar>.
eni <Kategorie> checkout "<Kommentar>" <Objekttyp>:<Bausteinname> <Objekttyp>:<Bausteinname> (kategoria, komentarz, typ obiektu, nazwa modułu, typ obiektu, nazwa modułu)	Obiekty odpowiedniej kategorii podane jako typ obiektu Nazwa modułu zawarte na liście rozdzielonej za pomocą spacji są wyrejestrowywane z bazy danych („Wyrejestruj”). Proces wyrejestrowania jest zawsze opatrywany w historii wersji komentarzem <comment>. Przykład: "eni project checkout "do przetwarzania xy" pou:as_fub gvl:global_1" (POU "as_fub" oraz globalna lista zmiennych "global_1" zostają wyrejestrowane, zaś proces wyrejestrowania zostaje opatrzony komentarzem "zur Bearbeitung von xy" (do przetwarzania xy)
eni <Kategorie> checkinall "<Kommentar>" (kategoria, komentarz)	Rejestruje wszystkie obiekty projektu zarządzane w bazie danych projektu. Proces rejestracji zostaje opatrzony komentarzem <Kommentar> (komentarz).
eni <Kategorie> checkin "<Kommentar>" <Objekttyp>:<Bausteinname> <Objekttyp>:<Bausteinname> (kategoria, komentarz, typ obiektu, nazwa modułu, typ obiektu, nazwa modułu)	Obiekty podane jako typ obiektu Nazwa modułu zawarte na liście rozdzielonej za pomocą spacji są rejestrowane w bazie danych. Proces rejestracji zostaje opatrzony komentarzem <comment> (komentarz).

Plik poleceń (Cmdfile)

Słowa kluczowe dla parametrów komend:

Przy podawaniu parametrów komend można stosować następujące słowa kluczowe umieszczone w „\$”:

\$PROJECT_NAME\$	Nazwa aktualnego projektu CoDeSys (nazwa pliku bez rozszerzenia „.pro”, np. „project_2.pro”)
\$PROJECT_PATH\$	Ścieżka katalogu, w którym znajduje się aktualny plik projektu CoDeSys (bez danych napędu i bez znaku backslash na końcu, np. „projects\sub1”).
\$PROJECT_DRIVE\$	Napęd, w którym jest umieszczony aktualny plik projektu CoDeSys. (bez znaku backslash na końcu, np. „D:”)
\$COMPILE_DIR\$	Katalog kompilacji aktualnego projektu CoDeSys (z danymi napędu i bez znaku backslash na końcu, np. „D:\codesys\compile”)
\$EXE_DIR\$	Katalog, w którym jest umieszczony plik codesys.exe (z danymi napędu i bez znaku backslash na końcu, np. D:\codesys)

Przykład pliku komend <command file name>.cmd:

```
file open C:\projects\CoDeSys_test\ampel.pro
query off ok
watchlist load c:\work\w.wtc
online login
online run
delay 1000
watchlist read
watchlist save $PROJECT_DRIVE$\$PROJECT_PATH$\w_update.wtc
online logout
file close
```

Ten plik komend otwiera plik projektu ampel.pro, wczytuje listę kontrolną załadowaną w w.wtc, uruchamia program użytkownika, po 1 s zapisuje wartości zmiennych w listę kontrolną w_update.wtc, również zapisaną w katalogu „C:\projects\CoDeSys_test” i ponownie zamyka projekt.

Plik komend jest wywoływany w następujący sposób z listy poleceń:

```
"<Pfad CoDeSys-Exe-Datei>" /cmd "<Pfad cmd-Datei>" ("<ścieżka pliku CoDeSys-Exe>" /cmd "<ścieżka pliku cmd>")
```

Załącznik H Siemens, importowanie

W podmenu „Import Siemens” można znaleźć polecenia importu modułów i zmiennych z plików Siemens STEP5. Polecenie „Importuj plik symboli SEQ” służy do importu zmiennych globalnych z plików symboli STEP5. Polecenie to należy wywołać przed poleceniami „Importuj plik S5”, by przy imporcie modułów z adresów absolutnych można było utworzyć czytelne nazwy symboliczne. Polecenie to służy do importowania modułów z plików programu STEP5. Moduły są przy tym wstawiane w otwarty projekt CoDeSys. Można wybrać, czy moduły mają zostać w języku STEP5-IL, czy mają zostać przekonwertowane na język IEC.

Projekt CoDeSys, w który ma nastąpić import powinien być pusty. Należy jednak zapewnić, by biblioteka standard.lib była podłączona do projektu, w przeciwnym razie nie można będzie zaimportować liczników i timerów.

13.26 Importuj plik symboli &SEQ

Format SEQ to popularny format pliku symboli w projekcie STEP5. Z plików symboli SEQ (*.seq) można wczytać przyporządkowania symboliczne. Przyporządkowanie symboliczne zawiera adres absolutny elementu programu S5 (wejście, wyjście, znacznik itp.), odpowiedni identyfikator symboliczny i opcjonalnie komentarz do symbolu. Plik SEQ jest plikiem tekstowym, który w każdej linii zawiera takie przyporządkowanie, przy czym poszczególne „pola” przyporządkowania są oddzielone od siebie za pomocą tabulatorów. Linia może również zawierać sam komentarz; jednak musi się wtedy rozpoczynać od średnika.

Przyporządkowania symboliczne w pliku SEQ są skompilowane na globalne deklaracje zmiennych zgodne z IEC 1131-3. Zastosowane zostają przy tym nazwa symboliczna, adres i w razie potrzeby komentarz. Adres jest dostosowywany do IEC 1131-3 (znaki procentu itp.). Ponieważ plik symboli S5 może zawierać znaki niedopuszczalne w zakresie identyfikatora IEC, nazwa może w razie potrzeby zostać zmieniona. Niepoprawne znaki zostają zastąpione podkreśleniami; jeśli więcej podkreśleń miałyby następować po sobie, drugie z nich zostaje zastąpione poprawnym znakiem (np. „0”). W przypadku zmiany nazwy symbolicznej przy konwersji, nazwa oryginalna zostaje dołączona w komentarzu na końcu. Linie komentarzy SEQ są stosowane jako komentarze. Można utworzyć większą ilość bloków zmiennych globalnych. Każdy blok zawiera do 64 kB tekstu.

Opisany format SEQ jest stosowany przez Siemens-STEP5-PG oraz ACCON-PG firmy DELTALOGIC.

Użytkownik wybiera plik SEQ w standardowym oknie dialogowym systemu Windows. Następnie przeprowadzany jest import, a potem skompilowana jest lista zmiennych globalnych. Mogą przy tym wystąpić błędy warunkowane przez przekształcenie identyfikatorów STEP5 na identyfikatory zgodne z IEC1131-3. Przykładowo identyfikatory STEP5 „A!” i „A?” zostają przekształcone na identyfikatory „A_”, przy czym pojawia się komunikat „Kilka deklaracji o tym samym identyfikatorze A_”. Należy wtedy zmienić jeden z nich.

W żadnym wypadku nie należy podejmować innych zmian w zakresie globalnej listy zmiennych. Jeśli są widoczne adresy poprawne w PLC firmy Siemens, które jednak nie są poprawne dla danego sterownika: Należy je tymczasem zignorować, nawet jeśli w trakcie kompilacji pojawia się ogromna liczba komunikatów o błędach. Adresy są potrzebne do importu modułów!

Jeśli projekt, do którego następuje import zawiera już deklarację zmiennej globalnej x z adresem (np. „%MX4.0”), może się zdarzyć, że w trakcie importu SEQ dla tego adresu jest definiowana jeszcze inna zmienna z tym samym adresem. Zgodnie z IEC 1131-3 jest to dopuszczalne, jednak przeważnie niezgodnie z zamiarem użytkownika. Nie pojawiają się wtedy komunikaty o błędach, jednak program może nie działać zgodnie z oczekiwaniami, ponieważ adres jest wykorzystywany w różnych modułach bez związku. Import należy z tego względu przeprowadzać do pustego projektu, lub do projektu, w którym nie są (jeszcze) stosowane adresy absolutne.

Po imporcie SEQ można zaimportować moduły STEP5. Również wtedy można wstawić do konfiguracji sterownika stosowane wejścia i wyjścia. Ich istnienia nie zakłada się przy imporcie STEP5, lecz po ponownej kompilacji projektu są sprawdzane nowe adresy i w razie potrzeby zgłaszane jako błędy.

13.27 Importuj plik S5

Z plików programu Siemens-S5 (*.s5d) można wczytać moduły. Zawarty w nich kod to kod MC5, który może być wykonany przez PLC S5. Kod MC5 ogólnie odpowiada bezpośrednio liście instrukcji STEP5 (bez nazw symbolicznych) znanej programiście. Ponadto S5D zawiera komentarze do linii listy instrukcji STEP5. Ponieważ plik S5D nie zawiera nazw symbolicznych lecz wyłącznie absolutne, CoDeSys szuka nazwy symbolicznej dla danego adresu w zakresie już istniejących zmiennych projektu CoDeSys. Jeśli nie zostanie ona znaleziona, zostaje zachowany adres absolutny. Jeśli znaczenie ma nazwa symboliczna, import pliku SEQ należy przeprowadzić przed importem pliku S5.

Użytkownik wybiera plik S5D w standardowym oknie dialogowym systemu Windows. Następnie w kolejnym oknie dialogowym pojawia się lista zawartych modułów do wyboru. Najlepiej jest wybrać wszystkie z nich. W tym miejscu można również zdecydować, czy moduły mają pozostać w języku STEP5-IL, czy należy je skonwertować w IL, LD lub FBD.

O ile jest to możliwe, w trakcie importu zamiast adresów absolutnych stosowane są nazwy symboliczne. Jeśli CoDeSys znajduje w trakcie importu instrukcję taką jak „U M12.0”, szukana jest zmienna globalna, na którą jest ustawiony znacznik M12.0. Pobierana jest pierwsza pasująca deklaracja, zaś instrukcja zostaje zaimportowana jako „U-Nazwa” a nie jako „U M12.0” (przy czym nazwa jest symbolicznym identyfikatorem znacznika M12.0).

Niekiedy do importu lub konwersji kodu potrzebne są dodatkowe zmienne. Są one deklarowane globalnie. Przykładowo do imitacji wejść skierowanych na krawędzie (np. w liczniku S5) konieczne są instancje R_TRIG.

13.28 Konwersja S5 zgodnie z IEC 1131-3

W przypadku wybrania jako języka docelowego w trakcie importu STEP5 języka IEC należy założyć, że nie cały projekt zgodny z IEC 1131-3 można poddać konwersji. Jeśli część modułu S5 zawiera kod, którego nie można skonwertować zgodnie z IEC 1131-3, pojawia się odpowiedni komunikat o błędzie i krytyczny oryginalny kod STEP5-IL jest stosowany w module IEC jako komentarz. Kod ten z reguły należy zastąpić, tzn. ponownie wpisać. Niemożliwa jest zgodna z IEC konwersja komend systemowych działających wyłącznie na określonym procesorze S5. „Zestaw komend jądra STEP5” można za pomocą naciśnięcia klawisza przekształcić w kod IEC, pomimo że STEP5 wykazuje ogromne, zasadnicze różnice.

Zestaw komend jądra do konwersji zgodnej z IEC 1131-3 obejmuje wszystkie komendy, do przekształcania w systemie programowania STEP5 zgodnie z LD lub FBD oraz również wszystkie komendy dozwolone w STEP5-PB (moduł programu). Ponadto z komend STEP5, dozwolonych wyłącznie w IL lub FB (modułach funkcji) zasadniczo jest możliwa konwersja zgodna z IEC tych modułów, które rzeczywiście są dostępne w każdym procesorze S5, czyli np. absolutnych i warunkowych skoków, komend przeniesienia itp.

Jedyny wyjątek lub ograniczenie w zakresie konwersji dotyczy zerowania timerów, co jest możliwe w STEP5, jednak nie w zakresie normy IEC 1131-3.

Opis poszczególnych komend do konwersji:

U, UN, O, ON, S, R, = z następującymi argumentami bitowymi: E (wejścia), A (wyjścia), M (znaczniki), S (znaczniki S), D (dane w modułach danych)

U, UN, O, ON z następującymi argumentami: T (timer), Z (licznik)

S, R z następującymi argumentami: Z

SU, RU, P, PN z następującymi argumentami: E, A, M, D

O, O(, U(,)

L, T z zakresem argumentów: E, A, M, D, T, Z, P (peryferia) i rozmiary argumentów: B (bajt), W (słowo), D (podwójne słowo), L (lewy bajt), R (prawy bajt)

L z następującymi formatami stałych: DH, KB, KF, KH, KM, KT, KZ, KY, KG, KC

SI, SE, SA z następującymi argumentami: T

ZV, ZR z następującymi argumentami: Z

+, -, X, : z następującymi argumentami: F (liczba stałoprzecinkowa), G (liczba zmiennoprzecinkowa)

+, - z następującymi argumentami: D (32 bitowa liczba stałoprzecinkowa)

!=, ><, >, <, >=, <= z następującymi argumentami: F, D, G

ADD z następującymi argumentami: BF, KF, DH

SPA, SPB z następującymi argumentami: PB, FB (z większością typów parametrów), SB

A, AX z następującymi argumentami: DB, DX

BE, BEA, BEB

BLD, NOP, ***

UW, OW, XOW

KEW, KZW, KZD

SLW, SRW, SLD, RRD, RLD

SPA=, SPB=

SPZ=, SPN=, SPP=, SPM=

TAK

D, I

Większość komend do argumentów formalnych

Komendy nie do konwersji

U, UN, O, ON, S, R, = z następującymi argumentami bitowymi: Bity timera i licznika (T0.0, Z0.0)

L, T z następującymi zakresami argumentów: Q (peryferia rozszerzone)

LC z następującymi argumentami: T, Z

SV, SS, R, FR z następującymi argumentami: T

FR z następującymi argumentami: Z

Komendy do argumentów formalnych do uruchamiania, zerowania, zwolnienia timerów

Wszystkie komendy z argumentami z zakresów BA, BB, BS, BT (dane systemu operacyjnego).

SPA, SPB z następującymi argumentami: OB (działa tylko na niektórych S5 z niektórymi OB)

BA, BAB z następującymi argumentami: FX

E, EX z następującymi argumentami: DB, DX

STP, STS, STW

DEF, DED, DUF, DUD

SVW, SVD

SPO=, SPS=, SPR

AS, AF, AFS, AFF, BAS, BAF

ENT

SES, SEF

B z następującymi argumentami: DW, MW, BS

LIR, TIR, LDI, TDI, TNW, TXB, TXW

MAS, MAB, MSA, MSB, MBA, MBS

MBR, ABR

LRW, LRD, TRW, TRD

TSG

LB, TB, LW, TW z następującymi argumentami: GB, GW, GD, CB, CW, CD
ACR, TSC
BI
SIM, LIM

Po obejrzeniu komend nie do konwersji można stwierdzić, iż większość to komendy specjalne do dyspozycji tylko na niektórych procesorach. Komendy standardowe nie do konwersji zgodnej z IEC to: Wczytywanie zakodowanych przez BCD wartości timera lub licznika (LC T, LC Z), typy timerów SV, SS oraz zerowanie timerów.

Moduły danych

Moduły danych STEP5 są konwertowane na moduły z nagłówkiem, lecz bez kodu. Jest to korzystne, jeśli moduły danych są stosowane jako zwykły zakres zmiennych, lecz niekorzystne w przypadku podjęcia w programie STEP5 próby ręcznej implementacji koncepcji takich jak moduły danych instancji.

Inne problemy w trakcie importu STEP5

Istnieją następujące miejsca, w których można ręcznie poprawić import STEP5.

1. Wartości czasu w zmiennych słownych

W STEP5 wartość czasu może być umieszczona w każdym adresie słowa, czy jest to zakres znacznika, czy moduł danych. W IEC 1131-3 jest to zabronione, zmiennych lub stałych TIME nie można łączyć z adresami WORD. W trakcie importu STEP5 może się zdarzyć utworzenie takiej błędnej sekwencji komend. Nie stanie się tak w przypadku skorzystania z modułu danych i wybrania dla odpowiedniego adresu formatu czasu (KT). Błąd występuje więc jedynie wtedy, gdy program STEP5 co najmniej wymaga poprawek. Błąd można rozpoznać po komunikacie „Niekompatybilne typy: WORD nie można skonwertować w TIME.” lub „Niekompatybilne typy: TIME nie można skonwertować w WORD.” W trakcie dodatkowej obróbki należy zmienić deklarację zmiennej WORD (jeśli taka istnieje) i zrobić z niej zmienną TIME.

2. Błąd dostępu do modułów danych

W IEC 1131-3 brak jest modułów danych i nie można ich w niej kompletnie zobrazować. W STEP5 są one raz stosowane jako zwykły zakres zmiennych (jak np. zakres znaczników), zaś raz w formie tablicy (B DW), znacznika (B MW100, A DB 0) lub union (dostęp bajtowy, słowa lub słowa podwójnego w modułach danych). Konwersja STEP5 może być przeprowadzana w stosunku do dostępu do modułu danych tylko wtedy, jeśli dostęp ten następuje w sposób strukturalny. W sposób konkretny w przypadku dostępu musi być znane, który moduł danych jest właśnie otwarty (A DB). Jest tak, jeśli operacja na module danych A w tym samym module jest bardziej z przodu, lub jeśli numer modułu danych zostaje dodany do modułu jako parametr formalny. Jeśli przed pierwszym dostępem do modułu danych nie znajduje się A DB, modułu nie można poddać konwersji. Można to rozpoznać po ostrzeżeniu „Nie otwarto modułu danych (dodaj moduł danych A DB)”. W skonwertowanym module widoczne są następnie dostępy do niezdefiniowanych zmiennych o przykładowych nazwach „ErrorDW0”, które w trakcie kompilacji właśnie skonwertowanego modułu powodują komunikaty o błędach. Zmienne można następnie zastąpić dostęпами do właściwego modułu danych np. wszędzie wpisać „DB10.DW0” zamiast „ErrorDW0”. Inną możliwością stanowi, wyrzucenie jeszcze raz skonwertowanego modułu i wstawienie A DB na początku modułu STEP5.

W każdym przypadku moduł STEP5 korzystający ze słów (bajtów itp.) z danymi powinien najpierw otwierać moduł danych. W razie potrzeby moduł należy poprawić przed importem wstawiając odpowiednią komendę A DB, najlepiej na początku modułu. W innym przypadku skonwertowany moduł należy poddać dodatkowej obróbce.

Jeśli istnieje więcej operacji A DB, przez które częściowo przeskakuje się, konwersja może być niepoprawna, tzn. być może jest tworzony kod korzystający z niewłaściwego modułu danych.

3. Wyższe koncepcje dostępu do modułów danych

W STEP5 istnieje możliwość ręcznej realizacji czegoś takiego jak instancje przez otwieranie przez moduł kodu w sposób ukierunkowany modułu danych. Coś takiego dzieje się np. w przypadku poniższej sekwencji kodu:

L KF +5
 T MW 44
 B MW 44
 A DB 0

W tym miejscu zostaje ostatecznie otwarte DB5 (ogólnie zostaje otwarty moduł danych, którego numer znajduje się w słowie ze znacznikiem %MW44). Dostępny tego typu nie zostają ujęte w trakcie konwersji, tzn. należy dokonać niżej opisanej dodatkowej obróbki:

Najpierw należy zaimportować wszystkie moduły danych służące jako moduły danych instancji, np. DB5 oraz DB6. Są one importowane jako zwykłe moduły IL, LD lub FBD, zgodnie z wymaganiami. Moduły te nie mają kodu, lecz tylko nagłówek z definicjami zmiennych lokalnych. Z tych modułów można następnie zrobić instancje typów. Należy utworzyć typ zdefiniowany przez użytkownika (o nazwie np. DBTyp) i wstawić w niego jako składniki zmienne lokalne skonwertowanych modułów danych. Następnie należy utworzyć globalne instancje tego typu zapisując w liście zmiennych globalnych:

```
VAR_GLOBAL
  DB5, DB6 : DBTyp;
END_VAR
```

Po tym można usunąć z projektu skonwertowane moduły danych.

Następnie należy jeszcze zobrazować ukierunkowane otwieranie modułów danych przez nadanie odpowiedniemu modułowi jeszcze parametru VAR_INPUT typu DBTyp. Dostęp do danych w obrębie tego modułu należy następnie zmienić na tę instancję. Przy wywołaniu należy następnie jako aktualny parametr podać jeden z modułów danych tej instancji.

4. Specjalną rolę odgrywają tzw. zintegrowane moduły funkcji S5 posiadające interfejs do wywoływania STEP5, którego implementacja nie jest jednak zapisana w STEP5 (lub MC5) lub jest chroniona za pomocą specjalnego mechanizmu. Moduły takie istnieją przeważnie jako firmware i można je „zaimportować wyłącznie jako interfejs”. Część implementacji takiego modułu pozostaje pusta. W trakcie konwersji moduły takie zasadniczo należy zaprogramować na nowo.

5. Istnieją również firmware-OB nie posiadające wprawdzie interfejsu, których kod nie jest w STEP5, lecz np. w assemblerze 805xx. Dotyczy to w pierwszym rzędzie wykonywanego jako OB251 regulatora PID uzyskującego swoje parametry i zmienne lokalne za pomocą wybranego innego modułu (danych). Regulatora PID oraz odpowiedniego modułu danych oraz innych modułów wykorzystujących regulator przy pomocy modułu danych nie można skonwertować zgodnie z IEC. Kod IEC tworzony dla modułów danych i innych modułów w trakcie konwersji nie ma sensu bez regulatora PID. Znaczenie części programu wynika z odpowiedniego podręcznika programowania CPU.

6. Do konfiguracji S5-CPU a czasem również podzespołów istnieją niekiedy moduły danych konfiguracji takie jak DB1 (S5-95U), DX0, DX2, przekształcane w trakcie konwersji w moduły IEC bez znaczenia. Znaczenia takich danych wynika częściowo z odpowiedniego podręcznika programowania CPU, częściowo należy skorzystać z systemu programowania S5, do przeanalizowania za pomocą modułów danych konfiguracji. Konfiguracja dotyczy np. komunikacji, przetwarzania wartości analogowych, pracy w trybie wieloprocesorowym itp. Z tego względu nie opłaca się, w ogóle ruszać tych modułów przy zamianie na PLC innej firmy niż Siemens.

Po imporcie należy przejść zgłoszone błędy i poprawić odpowiednie miejsca, uzupełnić je lub napisać na nowo. Miejsca te są oznaczone komentarzami zgodnymi z:

(* UWAGA! Jako komentarz kod STEP5 nie do konwersji: *)

Następnie kod nie do konwersji, również jako komentarz.

Na koniec należy jeszcze zwrócić uwagę na adresy. W trakcie importu są tworzone oryginalne adresy Siemens. Schemat tych adresów jest następujący:

Bity: Offset bajtu.Numer bitu

Nie bity:offset bajtu

Ponadto następujące po sobie adresy nakładają się na siebie (ponieważ liczby w adresach to offsety bajtów). Tak np. %MW32 i %MW33 mają nakładające się bajty, mianowicie %MB33 (oczywiście tylko w PLC firmy). W Państwa PLC zwykle %MW32 oraz %MW33 nie mają ze sobą nic wspólnego.

Państwa sterownik ewentualnie ma więcej hierarchii, np. nie bity mają poziomy zagnieżdżenia („%MW10.0.0” jako WORD). Można albo uzupełnić adresy, by były one kompatybilne ze sterownikiem lub spróbować je opuścić. Należy jednak przy tym postępować bardzo ostrożnie. Często może się zdarzyć, że w pierwotnym programie firmy Siemens następuje do tego samego zakresu pamięci dostęp do słowa, dostęp bitowy lub bajtowy. W trakcie importu do CoDeSys dostępy takie są poprawnie kompilowane wyłącznie w modułach danych. W tym miejscu CoDeSys tworzy dla słów w modułach danych zmienne WORD. W przypadku dostępu WORD do słowa X w module danych y nie ma problemów. Dostępy do lewego lub prawego bajtu w słowie X, dostępy do podwójnego słowa zostają następnie skompilowane w bardziej złożone wyrażenia. Nie można tego zrobić w przypadku znaczników, wejść i wyjść, ponieważ nie można w tym przypadku wyjść ze standardowej metody dostępu (np. dostęp do słowa). Jeśli w zakresie programu pracuje się raz z %MX33.3 a raz z %MB33 lub %MW32 lub %MD30, należy to wykonać – w żmudny sposób – ręcznie. Utworzony przez CoDeSys w trakcie importu program IEC nie będzie z pewnością działać w tym przypadku poprawnie.

Należy wyświetlić listę odsyłaczy ze wszystkimi wejściami, wyjściami i znacznikami, by znaleźć, które dostępy są krytyczne. Mieszane dostępy należy usunąć ręcznie.

Załącznik I Okna dialogowe ustawień systemu docelowego

Zdefiniowane wstępnie w pliku docelowym ustawienia systemu docelowego użytkownik może jeszcze w razie potrzeby zmienić w CoDeSys. Zależy to od tego, czy w pliku docelowym zostały one wpisane jako „widoczne” i „edytowalne”.

W zakładce „Zasoby” w „Ustawieniach systemu docelowego” można znaleźć okno dialogowe „**Ustawienia systemu docelowego**”, wyświetlające w polu za **Konfiguracją** co najmniej nazwę konfiguracji dla aktualnie ustawionych celów. Jeśli taka jest konfiguracja, widoczne jest dodatkowo pięć zakładek, w których widoczne i edytowalne są różne ustawienia dla tego celu odnoszące się do

- platformy docelowej,
- podziału pamięci,
- ogólnych,
- funkcji sieciowych,
- wizualizacji.

Uwaga: Każda zmiana zadanych przez producenta ustawień może prowadzić do znaczących zmian w zachowaniu systemu docelowego!

Za pomocą przycisku **Ustawienie domyślne** po zmianie ustawienia można wyzerować do wartości z konfiguracji standardowej.

Opis możliwej zawartości okien dialogowych kategorii Podział pamięci, Ogólne i Funkcje sieciowe odnosi się do wszystkich platform. **Dla różnych systemów docelowych różne są punkty w oknie dialogowym odnoszące się do kategorii Platforma docelowa.**

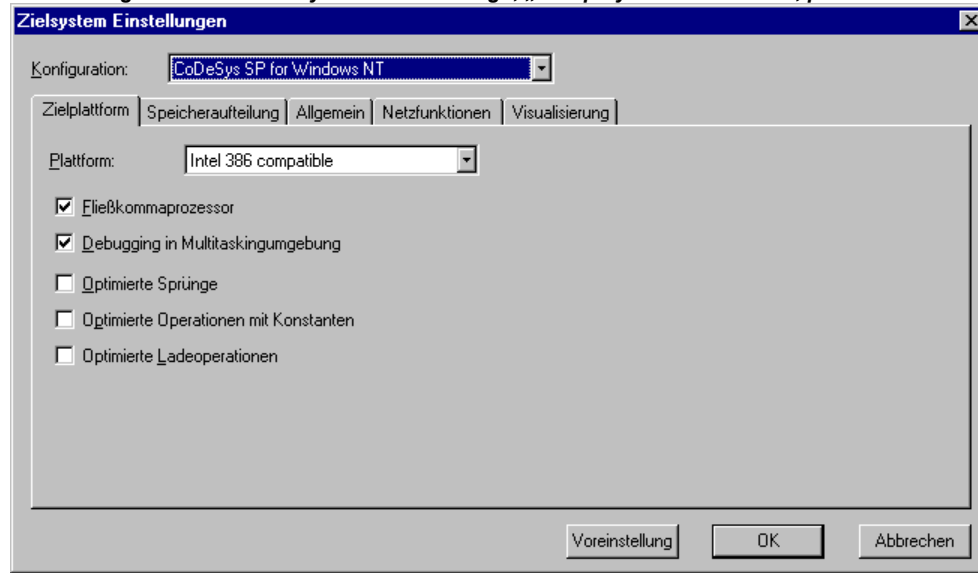
Aktualnie obsługiwane są następujące platformy:

- kompatybilne z Intel 386
- Motorola 68K
- Infineon C16x
- PowerPC
- Intel StrongARM
- MIPS III ISA
- Hitachi SH
- 8051

13.29 Ustawienia kategorii Platforma docelowa

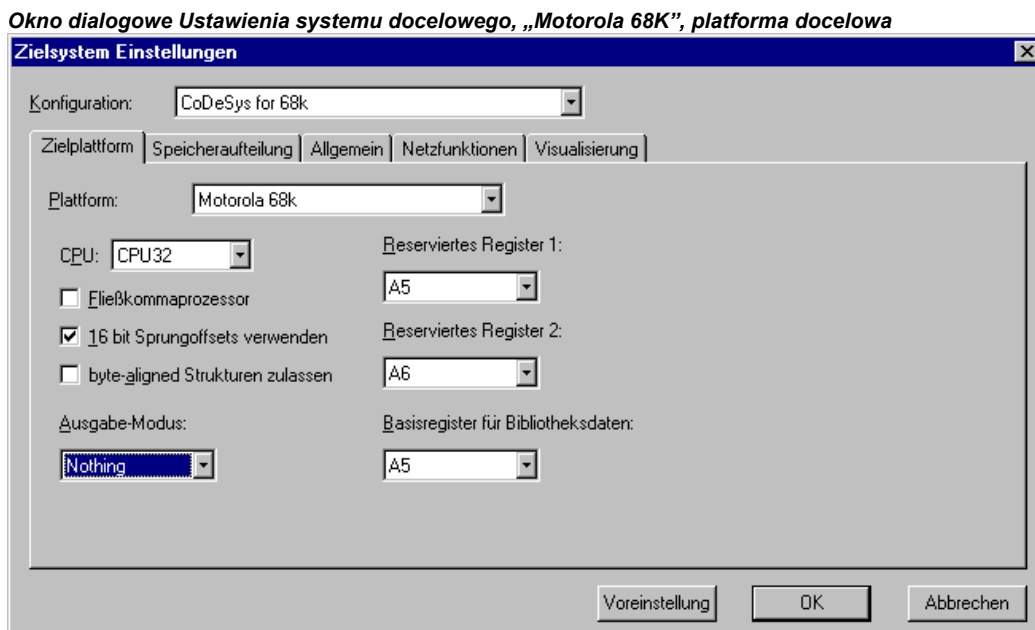
13.29.1 System docelowy „Intel 386 compatible”, platforma docelowa

Okno dialogowe Ustawienia systemu docelowego, „Kompatybilne z Intel 386”, platforma docelowa



Punkt w oknie dialogowym	Znaczenie
Platforma	Typ systemu docelowego.
Opcja procesora zmiennoprzecinkowego	Komendy FPU do operacji zmiennoprzecinkowych są generowane na platformie x86.
Debugowanie w środowisku wielozadaniowym	Generowanie dodatkowego kodu zezwalającego na debugowanie w systemach wielozadaniowych.
Skoki zoptymalizowane	Poprawione skoki warunkowe po porównywaniu; szybszy i krótszy kod (szczególnie na 386/486); linie z warunkami przed skokami, są wyświetlane w kontroli przepływu w kolorze szarym.
Zoptymalizowane operacje se stałymi	Poprawione operacje na stałych ($A = A + 1$, $A < 500$ itd.); szybszy i krótszy kod (szczególnie na 386/486); stałe są wyświetlane w kontroli przepływu w kolorze szarym.
Zoptymalizowane operacje wczytywania	Operacje wczytywania nie korzystają w przypadku wielokrotnego dostępu ze zmiennej/stałej; szybszy i krótszy kod.

13.29.2 System docelowy Motorola 68K, kategoria Platforma docelowa



Punkt w oknie dialogowym	Znaczenie
Platforma	Typ systemu docelowego.
Procesor zmiennoprzecinkowy	Generowanie komend FPU do operacji zmiennoprzecinkowych.
Rejestr zarezerwowany 1:	Podany rejestr adresu jest zarezerwowany i nie jest stosowany. W przypadku „None” może być stosowane przez generator kodu; Poprawne wartości: None, A2, A4, A5, A6.
Rejestr zarezerwowany 2:	Kolejny zarezerwowany rejestr adresu. Podany rejestr adresu jest zarezerwowany i nie jest stosowany. W przypadku „None” może być stosowane przez generator kodu; poprawne wartości: None, A2, A4, A5, A6.
CPU:	Wariant 68K CPU; wersja podstawowa 68000 lub CPU32 i wyższa; poprawne wartości: CPU32 68K.
Zastosuj 16-bitowe przeskoków	Aktywne: Skoki do analizy wyrażeń boolean pracują ze względnymi offsetami 16 bitowymi (możliwe bardziej złożone wyrażenia, lecz więcej kodu); Nnieaktywne: stosowane są offsety 8 bitowe.
Rejestr podstawowy dla danych biblioteki	Rejestr do adresowania danych statystycznych w obrębie bibliotek C (jest wczytywany przed wywołaniem funkcji bibliotek za pomocą adresu wolnej pamięci).
Zezwalaj na struktury byte-aligned	Adresowanie bajtowe (możliwe również adresy nieparzyste).
Tryb prezentacji	Ustawienie „Assembler” lub „Disassembler”: W trakcie kompilacji w ustawionym katalogu kompilacji („Projekt” „Opcje” „Katalogi”) zostaje utworzony plik szesnastkowy. Zawiera on utworzony kod assemblera lub dodatkowo rozbitny kod. Ustawienie „Nothing”: Nie jest tworzony plik szesnastkowy.

13.29.3 System docelowy Infineon C16x, kategoria Platforma docelowa

Okno dialogowe Ustawienia systemu docelowego, „Infineon C16x”, platforma docelowa

Punkt w oknie dialogowym

Znaczenie

- Platforma** Typ systemu docelowego.
- Kod / kompilator:** Kompilator stosowany w kompilacji systemu docelowego i bibliotek (w związku z konwencjami wywołania C).
BSO-Tasking|Klin
- Kod / wielkość stosu** Maksymalna głębokość wywołania (zagnieżdżenie).
Poprawne wartości: 32, 64, 96, 128, 160, 192, 224, 256
- Kod / dane** Model pamięci dla danych.
Poprawne wartości: Huge, Far, Near.
- Kod / funkcje** Model pamięci dla kodu.
Poprawne wartości: Huge, Near.
- Inicjuj funkcje** Aktywne = funkcje zawierają kod inicjalizacji dla zmiennych lokalnych.
- Optymalizuj** Aktywne = optymalizacje kodu dla stałych wskaźników tablicy.
- Output HEX** Aktywne = wydawanie pliku szesnastkowego kodu.
- Output BIN** Aktywne = wydawanie pliku binarnego kodu.
- Output MAP** Aktywne = wydawanie pliku MAP kodu.
- Output LST** Aktywne = wydawanie pliku List kodu.
- Output LST, z adresami** Aktywne = wydawanie listy adresów kodu.
- Biblioteki / kod** Ustawienia bibliotek:
- tabele** adres początkowy kodu,
- dane** adres początkowy tabel,
- długość danych** adres początkowy danych,
- moduły** długość danych bibliotek,
- referencje** maks. ilość modułów bibliotek: Poprawne wartości: 0-512
maks. ilość referencji.

DPPs / DPP0..DPP2 Ustawianie Data Page Pointer 0 do 2.
W instancjach Poprawne wartości: None, Auto, Page 0, Page 1, ..., Page 255.
 DPP do krótkiego adresowania instancji bloków funkcji.
 Poprawne wartości: None, DPP0, DPP1, DPP2.

13.29.4 Systemy docelowe Intel StrongARM i Power PC, kategoria Platforma docelowa

Punkty w oknie dialogowym dla tych obu systemów są identyczne.

Okno dialogowe Ustawienia systemu docelowego, Power PC, platforma docelowa

Punkt w oknie dialogowym	Znaczenie
Platforma	Typ systemu docelowego.
Procesor zmiennoprzecinkowy	Aktywne: Generowanie komend FPU do operacji zmiennoprzecinkowych.
Pierwszy rejestr parametrów (Integer):	Rejestr, do którego jest przesyłana pierwsza (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr dla ostatniego parametru (Integer)	Rejestr, do którego jest przesyłana ostatnia (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr zwracanej wartości (Integer)	Rejestr do którego są zwracane wartości całkowite wywołań funkcji C (zakres zależny od systemu operacyjnego).
Pierwszy rejestr parametrów (Float)	Rejestr, do którego jest przesyłana pierwsza (Float) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr dla ostatniego parametru (Float)	Rejestr, do którego jest przesyłana ostatnia (Float) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr zwracanej wartości (Float)	Rejestr do którego są zwracane parametry Float wywołań funkcji C (zakres zależny od systemu operacyjnego).
Intel byte order	Aktywne: Stosowany jest schemat adresów Intel Byte.
Maks. wielkość argumentów w stosie (w bajtach)	Wielkość stosu dla argumentów w bajtach. Domyślnie: 40

13.29.5 System docelowy MIPS III ISA, kategoria Platforma docelowa

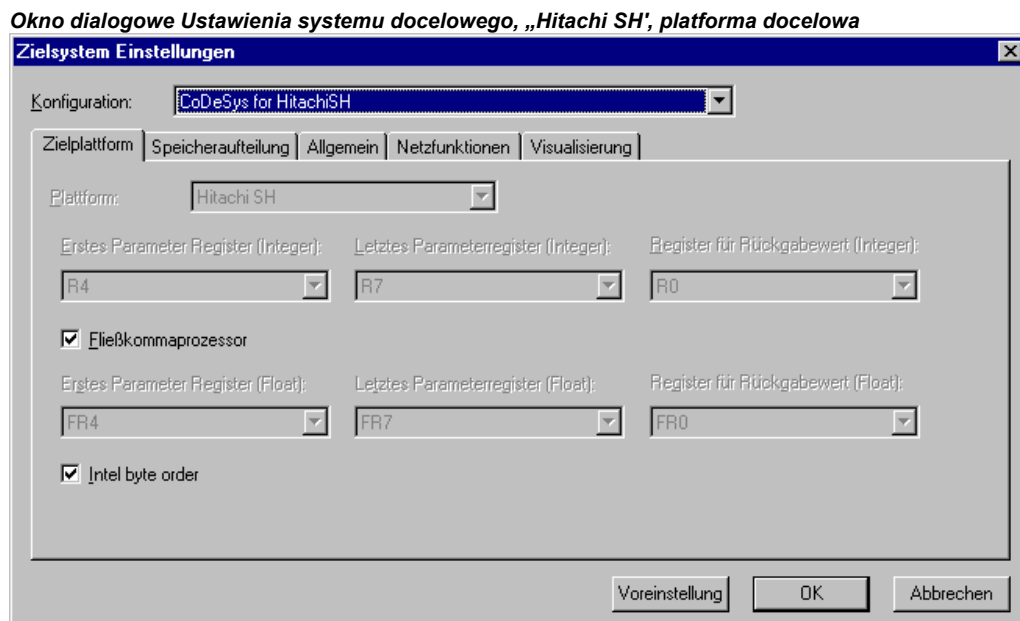
Okno dialogowe Ustawienia systemu docelowego, „MIPS III ISA”, platforma docelowa

Punkt w oknie dialogowym

Znaczenie

Platforma	Typ systemu docelowego.
Pierwszy rejestr parametrów (Integer)	Rejestr, do którego jest przesyłana pierwsza (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr dla ostatniego parametru (Integer)	Rejestr, do którego jest przesyłana ostatnia (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr zwracanej wartości (Integer)	Rejestr do którego są zwracane wartości całkowite wywołań funkcji C (zakres zależny od systemu operacyjnego).
Maks. wielkość argumentów w stosie (w bajtach):	Zależne od systemu operacyjnego: Maksymalna wielkość argumentów, które można przesłać do projektu w stosie (w bajtach).

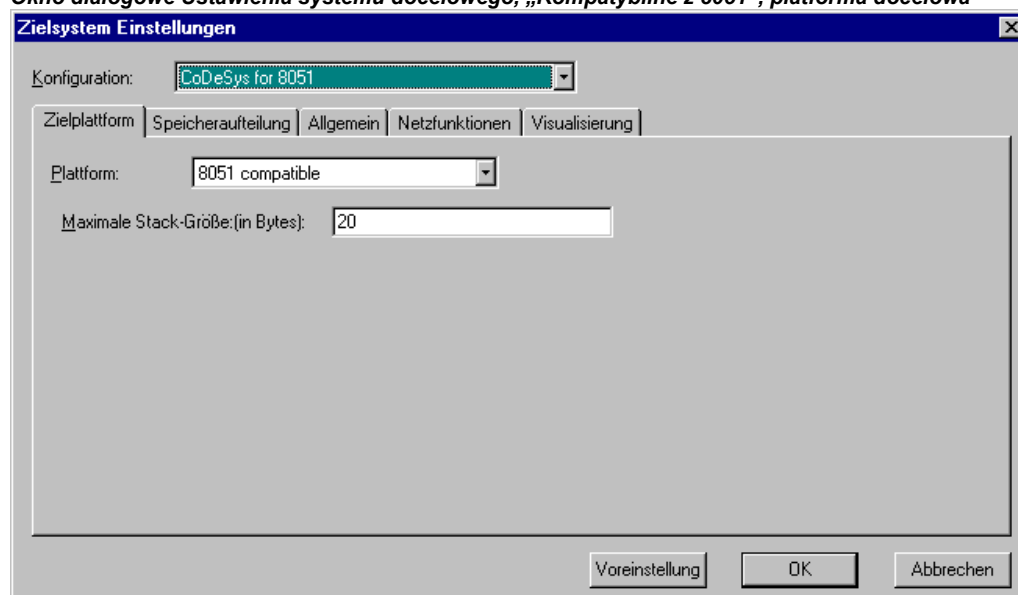
13.29.6 System docelowy Hitachi SH, kategoria Platforma docelowa



Punkt w oknie dialogowym	Znaczenie
Platforma	Typ systemu docelowego.
Pierwszy rejestr parametrów (Integer)	Rejestr, do którego jest przesyłana pierwsza (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr dla ostatniego parametru (Integer)	Rejestr, do którego jest przesyłana ostatnia (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr zwracanej wartości (Integer)	Rejestr do którego są zwracane wartości całkowite wywołań funkcji C (zakres zależny od systemu operacyjnego).
Procesor zmiennoprzecinkowy	Aktywne: Generowanie komend FPU do operacji zmiennoprzecinkowych.
Pierwszy rejestr parametrów (Float)	Pierwszy rejestr dla parametrów całkowitych (zakres zależny od systemu operacyjnego).
Rejestr dla ostatniego parametru (Float)	Pierwszy rejestr dla parametrów Float (zakres zależny od systemu operacyjnego).
Rejestr zwracanej wartości (Float)	Pierwszy rejestr dla wartości zwracanych Float (zakres zależny od systemu operacyjnego).
Intel Byte Order	Aktywne: Stosowany jest schemat adresów Intel Byte.

13.29.7 System docelowy 8051 compatible, kategoria Platforma docelowa

Okno dialogowe Ustawienia systemu docelowego, „Kompatybilne z 8051”, platforma docelowa



Punkt w oknie dialogowym	Znaczenie
Platforma	Typ systemu docelowego.
Maks. wielkość stosu (w bajtach):	Maksymalna wielkość argumentów, które można przesłać do projektu w stosie (ilość w bajtach).

13.29.8 System docelowy TriCore, kategoria Platforma docelowa

Ustawienia dla systemu docelowego są zakodowane na stałe i plik docelowy nie ma na nie wpływu. W przypadku konieczności zmian należy się zwrócić do firmy 3S-Smart Software Solutions GmbH.

Wskazówka: System docelowy TriCore operacje native REAL, lecz nie operacje LREAL.

Najważniejsze ustawienia:

Ustawienie	Znaczenie
Platform=Tricore	Typ systemu docelowego.
Pierwszy rejestr parametrów (Integer) = 4	Rejestr, do którego jest przesyłana pierwsza (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr dla ostatniego parametru (Integer) = 7	Rejestr, do którego jest przesyłana ostatnia (całkowita) wartość parametru wywołania funkcji C (zakres zależny od systemu operacyjnego).
Rejestr zwracanej wartości (Integer) = 2	Rejestr do którego są zwracane wartości całkowite wywołań funkcji C (zakres zależny od systemu operacyjnego).
Pozostałe:	<ul style="list-style-type: none"> – wywołanie funkcji nie jest stosowane do implementacji punktów wstrzymania, – nie jest stosowane Motorola ByteOrder, – Alignment: 4 bajty (ważne przede wszystkim dla tablic).

13.30 Ustawienia kategorii Podział pamięci

Opisane w tej zakładce ustawienia mogą pojawić się we wszystkich platformach standardowych.

Okno dialogowe Ustawienia systemu docelowego, Podział pamięci

The screenshot shows the 'Zielsystem Einstellungen' dialog box for 'CoDeSys SP for Windows NT Realtime'. The 'Speicheraufteilung' tab is selected. The 'Basis' section has 'Automatisch' checked for all items. The 'Größe' section shows segment sizes: 16#400000, 16#80000, 16#8000, 16#4000, 16#4000, and 16#4000. The 'Bereich' section has dropdown menus set to 1, 1, 1, 1, and 2. At the bottom, 'Eigenes Retainsegment' is checked, 'Maximale Anzahl von Bausteinen' is 512, and 'Größe des gesamten Datenspeichers' is 16#10000.

Punkt w oknie dialogowym	Znaczenie
Kod (podstawowy)	Aktywne: Zakres kodu jest alokowany automatycznie; Nieaktywne: Zakres kodu znajduje się w podanym adresie absolutnym (podstawowy).
Globalny (podstawowy)	Aktywne: Zakres danych (dane globalne) jest alokowany w odpowiednim zakresie automatycznie; Nieaktywne: Zakres kodu (dane globalne) znajduje się w podanym adresie absolutnym.
Pamięć (podstawowa)	Aktywne: Znaczniki są alokowane w odpowiednim zakresie automatycznie.; Nieaktywne: Zakres znaczników znajduje się w podanym adresie absolutnym.
Dane wejściowe (podstawowe)	Aktywne: Obraz procesu wejściowego jest alokowany w odpowiednim zakresie automatycznie; Nieaktywne: Obraz procesu wejściowego znajduje się w podanym adresie absolutnym.
Dane wyjściowe (podstawowe)	Aktywne: Obraz procesu wyjściowego jest alokowany w odpowiednim zakresie automatycznie; Nieaktywne: Obraz procesu wyjściowego znajduje się w podanym adresie absolutnym.
Retain (podstawowe)	Aktywne: Dane remanentne są alokowane w odpowiednim zakresie automatycznie; Nieaktywne: Dane remanentne znajdują się w podanym adresie absolutnym.
Zakres (kod)	Zakres numerów zakresu danych (kod); Poprawne wartości: 1, 2, 3, 4, 5, 6
Zakres (globalny)	Zakres numerów danych (dane globalne); Poprawne wartości: 1, 2, 3, 4, 5, 6
Zakres (pamięć)	Zakres numerów znacznika; Poprawne wartości: 1, 2, 3, 4, 5, 6
Zakres (dane wejściowe)	Zakres numerów obrazu procesu wejściowego; Poprawne wartości: 1, 2, 3, 4, 5, 6

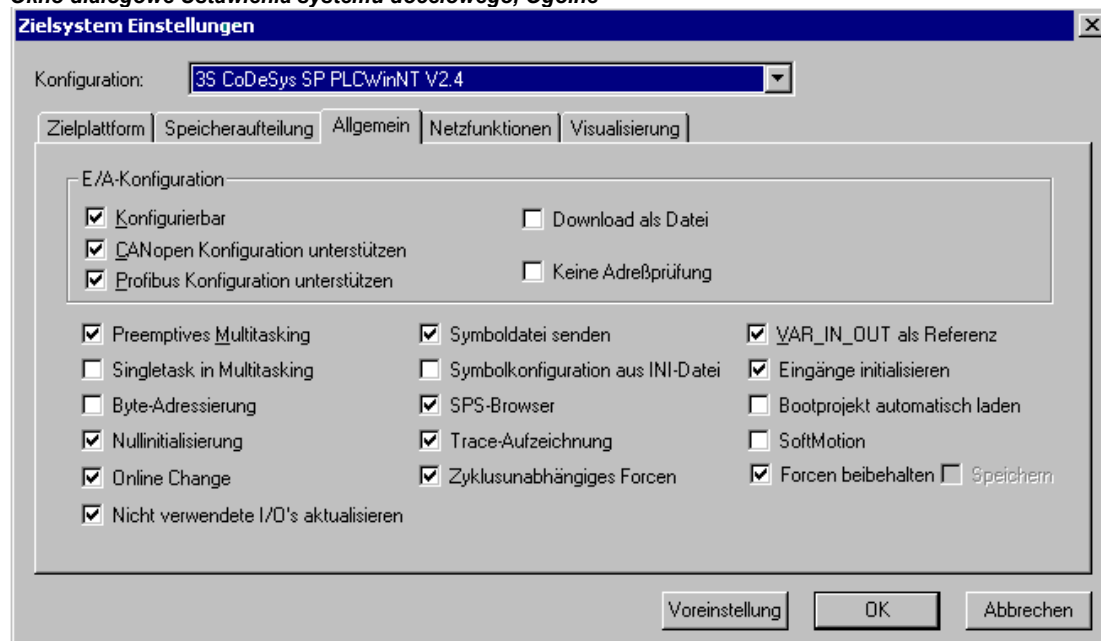
Ustawienia kategorii Podział pamięci

Punkt w oknie dialogowym	Znaczenie
Zakres (dane wyjściowe)	Zakres numerów obrazu procesu wyjściowego; Poprawne wartości: 1, 2, 3, 4, 5, 6
Zakres (Retain)	Zakres numerów danych remanentnych; Poprawne wartości: 1, 2, 3, 4, 5, 6
Podstawa (kod)	Adres segmentu kodu (poprawne tylko jeśli „Automatycznie” nie jest nieaktywne).
Podstawa (globalne)	Adres zakresu danych (dane globalne); (poprawne tylko jeśli „Automatycznie” nieaktywne).
Podstawa (pamięć)	Adres zakresu znacznika; (poprawne tylko jeśli „Automatycznie” nieaktywne).
Podstawa (dane wejściowe)	Adres obrazu procesu wejściowego; (poprawne tylko jeśli „Automatycznie” nieaktywne).
Podstawa (dane wyjściowe)	Adres obrazu procesu wyjściowego; (poprawne tylko jeśli „Automatycznie” nieaktywne).
Podstawa (Retain)	Adres zakresu danych remanentnych; (poprawne tylko jeśli „Automatycznie” nieaktywne).
Wielkość (kod)	Wielkość zakresu kodu.
Wielkość segmentu (globalnie)	Wielkość segmentu danych.
Wielkość (pamięć)	Wielkość zakresu znacznika.
Wielkość (dane wejściowe)	Wielkość obrazu procesu wejściowego.
Wielkość (dane wyjściowe)	Wielkość obrazu procesu wyjściowego.
Wielkość (Retain)	Wielkość zakresu danych remanentnych.
Oddzielny segment Retain on/off	Aktywne: Dane remanentne są zarządzane w oddzielnym segmencie.
Rozmiar całej pamięci danych	Rozmiar całej pamięci danych
Maks. liczba segmentów danych globalnych	Maks. ilość globalnych segmentów danych do ustawienia w opcjach projektu.
Maks. ilość modułów	Maks. ilość modułów w projekcie.

13.31 Ustawienia kategorii Ogólne

Opisane w tym miejscu wpisy można stosować do wszystkich platform standardowych.

Okno dialogowe Ustawienia systemu docelowego, Ogólne



Punkt w oknie dialogowym	Znaczenie
Konfigurowalne	Aktywne: Obsługa konfigurowalnej konfiguracji I/O i wczytywanie opisu konfiguracji do sterownika.
Obsługa konfiguracji CANopen	Aktywne: Obsługa konfiguracji CANopen i wczytywanie opisu konfiguracji do sterownika.
Obsługa konfiguracji Profibus	Aktywne: Obsługa konfiguracji Profibus i wczytywanie opisu konfiguracji do sterownika.
Pobierz jako plik	Aktywne: W trakcie pobierania konfiguracja I/O jest wczytywana do sterownika jako plik.
Obsługa multitaskingu preemptywnego	Aktywne: Obsługa konfiguracji zadania i wczytywanie opisu zadania do sterownika.
Bez sprawdzania adresów	Aktywne: W trakcie kompilacji projektu nie są sprawdzane adresy IEC.
Zmiana online	Aktywne: Funkcjonalność zmiany online.
Aktualizuj nieużywane wej./wyj.	Aktywne: CoDeSys tworzy zadanie aktualizujące i ilustrujące aktualnie nie stosowane wejścia i wyjścia w monitorowaniu.
Pojedyncze zadanie w multitaskingu	Jeszcze nie zaimplementowane.
Adresowanie bajtów	Aktywne: adresowanie następuje w bajtach (np. var1 AT %QD4 otrzymuje adres początkowy %QB4).
Inicjalizacja zerowa	Aktywne: Zasadnicza inicjalizacja za pomocą zera.
Wyślij plik symboli	Aktywne: Jeśli w trakcie pobierania zostaje utworzony plik symboli, zostaje on wczytany do sterownika.
Konfiguracja symboli z pliku INI	Aktywne: Parametry konfiguracji symboli nie są wczytywane z okna dialogowego do opcji projektu, lecz z pliku codesys.ini lub, jeśli tam podano, z innego pliku ini (patrz załącznik, konfiguracja symboli z pliku ini).

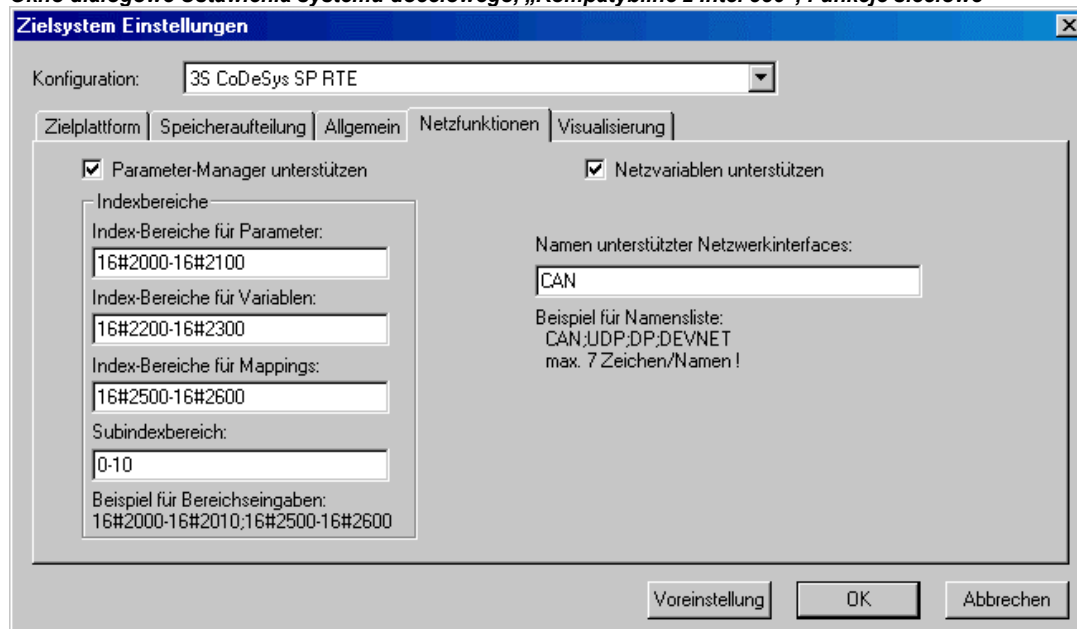
Ustawienia kategorii Ogólne

Punkt w oknie dialogowym	Znaczenie
Przeglądarka PLC	Aktywne: Funkcjonalność przeglądarki PLC.
Zapis śladu	Aktywne: Zapis śladu.
VAR_IN_OUT jako referencja	Aktywne: VAR_IN_OUT są przesyłane przy wywołaniu funkcji jako referencja (wskaźnik); dlatego też nie jest możliwe przydzielenie stałych i dostęp do odczytu i zapisu z zewnątrz.
Zainicjuj wejścia	Nieaktywne: Ze względów optymalizacji nie jest tworzony kod inicjalizacji dla wejść zadeklarowanych za pomocą „AT %IX” (-> do pierwszego cyklu magistrali wartości niezdefiniowane!).
Wczytaj autom. projekt boot.	Aktywne: Po pobraniu z nowego programu zostaje automatycznie utworzony projekt bootowalny i przesłany do sterownika.
Softmotion	Aktywne: Funkcjonalność SoftMotion jest aktywna, tzn. dostępna w zakładce Zasoby (Lista programów CNC, Krzywki tarczowe).
Utrzymać wymuszanie	Aktywne: Po wylogowaniu lista wymuszania zostaje zachowana w systemie czasu przebiegu; dla użytkownika przy wylogowaniu otwiera się odpowiednie okno dialogowe z zapytaniem; (aktualnie obsługiwane od LZS CoDeSys SP 32bit full, V2.4, Patch 4 oraz CoDeSys SP 32bit embedded V2.3, Patch 1); opcja pojawia się w oknie dialogowym, jeśli SupportPersistentForce = 1 (patrz poniżej).
Zapisz	Aktywne: System czasu przebiegu utrzymuje wymuszanie również w przypadku restartu. Opcja jest dostępna tylko dla odpowiedniego celu i można ją wybrać tylko jeśli jest aktywne „Utrzymaj wymuszanie” (patrz wyżej).
Wymuszanie niezależne od cyklu	Aktywne: Wymuszanie następuje nie tylko na początku i na końcu cyklu, lecz zostają także wyłączone wszystkie dostępy do zapisu w przebiegu programu.

13.32 Ustawienia kategorii Funkcje sieciowe

Opisane w tym miejscu wpisy można stosować do wszystkich platform standardowych.

Okno dialogowe Ustawienia systemu docelowego, „Kompatybilne z Intel 386”, Funkcje sieciowe

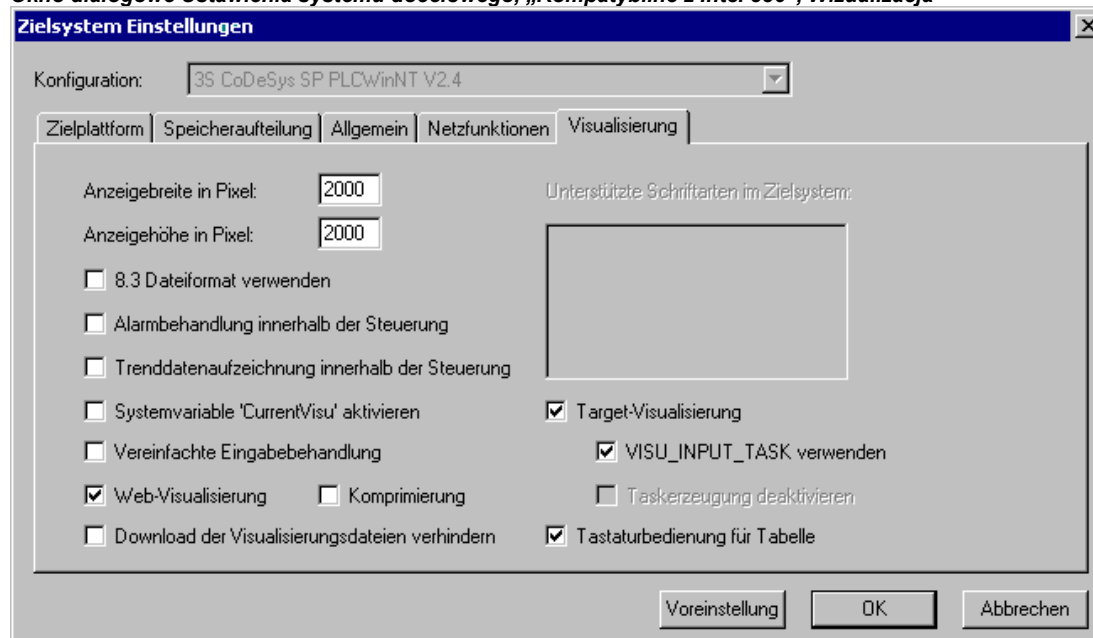


Punkt w oknie dialogowym	Znaczenie
Obsługuj menedżera parametrów	Aktywne: W zakładce Zasoby pojawia się wpis Menedżer parametrów. Umożliwia on utworzenie katalogu obiektów na zmienne i parametry służące do ukierunkowanej i aktywnej wymiany danych z innymi sterownikami.
Obsługuj zmienne sieciowe	Aktywne: Zmienne sieciowe mogą być stosowane do automatycznej wymiany danych w obrębie sieci (patrz rozdział Zasoby, Zmienne sieciowe).
Nazwy obsługiwanych interfejsów sieciowych	Lista obsługiwanych typów sieci (np. CAN, UDP).
Zakresy indeksu dla parametrów	Zakresy indeksu dla parametrów (patrz rozdział 9.11, Zasoby, Menedżer parametrów).
Zakresy indeksu dla zmiennych	Zakresy indeksu dla zmiennych (patrz rozdział 9.11, Zasoby, Menedżer parametrów).
Zakresy indeksu dla mapowania	Zakresy indeksu dla mapowania (patrz rozdział 9.11, Zasoby, Menedżer parametrów). Uwaga: Jeśli zakres tej jest zdefiniowany w tym miejscu, CanDevice uwzględni do mapowania tylko to; tzn. jeśli dodatkowo jest zdefiniowany zakres indeksu dla parametrów (patrz wyżej), nie jest on uwzględniany.
Zakres subindeksu	Zakres indeksu, który może być stosowany do subindeksu w/wym. zakresu indeksu do parametrów lub zmiennych SDO.

13.33 Ustawienia w kategorii Wizualizacja

Opisane w tym miejscu wpisy można stosować do wszystkich platform standardowych.

Okno dialogowe Ustawienia systemu docelowego, „Kompatybilne z Intel 386”, Wizualizacja



Punkt w oknie dialogowym

Znaczenie

Szer. widoku w pikselach
Wys. widoku w pikselach

Optyczne ograniczenie podanych w tym miejscu wartości jest wyświetlane przy tworzeniu wizualizacji w oknie edytora, by przykładowo zaznaczyć powierzchnię ekranu, na której ma potem być wyświetlona wizualizacja.

Zastosuj format danych 8.3

Stosowane w wizualizacji nazwy bitmap i plików języka zostają automatycznie skrócone do formatu notacji 8.3 i w ten sposób są przesyłane do systemu docelowego.

Obsługa alarmów w obrębie sterownika

Zadanie obsługi alarmów jest automatycznie tworzone w konfiguracji zadania. Przetwarzana ono utworzony pośrednio kod ST analizujący stan poszczególnych alarmów i w razie potrzeby wykonujący odpowiednie czynności. Kod ST wymaga funkcji pomocniczych z biblioteki SysLibAlarmTrend.lib, która zostaje wczytana automatycznie. (Ponadto wczytane zostają potrzebne pośrednio biblioteki SysLibSockets.lib, SysLibMem.lib, SysLibTime.lib, SysLibFile.lib. Muszą one być obsługiwane przez system docelowy.)

Jeśli opcja ta nie jest aktywna zaś aktywna jest wizualizacja docelowa, pojawia się błąd kompilacji.

Wskazówka: „Obsługę alarmów w obrębie sterownika” można używać również wtedy, gdy nie jest włączona wizualizacja docelowa lub sieciowa (patrz poniżej). Następnie również pośrednio zostaje utworzony kod ST, przejmujący analizę alarmu.

Punkt w oknie dialogowym	Znaczenie
Zapis danych trendów w obrębie sterownika	<p>W konfiguracji zadania jest tworzone zadanie TREND_TASK. Wykonuje ono pośrednio utworzony kod ST, rejestrujący dane trendów w obrębie pamięci pierścieniowej i dodatkowo, jeśli w obrębie trendu jest ustawiona opcja historia zapisuje wartości w system plików. Kod ST wymaga funkcji pomocniczych z biblioteki SysLibAlarmTrend.lib, która zostaje wczytana automatycznie. (Ponadto wczytane zostają potrzebne pośrednio biblioteki SysLibSockets.lib, SysLibMem.lib, SysLibTime.lib, SysLibFile.lib. Muszą one być obsługiwane przez system docelowy.)</p> <p>Jeśli opcja ta nie jest aktywna zaś aktywna jest wizualizacja docelowa, pojawia się błąd kompilacji.</p> <p>Wskazówka: „Zapis danych trendów w obrębie sterownika” można używać również wtedy, gdy nie jest włączona wizualizacja docelowa lub sieciowa (patrz poniżej). Następnie również pośrednio zostaje utworzony kod ST, przejmujący zapis danych trendów.</p>
Uaktywnij zmienną syst. „CurrentVisu”	<p>Jeśli ta opcja jest aktywna, do przełączania wizualizacji można stosować zmienną systemową CurrentVisu.</p>
Czcionki obsługiwane w systemie docelowym	<p>Lista fontów obsługiwanych przez system docelowy.</p>
Uproszczona obsługa wprowadzania	<p>Aktywne: W trybie online wpisy są traktowane w uproszczony sposób: aby przejść z danego pola wprowadzania danych do kolejnego, nie trzeba naciskać <Tabulatora> i <Spacji>. Wybór zostaje automatycznie przeniesiony w kolejne pole po wpisaniu danych i potwierdzeniu ich za pomocą klawisza <Enter>.</p> <p>Dodatkowo w pole wprowadzania danych można wejść za pomocą klawiszy strzałek lub tabulatora i następnie wpisać dane.</p> <p>Nieaktywne: <Tabulator> i <Spację> stosuje się do przechodzenia do kolejnego pola i przygotowania go do wpisania danych.</p>
Wizualizacja sieciowa	<p>Aktywne: Wszystkie obiekty wizualizacji w zakresie projektu zostają poddane kompilacji, by można je było stosować w wizualizacji sieciowej. (Każdy z nich można jednak w oknie dialogowym właściwości obiektu z tego wykluczyć).</p>
Kompresja	<p>Jeśli opcja ta jest aktywna, pliki do wizualizacji sieciowej do przesłania przez Codesys do serwera sieciowego/PLC są przesyłane w formie skompresowanej (zip) (w innym przypadku w formacie oryginalnym):</p> <ul style="list-style-type: none"> – XML wizualizacji – Pliki graficzne (w tym przypadku wyłącznie *.bmp ze względu na kompresję) – Plik języka (*.xml do tekstów dynamicznych, *.tlt, *.vis) <p>Pliki do istniejącej nazwy otrzymują rozszerzenie „.zip”. Kropka w nazwie pliku zostaje zastąpiona przez podkreślenie. (Przykład: nazwa „PLC_VISU.xml” zmienia się w „PLC_VISU_xml.zip”)</p> <p>Archiwa Java nie podlegają kompresji (minml.jar, webvisu.jar), to samo dotyczy strony głównej webvisu.htm.</p>
Zapobiegaj pobieraniu plików wizualizacji	<p>Jeśli opcja ta jest aktywna, pliki wizualizacji stosowane w aktualnej wizualizacji nie są wczytywane do systemu docelowego w trakcie pobierania projektu. Pliki wizualizacji są przenoszone tylko dla wizualizacji elementu docelowego lub dla wizualizacji sieciowej. Mogą to być pliki mapy bitowej, pliki języka i dla wizualizacji sieciowych także pliki opisu XML.</p>
Obsługa tabeli za pomocą klawiatury	<p>Jeśli opcja jest aktywna, w trybie online działa obsługa tabeli za pomocą klawiatury w zakresie wizualizacji (CoDeSys HMI, wizualizacja sieciowa, wizualizacja elementu docelowego).</p> <p>Wyłączenie tej opcji powoduje, że nie jest generowany kod funkcji klawiszy, co może być uzasadnione w związku z wydajnością przy wizualizacji elementu docelowego.</p>
Wizualizacja elementu docelowego	<p>Aktywne: Wszystkie obiekty wizualizacji w zakresie projektu zostają poddane kompilacji, by można je było stosować w wizualizacji elementu docelowego. (Każdy z nich można jednak w oknie dialogowym właściwości obiektu z tego wykluczyć).</p>

Punkt w oknie dialogowym	Znaczenie
Zastosuj VISU_INPUT_TASK	<p>(do włączenia, tylko przy włączonej wizualizacji elementu docelowego, patrz wyżej);</p> <p>Jeśli opcja ta jest włączona, zaś opcja „Wyłącz tworzenie zadań” (patrz niżej) jest wyłączona, do wizualizacji elementu docelowego są automatycznie generowane dwa zadania:</p> <p>VISU_INPUT_TASK steruje za pomocą ukrytego modułu MAINTARGETVISU_INPUT_CODE przetwarzaniem wpisów użytkownika, VISU_TASK steruje za pomocą ukrytego modułu MAINTARGETVISU_PAINT_CODE rysowaniem elementów wizualizacji.</p> <p>Jeśli ta opcja nie jest włączona, zostaje utworzone tylko VISU_TASK i stosowany jest wyłącznie moduł MAINTARGETVISU_PAINT_CODE, który w tym przypadku jednak przejmuje zadania MAINTARGETVISU_INPUT_CODE.</p>
Wyłącz tworzenie zadań	<p>(do włączenia, tylko przy włączonej wizualizacji elementu docelowego, patrz wyżej);</p> <p>Jeśli ta opcja jest aktywna, nie są tworzone zadania VISU_INPUT_TASK i VISU_TASK (patrz wyżej „Zastosuj VISU_INPUT_TASK”). W ten sposób można w programie sterownika bezpośrednio wywołać oba w/w. moduły lub jeśli nie jest włączona opcja „Zastosuj VISU_INPUT_TASK” tylko moduł MAINTARGETVISU_PAINT_CODE, lub w razie potrzeby może to być sterowane z dowolnego innego zadania. Patrz opis wizualizacji elementu docelowego (w podręczniku wizualizacji CoDeSys).</p>

Załącznik J Obsługa za pomocą klawiatury

13.34 Obsługa za pomocą klawiatury

W celu obsługi CoDeSys wyłącznie za pomocą klawiatury należy skorzystać z kilku komend, których nie można znaleźć w menu.

- Za pomocą klawisza funkcyjnego <F6> można przechodzić w otwartym module między deklaracją a instrukcjami. W Menedżerze parametrów przechodzi się w ten sposób między oknem nawigacji a edytorem listy.
- Za pomocą <Alt> + <F6> przechodzi się z w otwartego obiektu do Organizera obiektów i stamtąd do Okna komunikatów, jeśli jest ono otwarte. Jeśli jest otwarty dialog wyszukiwania, za pomocą <Alt> + <F6> przechodzi się w Organizera obiektów do dialogu wyszukiwania i stamtąd do obiektu.
- Za pomocą <Tabulatora> przechodzi się w dialogach do pól wprowadzania danych i ikon.
- Za pomocą Klawiszy strzałek można poruszać się w obrębie Organizera obiektów i Menedżera bibliotek przez zakładki i obiekty.

Wszystkie inne akcje można wykonywać za pomocą komend menu lub skrótów umieszczonych za komendami menu. Za pomocą <Shift> + <F10> można otworzyć menu kontekstowe z najczęściej stosowanymi komendami dotyczącymi zaznaczonego obiektu lub aktywnego edytora.

13.35 Kombinacje klawiszy

W tym miejscu można znaleźć zestawienie wszystkich kombinacji klawiszy i klawiszy funkcyjnych:

Ogólna obsługa

Przechodzenie z deklaracji do instrukcji modułu	<F6>
Przechodzenie z Organizera obiektów do obiektów i okna komunikatów	<Alt> + <F6>
Menu podręczne	<Shift> + <F10>
Przechodzenie do kolejnego otwartego okna edytora	<Ctrl> + <F6>
Przechodzenie do poprzedniego otwartego okna edytora	<Ctrl> + <Shift> + <F6>
Tryb skróconej formy deklaracji	<Ctrl> + <Enter>
Przechodzenie od komunikatu w oknie komunikatów do odpowiedniego miejsca w edytorze	<Enter>
Rozwijanie i zwijanie zmiennych wielopoziomowych	<Enter>
Otwieranie i zamykanie folderów	<Enter>
Przechodzenie pomiędzy zakładkami w Organizercie obiektów i Menedżerze obiektów	<Klawisze strzałek>
Poruszanie się w oknach dialogowych	<Tabulator>
Pomoc kontekstowa	<F1>

Polecenia ogólne

„Plik” „Zapisz”	<Ctrl> + <S>
„Plik” „Drukuj”	<Ctrl> + <P>
„Plik” „Zakończ”	<Alt> + <F4>
„Projekt” „Sprawdź wszystko”	<Ctrl> + <F11>

Kombinacje klawiszy

„Projekt” „Kompiluj”	<Shift> + <F11>
„Projekt” „Kompiluj wszystko”	<F11>
„Projekt” „Usuń obiekt”	
„Projekt” „Wstaw obiekt”	<Ins>
„Projekt” „Zmień nazwę obiektu”	<Spacja>
„Projekt” „Edytuj obiekt”	<Enter>
„Edycja” „Cofnij”	<Ctrl> + <Z>
„Edycja” „Przywróć”	<Ctrl> + <Y>
„Edycja” „Wytnij”	<Ctrl> + <X> lub <Shift> +
„Edycja” „Kopiuj”	<Ctrl> + <C>
„Edycja” „Wklej”	<Ctrl> + <V>
„Edycja” „Usuń”	
„Edycja” „Znajdź następny”	<F3>
„Edycja” „Asystent deklaracji”	<F2>
„Edycja” „Deklaracja zmiennych”	<Shift><F2>
„Edycja” „Następny błąd”	<F4>
„Edycja” „Poprzedni błąd”	<Shift> + <F4>
„Online” „Zaloguj”	<Alt> + <F8>
„Online” „Wyloguj”	<Ctrl> + <F8>
„Online” „Start”	<F5>
„Online” „Wstaw/usuń punkt wstrzymania”	<F9>
„Online” „Krok pojedynczy nad”	<F10>
„Online” „Krok pojedynczy w”	<F8>
„Online” „Cykl pojedynczy”	<Ctrl> + <F5>
„Online” „Zapisz wartości”	<Ctrl> + <F7>
„Online” „Wymuszanie wartości”	<F7>
„Online” „Anuluj wymuszanie”	<Ctrl> + <Shift> + <F7>
„Online” „Okno dialog. zapisywanie/wymuszanie”	<Shift> + <F7>
„Okno” „Komunikaty”	<Shift> + <Esc>

Polecenia edytora FBD

„Wstaw” „Sieć (za)”	<Shift> + <T>
„Wstaw” „Przyporządkowanie”	<Ctrl> + <A>
„Wstaw” „Skok”	<Ctrl> + <L>
„Wstaw” „Powrót”	<Ctrl> + <R>
„Wstaw” „Operator”	<Ctrl> + <O>
„Wstaw” „Funkcja”	<Ctrl> + <F>
„Wstaw” „Blok funkcji”	<Ctrl> +
„Wstaw” „Wejście”	<Ctrl> + <U>

„Dodatki” „Negacja”	<Ctrl> + <N>
„Dodatki” „Powiększ”	<Alt> + <Enter>

Polecenia edytora CFC

„Wstaw” „Moduł”	<Ctrl> +
„Wstaw” „Wejście”	<Ctrl> + <E>
„Wstaw” „Wyjście”	<Ctrl> + <A>
„Wstaw” „Skok”	<Ctrl> + <G>
„Wstaw” „Etykieta”	<Ctrl> + <L>
„Wstaw” „Powrót”	<Ctrl> + <R>
„Wstaw” „Komentarz”	<Ctrl> + <K>
„Wstaw” „Wejście modułu”	<Ctrl> + <U>
„Dodatki” „Negacja”	<Ctrl> + <N>
„Dodatki” „Ustaw/Przywróć”	<Ctrl> + <T>
„Dodatki” „Połączenie”	<Ctrl> + <M>
„Dodatki” „EN/ENO”	<Ctrl> + <E>
„Dodatki” „Powiększ”	<Alt> + <Enter>

Polecenia edytora LD

„Wstaw” „Sieć (za)”	<Shift> + <T>
„Wstaw” „Styk”	<Ctrl> + <K>
„Wstaw” „Styk równoległy”	<Ctrl> + <R>
„Wstaw” „Blok funkcji”	<Ctrl> +
„Wstaw” „Cewka”	<Ctrl> + <L>
„Dodatki” „Wklej pod”	<Ctrl> + <U>
„Dodatki” „Negacja”	<Ctrl> + <N>
„Dodatki” „Powiększ”	<Alt> + <Enter>

Polecenia edytora AS

„Wstaw” „Przejście kroku (przed)”	<Ctrl> + <T>
„Wstaw” „Przejście kroku (po)”	<Ctrl> + <E>
„Wstaw” „Gałąź alternatywna (w prawo)”	<Ctrl> + <A>
„Wstaw” „Gałąź równoległa (w prawo)”	<Ctrl> + <L>
„Wstaw” „Skok” (AS)	<Ctrl> + <U>
„Dodatki” „Powiększ akcję/przejście”	<Alt> + <Enter>
Przejdź z Przeglądu AS z powrotem do edytora	<Enter>

Obsługa konfiguracji sterownika lub zadań

Rozwijanie i zwijanie elementów organizacji	<Enter>
Wstawianie obramowania wokół nazwy	<Spacja>

Kombinacje klawiszy

Obsługa Edytora Menedżera parametrów

Przechodzenie z okna nawigacji do Edytora listy

<F6>

Usuwanie wiersza z Edytora listy

<Ctrl> + , <Shift> +

Usuwanie pola

Załącznik K

Załącznik L Zalecenia odnośnie nadawania identyfikatorów

13.36 Nadawanie identyfikatorów w trakcie deklaracji

Identyfikatory są nadawane w trakcie deklaracji zmiennych ([Nazw zmiennych](#)), zdefiniowanych przez użytkownika [Typów danych](#) i w trakcie tworzenia [POU](#) (moduły: funkcje, bloki funkcji, programy) i wizualizacji. W celu jak największego ujednoczenia przy nadawaniu nazw zalecane jest stosowanie się następujących reguł:

13.37 Identyfikator zmiennych (nazw zmiennych)

Nazwy zmiennych we wszystkich aplikacjach powinny w miarę możliwości opierać się na **notacji węgierskiej**:

Dla każdej zmiennej należy znaleźć logiczny, możliwie krótki opis w języku angielskim **Nazwę podstawową**. Pierwsza litera słowa winna być napisana z wielkiej litery, pozostałe zaś z małych (przykład: FileSize). W razie potrzeby można dodatkowo utworzyć plik kompilacji na inne języki.

Do tej nazwy podstawowej, odpowiednio do typu danych w zakresie zmiennej zostaje dołączony **Prefiks (prefiksy)** pisany małymi literami.

<i>Typ danych</i>	<i>Dolna granica</i>	<i>Górna granica</i>	<i>Treść informacji</i>	<i>Prefiks</i>	<i>Wskazówka</i>
BOOL	FALSE	TRUE	1 bit	x	
				b	zarezerwowane
BYTE			8 bitów	by	Bitstring, nie do operacji arytmetycznych
WORD			16 bitów	w	Bitstring, nie do operacji arytmetycznych
DWORD			32 bity	dw	Bitstring, nie do operacji arytmetycznych
LWORD			64 bity	lw	nie do operacji arytmetycznych
SINT	-128	127	8 bitów	si	
USINT	0	255	8 bitów	usi	
INT	-32.768	32.767	16 bitów	i	
UINT	0	65.535	16 bitów	ui	
DINT	-2.147.483.648	2.147.483.647	32 bity	di	
UDINT	0	4.294.967.295	32 bity	udi	
LINT	-2^{63}	$2^{63} - 1$	64 bity	li	
ULINT	0	$2^{64} - 1$	64 bity	uli	
REAL			32 bity	r	
LREAL			64 bity	lr	

Identyfikator zmiennych (nazw zmiennych)

<i>Typ danych</i>	<i>Dolna granica</i>	<i>Górna granica</i>	<i>Treść informacji</i>	<i>Prefiks</i>	<i>Wskazówka</i>
STRING				s	
TIME				tim	
TIME_OF_DAY				tod	
DATETIME				dt	
DATE				date	
ENUM			16 bitów	e	

* Dla zmiennych typu BOOL świadomie wybrano jako prefiks x, by po pierwsze znaleźć odgraniczenie od BYTE, zaś po drugie by zastosować perspektywę programisty IEC (porównaj Adresowanie %IX0.0).

Przykłady:

```
bySubIndex: BYTE;
sFileName: STRING;
udiCounter: UDINT;
```

W przypadku **deklaracji zagnieżdżonych** prefiksy są dołączane zgodnie z kolejnością deklaracji:

<i>Typ</i>	<i>Dolna granica</i>	<i>Górna granica</i>	<i>Pamięć</i>	<i>Prefiks</i>	<i>Wskazówka</i>
POINTER				p	
ARRAY				a	

Przykład:

```
pabyTelegramData: POINTER TO ARRAY [0..7] OF BYTE;
```

Instancje bloków funkcji oraz **zmienne** zdefiniowanych przez użytkownika typów danych zawierają jako skrót identyfikatora nazwy bloku funkcji lub typu danych (przykład: sdo).

Przykład:

```
cansdoReceivedTelegram: CAN_SDOTelegram;
TYPE CAN_SDOTelegram :      (* prefix: sdo *)
STRUCT
  wIndex:WORD;
  bySubIndex:BYTE;
  byLen:BYTE;
  aby: ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE
```

Stałe lokalne (c) rozpoczynają się od prefiksu stałej c i dodatkowego podkreślenia, po którym następuje typ prefiksu i nazwa zmiennej.

Przykład:

```
VAR CONSTANT
  c_uiSyncID: UINT := 16#80;
END_VAR
```

Dla **Zmiennych globalnych** (g) oraz **Stałych globalnych** (gc) do prefiksu biblioteki zostaje dołączony dodatkowy prefiks + podkreślenie.

Przykłady:

```

VAR_GLOBAL
    CAN_g_iTest: INT;
END_VAR
VAR_GLOBAL CONSTANT
    CAN_gc_dwExample: DWORD;
END_VAR

```

13.38 Identyfikator zdefiniowanych przez użytkownika typów danych (DUT)

Nazwa każdego **Typu danych struktury** składa się z prefiksu biblioteki (przykład: CAN), podkreślenia i krótkiego, logicznego opisu (przykład: SDOTelegram) struktury. Odpowiedni prefiks utworzonej zmiennej tej struktury powinien nastąpić bezpośrednio po dwukropku jako komentarz.

Przykład:

```

TYPE CAN_SDOTelegram :      (* prefix: sdo *)
STRUCT
    wIndex:WORD;
    bySubIndex:BYTE;
    byLen:BYTE;
    abyData: ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE

```

Wartości enumeracji rozpoczynają się od prefiksu biblioteki (przykład: CAL), po którym następuje podkreślenie i identyfikator pisany wielkimi literami.

W poprzednich wersjach CoDeSys wartości ENUM > 16#7FFF powodowały błędy, ponieważ nie można było dokonać ich automatycznej konwersji na INT. Z tego względu ENUM należy zawsze definiować z poprawnymi wartościami INT.

Przykład:

```

TYPE CAL_Day : (
    CAL_MONDAY,
    CAL_TUESDAY,
    CAL_WEDNESDAY,
    CAL_THURSDAY,
    CAL_FRIDAY,
    CAL_SATURDAY,
    CAL_SUNDAY);

```

Deklaracja:

```
eToday: CAL_Day;
```

13.39 Identyfikator funkcji, bloków funkcji, programów (POU)

Funkcje, bloki funkcji i programy składają się z prefiksu biblioteki (przykład: CAN), podkreślenia i krótkiej, logicznej nazwy POU (przykład: SendTelegram). Jak w przypadku zmiennych pierwsza litera słowa w nazwie podstawowej winna być napisana wielkimi, zaś pozostałe małymi literami. Zalecane jest utworzenie nazwy POU z czasownika i rzeczownika.

Przykład:

```
FUNCTION_BLOCK CAN_SendTelegram (* prefix: canst *)
```

Identyfikator wizualizacji

Część deklaracji jako komentarz zawiera **Skrócony opis** modułu. Ponadto wszystkie **wejścia** i **wyjścia** są opatrzone **komentarzami**. W przypadku bloków funkcji odpowiednie prefiks utworzonej instancji powinien być umieszczony jako komentarz bezpośrednio po nazwie.

Czynności zasadniczo nie zawierają prefiksu; jedynie czynności, wywoływane wewnętrznie przez sam moduł rozpoczynają się od „prv_”.

Każda **funkcja** ze względu na zgodność z wcześniejszymi wersjami CoDeSys musi mieć co najmniej jeden parametr transferu. **Funkcje zewnętrzne** jako wartość zwracaną nie mogą stosować struktur.

13.40 Identyfikator wizualizacji

Wskazówka: Tymczasem należy zwrócić uwagę na to, by wizualizacja nie miała tej samej nazwy co inny moduł w zakresie projektu, ponieważ mogłoby to prowadzić do problemów przy zmianie wizualizacji.

Załącznik M Błędy i ostrzeżenia w zakresie kompilacji

Przy Kompilowaniu projektu w Oknie komunikatów pojawiają się komunikaty ewentualnych błędów lub ostrzeżeń. Za pomocą <F4> można przejść do kolejnej linii komunikatu – otwiera się przy tym okno z odpowiednim miejscem w programie. Komunikaty błędów i ostrzeżenia są poprzedzane w oknie komunikatów jednoznacznym identyfikatorem. Jeśli taka linia komunikatu jest zaznaczona, można za pomocą <F1> otworzyć odpowiednie okno pomocy.

13.41 Ostrzeżenia...

1100

„Nieznana funkcja «<Name>» w bibliotece.”

Stosowana jest zewnętrzna biblioteka. Należy sprawdzić, czy wszystkie funkcje podane w pliku .hex są zdefiniowane również w pliku .lib.

1101

„Niejasny symbol «<Symbol>».”

Generator kodu oczekuje modułu o nazwie <Symbol>. Nie jest on zdefiniowany w projekcie. Należy zdefiniować funkcję/program o odpowiedniej nazwie.

1102

„Nieprawidłowy interfejs dla symbolu «<Symbol>».”

Generator kodu oczekuje elementu o nazwie <Symbol> z dokładnie skalarnym wejściem lub programem o nazwie <Symbol> bez wejścia lub wyjścia.

1103

„Stała «<Name>» przy adresie kodu <%04X %04X> jest powyżej granicy strony 16K!”

Stała stringu powyżej granicy strony 16K. Obsługa przez system niemożliwa. Zależnie od systemu czasu przebiegu może istnieć możliwość obejścia za pomocą wpisu w pliku docelowym. W tej kwestii należy zwrócić się do producenta sterownika.

1200

„Zadanie «<Name>», wywołanie «<Name>». Zmienne dostępu do listy parametrów nie są aktualizowane”

Zmienne stosowane wyłącznie przy wywołaniu modułu funkcyjnego z konfiguracji zadań nie są wpisywane do listy odsyłaczy.

1300

„Nie znaleziono pliku «<Name>»”

Plik odsyłający do globalnego projektu zmiennych nie istnieje. Należy sprawdzić ścieżkę.

1301

„Nie znaleziono biblioteki analiz. Nie można utworzyć kodu analizy.”

Stosowana jest funkcja analizy, jednak brak jest biblioteki analization.lib. Bibliotekę należy dodać w zarządzaniu bibliotekami.

1302

„Wstawiono nowe funkcje z zewnętrznymi odwołaniami. Zmiana online nie jest możliwa!”

Po ostatnim pobraniu powiązано bibliotekę zawierającą funkcje, do których brak jest odwołań w systemie czasu przebiegu. Z tego względu konieczne jest pobranie całego projektu.

- 1400**
„Nieznana komenda kompilatora «<Name>» zostanie zignorowana!”
 Pragma nie jest obsługiwana przez kompilator. Odnośnie obsługiwanych komend patrz hasło „Pragma”.
- 1401**
„Struktura «<Name>» nie zawiera elementów.”
 Struktura nie zawiera żadnych elementów, zmienne tego typu zajmują jednak 1 bajt pamięci.
- 1410**
„«RETAIN» i «PERSISTENT» nie mają wpływu na funkcje.”
 Zmienne zadeklarowane w obrębie funkcji jako remanentne nie są zachowywane w zakresie Retain, lecz traktowane jak zwykle zmienne lokalne.
- 1411**
„Zmienna «<name>» w konfiguracji zmiennych nie jest aktualizowana w obrębie żadnego zadania”
 Najwyższa instancja zmiennej nie posiada w żadnym zadaniu odwołania poprzez wywołanie i tym samym nie jest kopiowana z odwzorowania procesu.
Przykład:
 Konfiguracja zmiennych:

```
VAR_CONFIG
  plc_prg.aprg.ainst.in AT %IB0 : INT;
END_VAR

plc_prg:
  index := INDEXOF(aprg);
```

 Istnieje wprowadzone odwołanie programu aprg, lecz nie jest ono wywoływane. Z tego względu plc_prg.aprg.ainst.in nigdy nie będzie zawierać rzeczywistej wartości %IB0.
- 1412**
„Nieoczekiwany token «<Name>» w pragmie {Pragmaname}”
 Stosowana jest nieprawidłowa nazwa pragmy lub pragma, której w tym miejscu nie można zastosować. W razie potrzeby należy przy korekcie wykorzystać opisy hasła „Pragma” zawarte w pomocy online lub w podręczniku użytkownika CoDeSys.
- 1413**
„«<Name>» nie jest prawidłowym kluczem do listy «<Name>». Klucz jest ignorowany”
 W pragmie podano nieistniejącą listę parametrów. Należy sprawdzić nazwę listy lub sprawdzić w menedżerze parametrów, jakie listy są dostępne.
- 1414**
Zbyt wiele definicji komponentów w pragmie „<pragmaanweisung>”
 Pragma zawiera więcej definicji (w nawiasach kwadratowych) niż liczba elementów w odpowiedniej tablicy, bloku funkcji lub strukturze elementów.
- 1500**
„To wyrażenie nie zawiera przyporządkowania. Kod nie jest generowany.”
 Nie jest stosowany rezultat tego wyrażenia. Dlatego dla całego wyrażenia nie jest generowany kod.

1501

„Stała stringu jest przekazywana jako VAR_IN_OUT: Nie wolno zastąpić «<Name>»!”

Stałej nie można opisać w korpusie modułu, ponieważ tam nie jest możliwy test wielkości.

1502

„Zmienna «<Name>» ma tę samą nazwę co moduł. Moduł nie jest wywoływany!”

Stosowana jest zmienna o tej samej nazwie co moduł.

Przykład:

PROGRAM a

...

VAR_GLOBAL

a: INT;

END_VAR

...

a; (* Nie jest wywoływany moduł a, lecz załadowywana jest zmienna a. *)

1503

„Moduł nie ma wyjść, połączenie jest kontynuowane z parametrem TRUE.”

Pin wyjściowy modułu zostaje połączony bez wyjść w FBD lub LD. Do połączenia zostaje automatycznie przypisana wartość TRUE.

1504

„Instrukcja może nie zostać wykonana, zależnie od wyrażenia logicznego”

W pewnych okolicznościach nie są wykonywane wszystkie odgałęzienia wyrażenia logicznego.

Przykład:

IF a AND funct(TRUE) THEN ...

Jeśli a jest FALSE, funct nie jest wywoływane.

1505

„Efekt uboczny w «<Name>»! Odgałęzienie może nie zostać obliczone”

Pierwsze wejście modułu to FALSE, dlatego odgałęzienie boczne, wchodzące do drugiego wejścia może ewentualnie nie być obliczane.

1506

„Zmienna «<Name>» ma taką samą nazwę, jak czynność lokalna. Czynność nie jest wywoływana!”

By zapewnić, że nie będą stosowane takie same nazwy należy zmienić nazwę zmiennej lub czynności.

1507

„Instancja «<Name>» ma nazwę jak funkcja. Instancja nie jest wywoływana!”

W ST jest wywoływana instancja o takiej samej nazwie co funkcja. Wywoływana jest funkcja! Należy nadać różne nazwy.

1550

„Wielokrotne wywołanie tego samego modułu «<Name>» może prowadzić do niepożądanych wzajemnych oddziaływań”

Należy sprawdzić, czy rzeczywiście jest konieczne wielokrotne stosowanie tego modułu. Wskutek wielokrotnego wywołania w jednym cyklu może dojść do niepożądanego nadpisywania wartości.

1600

„Otwarta baza danych niejasna (wygenerowany kod może być błędny)”

Z oryginalnego programu Siemens nie wynika, który moduł danych jest otwarty.

1700

„Wejście niepołączone.”

W CFC jest stosowany element wejściowy, który nie jest dalej połączony. Dla niego nie jest generowany kod.

1750

„Krok «<Name>»: czas minimalny jest większy od maksymalnego!”

W podanym kroku należy otworzyć okno dialogowe „Atrybuty kroku” i skorygować dane czasu.

1751

„Ostrożnie przy stosowaniu zmiennej «<Name>». Zmienna ta jest stosowana przez kod utworzony wewnętrznie i ma wpływ na przebieg łańcucha kroków!”

Ze względów bezpieczeństwa należy zmienić nazwę zmiennej, by zawierała ona jednoznaczny identyfikator.

1800

„<Name> (Element #<Elementnummer>) (nazwa, element, numer elementu): Nieprawidłowe wyrażenie kontrolne «<Name>»”

Element wizualizacji zawiera wyrażenie, którego nie można monitorować. Należy sprawdzić nazwę zmiennej i zastąpione wypełniacze.

1801

„Nie jest możliwy wpis do wyrażenia.”

W konfiguracji obiektu wizualizacji jako cel wpisu stosowane jest wyrażenie złożone. Należy je zastąpić jedną zmienną.

1802

„<Visualisierungsobjekt>(Elementnummer): Nie znaleziono bitmapy «<Name>»”

Należy się upewnić, że zewnętrzny plik bitmapy jest w lokalizacji, do której odwołuje się ścieżka w wizualizacji.

1803

„«<name>» («<nummer>»): Czynność drukowania nie jest obsługiwana w przypadku wizualizacji docelowej oraz sieciowej.”

Alarm skonfigurowany w wizualizacji jest połączony z czynnością drukowania. Nie jest to uwzględniane w wizualizacji elementu docelowego lub sieciowej.

1804

„«<name>» («<nummer>»): System docelowy nie obsługuje zestawu znaków «<name>».”

W wizualizacji jest stosowana czcionka, której nie obsługuje system docelowy. Obsługiwane przez system docelowy zestawy znaków są wyświetlane w ustawieniach systemu docelowego, kategoria „Wizualizacja”.

1807

„<name> (<nummer>): Brak okna komunikatu w przypadku alarmów w obrębie wizualizacji docelowej.”

Należy zwrócić uwagę, iż czynność „Okno komunikatu” nie jest obsługiwana w Wizualizacji elementu docelowego!

1808

„<<name>> (<<nummer>>): Wielokąt do wizualizacji docelowej składa się ze zbyt wielu punktów. W przypadku instrumentu wskaźnikowego należy sprawdzić konfigurację.”

Standardowo dozwolone jest maksymalnie 512 punktów, docelowo można zdefiniować inną najwyższą liczbę. Otwarcie konfiguracji powoduje optymalizację stosownie do dozwolonej liczby.

1809

„<<name>>(<<nummer>>): Dla zmiany wizualizacji został skonfigurowany nieprawidłowy cel: <<nummer>>”

Podana wizualizacja nie istnieje. Należy sprawdzić, czy zgadza się nazwa lub czy wizualizacja istnieje.

1850

„Zmienna wejściowa w %IB<nummer> jest używana w zadaniu <<Name>>, lecz aktualizowana w innym zadaniu.”

Należy sprawdzić, które zadania stosują podaną zmienną i czy aktualne programowanie nie prowadzi do niepożądanych efektów. Aktualizacja wartości zmiennej następuje z reguły za pomocą zadania o wyższym priorytecie.

1850

„Zmienna wyjściowa w %IQ<nummer> jest używana w zadaniu <<name>>, lecz aktualizowana w innym zadaniu.”

Należy sprawdzić, które zadania stosują podaną zmienną i czy aktualne programowanie nie prowadzi do niepożądanych efektów. Aktualizacja wartości zmiennej następuje z reguły za pomocą zadania o wyższym priorytecie.

1852

„CanOpenMaster w zadaniu zdarzeniu <<Name>> w pewnych okolicznościach nie jest wywoływane cyklicznie! Ustawić parametry modułu UpdateTask!”

Aktualnie CanOpen Master jest wywoływane za pomocą zadania zdarzenia. Jeśli wymagane jest wywołanie cykliczne, odpowiednie zadanie należy zdefiniować za pomocą parametru UpdateTask w konfiguracji sterownika w dialogu „Parametry modułu”.

1853

„PDO (Indeks: <Zahl>) w zadaniu zdarzeniu <<Name>> w pewnych okolicznościach nie jest aktualizowane cyklicznie.”

Aktualnie wymienione PDO jest wywoływane za pomocą wymienionego zadania zdarzenia. Jeśli jednak wymagane jest wywołanie cykliczne, PDO należy przydzielić odpowiednie zadanie poprzez przesunięcie referencji IO do tego zadania.

1900

„POU <<Name>> (funkcja wejściowa) nie jest dostępny w bibliotece”

Moduł wstawiany (np. PLC_PRG) nie jest dostępny w trakcie stosowania biblioteki.

1901

„Zmienne dostępu oraz zmienne konfiguracji nie są zapisywane w bibliotece!”

Zmienne dostępu oraz konfiguracja zmiennych nie są zapisywane w bibliotece.

- 1902**
„<<Name>>: Biblioteka nie jest odpowiednia dla bieżącego typu maszyny!”
Plik .obj w bibliotece został utworzony dla innego typu maszyny.
- 1903**
„<<Name>>: niepoprawna biblioteka”
Plik nie odpowiada wymaganemu formatowi pliku w systemie docelowym.
- 1904**
„Stała <<Name>> zasłania taką samą stałą w bibliotece”
W projekcie zadeklarowano stałą o takiej samej nazwie, jak stała zawarta w powiązanej bibliotece. Zmienna biblioteki zostaje nadpisana!
- 1970**
„Manager parametrów: nie można zaimportować listy <<Name>>, kolumna <<Name>>, wartość <<Name>>!”
Należy sprawdzić importowany plik *.prm pod kątem wpisów, nie pasujących do aktualnej konfiguracji (wartości standardowe lub plik opisów XML) menedżera parametrów.
- 1980**
„Globalne zmienne sieciowe <<name>> <<name>>: Równoczesny odczyt i zapis mogą spowodować utratę danych!”
W trakcie konfiguracji listy zmiennych sieciowych (wybrać listę z arkusza rejestru zasobów i otworzyć okno dialogowe Lista zmiennych globalnych za pomocą komendy „Właściwości” w menu kontekstowym) włączono zarówno opcję „Odczyt”, jak i „Zapis”. Należy pamiętać, że może to spowodować utratę danych w trakcie komunikacji.
- 1990**
„Brak <<VAR_CONFIG>> dla <<Name>>”
Dla tej zmiennej brak konfiguracji adresu w Konfiguracja zmiennej VAR_CONFIG. Należy otworzyć w zakładce „Zasoby” okno VAR_CONFIG i wstawić (komenda „Wstaw” „Wszystkie ścieżki instancji”).
- 2500**
„Zadanie <<task name>>: dla zadania cyklicznego nie podano czasu cyklu.”
W konfiguracji zadań zostało utworzone zadanie cykliczne, dla którego nie został zdefiniowany czas cyklu. W oknie dialogowym właściwości zadania w „Interwał” należy wpisać odpowiedni interwał czasu.

13.42 Błąd kompilacji...

- 3100**
„Program jest zbyt duży. Maksymalny rozmiar: <<Anzahl>> bajtów (<Anzahl>K)”
Przekroczono maksymalny rozmiar programu. Program należy zmniejszyć.
- 3101**
„Zakres danych jest zbyt duży. Maksymalny rozmiar: <<Anzahl>> bajtów (<Anzahl>K)”
Zbyt mało pamięci. Należy zmniejszyć zapotrzebowanie aplikacji na dane.

3110

„Błąd w pliku biblioteki «<Name>».”

Plik .hex nie odpowiada formatowi Hex firmy INTEL.

3111

„Biblioteka «<Name>» jest zbyt duża. Maksymalny rozmiar: 64K”

Plik .hex przekracza maksymalny możliwy rozmiar.

3112

„W bibliotece znajduje się instrukcja bez możliwości przeniesienia.”

Plik .hex zawiera instrukcję, której nie można relokować. Kodu biblioteki nie można powiązać.

3113

„Kod biblioteki nadpisuje tabele funkcji.”

Zakresy kodu i tabele informacji nakładają się na siebie.

3114

„Biblioteka wykorzystuje więcej niż jeden segment.”

Tabele i kody zawarte w pliku .hex wykorzystują więcej niż jeden segment.

3115

„Stałej nie można przypisać do VAR_IN_OUT. Niekompatybilne typy danych.”

Wewnętrznego formatu wskaźnika do stałych stringów nie można skonwertować na wewnętrzny format wskaźnika VAR_IN_OUT, ponieważ dane są zdefiniowane jako „near”, a stałe stringów jako „huge” lub „far”. Jeśli jest to możliwe, należy zmienić ustawienia systemu docelowego.

3116

„Tabele funkcji nadpisują kod biblioteki lub granicę segmentu.”

Kod 166x: Biblioteka zewnętrznej nie można stosować w ten sposób. Należy dopasować ustawienia systemu docelowego lub utworzyć nową bibliotekę z innymi ustawieniami.

3120

„Aktualny segment kodu przekracza rozmiar 64K.”

Właśnie wygenerowany kod systemu ma rozmiar ponad 64K. Może być potrzebna zbyt duża część kodu inicjalizacji.

3121

„Moduł jest zbyt duży.”

Rozmiar modułu nie może przekraczać 64K.

3122

„Inicjalizacja jest zbyt duża. Maksymalny rozmiar: 64K”

Kod inicjalizacji modułu funkcyjnego lub struktury nie może przekraczać 64K.

3123

„Segment danych jest zbyt duży: Segment «<Nummer>», rozmiar <Name des Datenbereichs> bajty(ów) (maksimum <Name des Datenbereichs> bajty(ów))”

Należy zwrócić się do producenta sprzętu.

- 3124**
„Stała stringu jest zbyt duża: <Anzahl> znaków (maksimum 253 znaki)”
Odpowiednią stałą należy skrócić do co najmniej 253 znaków.
- 3130**
„Stos aplikacji jest zbyt mały: «<Anzahl>» DWORD potrzebne, «<Anzahl>» DWORD dostępne.”
Zbyt duża głębokość zagnieżdżenia wywołań modułu. W ustawieniach systemu docelowego należy zwiększyć wielkość stosu lub skompilować program bez opcji kompilacji projektu „Debug”.
- 3131**
„Stos użytkownika jest zbyt mały: «<Anzahl>» WORD potrzebne, «<Anzahl>» WORD dostępne.”
Należy zwrócić się do producenta sterownika.
- 3132**
„Stos systemu jest zbyt mały: «<Anzahl>» WORD potrzebne, «<Anzahl>» WORD dostępne.”
Należy zwrócić się do producenta sterownika.
- 3150**
„Parametr <Zahl> funkcji «<Name>»: Rezultatu funkcji IEC nie można przekazać jako parametru stringu do funkcji C.”
Należy zastosować zmienną pośrednią, do której odnosi się rezultat funkcji IEC.
- 3160**
„Nie można otworzyć pliku biblioteki «<Name>».”
Nie można znaleźć pliku «<Name>» potrzebnego do biblioteki.
- 3161**
„Biblioteka «<Name>» nie zawiera segmentu kodu”
Plik .obj biblioteki musi zawierać co najmniej jedną funkcję C. Należy wstawić w plik .obj funkcję dummy, która nie jest zdefiniowana w pliku .lib.
- 3162**
„Nie można odczytać referencji w bibliotece «<Name>» (symbol «<Name>», klasa «<Name>», typ «<Name>»)”
Plik .obj zawiera nieczytelne odniesienie do innego symbolu. Sprawdzić ustawienia kompilatora C.
- 3163**
„Nieznany typ referencji w bibliotece «<Name>» (symbol «<Name>», klasa «<Name>», typ «<Name>»)”
Plik .obj zawiera nieczytelny dla generatora kodu typ referencji. Sprawdzić ustawienia kompilatora C.
- 3200**
„<Name> (<Zahl>): Wyrażenie logiczne jest zbyt złożone.”
Tymczasowa pamięć systemu docelowego nie jest wystarczająca dla rozmiaru wyrażenia. Wyrażenie należy podzielić na więcej wyrażen częściowych z odniesieniami do zmiennych pośrednich.
- 3201**
„<Name> (<Netzwerk>): Sieć może wytwarzać kod maks. 512-bajtowy”
Nie można wykonać wewnętrznych skoków. W ustawieniach docelowych 68k należy aktywować opcję „Stosować 16-bitowe offsety skoku”.

3202

„Przepełnienie stosu w przypadku powiązanych sekwencji / tablic / struktur wywołań funkcji”

Stosowane jest powiązane wywołanie funkcji w formie CONCAT(x, f(i)). Może to prowadzić do utraty danych. Wywołanie należy podzielić na dwa wyrażenia.

3203

„Przyporządkowanie zbyt złożone (zbyt wiele wymaganych rejestrów adresów)”

Przyporządkowanie należy podzielić na mniejsze części.

3204

„Skok ma długość ponad 32 kB”

Wielkość skoku nie może przekraczać 32 767 bajtów.

3205

„Błąd wewnętrzny: Zbyt wiele stałych stringów”

W module można zastosować maksymalnie 3000 stałych stringów.

3206

„Blok funkcji jest zbyt duży”

Blok funkcji może wytwarzać maks. kod 32 767-bajtowy.

3207

„Optymalizacja tablicy”

Optymalizacja dostępu do tablicy nieudana, ponieważ w trakcie obliczania indeksu została wywołana funkcja.

3208

„Przekształcenie nie zostało implementowane”

Stosowana jest funkcja konwersji, która jest nie implementowana w aktualnym generatorze kodu.

3209

„Operator nie został implementowany”

Stosowany jest operator, który nie jest implementowany dla tego typu plików w aktualnym generatorze kodu: MIN(string1,string2).

3210

„Nie znaleziono funkcji «<Name>»”

Wywoływana jest funkcja, która nie istnieje w projekcie.

3211

„Zmienna sekwencji używana zbyt często”

Zmienna typu string może być stosowana w generatorze kodu 68K tylko 10 razy w wyrażeniu.

3212

„Błędna kolejność bibliotek w module «<Bausteinname>»”

Kolejność bibliotek w module „<Bausteinname>” w menedżerze bibliotek nie jest zgodna z kolejnością w pliku cslib.hex. Odpowiednio skonfigurować kolejność. (tylko w systemach docelowych 68K, jeśli w pliku docelowym jest aktywny test kolejności.)

- 3250**
„Real nie jest obsługiwany w kontrolerach 8-bitowych”
System docelowy nie jest aktualnie obsługiwany.
- 3251**
„Kontrolery 8-bitowe nie obsługują typów «date of day»”
System docelowy nie jest aktualnie obsługiwany.
- 3252**
„Rozmiar stosu przekracza <Zahl> bajtów”
System docelowy nie jest aktualnie obsługiwany.
- 3253**
„Nie znaleziono pliku szesnastkowego: «<Name>»”
System docelowy nie jest aktualnie obsługiwany.
- 3254**
„Odczyt wywołania zewnętrznej funkcji biblioteki był niemożliwy.”
System docelowy nie jest aktualnie obsługiwany.
- 3255**
„Wskaźniki nie są obsługiwane w kontrolerach 8-bitowych.”
By móc używać program na komputerze 8-bitowym, należy obejść stosowanie wskaźników.
- 3260**
„Funkcja «name>» ma zbyt dużo argumentów: należy zwiększyć stos argumentów w ustawieniach docelowych.”
Jeśli to możliwe, należy zmienić wielkość stosu w oknie dialogowym „System docelowy” ustawień systemu docelowego (domyślnie: 40), patrz rozdział 13.29.4. Jeśli nie można zmienić ustawienia w CoDeSys, należy zwrócić się do producenta sterownika.
- 3400**
„Błąd podczas importowania zmiennych dostępu”
Plik .exp zawiera niepoprawny fragment ze zmiennymi dostępu.
- 3401**
„Błąd podczas importowania konfiguracji zmiennych”
Plik .exp zawiera niepoprawny fragment ze zmiennymi konfiguracji.
- 3402**
„Błąd podczas importowania zmiennych globalnych”
Plik .exp zawiera niepoprawny fragment ze zmiennymi globalnymi.
- 3403**
„Import <Name> był niemożliwy”
Fragment w pliku .exp dot. podanego obiektu jest niepoprawny.
- 3404**
„Błąd podczas importowania konfiguracji zadania”

Fragment w pliku .exp dot. konfiguracji zadania jest niepoprawny.

3405

„Błąd podczas importowania konfiguracji sterownika”

Fragment w pliku .exp dot. konfiguracji sterownika jest niepoprawny.

3406

„Dwa kroki o nazwie «<Name>». Drugi krok nie został zaimportowany”

Fragment modułu AS zawarty w pliku .exp zawiera dwa kroki o takiej samej nazwie. W pliku eksportu należy zmienić nazwę kroków.

3407

„Nie znaleziono kroku wejściowego «<Name>”

W pliku .exp brakuje podanego kroku.

3408

„Nie znaleziono kolejnego kroku «<Name>”

W pliku .exp brakuje podanego kroku.

3409

„Brak kolejnego przejścia dla kroku «<Name>”

W pliku .exp brak przejścia, które wymaga podanego kroku jako kroku wejściowego.

3410

„Brak kolejnego kroku dla przejścia «<Name>”

W pliku .exp brak kroku, który wymaga podanego przejścia.

3411

„Krok «<Name>» niedostępny z poziomu Init-Step”

W pliku .exp brak jest połączenia między podanym krokiem i Init-Step.

3412

„Nie można zaimportować makro «<Name>”

Należy sprawdzić plik eksportu.

3413

„Błąd podczas importowania krzywek tarczowych”

Zaimportowany został plik eksportu (*.exp), zawierający błędną informację o krzywce tarczowej. Należy sprawdzić plik eksportu.

3414

„Błąd podczas importowania listy programów CNC”

Zaimportowany został plik eksportu (*.exp), zawierający błędną informację o programie CNC. Należy sprawdzić plik eksportu.

3415

„Błąd podczas importowania konfiguracji alarmu”

Zaimportowany został plik eksportu (*.exp), zawierający błędną informację o konfiguracji alarmu. Należy sprawdzić plik eksportu.

- 3450**
„PDO<Name>: <Modulname> <Konfigurationsdialogname>-<PDO Name> brak COB-Id!”
Należy kliknąć przycisk Właściwości w oknie dialogowym konfiguracji <Konfigurationsdialogname> modułu <Modulname> i wpisać do PDO <PDO Name> ID COB.
- 3451**
„Błąd ładowania: nie można znaleźć pliku EDS «<Name>», jest on jednak używany w konfiguracji!”
Plik urządzenia potrzebny do konfiguracji CAN może znajdować się w nieodpowiednim folderze. Należy to sprawdzić na podstawie wpisu w folderze, odnoszącego się do plików konfiguracyjnych w „Projekt” „Opcje” „Katalogi”.
- 3452**
„Nie można utworzyć modułu «<Name>»!”
Plik urządzenia dla modułu <Name> nie jest już odpowiedni dla istniejącej konfiguracji. Możliwe, że został zmodyfikowany po utworzeniu konfiguracji lub jest uszkodzony.
- 3453**
„Nie można utworzyć kanału «<Name>»!”
Plik urządzenia dla kanału <Name> nie jest już odpowiedni dla istniejącej konfiguracji. Możliwe, że został zmodyfikowany po utworzeniu konfiguracji lub jest uszkodzony.
- 3454**
„Adres «<Name>» odsyła do zajętego obszaru pamięci”
W oknie dialogowym ustawień konfiguracji sterownika włączono opcję „Sprawdź nakładanie się adresów” i stwierdzono nakładanie się. Należy pamiętać, że podstawę sprawdzania zakresu stanowi rozmiar wynikający z typu danych w modułach, a nie wartość we wpisie «size» w pliku konfiguracji!
- 3455**
„Błąd ładowania: nie można znaleźć pliku GSD «<Name>», jest on jednak używany w konfiguracji!”
Plik urządzenia potrzebny do konfiguracji Profibus może nie być umieszczony w odpowiednim folderze. Patrz wpis dot. plików konfiguracji w „Projekt” „Opcje” „Katalogi”.
- 3456**
„Nie można utworzyć urządzenia Profibus «<Name>»!”
Plik urządzenia dla urządzenia <Name> nie jest już odpowiedni dla istniejącej konfiguracji. Możliwe, że został zmodyfikowany po utworzeniu konfiguracji lub jest uszkodzony.
- 3457**
„Błędny opis modułu: «<Name>»!”
Należy sprawdzić plik urządzenia przypisany do modułu.
- 3458**
„Nie można utworzyć konfiguracji sterownika! Należy sprawdzić pliki konfiguracji.”
Należy sprawdzić, czy w ustawionym katalogu z plikami konfiguracji („Projekt” „Opcje” „Katalogi”) znajdują się wszystkie potrzebne pliki konfiguracji i urządzeń.
- 3459**
„Ustawiona prędkość transmisji nie jest obsługiwana!”

Należy zmienić ustawienie w oknie dialogowym z parametrami CAN. Należy zwrócić uwagę na dane prędkości transmisji umieszczone w pliku GSD.

3460

3S_CanDrv.lib ma nieprawidłową wersję.

Należy upewnić się, że do projektu została podłączona aktualna wersja 3S_CanDrv.lib.

3461

„3S_CanOpenMaster.lib ma błędną wersję.”

Należy upewnić się, że do projektu została podłączona aktualna wersja 3S_CanOpenMaster.lib.

3462

„3S_CanOpenDevice.lib ma nieprawidłową wersję.”

Należy upewnić się, że do projektu została podłączona aktualna wersja 3S_CanOpenDevice.lib.

3463

„3S_CanOpenManager.lib ma nieprawidłową wersję.”

Należy upewnić się, że do projektu została podłączona aktualna wersja 3S_CanOpenManager.lib.

3464

„3S_CanNetVar.lib ma nieprawidłową wersję.”

Należy upewnić się, że do projektu została podłączona aktualna wersja 3S_CanNetVar.lib.

3465

„CanDevice: Subindeksy muszą być numerowane sekwencyjnie.”

Na listach parametrów stosowanych w CanDevice subindeksy należy numerować w sposób ciągły i bez przerw. Należy sprawdzić odpowiednią listę w menedżerze parametrów.

3466

„Zmienne sieciowe CAN: w konfiguracji sterownika nie znaleziono kontrolera CAN.”

Dla sieci CAN skonfigurowano zmienne sieciowe (zasoby, zmienne globalne), w konfiguracji sterownika projektu nie jest jednak dostępny kontroler CAN.

3468

„CanDevice: brak nazwy zadania aktualizacji w konfiguracji zadań.”

Zadanie aktualizacji (do wywołania CANdevice), zdefiniowane w oknie dialogowym „Ustawienia podstawowe” CANdevice w konfiguracji sterownika, musi być skonfigurowane w konfiguracji zadań w obrębie projektu.

3469

„Nie można wywołać modułu CanOpenMaster. Zadanie należy przyporządkować ręcznie.”

Za pomocą parametru zadania aktualizacji, w oknie dialogowym parametrów modułu w konfiguracji sterownika, należy przyporządkować do master zadanie, za pomocą którego ma on zostać wywołany.

3470

„Nieprawidłowa nazwa w parametrze zadania aktualizacji.”

Otworzyć okno dialogowe parametrów modułu konfiguracji sterownika CoDeSys do CAN-Master. Sprawdzić nazwę zadania podaną dla parametru zadania aktualizacji. Podane zadanie musi być dostępne w projekcie. Ewentualnie może wystarczyć ustawienie innego zadania, lecz należy sprawdzić również wartości zadane za pomocą pliku urządzenia.

- 3500**
„Brak VAR_CONFIG dla «<Name>»”
Dla podanej zmiennej w globalnej liście zmiennych zawierającej „Variablen_Konfiguration” należy wpisać deklarację.
- 3501**
„Brak adresu w VAR_CONFIG dla «<Name>»”
Dla podanej zmiennej w liście zmiennych globalnych zawierającej konfigurację zmiennej należy wpisać adres.
- 3502**
„Nieprawidłowy typ danych «<Name>» w VAR_CONFIG”
Podana zmienna w liście zmiennych globalnych zawierającej konfigurację zmiennej jest zadeklarowana z innym typem danych niż w module funkcyjnym.
- 3503**
„Nieprawidłowy typ adresu «<Name>» w VAR_CONFIG”
Podana zmienna w liście zmiennych globalnych zawierającej konfigurację zmiennej jest zadeklarowana z innym typem adresu niż w module funkcyjnym.
- 3504**
„Wartości początkowe zmiennej VAR_CONFIG nie są obsługiwane”
Zmienna Variablen_Konfiguration jest zadeklarowana z adresem i wartością początkową. Wartość początkowa może być jednak zdefiniowana tylko w zmiennych wejściowych bez przydzielonego adresu.
- 3505**
„<Name> nie jest prawidłową ścieżką instancji”
W konfiguracji zmiennych podano nieistniejącą zmienną.
- 3506**
„Oczekiwana ścieżka dostępu”
W liście zmiennych globalnych dla zmiennych dostępu brak jest dla jednej z nich prawidłowej ścieżki dostępu: <Bezeichner>:'<Zugriffspfad>':<Typ> <Zugriffsart> (<Identyfikator>:'<Ścieżka dostępu>':<Typ> <Typ dostępu>)
- 3507**
„Podawanie adresu dla VAR_ACCESS niedozwolone”
W liście zmiennych globalnych dla zmiennych dostępu istnieje przydział adresu dla jednej z nich.
Poprawna definicja: <Bezeichner>:'<Zugriffspfad>':<Typ> <Zugriffsart> (<Identyfikator>:'<Ścieżka dostępu>':<Typ> <Typ dostępu>)
- 3550**
„Nazwa zadania «<Name>» została użyta dwukrotnie”
Zdefiniowano dwa zadania o tej samej nazwie. Nazwę jednego z nich należy zmienić.
- 3551**
„Zadanie «<Name>» musi zawierać przynajmniej jedno wywołanie programu”
Należy wstawić wywołanie programu lub usunąć zadanie.

3552

„Nie zdefiniowano zmiennej zdarzenia «<Name>» w zadaniu «<Name>»”

W konfiguracji podanego zadania zdefiniowano zmienną zdarzenia niezadeklarowaną w projekcie globalnie. Należy zastosować inną zmienną lub zdefiniować wpisaną zmienną globalnie.

3553

„Zmienna zdarzenia «<Name>» w zadaniu «<Name>» musi być typu BOOL”

Jako zmienną zdarzenia należy zastosować zmienną typu BOOL.

3554

„Wpis zadania «<Name>» musi być programem lub globalną instancją bloku funkcji”

W polu wywołania programu wpisano funkcję lub niezdefiniowany moduł.

3555

„Wpis zadania «<Name>» ma niepoprawne parametry”

W polu wywołania programu podano parametry nieodpowiadające deklaracji modułu.

3556

„Ustawiony cel nie obsługuje zadań”

Istniejąca w projekcie konfiguracja zadania nie jest możliwa dla aktualnie ustawionego systemu docelowego. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.

3557

„Maksymalna liczba zadań (liczba) została przekroczona”

Aktualnie ustawiony system docelowy nie dopuszcza liczby istniejącej obecnie w konfiguracji zadania. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.

3558

„Priorytet zadania '<Name>' znajduje się poza prawidłowym zakresem pomiędzy «<Untergrenze>» i «<Obergrenze>»”

Aktualnie ustawiony system docelowy nie dopuszcza priorytetu zalecanego w konfiguracji zadania. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.

3559

„Zadanie «<Name>»: aktualny system docelowy nie obsługuje zadań interwałowych”

Aktualnie ustawiony system docelowy nie dopuszcza interwału zdefiniowanego obecnie w konfiguracji zadania. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.

3560

„Zadanie «<Name>»: aktualny system docelowy nie obsługuje zadań dowolnych”

Aktualnie ustawiony system docelowy nie dopuszcza zadań dowolnych zdefiniowanych obecnie w konfiguracji zadania. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.

3561

„Zadanie «<Name>»: aktualny system docelowy nie obsługuje zadań zdarzeń”

Aktualnie ustawiony system docelowy nie dopuszcza zadań zdarzeń zdefiniowanych obecnie w konfiguracji zadania. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.

- 3562**
„Zadanie «<Name>»: aktualny system docelowy nie obsługuje zadań sterowanych zewnątrznie przez zdarzenia”
Aktualnie ustawiony system docelowy nie dopuszcza zadań sterowanych zewnątrznie przez zdarzenia, jakie zdefiniowano obecnie w konfiguracji zadania. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.
- 3563**
„Interwał zadania <Taskname> znajduje się poza prawidłowym zakresem od <Bereichsgrenze> do <Bereichsgrenze>”
W oknie dialogowym właściwości w konfiguracji zadania należy skorygować dane interwału.
- 3564**
„Zewnętrzne zdarzenie <Ereignisname> zadania <Taskname> nie jest obsługiwane przez aktualny system docelowy”
Aktualnie ustawiony system docelowy nie dopuszcza zdarzeń zewnętrznych obecnie zdefiniowanych dla zadania. Należy odpowiednio zmienić konfigurację zadania.
- 3565**
„Maksymalna liczba zadań zdarzeń <max. Anzahl> została przekroczona”
Aktualnie ustawiony system docelowy nie dopuszcza obecnej liczby zadań zdarzeń. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.
- 3566**
„Maksymalna liczba zadań interwałowych <max. Anzahl> została przekroczona”
Aktualnie ustawiony system docelowy nie dopuszcza obecnej liczby zadań interwałowych. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.
- 3567**
„Maksymalna liczba dowolnych zadań <max. Anzahl> została przekroczona”
Aktualnie ustawiony system docelowy nie dopuszcza obecnej liczby zadań dowolnych. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.
- 3568**
„Maksymalna liczba zewnętrznych zadań zdarzeń <max. Anzahl> została przekroczona”
Aktualnie ustawiony system docelowy nie dopuszcza obecnej liczby zewnętrznych zadań zdarzeń. Należy ustawić odpowiedni system docelowy lub zmienić konfigurację zadania.
- 3569**
„Nie zdefiniowano modułu <Bausteinname> dla zdarzenia systemowego <Ereignisname>”
Podany moduł, wywoływany ze zdarzenia systemowego zdefiniowanego w konfiguracji zadania, nie jest dostępny w projekcie. Należy odpowiednio zmienić konfigurację zadania lub upewnić się, że wywoływany moduł z tą nazwą istnieje w projekcie.
- 3570**
„Zadania «<Name>» i «<Name>» mają taki sam priorytet”
Należy zmienić konfigurację zadania w taki sposób, by dla dwóch zadań nie był wpisany taki sam priorytet.

3571

„Biblioteka Syslibcallback nie jest podłączona! Nie można wygenerować zdarzeń systemowych.”

Do stosowania zadań sterowanych przez zdarzenia (Event Tasks) wymagana jest biblioteka SysLibCallback.lib. Należy podłączyć bibliotekę lub nie stosować w konfiguracji zadania (właściwości zadania) zadania typu „sterowane zdarzeniami”.

3572

„Interwał Watchdog zadania «<Name>» znajduje się poza prawidłowym zakresem «<Zahl> µs» do «<Zahl> µs»”

W konfiguracji zadania w oknie dialogowym „Właściwości zadania” czas Watchdog jest wpisany w mikrosekundach i znajduje się poza obszarem zdefiniowanym w pliku opisu XML.

3573

„Interwał Watchdog zadania «<Name>» znajduje się poza prawidłowym zakresem «<Zahl>%» do «<Zahl>%»”

W konfiguracji zadania w oknie dialogowym „Właściwości zadania” czas Watchdog jest wpisany w procentach i znajduje się poza zakresem zdefiniowanym w pliku opisu XML.

3574

„Zmienna zdarzenia «<Name>» lub jej bezpośredni adres mogą być zastosowane jako zdarzenie tylko jeden raz”

Zdarzenie Singleton jest stosowane w konfiguracji zadania wielokrotnie. Patrz opis w „Wstaw” „Wstaw zadanie” lub „Wstaw” „Podłącz zadanie”.

3575

„Zadanie «<Name>»: czas cyklu musi być wielokrotnością «<Zahl>» µs.”

Należy skorygować czas cyklu odpowiednio do okna dialogowego właściwości zadania. System docelowy definiuje czas podstawowy i wymaga, by jemu lub jego wielokrotności odpowiadał czas cyklu.

3600

„Nie znaleziono zmiennej bezpośredniej.”

Następnie należy zastosować komendę „Kompiluj wszystko”. Jeśli komunikat błędu pojawi się ponownie, należy zwrócić się do producenta sterownika.

3601

„<Name> to zarezerwowana nazwa zmiennej”

W projekcie zadeklarowano zmienną zarezerwowaną już dla generatora kodu. Należy zmienić nazwę tej zmiennej.

3610

„«<Name>» nie jest obsługiwane”

Podana właściwość nie jest obsługiwana w tej wersji.

3611

„Katalog kompilacji «<Name>» jest nieprawidłowy”

W opcje projektu/katalogu został wpisany nieprawidłowy katalog dla plików kompilacji.

- 3612**
„Przekroczono maksymalną liczbę modułów (<Anzahl>)! Kompilacja zostanie przerwana.”
W projekcie stosowane jest zbyt wiele modułów i typów plików. W ustawieniach systemu docelowego/podziału pamięci należy zmienić maks. liczbę modułów.
- 3613**
„Kompilację przerwano”
Kompilacja przerwana przez użytkownika.
- 3614**
„Projekt nie zawiera modułu «<Name>» (funkcja wejściowa) ani konfiguracji zadania”
W projekcie potrzebna jest funkcja wejściowa typu program (np. PLC_PRG) lub konfiguracja zadania.
- 3615**
„<Name> (funkcja wejściowa) musi być typu Program”
Stosowana jest funkcja wejściowa (np. PLC_PRG), która nie jest typu Program.
- 3616**
„Programy w bibliotekach zewnętrznych nie są obsługiwane”
Zapisywana biblioteka zawiera program. Nie będzie on dostępny w trakcie stosowania biblioteki.
- 3617**
„Za mało pamięci”
Należy zwiększyć pamięć wirtualną komputera.
- 3618**
„Aktualny generator kodu nie obsługuje dostępów bitowych”
Generator kodu do aktualnie ustawionego systemu docelowego nie obsługuje dostępów bitowych do zmiennych.
- 3619**
„Plik obiektu «<name>» oraz biblioteka «<name>» mają różne wersje!”
Należy upewnić się, że istnieją pasujące do biblioteki wersje plików *.lib oraz *.obj lub *.hex. Pliki muszą mieć dokładnie taki sam datownik.
- 3620**
„Moduł PLC_PRG nie może znajdować się w bibliotece”
Czy projekt ma zostać zachowany jako biblioteka wersji 2.1.? W tej wersji biblioteka nie może zawierać modułu PLC_PRG. Należy zastosować inną nazwę modułu.
- 3621**
„Nie można zapisać pliku kompilacji «<Name>”
Na przewidzianej dla podanego pliku kompilacji ścieżce (patrz Opcje projektu, Katalogi) prawdopodobnie znajduje się już plik o tej samej nazwie tylko do odczytu. Należy usunąć ten plik lub zmienić prawa dostępu.
- 3622**
„Utworzenie pliku symbolu «<Name>» było niemożliwe”
Na przewidzianej dla pliku symbolu ścieżce (zwykle katalog projektu) prawdopodobnie znajduje się już plik o tej samej nazwie tylko do odczytu. Należy usunąć ten plik lub zmienić prawa dostępu.

3623

„Nie można zapisać pliku projektu bootowalnego «<Name>»”

Na przewidzianej dla pliku symbolu ścieżce (zależna od systemu docelowego) prawdopodobnie znajduje się już plik o tej samej nazwie tylko do odczytu. Należy usunąć ten plik lub zmienić prawa dostępu.

3624

„Ustawienie systemu docelowego <Zielsystemeinstellung1>=<eingestellter Wert> nie daje się powiązać z <Zielsystemeinstellung2>=<eingestellter Wert>”

Należy sprawdzić i skorygować podane ustawienia w oknie dialogowym „Ustawienia systemu docelowego” (rejestr zasobów). Jeśli tam nie są wyświetlane ustawienia lub nie można ich edytować, należy zwrócić się do producenta sterownika.

3700

„Moduł o nazwie «<Name>» znajduje się już w bibliotece «<Name>»”

Stosowana jest nazwa modułu przydzielona już modułowi biblioteki. Należy zmienić nazwę modułu.

3701

„Nazwa modułu w nazwie deklaracji jest niezgodna z nazwą na liście obiektów”

Nazwę modułu należy zmienić za pomocą komendy menu „Projekt” „Zmień nazwę obiektu” lub zmienić nazwę modułu w części z deklaracją. Nazwa musi się znajdować bezpośrednio po słowach kluczowych PROGRAM, FUNCTION lub FUNCTIONBLOCK.

3702

„Zbyt wiele identyfikatorów”

Dla deklaracji zmiennych można podać maksymalnie 100 identyfikatorów.

3703

„Kilka deklaracji z tym samym identyfikatorem «<Name>»”

W części z deklaracją obiektu istnieje większa ilość identyfikatorów o tej samej nazwie.

3704

„Rekursja danych: <Baustein 0> -> <Baustein 1> -> ... -> <Baustein 0>”

Zastosowano instancję FB, dla której jest potrzebna ona sama.

3706

„Jako modyfikator «CONSTANT» są dozwolone tylko «VAR», «VAR_INPUT», «VAR_EXTERNAL» i «VAR_GLOBAL»”

Dla tej kategorii zmiennych nie można zadeklarować stałych.

3705

„<Name>: VAR_IN_OUT w modułach najwyższego poziomu jest niedozwolona, gdy brak jest konfiguracji zadań”

Należy ustawić konfigurację zadań lub upewnić się, że w PLC_PRG nie są stosowane zmienne VAR_IN_OUT.

3720

„Po «AT» musi znajdować się adres”

Po słowie kluczowym AT należy wstawić poprawny adres lub zmienić słowo kluczowe AT.

- 3721**
„W adresach można umieścić tylko VAR i VAR_GLOBAL”
Deklarację należy skopiować w zakres VAR lub VAR_GLOBAL.
- 3722**
„Pod podanym adresem mogą znajdować się tylko proste zmienne boolean”
Należy zmienić adres lub podany w deklaracji typ zmiennej.
- 3726**
„Stałej nie można umieścić w adresie bezpośrednim”
Należy odpowiednio zmienić przydział adresu.
- 3727**
„W tych adresach nie można umieszczać tablic”
Należy odpowiednio zmienić przydział adresu.
- 3728**
„Niepoprawny adres: «<Adresse>»”
Konfiguracja sterownika nie obsługuje tego adresu. Należy sprawdzić konfigurację lub skorygować adres.
- 3729**
„Niepoprawny typ «<Name>» w adresie: «<Name>»”
Typu tej zmiennej nie można umieścić w podanym adresie. Przykład: w odniesieniu do systemu docelowego pracującego z Alignment 2 niepoprawna jest następująca deklaracja: var1 AT %IB1:WORD.
Komunikat błędu pojawia się ponadto, jeśli tablica jest przypisana do adresu zmiennej bezpośredniej
- 3740**
„Nieznany typ: «<Name>»”
Dla deklaracji zmiennych stosowany jest niepoprawny typ.
- 3741**
„Oczekiwany identyfikator typu”
Zamiast poprawnego identyfikatora typu stosowane jest słowo kluczowe lub operator.
- 3742**
„Oczekiwana wartość wyliczenia”
W definicji typu wyliczenia brak jest identyfikatora po nawiasie otwierającym lub po przecinku w nawiasie.
- 3743**
„Oczekiwana liczba całkowita”
Wartości enumeracji można inicjalizować wyłącznie za pomocą liczb całkowitych typu INT.
- 3744**
„Stała Enum «<Name>» jest już zdefiniowana”
Należy sprawdzić, czy przy przydzielaniu wartości wyliczenia przestrzegano następujących reguł:
– W zakresie definicji Enum wszystkie wartości muszą być jednoznaczne.

- W zakresie wszystkich globalnych definicji Enum wszystkie wartości muszą być jednoznaczne.
- W zakresie wszystkich lokalnych definicji Enum modułu wszystkie wartości muszą być jednoznaczne.

3745

„Granice zakresów są dozwolone tylko dla typu danych scalonych!”

Typy podzakresów można zdefiniować wyłącznie na podstawie typu danych scalonych.

3746

„Granica zakresu «<Name>» nie jest kompatybilna z typem danych «<Name>»”

Granica zakresu podanego dla typu podzakresu znajduje się poza granicami dozwolonymi dla typu podstawowego.

3747

„Nieznana długość stringu: «<Name>»”

W definicji długości stringu stosowana jest nieznana stała.

3748

„Więcej niż trzy wymiary w jednej tablicy są niedozwolone”

W tablicy są stosowane ponad trzy dopuszczalne wymiary. W razie potrzeby należy zastosować ARRAY OF ARRAY.

3749

„Dolna granica «<Name>» nieznana”

Jako dolna granica podzakresu lub typu maczyzy stosowana jest niezdefiniowana stała.

3750

„Górna granica «<Name>» nieznana”

Jako górna granica podzakresu lub typu tablicy stosowana jest niezdefiniowana stała.

3751

„Niepoprawna długość stringu <Anzahl Zeichen>”

Dozwolona w ustawionym systemie docelowym maksymalna długość stringu (liczba znaków) została przekroczona w niniejszej deklaracji.

3752

„Ponad 9 wymiarów nie jest dozwolone dla tablic połączonych”

Tablica może być maksymalnie trójwymiarowa. Rezultat połączenia tablic (np. „arr: ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2, 0..2] OF DINT” może stanowić liczba maksymalnie 9 wymiarów, którą przekroczono w tym przypadku. Należy zmniejszyć liczbę do maksymalnie dziewięciu wymiarów.

3760

„Nieprawidłowa wartość początkowa”

Należy zastosować wartość początkową odpowiadającą definicji typu. Można skorzystać z dialogu deklaracji zmiennych (Shift/F2 lub „Edytuj” „Deklaracja zmiennych”).

3761

„Zmienne VAR_IN_OUT nie mogą mieć wartości początkowej.”

Przy deklarowaniu zmiennych należy usunąć inicjalizację.

- 3780**
„Oczekiwano VAR, VAR_INPUT, VAR_OUTPUT lub VAR_IN_OUT”
Pierwszy wiersz po nazwie modułu musi zawierać słowa kluczowe.
- 3781**
„Oczekiwano END_VAR lub identyfikatora”
Na początku wiersza deklaracji należy wpisać poprawny identyfikator lub END_VAR.
- 3782**
„Nieoczekiwane zakończenie”
W części z deklaracją: W razie potrzeby na koniec części z deklaracją należy wstawić słowo kluczowe END_VAR. Komunikat błędu jest zwracany również w związku z błędem 3703, ponieważ dwie takie same deklaracje znajdują się na końcu części z deklaracją.
W edytorze tekstu: wstawić instrukcje kończące ostatnią sekwencję instrukcji (np. END_IF).
- 3783**
„Oczekiwano END_STRUCT lub identyfikatora”
Należy upewnić się, że deklaracja typu została poprawnie zakończona.
- 3784**
„System docelowy nie obsługuje atrybutu «<Name>»”
System docelowy nie obsługuje tego typu zmiennej. (np. RETAIN, PERSISTENT)
- 3800**
„Zmienne globalne wymagają zbyt wiele pamięci. W opcjach projektu należy zwiększyć ilość dostępnej pamięci”
Zwiększyć liczbę segmentów ustawioną w opcjach projektu w zakresie opcji kompilacji.
- 3801**
„Zmienna «<Name>» jest zbyt duża. (<Größe> bajtów)”
Zmienna stosuje typ większy od jednego segmentu danych. Zależnie od systemu docelowego można ustawić rozmiar segmentu w ustawieniach systemu docelowego/podziale pamięci. Jeśli nie można znaleźć tam możliwości wpisania danych, należy zwrócić się do producenta sterownika.
- 3802**
„Pamięć dla zachowywanych zmiennych została wykorzystana. Zmienna «<Name>», %u bajtów.”
Brak pamięci na zachowywane zmienne. Zależnie od systemu docelowego można ją ustawić w ustawieniach systemu docelowego/podziale pamięci. Jeśli nie można znaleźć tam możliwości wpisania danych, należy zwrócić się do producenta sterownika. (W instancjach bloków funkcji, w których stosowana jest zachowywana zmienna, całość instancji jest zarządzana w pamięci Retain!)
- 3803**
„Brak pamięci na zmienne globalne. Zmienna «<Name>», «<Anzahl>» bajtów.”
Brak dostępnej pamięci na zmienne globalne. Zależnie od systemu docelowego można ją ustawić w ustawieniach systemu docelowego/podziale pamięci. Jeśli nie można znaleźć tam możliwości wpisania danych, należy zwrócić się do producenta sterownika.
- 3820**
„VAR_OUTPUT oraz VAR_IN_OUT są niedozwolone w funkcjach.”

Nie mogą one definiować w funkcji parametrów wyjściowych/referencyjnych.

3821

„W przypadku jednej funkcji jest niezbędne co najmniej jedno wejście”

Dla funkcji należy wstawić co najmniej jeden parametr wprowadzania.

3840

„Nieznana zmienna globalna «<Name>»!”

W module jest stosowana zmienna VAR_EXTERNAL, dla której nie została zadeklarowana odpowiednia zmienna globalna.

3841

„Deklaracja «<Name>» jest niezgodna z deklaracją globalną!”

Dane typu w deklaracji zmiennej VAR_EXTERNAL są niezgodne z danymi w deklaracji globalnej.

3850

„Deklaracja niespakowanej struktury «<name>» w spakowanej strukturze «<name>» jest niedozwolona!”

Podana definicja struktury prowadzi do „misalignment” w obrębie pamięci. Należy odpowiednio zmienić definicję struktury.

3900

„Wielokrotne podkreślenia w identyfikatorze”

Z identyfikatora należy usunąć wielokrotne podkreślenia.

3901

„Dopuszczalne są maks. 4 pola adresowe”

Stosowany jest bezpośredni przydział do adresu zawierającego ponad cztery poziomy (np. %QB0.1.1.0.1).

3902

„Słowa kluczowe muszą być pisane wielkimi literami”

Słowo kluczowe należy napisać wielką literą lub włączyć opcję „Formatuj automatycznie”.

3903

„Nieprawidłowa stała czasu”

Stała nie została podana odpowiednio do formatu IEC61131-3.

3904

„Przepelnienie w stałej czasu”

Dla stałej czasu jest stosowana wartość, której nie można przedstawić w formacie wewnętrznym. Maksymalna wartość do przedstawienia to t#49d17h2m47s295ms.

3905

„Nieprawidłowa stała daty”

Stała nie została podana odpowiednio do formatu IEC61131-3.

3906

„Nieprawidłowa stała czasu dziennego”

Stała nie została podana odpowiednio do formatu IEC61131-3.

- 3907**
„Nieprawidłowa stała daty/czasu”
Stała nie została podana odpowiednio do formatu IEC61131-3.
- 3908**
„Nieprawidłowa stała ścieżki”
Stała stringu zawiera niepoprawny znak.
- 4000**
„Oczekiwany identyfikator”
W tym miejscu należy wpisać poprawny identyfikator.
- 4001**
„Zmienna «<Name>» (nazwa) nie została zadeklarowana”
Zmienną należy zadeklarować lokalnie lub globalnie.
- 4010**
„Wykluczające się typy: nie można skonwertować «<Name>» na «<Name>».”
Należy sprawdzić wymagany typ operatora (operator należy znaleźć w pliku pomocy) i zmienić typ zmiennej, która spowodowała błąd, na typ dozwolony lub wybrać inną zmienną.
- 4011**
„Niedozwolony typ w parametrze <Parameter> «<Name>»: nie można skonwertować «<Name>» na «<Name>».”
Typ aktualnego parametru nie może zostać przeniesiony do parametru formalnego. Należy zastosować konwersję typu lub zastosować odpowiedni typ zmiennej.
- 4012**
„Niedozwolony typ dla wejścia «<Name>» z «<Name>»: nie można skonwertować «<Name>» na «<Name>».”
Zmiennej «<Name>» jest przydzielana wartość niedozwolonego typu <Typ2>. Należy zmienić zmienną lub stałą na zmienną lub stałą typu <Typ1> lub zastosować konwersję typu lub stałą z prefiksem typu.
- 4013**
„Niedozwolony typ dla wyjścia «<Name>» z «<Name>»: nie można skonwertować «<Name>» na «<Name>».”
Zmiennej «<Name>» jest przydzielana wartość niedozwolonego typu <Typ2>. Należy zmienić zmienną lub stałą na zmienną lub stałą typu <Typ1> lub zastosować konwersję typu lub stałą z prefiksem typu.
- 4014**
„Stała z prefiksem typu: Nie można skonwertować «<Name>» na «<Name>».”
Typ stałej nie jest kompatybilny z typem prefiksu. Przykład: SINT#255
- 4015**
„Niedozwolony typ danych «<Name>» dla bezpośredniego dostępu bitowego”
Bezpośredni adres bitowy jest dopuszczalny wyłącznie dla danych typu integer i bitstring i nie dozwolony dla zmiennych bezpośrednich. Należy w dostępie bitowym <var1>.<bit> zastosować zmienną var1 typu REAL/LREAL lub stałą lub spróbować dostępu bitowego do zmiennej bezpośredniej.

4016

„Indeks bitu «<<Zahl>>» poza prawidłowym zakresem dla zmiennej typu «<Name>»”
Próba dostępu do bitu niezdefiniowanego dla typu danych zmiennej.

4017

„MOD dla REAL jest niezdefiniowany”
Operator MOD może być stosowany wyłącznie dla typów danych integer i bitstring.

4020

„Argumenty ST, STN, S, R muszą być zmiennymi z dostępem zapisu”
Pierwszy argument należy zastąpić zmienną, którą można nadpisać.

4021

„Brak dostępu zapisu do «<Name>»”
Zmienną należy zastąpić zmienną z dostępem zapisu.

4022

„Oczekiwany argument”
Argument za istniejącą komendą należy uzupełnić.

4023

„Po «+» lub «-» oczekiwana jest liczba”
Należy wpisać liczbę.

4024

„Oczekiwane <Operator 0> lub <Operator 1> lub ... przed «<Name>»”
W podane miejsce należy wpisać odpowiedni operator.

4025

„Oczekiwane «:=» lub «=>» przed «<Name>»”
W podane miejsce należy wpisać jeden z dwóch operatorów.

4026

„BITADR oczekuje adresu bitu lub zmiennej do adresu bitu”
Należy zastosować poprawny adres bitu (np. %IX0.1).

4027

„Oczekiwana liczba całkowita lub stała symboliczna”
Należy wstawić liczbę całkowitą lub identyfikator poprawnej stałej.

4028

„Operator INI wymaga instancji bloku funkcji lub zmiennej strukturalnej”
Należy sprawdzić typ zmiennej, dla którego ma być zastosowany operator INI.

4029

„Nie są możliwe zagnieżdżone wywołania tej samej funkcji.”
W systemach docelowych bez możliwości wielokrotnego dostępu i w trybie symulacji wywołanie funkcji nie może zawierać jako parametr wywołania siebie samej.

Przykład: fun1(a,fun1(b,c,d),e);

Należy zastosować zmienną pośrednią.

4030

„Stałe oraz wyrażenia nie są dozwolonymi argumentami ADR”

Stałą lub wyrażenie należy zastąpić zmienną lub adresem bezpośrednim.

4031

„Operator adresu jest niedozwolony dla bitów! Zamiast niego należy zastosować BITADR”

Należy zastosować BITADR. Należy pamiętać: BITADR nie zapewnia fizycznego adresu pamięci

4032

„«<Anzahl>» argumentów to zbyt mało dla «<Name>». Potrzebne jest co najmniej «<Anzahl>»”

Należy sprawdzić, ilu argumentów potrzebuje operator „<Name>” i wstawić brakujące.

4033

„«<Anzahl>» argumentów to zbyt wiele dla «<Name>». Potrzebne jest dokładnie «<Anzahl>»”

Należy sprawdzić, ilu argumentów potrzebuje operator «<Name>» i usunąć zbędne.

4034

„Dzielenie przez 0”

W wyrażeniu stałym jest stosowane dzielenie przez 0. W razie potrzeby należy zastosować zmienną wartości 0 i w ten sposób wymusić błąd czasu przebiegu.

4035

„ADR nie można zastosować do «VAR CONSTANT», jeśli włączono opcję «Zastępuj stałe»”

Dostęp adresu do stałych, dla których są stosowane wartości bezpośrednie, jest niemożliwy. W razie potrzeby należy wyłączyć w opcjach projektu opcję „Zastąp stałe” kategoria Opcje kompilacji.

4040

„Nie zdefiniowano znacznika skoku <LabelName> (etykieta nazwy)”

Należy zdefiniować znacznik o nazwie <LabelName> (etykieta nazwy) lub zmienić <LabelName> (etykieta nazwy) na znacznik zdefiniowany.

4041

„Wielokrotna definicja znacznika skoku «<Name>»”

Znacznik skoku «<Name>» jest wielokrotnie zdefiniowany w module. Należy odpowiednio zmienić nazwę lub usunąć definicję.

4042

„Kolejno można umieścić maks. «<Anzahl>» znaczników skoku”

Liczba znaczników skoku w instrukcji jest ograniczona do «<Anzahl>». Należy wstawić instrukcję dummy.

4043

„Niepoprawny format etykiety. Etykieta musi być identyfikatorem, po którym można wstawić dwukropek.”

Nazwa stosowana dla etykiety nie jest poprawnym identyfikatorem lub w definicji brakuje dwukropka.

4050

„Moduł «<Name>» nie istnieje w projekcie”

Należy zdefiniować moduł o nazwie „<Name>” za pomocą komend menu „Projekt” „Wstaw obiekt” lub zmienić „<Name>” na nazwę zdefiniowanego modułu

4051

„«<Name>» nie jest funkcją”

Jako «<Name>» należy zastosować zdefiniowane bloku funkcji w projekcie lub w bibliotekach.

4052

„«<Instanzname>» musi być zadeklarowaną instancją bloku funkcji «<Name>»”

Jako <Instanzname> należy zastosować zdefiniowaną w projekcie instancję typu «<Name>» lub zmienić typ <Instanzname> na «<Name>».

4053

„<Name> nie jest prawidłowym modułem ani operatorem”

«<Name>» należy zastąpić nazwą zdefiniowanego w projekcie modułu lub operatora.

4054

„Oczekiwana nazwa modułu jako parametr «INDEXOF»”

Podany parametr nie jest poprawną nazwą modułu.

4060

„Parametr «VAR_IN_OUT» «<Name>» z «<Name>» wymaga zmiennej z dostępem zapisu jako wpisu.”

Do parametru VAR_IN_OUT należy przekazać zmienne z dostępem zapisu, ponieważ można je modyfikować w obrębie modułu.

4061

„Parametr VAR_IN_OUT «<Name>» z «<Name>» musi być przypisany.”

Parametry VAR_IN_OUT muszą być przypisane do zmiennych z dostępem zapisu, ponieważ można je modyfikować w obrębie modułu.

4062

„Brak dostępu z zewnątrz do parametru VAR_IN_OUT «<Name>» z «<Name>».”

Parametry VAR_IN_OUT mogą być nadpisywane lub wczytywane tylko w obrębie modułu, ponieważ chodzi o przekazanie poprzez referencję.

4063

„Parametr VAR_IN_OUT «<Name>» z «<Name>» nie może być przypisany do adresów bitowych.”

Adres bitowy nie jest poprawnym adresem fizycznym. Należy przekazać zmienną lub bezpośredni adres niebitowy.

4064

„VAR_IN_OUT nie można zastąpić podczas lokalnego wywołania akcji!”

Należy usunąć przypisanie zmiennych VAR_IN_OUT do lokalnego wywołania czynności.

4070

„Moduł zawiera zbyt złożone wyrażenie.”

Należy uprościć wyrażenie poprzez podział na kilka mniejszych wyrażzeń i za pomocą przydziałów do zmiennych pośrednich.

- 4071**
„Sieć jest zbyt duża”
Sieć należy podzielić na większą ilość sieci.
- 4072**
„Niespójne wykorzystanie identyfikatora akcji w typie FB «<Name>» oraz w instancji «<Name>»”
Zostały zdefiniowane dwie czynności bloku funkcji fb: np. a1 i a2, stosują przy wywołaniu jednej z czynności w FBD i w informacji o typie (określenie w obrębie okna) inne określenie czynności (np. fb.a1) niż w określeniu instancji (np. inst.a2, poza oknem). Należy odpowiednio skorygować identyfikator wymaganej akcji.
- 4100**
„«^» wymaga typu wskaźnika”
Próba usunięcia referencji zmiennej niezadeklarowanej jako POINTER TO.
- 4110**
„«<index>» jest dozwolony tylko dla zmiennych matrycy”
Stosowane jest [<index>] dla zmiennej niezadeklarowanej jako ARRAY OF.
- 4111**
„Wyrażenie w indeksie tablicy musi stanowić wynik typu INT”
Należy zastosować wyrażenie odpowiedniego typu lub konwersję typu.
- 4112**
„Zbyt wiele wskaźników tablicy”
Należy sprawdzić liczbę indeksów (1, 2, lub 3), zadeklarowanych dla matrycy i usunąć zbędne.
- 4113**
„Zbyt mało wskaźników tablicy”
Należy sprawdzić liczbę indeksów (1, 2, lub 3), zadeklarowanych dla matrycy i uzupełnić potrzebne.
- 4114**
„Stały indeks nie mieści się w zakresie tablicy”
Należy się upewnić, że zastosowane indeksy znajdują się w granicach tablicy.
- 4120**
„Przed znakiem «.» musi znajdować się zmienna strukturalna”
Identyfikator z lewej strony punktu musi być zmienną typu STRUCT, lub FUNCTION_BLOCK lub nazwą funkcji FUNCTION lub PROGRAM.
- 4121**
„«<Name>» nie jest składnikiem <Objektname>”
Składnik «<Name>» nie zawiera się w definicji obiektu <Objektname>.
- 4122**
„«<Name>» nie jest parametrem wprowadzania wywołanego bloku funkcji”

Należy sprawdzić zmienne wejściowe wywołanego bloku funkcji i zmienić «<Name>» na jedną z tych zmiennych.

4200

„Oczekiwany «LD»”

Należy wstawić w okno edytora modułu IL lub po znaczniku skoku co najmniej jedną instrukcję LD.

4201

„Oczekiwany operator IL”

Każda instrukcja IL musi się rozpoczynać od operatora lub znacznika skoku.

4202

„Nieoczekiwane zakończenie wyrażenia w nawiasach”

Należy zamknąć nawias.

4203

„<Name> w nawiasie niedozwolone”

Podany operator jest niedopuszczalny w obrębie wyrażenia IL w nawiasie.

(niedopuszczalne są: «JMP», «RET», «CAL», «LDN», «LD», «TIME»)

4204

„Nawias zamykający bez odnośnego nawiasu otwierającego”

Należy wstawić nawias otwierający lub usunąć zamykający.

4205

„Po «)» przecinek jest niedozwolony”

Usunąć przecinek po nawiasie zamykającym.

4206

„Brak znaczników skoku w obrębie wyrażeń w nawiasach”

Należy przesunąć znacznik skoku, by znajdował się poza wyrażeniem w nawiasie.

4207

„Modyfikator «N» wymaga operatora typu BOOL, BYTE, WORD or DWORD”

Modyfikator N potrzebuje typu danych, dla którego można wykonać zaprzeczenie boolean.

4208

„Wyrażenie polecenia warunkowego musi dawać wynik typu BOOL”

Należy upewnić się, że wyrażenie daje wynik typu BOOL lub zastosować konwersję typu.

4209

„W tym miejscu nazwa funkcji jest niedozwolona”

Należy zmienić wywołanie funkcji na zmienną lub stałą.

4210

„«CAL», «CALC» i «CALN» wymagają jako argumentu instancji bloku funkcji”

Należy zadeklarować instancję bloku funkcji do wywołania.

- 4211**
„Komentarz w IL jest dopuszczalny tylko na końcu wiersza”
Komentarz należy przesunąć na koniec lub do oddzielnego wiersza.
- 4212**
„Akumulator jest nieważny przed instrukcją warunkową”
Zawartość akumulatora nie jest zdefiniowana. Przypadek instrukcji bez wyniku (np. „CAL”).
- 4213**
„«S» i «R» wymagają argumentu typu BOOL”
W tym miejscu należy zastosować zmienną boolean.
- 4250**
„Brak prawidłowego początku dla instrukcji ST”
Wiersz nie rozpoczyna się od poprawnej instrukcji ST.
- 4251**
„Funkcja «<Name>» ma zbyt wiele parametrów”
Podano więcej parametrów, niż zadeklarowano w definicji funkcji.
- 4252**
„Funkcja «<Name>» ma zbyt mało parametrów”
Podano mniej parametrów, niż zadeklarowano w definicji funkcji.
- 4253**
„«IF» I «ELSIF» wymagają wyrażenia boolean jako warunku”
- 4254**
„«WHILE» wymaga wyrażenia boolean jako warunku”
Należy się upewnić, że wyrażenie po «WHILE» jest wyrażeniem boolean.
- 4255**
„«UNTIL» wymaga wyrażenia boolean jako warunku”
Należy się upewnić, że wyrażenie po «UNTIL» jest wyrażeniem boolean.
- 4256**
„«NOT» wymaga argumentu boolean”
*Należy się upewnić, że wyrażenie po «NOT» jest wyrażeniem boolean.
- 4257**
„Licznik polecenia «FOR» musi być typu INT”
Należy się upewnić, że zmienna licznika ma typ danych integer lub bitstring (np. DINT, DWORD).
- 4258**
„Licznik instrukcji «FOR» nie jest zmienną z dostępem zapisu”
Zmienną licznika należy zastąpić zmienną z dostępem zapisu.

4259

„Wartość początkowa polecenia «FOR» musi być typu INT”

Wartość początkowa polecenia «FOR» musi być kompatybilna z typem zmiennej licznika.

4260

„Wartość końcowa polecenia «FOR» musi być typu INT”

Wartość końcowa polecenia «FOR» musi być kompatybilna z typem zmiennej licznika.

4261

„Wartość przyrostowa polecenia «FOR» musi być typu INT”

Wartość przyrostowa polecenia «FOR» musi być kompatybilna z typem zmiennej licznika.

4262

„«EXIT» jest dozwolony tylko w obrębie pętli”

«EXIT» można stosować wyłącznie w obrębie polecenia «FOR», «WHILE» lub «UNTIL».

4263

„Oczekiwana liczba «ELSE» lub «END_CASE»”

W obrębie «CASE» można podać wyłącznie liczbę lub polecenie «ELSE» lub polecenie końcowe «END_CASE».

4264

„Selektor polecenia «CASE» musi być typu INT”

Należy się upewnić, że selektor ma typ danych integer lub bitstring (np. DINT, DWORD).

4265

„Po «,» oczekiwana jest liczba”

W wyliczeniu selektorów CASE po przecinku musi być podany kolejny selektor.

4266

„Niezbędna jest co najmniej jedna instrukcja”

Należy podać instrukcję, przynajmniej średnik.

4267

„Wywołanie modułu funkcji musi rozpoczynać się od nazwy instancji”

Oznaczenie w wywołaniu modułu funkcji nie jest instancją. Należy zadeklarować instancję wymaganego modułu funkcji lub zastosować nazwę już zadeklarowanej instancji.

4268

„Oczekiwane wyrażenie”

W tym miejscu należy podać wyrażenie.

4269

„Po gałęzi «ELSE» oczekiwany jest «END_CASE»”

Po gałęzi «ELSE» należy zakończyć polecenie «CASE» za pomocą «END_CASE».

4270

„Stała «CASE» «%ld» jest już używana”

Selektor «CASE» można zastosować w obrębie polecenia «CASE» tylko jeden raz.

- 4271**
„Dolna granica podanego zakresu jest większa niż granica górna.”
Należy skorygować granice zakresów selektorów w taki sposób, by dolna granica nie była większa od górnej.
- 4272**
„Oczekiwany parametr «<Name>» w miejsce <Position> przy wywołaniu «<Name>!»”
Jeśli parametry funkcji są stosowane w wywołaniu funkcji z podaniem nazw parametrów, dodatkowo pozycja parametrów (kolejność) musi być zgodna z kolejnością podaną w definicji funkcji.
- 4273**
„Zakres CASE «<Bereichsgrenzen>» pokrywa się z aktualnie używanym zakresem «<Bereichsgrenzen>»”
Należy się upewnić, że podane w poleceniu CASE zakresy selektorów nie pokrywają się.
- 4274**
„Wielokrotna gałąź «ELSE» w poleceniu CASE”
Instrukcja CASE nie może zawierać więcej niż jednej gałęzi «ELSE».
- 4300**
„Skok lub powrót wymagają wpisu wyrażenia boolean”
Należy się upewnić, że wejście do skoku lub polecenia powrotu jest wyrażeniem boolean.
- 4301**
„Moduł «<Name>» wymaga dokładnie «<Anzahl>» wejść”
Liczba wejść nie odpowiada liczbie w definicji modułu zmiennych VAR_INPUT i VAR_IN_OUT.
- 4302**
„Moduł «<Name>» wymaga dokładnie «<Anzahl>» wyjść”
Liczba wyjść nie odpowiada liczbie w definicji modułu zmiennych VAR_OUTPUT.
- 4303**
„<Name> nie jest operatorem”
„<Name>” należy zastąpić prawidłowym operatorem.
- 4320**
„Dla styku zastosowano wyrażenie inne niż wyrażenie boolean «<Name>»”
Sygnał włączenia styku musi być wyrażeniem boolean.
- 4321**
„Dla cewki zastosowano wyrażenie inne niż wyrażenie boolean «<Name>»”
Zmienna wyjściowa cewki musi być typu BOOL.
- 4330**
„Oczekiwane wyrażenie na wejściu «EN» modułu «<Name>»”
Połączyć wejście EN modułu «<Name>» z wejściem lub wyrażeniem.
- 4331**
„Oczekiwane wyrażenie na wejściu «<Anzahl>» modułu «<Name>»”

Wejście modułu operatora nie jest podłączone.

4332

Oczekiwane wyrażenie na wejściu «<Name>» modułu «<Name>»

Wejście modułu ma typ VAR_IN_OUT i nie jest podłączone.

4333

„Oczekiwany identyfikator w skoku”

Podany punkt docelowy skoku nie jest poprawnym identyfikatorem.

4334

„Oczekiwane wyrażenie na wejściu skoku”

Połączyć wejście skoku z wyrażeniem boolean. Jeśli jest ono TRUE, zostaje wykonany skok.

4335

„Oczekiwane wyrażenie na wejściu powrotu”

Połączyć wejście instrukcji powrotu z wyrażeniem boolean. Jeśli jest ono TRUE, zostaje wykonany skok.

4336

„Oczekiwane wyrażenie na wyjściu wyjścia”

Połączyć wyjście z wyrażeniem, które może być przydzielone do tego wyjścia.

4337

„Oczekiwany identyfikator wejścia”

Wstawić do elementu wejściowego poprawne wyrażenie lub identyfikator.

4338

„Moduł «<Name>» nie ma prawdziwych wejść”

Żadne z wejść modułu operatora «<Name>» nie jest połączone z poprawnym wyrażeniem.

4339

„Wykluczające się typy na wyjściu: nie można skonwertować «<Name>» na «<Name>».”

Wyrażenie w elemencie wyjściowym nie jest kompatybilne pod względem typu z wyrażeniem, które ma mu zostać przydzielone.

4340

„Skok wymaga wpisu wyrażenia boolean”

Należy się upewnić, że wejście dla skoku jest wyrażeniem boolean.

4341

„Powrót wymaga wpisu wyrażenia boolean”

Należy się upewnić, że wejście dla instrukcji powrotu jest wyrażeniem boolean.

4342

„Wejście «EN» okna dialogowego wymaga wpisu operatora boolean”

Wejście EN modułu należy połączyć z poprawnym wyrażeniem boolean.

- 4343**
„Przyporządkowanie stałych: Niedozwolony typ dla parametrów «<Name>» z «<Name>»: nie można skonwertować «<Typ>» na «<Typ>».”
Zadeklarowane zostało wejście «<Name>» modułu «<Name>» jako VAR_INPUT CONSTANT. Jednak przydzielono do niego w oknie dialogowym «Edycja parametrów» wyrażenie niekompatybilne pod względem typu.
- 4344**
„«S» i «R» wymagają argumentów boolean”
Za instrukcjami set lub reset należy wstawić poprawne wyrażenie boolean.
- 4345**
„Niedozwolony typ dla parametrów «<Name>» z «<Name>»: nie można skonwertować «<Typ>» na «<Typ>».”
Do wejścia «<Name>» modułu «<Name>» przydzielono wyrażenie niekompatybilne pod względem typu.
- 4346**
„Wyjście nie może być stałą”
Celem przyporządkowania musi być zmienna lub bezpośredni adres z dostępem do zapisu.
- 4347**
„Parametr VAR_IN_OUT wymaga zmiennej z dostępem zapisu”
Do parametru VAR_IN_OUT należy przekazać zmienne z dostępem zapisu, ponieważ można je modyfikować w obrębie modułu.
- 4348**
"Niedozwolona nazwa programu, <name>. Istnieje już zmienna o tej nazwie."
W edytorze CFC wstawiono element programu z taką samą nazwą, jak już istniejąca (globalna) zmienna. Należy odpowiednio zmienić nazwę.
- 4349**
„Wejście lub wyjście w module <name> zostały usunięte: Należy sprawdzić połączenia z elementem. Ten komunikat błędu znika dopiero po edycji CFC.”
Należy odpowiednio poprawić moduł CFC.
- 4350**
„Akcja AS nie może zostać wywołana z zewnątrz!”
Czynności AS można wywołać wyłącznie w zakresie modułu AS, w którym są one zdefiniowane. Komunikat błędu pojawia się jednak również, jeśli czynność jest wywoływana w dopuszczalny sposób z zakresu modułu AS, lecz nie są stosowane kroki IEC, a biblioteka iecsf.lib jest jeszcze połączona z projektem. W takim przypadku należy w menedżerze biblioteki usunąć tę bibliotekę z projektu i ponownie wykonać kompilację.
- 4351**
„Nazwa kroku nie jest dozwolonym identyfikatorem: «<Name>»”
Należy zmienić nazwę kroku i wybrać dla nazwy poprawny identyfikator.
- 4352**
„Niedozwolone znaki następują po dozwolonej nazwie kroku: «<Name>»”
Usunąć niedozwolone znaki z nazwy kroku.

4353

„Nazwy kroków są podwójne: «<Name>»”

Zmienić nazwę jednego z kroków.

4354

„Skok do niezdefiniowanego kroku: «<Name>»”

Należy wybrać jako punkt docelowy skoku istniejącą nazwę kroku lub wstawić krok o jeszcze niezdefiniowanej nazwie.

4355

„Przejście nie może mieć efektów pobocznych (przyporządkowań, wywołań bloków funkcji itp.)”

Przejście może zawierać tylko jedno wyrażenie boolean.

4356

„Skok bez prawidłowych nazw kroku: «<Name>»”

Jako punkt docelowy skoku należy zastosować poprawny identyfikator.

4357

„Nie znaleziono biblioteki IEC”

Należy sprawdzić, czy w menedżerze biblioteki podłączono bibliotekę iecsfc.lib i czy ścieżki wpisane w opcjach projektu są poprawne.

4358

„Czynność niezadeklarowana: «<Name>»”

Czynność kroku IEC ma być wstawiona w Object Organizer poniżej modułu AS i nazwa czynności ma być wpisana w polu po prawej stronie kwalifikatora.

4359

„Niepoprawny kwalifikator: «<Name>»”

Wpisać kwalifikator dla czynności IEC w pole po lewej stronie nazwy czynności.

4360

„Oczekiwana stała czasu wg kwalifikatora: «<Name>»”

Wpisać stałą czasu dla czynności IEC po kwalifikatorze w pole po lewej stronie nazwy czynności.

4361

„Identyfikator «<Name>» nie określa żadnej czynności”

Wpisać nazwę czynności zdefiniowanej w projekcie lub zmienną boolean dla czynności IEC w pole po prawej stronie kwalifikatora.

4362

„Wyrażenie w czynności jest inne niż wyrażenie boolean: «<Name>»”

Należy wpisać zmienną boolean lub poprawną nazwę akcji.

4363

„Nazwa kroku IEC została już użyta dla zmiennej: «<Name>»”

Należy zmienić nazwę kroku lub zmiennej.

- 4364**
„Przejście musi być wyrażeniem boolean”
Wynik wyrażenia przejścia musi być typu BOOL.
- 4365**
„Krok «<Name>» ma błędną wartość graniczną czasu”
Należy otworzyć okno dialogowe atrybutów kroku dla kroku «<Name>» i wpisać poprawne zmienne lub stałe czasu.
- 4366**
„Znacznik kroku równoległego nie jest dozwolonym identyfikatorem: «<Name>»”
Należy wpisać obok trójkąta wskazującego znacznik skoku dopuszczalny identyfikator.
- 4367**
„Znacznik «<Name>» już istnieje”
Znacznik skoku lub krok o tej nazwie został już określony. Należy odpowiednio zmienić nazwę.
- 4368**
„Czynność «<Name>» jest używana na wielu powiązanych ze sobą poziomach SFC!”
Czynność «<Name>» jest stosowana zarówno w module, jak i w jednej lub większej liczbie czynności tego modułu.
- 4369**
„Do przejść jest niezbędna dokładnie jedna sieć”
Do przejścia zastosowano więcej sieci FBD lub LD. Ich liczbę należy zredukować dokładnie do jednej.
- 4370**
„Znaleziono zbędne wiersze po prawidłowym przejściu IL”
Należy usunąć niepotrzebne wiersze na końcu przejścia.
- 4371**
„Zbędne znaki po prawidłowym wyrażeniu: «<Name>»”
Należy usunąć niepotrzebne znaki na końcu przejścia.
- 4372**
„Krok «<Schrittname>»: Wartość graniczna czasu musi być typu «TIME»”
W atrybutach kroku należy zdefiniować granice czasowe kroku za pomocą zmiennej typu TIME lub za pomocą danych czasu w poprawnym formacie (np. „t#200ms”).
- 4373**
„Akcje IEC są dozwolone tylko w przypadku modułów SFC”
Poniżej modułu Nie-AS utworzono czynność zaprogramowaną w AS i zawierającą czynności IEC. Czynność tę należy zastąpić inną czynnością niezawierającą czynności IEC.
- 4374**
„Oczekiwany krok zamiast przejścia <Transitionsname>”
Moduł AS jest uszkodzony, możliwe, że wskutek akcji eksport/import.

4375

„Oczekiwano przejścia zamiast kroku <schrittname>”

Moduł AS jest uszkodzony, możliwe, że wskutek akcji eksport/import.

4376

„Oczekiwany krok po przejściu «<Name>»”

Moduł AS jest uszkodzony, możliwe, że wskutek akcji eksport/import.

4377

„Oczekiwane przejście po kroku «<Name>»”

Moduł AS jest uszkodzony, możliwe, że wskutek akcji eksport/import.

4400

„Moduł «<Name>» jest niekompletny / został zaimportowany lub skonwertowany z błędami.”

Modułu nie można w całości skonwertować wg IEC 61131-3.

4401

„Stała czasu S5 jest zbyt duża o «<Anzahl>» sekund (maks. 9990 s).”

W akumulatorze brak jest poprawnego zakodowanego czasu BCD.

4402

„Bezpośredni dostęp dozwolony tylko do we./wy.”

Należy się upewnić, że istnieje dostęp tylko do jednej zmiennej zdefiniowanej jako wejście lub wyjście.

4403

„Polecenie STEP5/7 niedozwolone lub niemożliwe do skonwertowania wg IEC 61131-3.”

Nie każde polecenie STEP5/7 można skonwertować wg IEC 61131-3, np. polecenia CPU, takie jak MAS.

4404

„Operator STEP5/7 niedozwolony lub niemożliwy do skonwertowania wg IEC 61131-3.”

Nie każdy argument STEP5/7 można skonwertować wg IEC 61131-3, wzgl. brak jest argumentu.

4405

„Zerowanie timera STEP5/7 nie może być konwertowane wg IEC 61131-3.”

Odpowiednie timery IEC nie mają wejścia do zerowania.

4406

„Stała licznika STEP5/7 zbyt duża (maks. 999).”

W akumulatorze brak jest poprawnej stałej licznika zakodowanej w BCD.

4407

„Brak możliwości konwersji instrukcji STEP5/7 wg IEC 61131-3”

Nie każdą instrukcję STEP5/7 można skonwertować wg IEC 61131-3, np. DUF.

4408

„Dostęp bitowy do timera/licznika słów nie może być konwertowany wg IEC 61131-3.”

Specjalne polecenia timera/licznika nie mogą być konwertowane wg IEC 61131-3.

- 4409**
„Zawartość Akku1 lub Akku2 niezdefiniowana, brak możliwości konwersji wg IEC 61131-3.”
Polecenie łączące oba akumulatory nie może zostać skonwertowane, ponieważ nie jest znana zawartość akumulatorów.
- 4410**
„Wywołany moduł nie znajduje się w projekcie.”
Należy najpierw zaimportować wywołany moduł.
- 4411**
„Błąd w globalnej liście zmiennych.”
Należy sprawdzić plik SEQ.
- 4412**
„Błąd wewn. nr 11”
Należy zwrócić się do producenta sterownika.
- 4413**
„Niepoprawny format linii w module danych”
Kod do zaimportowania zawiera błędną datę.
- 4414**
„Brak nazwy FB/FX”
W wyjściowym pliku S5D brak jest nazwy symbolicznej (poszerzonego) modułu funkcji.
- 4415**
„Polecenie po zakończeniu modułu niedozwolone”
Nie można zaimportować chronionego modułu.
- 4416**
„Nieprawidłowe polecenie”
Nie można rozłożyć polecenia S5/S7.
- 4417**
„Komentarz nie został zamknięty”
Komentarz należy zamknąć za pomocą „*)”.
- 4418**
„Nazwa FB/FX zbyt długa (maks. 8 znaków)”
Nazwa symboliczna (poszerzonego) modułu funkcji jest zbyt długa.
- 4419**
„Oczekiwany format wiersza «"(* nazwa: <FB/FX-Name> *)"»”
Wiersz należy odpowiednio skorygować.
- 4420**
„Brak nazwy parametru FB/FX”
Należy sprawdzić moduły funkcji.

4421

„Dane parametru FB/FX nieprawidłowe”

Należy sprawdzić moduły funkcji.

4422

„Nie podano rodzaju parametru FB/FX”

Należy sprawdzić moduły funkcji.

4423

„Nieprawidłowy argument aktualny”

Należy sprawdzić interfejs modułu funkcji.

4424

„Uwaga: Wywołany moduł nie istnieje, nagłówek jest błędny lub brak jest parametrów”

Wywołany moduł funkcji nie został zaimportowany, jest błędny lub brak jest w nim parametrów (w ostatnim przypadku można zignorować komunikat).

4425

„Nie zdefiniowano znacznika skoku”

Nie został podany punkt docelowy skoku.

4426

„Moduł nie ma prawidłowej nazwy STEP5, np. PB10”

Należy zmienić nazwę modułu.

4427

„Nie podano typu timera”

Deklarację timera należy wstawić w listę zmiennych globalnych.

4428

„Maksymalna głębokość nawiasu STEP5/7 przekroczona”

Nie można stosować więcej niż siedem nawiasów otwierających.

4429

„Błąd w nazwie parametru formalnego”

Nazwa parametru nie może być dłuższa niż cztery znaki.

4430

„Typ parametru formalnego niemożliwy do skonwertowania wg IEC”

Timer, licznik i moduły nie mogą być konwertowane jako parametry formalne w IEC 61131-3.

4431

„Zbyt wiele parametrów VAR_OUTPUT dla jednego wywołania w STEP5/7-IL”

Moduł nie może zawierać więcej niż szesnaście parametrów formalnych jako wyjść.

4432

„Znaczniki skoku w środku wyrażenia są niedozwolone”

W IEC 61131-3 znaczniki skoku nie mogą znajdować się na dowolnym miejscu.

- 4434**
„Zbyt wiele etykiet”
Moduł nie może zawierać więcej niż 100 etykiet.
- 4435**
„Po skoku/wywołaniu nie można wykonać dalszego powiązania”
Po skoku lub wywołaniu musi znajdować się polecenie załadowania.
- 4436**
„Zawartość VKE niezdefiniowana, brak możliwości konwersji wg IEC 61131-3.”
Polecenie stosujące VKE nie może być skonwertowane, ponieważ nie jest znana wartość VKE.
- 4437**
„Typ polecenia jest niezgodny z argumentem”
Komenda bitowa została zastosowana do argumentu Word lub na odwrót.
- 4438**
„Nie otwarto modułu danych (dodaj moduł danych A)”
Dodaj moduł danych A.
- 4500**
„Nieznana zmienna lub adres”
Ta zmienna Watch nie jest zadeklarowana w projekcie. Naciśnięcie <F2> powoduje wyświetlenie pomocy do wpisywania zadeklarowanych zmiennych.
- 4501**
„Po prawidłowym wyrażeniu kontrolnym następują niedozwolone znaki”
Należy usunąć zbędne znaki.
- 4520**
„Błąd w komendzie kompilatora: znacznik oczekiwany przed «<Name>»!”
Nieprawidłowo wprowadzono pragme. Należy sprawdzić, czy «<Name>» jest prawidłowym znacznikiem.
- 4521**
„Błąd w komendzie kompilatora: Niespodziewany element «<Name>»!”
Należy sprawdzić pragme pod kątem prawidłowej składni.
- 4522**
„Oczekiwane polecenie «flag off»!”
Brak wyłączenia pragmy, należy wstawić instrukcję «flag off».
- 4523**
„Pragma {<Pragmaname>} w «<name>» nie jest dopuszczalna”
Pragma nie może być stosowana w tym miejscu. Należy sprawdzić w pomocy online i podręczniku hasło «Pragma» i znaleźć tam informacje dotyczące prawidłowego stosowania.
- 4550**
„Indeks poza dozwolonym zakresem: zmienne OD <Nummer>, wiersz <Zeilennummer>.”

Należy się upewnić, czy indeks znajduje się w zakresie ustalonym w ustawieniach systemu docelowego/funkcjach sieci.

4551

„Subindeks poza dozwolonym zakresem: zmienne OD <Nummer>, wiersz <Zeilennummer>.”

Należy się upewnić, czy subindeks znajduje się w zakresie ustalonym w ustawieniach systemu docelowego/funkcjach sieci.

4552

„Indeks poza dozwolonym zakresem: parametr OD <Nummer>, wiersz <Zeilennummer>.”

Należy się upewnić, czy indeks znajduje się w zakresie ustalonym w ustawieniach systemu docelowego/funkcjach sieci.

4553

„Subindeks poza dozwolonym zakresem: parametr OD <Nummer>, wiersz <Zeilennummer>.”

Należy się upewnić, czy subindeks znajduje się w zakresie ustalonym w ustawieniach systemu docelowego/funkcjach sieci.

4554

„Niepoprawna nazwa zmiennej: zmienne OD <Nummer>, wiersz <Zeilennummer>.”

W pole zmiennej należy wpisać poprawną zmienną projektu. Należy zastosować typ pisowni <Bausteinname>.<Variablenname> lub dla zmiennych globalnych .<Variablenname>

4555

„Puste pole tabeli, wpis nie jest opcjonalny: parametr OD <Zahl>, wiersz <Zahl>.”

W tym polu musi znaleźć się wpis.

4556

„Puste pole tabeli, wpis nie jest opcjonalny: Zmienne OD <Zahl>, wiersz <Zahl>”

W tym polu musi znaleźć się wpis.

4557

„Wymagana pamięć parametrów jest zbyt duża.”

Przekroczono zdefiniowany w systemie docelowym maksymalny rozmiar danych, które mogą być załadowane do systemu docelowego za pomocą list parametrów typu Parametr. Rozmiar danych jest podawany przy kompilacji projektu w oknie komunikatów. Należy zmniejszyć objętość listy parametrów.

4558

„Wymagana pamięć zmiennych jest zbyt duża.”

Przekroczono zdefiniowany w systemie docelowym maksymalny rozmiar danych, które mogą być załadowane do systemu docelowego za pomocą list parametrów typu Parametr. Rozmiar danych jest podawany przy kompilacji projektu w oknie komunikatów. Należy zmniejszyć objętość listy parametrów.

4560

„Niepoprawna wartość: Katalog «<Name>», wiersz «<Zeilennummer>”

Należy sprawdzić wpis. To, jakie wpisy są dopuszczalne dla tego pola, zależy od definicji kolumny (atrybut) we właściwym dla systemu docelowego pliku opisowym XML menedżera parametrów lub od ustawień standardowych stosowanych w przypadku braku pliku opisowego.

- 4561**
„Kolumna niezdefiniowana: «<Name>»”
Wpisy w kolumnie listy parametrów odnoszą się do podanej w tym miejscu kolumny, która nie jest zdefiniowana. Definicje kolumn są zawarte w pliku opisowym (XML) menedżera parametrów aktualnego systemu docelowego. Jeśli brak jest tego pliku, obowiązują ustawienia standardowe.
- 4562**
„Indeks/subindeks «<Name>» jest już używany: Folder «<Name>», wiersz «<Zeilennummer>»”
Para indeks/subindeks musi być jednoznaczna w obrębie wszystkich list parametrów, ponieważ może być stosowana do celów dostępu. Należy odpowiednio zmienić indeksowanie.
- 4563**
„Nazwa «<Name>» jest już używana: Katalog «<Name>», wiersz «<Zeilennummer>»”
Nazwa wpisu musi być jednoznaczna w obrębie wszystkich list parametrów, ponieważ może być stosowana do celów dostępu. Należy zastosować inną nazwę.
- 4564**
„Indeks «<Name>» poza dozwolonym zakresem: Katalog «<Name>», wiersz «<Zeilennummer>»”
W tym polu należy podać indeks znajdujący się w tym zakresie, który zdefiniowano w ustawieniach systemu docelowego, kategoria funkcje sieci w polu „Zakres indeksu” dla danego typu listy (zmienne, parametry, mapowanie).
- 4565**
„Subindeks «<Name>» poza dozwolonym zakresem: Katalog «<Name>», wiersz «<Zeilennummer>»”
W tym polu należy podać subindeks znajdujący się w tym zakresie, który zdefiniowano w ustawieniach systemu docelowego, kategoria funkcje sieci w polu „Zakres subindeksu”.
- 4566**
„Błąd podczas importowania menedżera parametrów”
Do projektu został zaimportowany plik eksportu zawierający błędne informacje na temat menedżera parametrów. W tym celu należy sprawdzić plik *.exp.
- 4600**
„Zmienne sieciowe: «<Name>» nie jest wyrażeniem boolean!”
Należy się upewnić, że zmienna podana w oknie dialogowym właściwości listy zmiennych sieciowych w opcji „Transmisja sterowana zdarzeniem” jest typu BOOL.
- 4620**
W projekcie znaleziono nieużywane zmienne. Patrz opis polecenia 6.3 „Projekt” „Sprawdź” „Nieużywane zmienne”.
- 4621**
Przy przyporządkowaniu zmiennych za pomocą deklaracji „AT” do określonych zakresów pamięci stwierdzono nakładanie się. Patrz opis polecenia 6.3 „Projekt” „Sprawdź” „Nakładające się obszary pamięci”.
- 4622**
W więcej niż jednym zadaniu znajdują się referencje do adresów IEC, wskazujące na ten sam zakres pamięci. Patrz opis polecenia 6.3 „Projekt” „Sprawdź” „Konflikt dostępu”.

4623

W projekcie w więcej niż w jednym miejscu nastąpił dostęp z zapisem do tego samego zakresu pamięci. Patrz opis polecenia 6.3 „Projekt” „Wielokrotny zapis na wyjściu”.

4650

„AxisGroup «<Name>»: Zadanie «<Name>» nie istnieje.”

W konfiguracji sterownika w definicji grupy osi (okno dialogowe „Parametry modułu”, kolumna Value) dla zadania sterującego transferem danych grupy osi jest podana nazwa, która nie jest znana w konfiguracji zadania. Należy odpowiednio skorygować konfigurację zadania lub sterownika.

4651

„AxisGroup «<Name>»: Czas cyklu (dwCycle) nie jest ustawiony.”

W oknie dialogowym „Parametry modułu” grupy osi w konfiguracji sterownika należy wpisać wartość dla czasu cyklu (dwCycle).

4670

„Program CNC «<Name>»: Nie znaleziono zmiennej globalnej «<Name>».”

W programie CNC jest stosowana zmienna globalna (np. \$glob_var\$), która nie jest zdefiniowana w projekcie. Należy ją odpowiednio zadeklarować lub skorygować przydział zmiennej w programie CNC.

4671

„Program CNC «<Name>»: Niewłaściwy typ zmiennej «<Name>».”

W programie CNC w komendzie jazdy jest stosowana zmienna, którą zadeklarowano w tym miejscu z niewłaściwym typem. Należy skorygować zastosowanie lub deklarację typu.

4685

„Krzywka tarczowa «<Name>»: Nieznany typ danych w tabeli punktów wznowienia.”

Należy sprawdzić w dialogu „Opcje kompilacji” w edytorze CAM typ danych wpisanych dla równoodległych lub zoptymalizowanych tabel punktów wznowienia.

4686

„Krzywka tarczowa «<Name>»: Punkt wznowienia przekracza zakres typów danych.”

W krzywce tarczowej są stosowane punkty wznowienia, nie znajdujące się w zdefiniowanym dla tabeli punktów wznowienia zakresie danych. Patrz definicja w oknie dialogowym „Opcje kompilacji” w edytorze CAM.

4700

„«<Nummer>» («<Name>»): Wyrażenie kontrolne «<name>» nie jest numerem.”

W konfiguracji wizualizacji jest stosowana zmienna, której nie definiuje żadna liczba, chociaż byłoby to wymagane w tym miejscu (np. w konfiguracji XOffset lub kąta itd.).

4701

„«<name>» («<zahl>»): Wyrażenie kontrolne «<name>» nie jest typu BOOL.”

W konfiguracji wizualizacji jest stosowana zmienna, która nie jest typu BOOL, chociaż byłoby to wymagane w tym miejscu.

4702

„«<name>» («<zahl>»): Wyrażenie kontrolne «<Name>» nie jest typu STRING.”

W konfiguracji wizualizacji jest stosowana zmienna, która nie jest typu STRING, chociaż byłoby to wymagane w tym miejscu.

- 4703**
„<<Name>>(<<Nummer>>): Nieprawidłowe wyrażenie kontrolne <<Name>>”
Wizualizacja zawiera niepoprawną zmienną.
- 4704**
„<<name>> (<<nummer>>): Nieprawidłowa wartość początkowa w obrębie listy kontrolnej <<Name>>.”
Wizualizowana lista kontrolna (polecenie INTERN w kategorii wpisów) zawiera nieprawidłową wartość początkową. Należy sprawdzić stosowaną listę.
- 4705**
„<<name>> (<<nummer>>): Do tabeli alarmów nie jest przyporządkowana poprawna grupa alarmów.”
W oknie dialogowym konfiguracji tabeli alarmów w kategorii „Tabela alarmów” należy zdefiniować poprawną grupę alarmów.
- 4706**
„<<name>> (<<nummer>>): W celu wykorzystania tabeli alarmów musi być włączone ustawienie systemu docelowego «Obsługa alarmów w obrębie sterownika».”
Należy utworzyć ustawienia systemu docelowego w zakładce rejestru Zasoby i aktywować w oknie dialogowym Wizualizacja opcję „Obsługa alarmów w obrębie sterownika”. Jest to konieczne do działania tabeli alarmów stosowanej jako wizualizacja elementu docelowego, która również jest aktywowana w ustawieniach systemu docelowego.
- 4707**
„<<name>> (<<nummer>>): Tabele alarmów nie są obsługiwane przez sterownik. Należy je usunąć z wizualizacji docelowej.”
System docelowy nie pozwala na przetwarzanie alarmów (nie można włączyć ustawienia systemu docelowego „Obsługa alarmów w obrębie sterownika”). Dlatego do stosowania jako wizualizacja elementu docelowego, którą aktywowano w ustawieniach systemu docelowego (zakładka rejestru Wizualizacja), należy usunąć z wizualizacji elementy tabeli alarmów.
- 4708**
„<<name>> (<<nummer>>): W celu wykorzystania trendów należy włączyć ustawienie systemu docelowego «Zapis danych trendów w obrębie sterownika».”
Należy utworzyć ustawienia systemu docelowego w zakładce rejestru Zasoby i aktywować w oknie dialogowym Wizualizacja opcję „Zapis danych trendów w obrębie sterownika”. Jest to konieczne, by działał element trendów przy stosowaniu jako wizualizacja elementu docelowego, która jest również włączona w ustawieniach systemu docelowego.
- 4709**
„<<name>> (<<nummer>>): Trendy nie są obsługiwane przez sterownik. Należy je usunąć z wizualizacji docelowej.”
System docelowy nie pozwala na przetwarzanie trendów (nie można włączyć ustawienia systemu docelowego „Zapis danych trendów w obrębie sterownika”). Dlatego do stosowania jako wizualizacja elementu docelowego, którą włączono w ustawieniach systemu docelowego (zakładka rejestru Wizualizacja), należy usunąć z wizualizacji elementy trendów.
- 4900**
„Niedozwolony typ do konwersji”
Stosowana jest konwersja typu, która nie jest obsługiwana przez ustawiony generator kodu.

4901

„Błąd wewnętrzny: przepełnienie dostępu do tablicy!”

Granice tablic są zbyt duże dla zmiennej 32-bitowej. Należy zmniejszyć zakres indeksu tablicy.

14 Indeks

A

ABS, 13-21
 ACOS, 13-25
 ADD, 13-1
 Administrator instrukcji, 9-64
 ADR, 13-14
 Adres
 DeviceNet Master, 9-52
 DeviceNet Slave, 9-52
 Adres, 13-33
 Adres, 13-34
 Adres diagnostyczny
 DeviceNet Master, 9-51
 Adres diagnostyczny, 9-29, 9-36
 Adres instancji, 13-14
 Adres wejściowy
 DeviceNet Master, 9-51
 Adres wyjściowy
 DeviceNet Master, 9-51
 Adresowanie bajtów, 13-105
 Adresowanie bitów, 13-32
 Adresy IEC, 9-26
 ADRINST, 13-14
 Akcja, 2-7, 2-17
 Akcja powiązana w SFC, 2-19
 Akcja w krokach IEC w języku SFC, 2-19, 2-20
 Akcja w SFC, 2-18
 Akcja wejściowa, 2-18, 8-48
 Akcja wejściowa i akcja wyjściowa, 2-18
 Akcja wyjściowa, 2-18, 8-48
 Alarm
 Dezaktywacja, 9-16
 Kolor, 9-15
 Potwierdzenie, 9-11
 Priorytet, 9-11, 9-16
 Stan alarmu, 9-11
 Stan podrzędny, 9-11
 Alarm, 9-11
 ALARM_TASK, 13-108
 ALIAS, 13-44
 Analiza wyrażeń, 2-22, 13-70
 AnalyzationNew.lib, 2-22, 13-70
 AND, 13-4
 Anuluj polecenie, 9-90
 Anuluj wymuszanie, 6-85
 Archiwum, 6-27, 13-81
 Arcus cosinus, 13-25
 Arcus sinus, 13-25
 Arcus tangens, 13-25
 Argument, 2-1, 8-22
 Argument ST, 2-12
 Argumenty, 2-6
 ARRAY, 13-40
 AS
 Analiza wyrażeń przejścia, 2-22
 ASIN, 13-25
 Asystent deklaracji
 Bez struktury, 6-72
 Widok bez struktury, 6-72
 Widok struktury, 6-73
 Ze strukturą, 6-72
 Asystent deklaracji, 6-72
 AT, 8-7, 8-8
 ATAN, 13-25

Atrybuty kroków, 8-50
 Auto Declare, 6-6, 8-9
 Automatyczna kopia zapasowa, 6-4
 Automatyczne deklarowanie, 6-6, 8-9
 Automatyczne zapisywanie kopii zapasowej przed kompilacją, 6-5

B

Batch (linia komend), 13-79
 Baza danych projektu
 Automatyczne funkcje bazy danych, 6-19
 Cofnij wyrejestrowanie, 6-54
 Cofnij wyrejestrowanie wielokrotne, 6-56
 Historia wersji projektu, 6-56
 Kategorie, 10-2
 Login, 6-58
 Odśwież status, 6-58
 Opcje obiektów projektu, 6-19
 Opcje obiektów wspólnych, 6-19
 Opcje plików kompilacji, 6-21
 Pokaż historię wersji, 6-54
 Pokaż różnice, 6-54
 Praca z, 10-2
 Przywołaj, 6-53
 Przywołaj wszystko, 6-56
 Rejestrowanie wielokrotne, 6-56
 Ustal, 6-53
 Ustal wielokrotnie, 6-55
 Wersja etykiet, 6-57
 Właściwości obiektu, 6-65
 Wstaw wspólne obiekty, 6-57
 Wyrejestrowanie wielokrotne, 6-56
 Wyrejestruj, 6-54
 Zarejestruj, 6-54
 Baza danych projektu, 6-51
 Baza danych projektu ENI, 6-18
 BCD_TO_INT, 13-60
 Biblioteka
 AnalyzationNew.lib, 13-70
 Definiowanie, 9-19
 Informacja o licencji, 9-21
 SysTaskInfo.lib, 9-62
 SysTime.lib, 9-62
 Usuwanie, 9-20, 9-21
 Wewnętrzna, 9-19
 Zapisz plik jako, 6-26
 Zewnętrzna, 9-19
 Biblioteka makr
 Podłączanie, 6-23
 Tworzenie, 6-23
 Biblioteka makr, 6-23
 Biblioteka SFC, 2-19
 Biblioteka standardowa, 9-19, 13-49
 Biblioteka wewnętrzna, 6-26, 9-19
 Biblioteka zabezpieczona licencją, 12-1
 Biblioteka zewnętrzna, 6-26, 9-19
 Biblioteki, 2-9
 Biblioteki użytkownika, 9-19
 Bieżący krok, 2-19
 Bitaccess, 8-16
 BITADR, 13-15
 Błąd, 13-80
 Błąd kompilacji, 6-33, 13-121
 BLINK, 13-66

- Blok funkcji
 - Instancja, 2-3
 - Wywołanie, 2-3
- Blok funkcji, 2-2
- Blok funkcji w LD, 8-42
- Blok funkcji w schemacie styków, 2-25
- BOOL, 13-38
- Bramka, 13-85
- Bufor zapisu śledzenia, 9-68, 9-71
- BusDiag.lib, 9-29
- BY, 2-15
- BYTE, 13-38
- Byte-Alignment, 13-97

C

- CAL, 13-15
- CALC, 2-11
- CALCN, 2-11
- Call, 13-83
- CAN
 - Modularne slave, 9-44
- CAN, 9-51
- CanDevice
 - Mapowanie PDO, 9-50
 - Ustawienia CAN, 9-49
 - Ustawienia podstawowe, 9-47
- CanDevice, 9-47
- CanDevice, 9-47
- CanDrv.lib, 9-47
- CANopen slave, 9-47
- CanOpenDevice.lib, 9-47
- CanOpenManager.lib, 9-47
- CASE, 2-13, 2-14
- Cel, 9-85
- Cewka, 2-25, 8-42
- Cewki Set/Reset, 2-25
- CFC
 - Dodaj wejścia/wyjścia, 8-64
 - EN/ENO, 8-58
 - Kolejność, 8-65, 8-66
 - Kolejność przetwarzania, 8-64
 - Kolejność topologicznie, 8-64
 - Kopiowanie elementów, 8-61
 - Makro, 8-67
 - Negowanie, 8-58
 - Pin wejścia, 8-58
 - Pin wyjścia, 8-58
 - Pokaż Kolejność, 8-64
 - Poziomy makr, 8-68
 - Pozycje kursora, 8-55
 - Przejdź do makra, 8-68
 - Przesuwanie elementów, 8-61
 - Rozszerz makro, 8-68
 - Sprzężenie zwrotne, 8-69
 - Tryb online, 8-69
 - Ustaw/Przywróć, 8-58
 - Usuwanie połączeń, 8-62
 - Właściwości, 8-59
 - Wstaw moduł, 8-56
 - Wstaw powrót, 8-57
 - Wstaw skok, 8-57
 - Wstaw wejście, 8-57
 - Wstaw wejście modułu, 8-57
 - Wstaw wyjście, 8-57
 - Wstaw znacznik, 8-57
 - Wstawianie komentarza, 8-57
 - Zaznaczanie elementów, 8-61

- Zmienianie połączeń, 8-62
- Znacznik połączenia, 8-62
- CFC, 2-24, 8-54
- CHARCURVE, 13-68
- Check.lib, 13-46
- CheckBounds, 13-41
- CheckDivByte, 2-2
- CheckDivDWord, 2-2, 13-2
- CheckDivReal, 2-2, 13-2
- CheckDivWord, 2-2, 13-2
- CheckRangeSigned, 13-44
- CheckRangeUnsigned, 13-44
- Cmd (linia poleceń), 13-80
- COB-ID, 9-44, 9-46
- CoDeSys, 1-1
- CoDeSys HMI\pobieranie projektu, 13-79
- Cofnij, 6-68
- Cofnij wyrejestrowanie, 6-54
- Cofnij wyrejestrowanie wielokrotne, 6-56
- Com Cycle Period, 9-42
- CONCAT, 13-50
- CONSTANT, 8-7
- COS, 13-23
- Cosinus, 13-23
- CreateBasicSDOs, 9-44
- CreateCommSDOs, 9-44
- CreateMappingSDOs, 9-44
- CTD, 13-56
- CTU, 13-55
- CTUD, 13-56
- CurrentVisu, 13-109
- Cykl pojedynczy, 2-26, 6-83
- Część deklaracyjna, 2-1, 8-1
- Częstość próbkowania, 9-70, 9-71
- Czynność, 6-66
- Czynności przesłaniają programy, 6-13
- Czyszczenie, 6-34, 13-81

D

- Dane trendów, 13-109
- DATE, 13-39
- DATE_AND_TIME, 13-39
- Debugowanie, 1-1, 2-25, 6-12, 8-23, 13-96
- Default.chk, 6-94
- Default.prg, 6-94
- Default.sts, 6-94
- Deklaracja
 - Pole, 8-10
 - Tablica, 8-10
 - Znacznik, 8-14
- Deklaracja, 8-4, 8-7, 8-8, 8-9
- Deklaracja, 8-14
- Deklaracja, 13-117
- Deklaracja AT, 8-8
- Deklaracja funkcji, 2-1
- Deklaracja z dyrektywami pragma, 8-14
- Deklaracja zmiennych, 6-74, 8-7, 8-14
- Deklaracje jako tabela, 6-7, 8-11
- Deklarowanie
 - Automatyczne, 6-6, 8-9
- Deklarowanie, 8-9
- Deklarowanie zmiennych, 8-4
- Delay, 13-83
- DELETE, 13-51
- DERIVATIVE, 13-61
- Device guid, 13-85
- Device instance, 13-85

- Device name, 13-85
- Device parameter, 13-85
- DeviceNet, 9-51
- DeviceNet Master
 - Parametry, 9-52
 - Parametry modułu, 9-52
 - Parametry podstawowe, 9-51
- DeviceNet Slave
 - Konfiguracja połączenia We/Wy
 - DeviceNet Slave, 9-54
 - Konfiguracja połączenia We/Wy, 9-54
 - Parametry, 9-56
 - Parametry DeviceNet, 9-52
 - Parametry podstawowe, 9-52
- DeviceNet-Slave
 - Modulparameter, 9-56
- Dezaktywowanie alarmów, 9-16, 9-18
- DiagGetState, 9-29
- Diagnoza magistrali, 9-29
- DINT, 13-38
- Dir lib, 13-83
- DIV
 - CheckDivByte, 13-2
 - CheckDivDWord, 13-2
 - CheckDivReal, 13-2
 - CheckDivWord, 13-2
- DIV, 13-2
- DO, 2-16
- Dodaj akcję wejściową, 8-48
- Dodaj akcję wyjściową, 8-48
- Dodaj czynność, 6-66
- Dodaj gałąź równoległą (z prawej), 8-48
- Dodaj język docelowy, 6-37
- Dodaj plik konfiguracyjny, 9-26
- Dodaj zadanie, 9-59
- Dodaj znacznik do gałęzi równoległej, 8-48
- Dodatki
 - Atrybuty kroków, 8-50
 - Czynności synchroniczne, 9-84
 - Dodaj plik konfiguracyjny, 9-26
 - Drukuj bieżące polecenie, 9-90
 - EN/ENO, 8-58
 - Hierarchia wywołań, 9-64
 - Historia do przodu, 9-90
 - Historia do tyłu, 9-90
 - Kanał wielokrotny, 9-72
 - Kolejność Jeden do przodu, 8-65
 - Kolejność Jeden do tyłu, 8-65
 - Kolejność Przejdź Na koniec, 8-66
 - Kolejność Przejdź Na początek, 8-65
 - Kolejność Rozmieść topologicznie, 8-64
 - Kolejność Rozmieść wszystko wg przepływu danych, 8-66
 - Kompresuj, 9-74
 - Konfiguracja śledzenia, 9-69
 - Konfiguracja standardowa, 9-27
 - Monitoring aktywny, 9-67
 - Negacja w LD, 8-44
 - Negowanie w CFC, 8-58
 - Oblicz adresy, 9-26
 - Odczytaj instrukcję, 9-68
 - Opcje w LD, 8-29
 - Opcje w SFC, 8-51
 - Otwórz instancję, 8-2
 - Pokaż Kolejność, 8-64
 - Powiązanie akcji, 8-52
 - Powiększ akcję/przejście, 8-49
 - Powiększanie wywołanego modułu, 8-2, 8-37
 - Powiększenie w CFC, 8-70
 - Powrót o jeden poziom makr, 8-68
 - Przegląd czasów, 8-50
 - Przejdź do makra, 8-68
 - Przekształć, 9-27
 - Rozciągnij, 9-73
 - Rozszerz makro, 8-68
 - Siatka współrzędnych, 9-72
 - Skalowanie Y, 9-72
 - Uruchom śledzenie, 9-71
 - Ustalanie debugowania zadania, 9-64
 - Ustaw/Przywróć w CFC, 8-58
 - Ustaw/Przywróć w FBD, 8-37
 - Ustaw/Przywróć w LD, 8-45
 - Ustawienia monitoringu, 8-23
 - Usuń akcję/przejście, 8-49
 - Usuń znacznik skoku, 8-49
 - Utwórz makro w CFC, 8-67
 - Utwórz szablon dokumentacji, 9-9
 - Wczytaj listę kontrolną, 9-66
 - Wczytaj śledzenie, 9-71
 - Wczytaj śledzenie automatycznie, 9-71
 - Wczytaj wartości, 9-74
 - Wczytaj z programem, 9-83
 - Wklej po, 8-49
 - Włączanie/wyłączanie zadania, 9-64
 - Właściwości w CFC, 8-59
 - Wszystkie poziomy makr wstecz, 8-68
 - Wybierz szablon dokumentacji, 9-10
 - Wyświetl kursor, 9-72
 - Zapis śledzenia w pliku ASCII, 9-74
 - Zapisz historię, 9-90
 - Zapisz instrukcję, 9-68
 - Zapisz listę kontrolną, 9-66
 - Zapisz wartości, 9-74
 - Zapisz wartości śledzenia, 9-74
 - Zastosuj kroki IEC, 8-52
 - Zatrzymaj śledzenie, 9-71
 - Zaznacz wszystko, 8-61
 - Zmień nazwę listy kontrolnej, 9-66
 - Znacznik połączenia, 8-62
- Dodatki, 6-46
- Dodawanie wywołania programu, 9-61
- Dodawanie zabezpieczenia licencyjnego do biblioteki, 6-27
- DOKUFILE, 9-9
- Dokumentacja, 6-31
- Dokumentowanie, 6-40, 13-83
- Dołącz zadanie, 9-59
- Dołączanie
 - Dodaj zadanie Konfiguracja zadań\Dodaj zadanie lub Dołącz zadanie, 9-59
- Dołączanie wywołania programu, 9-61
- Dostęp bitowy, 8-16, 13-32
- Dostęp do zapisu, 6-18
- Dowolne zadanie, 9-60
- Drukuj, 6-30
- Drukuj bieżące polecenie, 9-90
- Drzewo wywołań, 6-33
- DSP301, DSP306, 9-42
- DT, 13-39
- DWORD, 13-38
- Dyrektywa pragma
 - Ogólne, 8-13
- Dyrektywa pragma, 8-14
- Dzielenie przez 0, 13-2
- Dziennik
 - Menu, 9-22

Zapisz, 9-23
Dziennik, 2-28, 6-12, 9-21

E

Echo, 13-82

Edycja

Asystent deklaracji, 6-72
Cofnij, 6-68
Deklaracja zmiennych, 6-74
Kopiuj, 6-69
Makra, 6-75
Następny błąd, 6-74
Poprzedni błąd, 6-75
Przywróć, 6-69
Usuń, 6-70
Wklej, 6-70, 8-37
Wytnij, 6-69, 8-37
Zamień, 6-71
Znajdź, 6-71
Znajdź następny, 6-71

Edycja pliku kompilacji, 6-38

Edytor

CFC, 8-54
Część deklaracyjna, 8-4
FBD, 8-32
IL, 8-25, 8-26
LD, 8-39
ST, 8-27

Edytor deklaracji, 8-4

Edytor IL, 8-25, 8-26

Edytor języka przebiegu, 8-45

Edytor LD, 8-39

Edytor list instrukcji, 8-25, 8-26

Edytor schematów funkcji, 8-32

Edytor SFC, 8-45

Edytor ST, 2-12, 8-27

Edytor tabel, 8-11

Edytor tabeli

Nowa deklaracja, 8-12

Edytory

Granice wydruku, 8-1

Edytory, 8-1

Edytory deklaracji w trybie online, 8-12

Edytory graficzne, 8-28

Edytory sieci

Tryb online, 8-31

Edytory sieci, 8-31

Edytory tekstowe

Online, 8-23

Edytory tekstowe, 8-22

Edytory tekstowe, 8-23

Edytuj informację o licencji, 6-27

Edytuj obiekt, 6-64

Eigenschaften eines Objekts, 6-64

Ekran startowy, 13-79

Eksport list parametrów, 9-85

Eksportowanie, 6-41

Eksportowanie modułu, 9-27

Eksportuj, 6-41, 13-81

Eksportuj moduł, 9-27

ELSE, 2-14

ELSIF, 2-14

EN/ENO w CFC, 8-58

END_CASE, 2-14

END_FOR, 2-15

END_IF, 2-14

END_PROGRAM, 2-5

END_REPEAT, 2-16

END_TYPE, 13-43, 13-44

END_VAR, 8-4, 8-5

END_WHILE, 2-16

Engineering Interface ENI, 6-18, 6-19, 6-21

ENI, 6-18, 6-51, 13-86

Enumeracja, 9-69, 13-43

EQ, 13-13

Etykieta

Edytory, 8-1

Monitorowanie, 8-23

Organizer obiektów, 6-60

Pasek funkcji, 6-2

SFC, 8-45, 8-52

Etykieta

Przeładowanie PLC, 9-88

EXIT, 2-13, 2-17

EXP, 13-23

EXPT, 13-26

EXTERNAL, 8-7

EXTRACT, 13-60

F

F_TRIG, 13-54

FBD

Pozycja kursora, 8-33

FBD, 2-23, 8-32

FBD w trybie online, 8-38

FIND, 13-52

Flag

Noinit, 8-14

Noread, 8-14

Nowatch, 8-14

Nowrite, 8-14

Flaga systemowa, 13-32

Folder, 6-60, 6-61

Fonty, 13-109

FOR, 2-15

Format danych 8.3, 13-108

Formatowanie

Automatyczne, 6-6

Formatowanie, 6-6

Formatuj automatycznie, 6-6

FREQ_MEASURE, 13-67

FUNCTION, 2-1

Funkcja, 2-1

Funkcja adresu, 13-14, 13-15

Funkcja callback, 9-62

Funkcja CheckPointer, 13-42

Funkcja CheckPointerAligned, 13-42

Funkcja CheckPointerAligned, 13-42

Funkcja Intellisense, 6-7, 8-2

Funkcja standardowa, 9-19, 13-49

Funkcja TIME, 13-34

Funkcja wykładnicza, 13-23

Funkcje edycji, 6-68

Funkcje online, 6-75

Funkcje sieciowe, 13-95

Funkcje String, 13-49

Funkcjonalność sieci, 9-3

G

Gałąź alternatywna (w prawo), 8-47

Gałąź alternatywna (z lewej), 8-47

Gałąź alternatywna w AS, 2-23

Gałąź alternatywna w SFC, 8-47

Gałąź równoległa w AS, 2-23
 Gałąź równoległa w SFC, 8-47
 Gateway
 About, 6-89
 Change Password, 6-89
 Exit, 6-89
 GatewayDDE Server, 11-1
 GE, 13-13
 GEN, 13-67
 Generuj wszystkie SDO, 9-44
 GetBusState, 9-30
 Graficzny edytor schematów funkcji, 2-24, 8-54
 Granice wydruku, 8-1
 Grupa alarmów, 9-15
 Grupa robocza, 13-79
 Grupa robocza, 6-50
 Grupa robocza\hasło za pomocą linii poleceń, 13-79
 GT, 13-11

H

Hasła, 6-14
 Hasła dla grup roboczych, 6-51
 Hasło, 6-15, 6-50, 13-81
 Hasło zabezpieczenia przed zapisem, 6-15
 Heartbeat, 9-44
 Heartbeat Master, 9-42
 Hierarchia wywołań, 6-87, 9-64
 Historia do przodu, 9-90
 Historia do tyłu, 9-90
 Historia wersji, 6-54
 Historia wersji projektu, 6-56
 Hitachi SH, 13-101
 HYSTERESIS, 13-70

I

ID modułu, 9-29
 ID narzędzia, 9-94
 ID partnera, 12-1
 Identifier
 Offline tooltip, 8-2
 Identyfikator, 8-7, 13-31, 13-117
 IEC61131-3, 2-28
 IF, 2-14
 Ignorowanie ostrzeżeń – F4, 6-8
 Ignoruj komentarze, 6-43
 Ignoruj spacje, 6-43
 Ignoruj właściwości, 6-43
 IL, 2-9, 8-25, 8-26
 IL w trybie online, 8-27
 Import
 Plik S5, 13-90
 Pliki Siemens, 13-89
 Import, 6-41
 Import list parametrów, 9-85
 Import pliku symboli SEQ, 13-89
 Import Siemens
 Konwersja, 13-90
 Plik S5, 13-90
 Plik symboli SEQ, 13-89
 Importowanie modułu, 9-27
 Importowanie plików Siemens, 6-42
 Importuj, 6-41, 13-81
 Importuj moduł, 9-27
 Indeksy, 6-33
 Indeksy modułu, 6-33
 INDEXOF, 13-4

Infineon C16x, 13-98
 Info
 DeviceNet Slave, 9-52
 Informacja diagnostyczna, 9-56
 Informacja o licencji, 6-47
 Informacja o licencji biblioteki, 12-1
 Informacja o licencji dla bibliotek, 9-19
 Informacja o pozycji, 6-36
 Informacja o projekcie, 6-47
 Informacje licencyjne, 12-1
 Informacje o kompilacji, 6-94
 Informacje o pobieraniu, 6-34, 6-79, 6-80, 6-94
 Informacje o użytkowniku, 6-5
 Informacje/diagnoza z systemu docelowego, 9-56
 Inicjalizacja, 6-79, 8-7, 8-14
 Inicjalizacja zerowa, 13-105
 Inicjalizacja zmiennych retain, 13-26
 INSERT, 13-51
 Instancja bloku funkcji
 Adres, 13-14
 Instancje bloków funkcji, 2-3
 Instancje w menedżerze parametrów, 9-79
 Instrukcja, 2-9, 2-12
 Instrukcja CASE, 2-14
 Instrukcja dot. kompilacji, 8-14
 Instrukcja IF, 2-13, 2-14
 Instrukcja kompilacji, 8-14
 Instrukcja RETURN, 2-14
 INT, 13-38
 INT_TO_BCD, 13-60
 INTEGRAL, 13-62
 Intel Byte Order, 13-99, 13-101
 Interfejs DDE
 Łączenie za pomocą Intouch, 11-2
 Łączenie za pomocą programu EXCEL, 11-2
 Łączenie za pomocą programu WORD, 11-1
 Obsługa serwera GatewayDDE, 11-2
 Opcje linii poleceń serwera GatewayDDE, 11-4
 Serwer GatewayDDE i EXCEL, 11-4
 Serwer GatewayDDE i WORD, 11-4
 Uruchom, 11-1
 Wczytywanie serwera GatewayDDE, 11-3
 Zapytanie, 11-1
 Zapytanie do serwera GatewayDDE, 11-3
 Interfejs DDE, 11-1
 Interfejs symboli, 6-16

J

Język
 Pokaż skompilowany projekt, 6-39
 Przełącz kompilację, 6-40
 Język, 6-9
 Język przebiegu, 2-17, 8-45
 Języki IEC, 2-9

K

Kanał Gateway, 6-91, 6-92
 Kanał parametry podstawowe, 9-31
 Kanał Profibus, 9-37
 Kanał wielokrotny, 9-72
 Kanały bitowe, 9-32
 Kasowanie wyjścia, 8-45
 Katalog, 6-11
 Katalog biblioteki, 6-11, 9-20, 13-83
 Katalog plików kompilacji, 6-11, 13-83
 Katalog plików konfiguracyjnych, 6-11, 13-83

- Katalog plików upload, 6-11, 13-83
- Katalog plików wizualizacji, 6-11
- Katalog projektu, 6-11
- Kategoria obiektu, 10-2
- Klasa alarmów
 - Plik zapisu, 9-12
- Klasa alarmów, 9-12
- Kod projektu, 6-94
- Kodowanie XML, 6-9
- Kolejność przetwarzania w CFC, 8-64
- Kolory, 6-10
- Komentarz
 - Moduł, 9-29
 - SFC, 8-50
- Komentarz, 8-1, 8-11, 8-29
- Komentarz w CFC, 8-57
- Komentarze zagnieżdżone, 6-13
- Kompatybilne z Intel 386, 13-96
- Kompilacja na inny język, 6-35
- Kompilowanie, 6-13, 6-33, 6-34
- Kompilowanie LREAL jako REAL, 6-14
- Kompilowanie przyrostowe, 6-33
- Kompiluj
 - Pokaż skompilowany projekt, 6-39
- Kompiluj, 6-33, 6-34
- Kompiluj, 6-78
- Kompiluj, 6-78
- Kompiluj, 13-81
- Kompiluj projekt (na inny język), 6-38
- Kompiluj wszystko, 6-34
- Kompresja, 13-109
- Kompresuj, 9-74
- Komunikacja
 - Interfejs symboli, 6-17
- Komunikacja, 6-17
- Komunikacja DDE z CoDeSys, 11-1
- Komunikaty, 13-82
- Komunikaty o błędach, 13-121
- Konfiguracja CAN
 - Mapowanie PDO, 9-45
- Konfiguracja CAN, 9-41
- Konfiguracja CANopen, 13-105
- Konfiguracja klas alarmów, 9-12
- Konfiguracja klasy alarmów, 9-12
- Konfiguracja pliku symboli, 6-17
- Konfiguracja śledzenia
 - Częstość próbkowania, 9-70
 - Krawędź wyzwalająca, 9-70
 - Poziom wyzwalania, 9-70
 - Pozycja wyzwalania, 9-70
 - Zewnętrzne konfiguracje śledzenia, 9-75
 - Zmienna wyzwalająca, 9-70
- Konfiguracja śledzenia, 9-69
- Konfiguracja standardowa, 9-27
- Konfiguracja sterownika
 - CAN, 9-41
 - Dodawanie elementów, 9-26
 - Dodawanie pliku konfiguracyjnego, 9-26
 - Eksportuj moduł, 9-27
 - Formaty, 9-23
 - Importuj moduł, 9-27
 - Kanały bitowe, 9-32
 - Mapowanie PDO, 9-45
 - Master DP, 9-33, 9-34
 - Moduł główny, 9-25
 - Moduły CANopen, 9-41
 - Parametry modułu, 9-31
 - Parametry niestandardowe, 9-31
 - Parametry podstawowe, 9-36
 - Parametry podstawowe kanału, 9-31
 - Plik konfiguracyjny, 9-23
 - Pobierz jako plik, 13-105
 - Profibus, 9-32
 - Przeгляд, 9-25
 - Przekształcanie starych konfiguracji sterownika, 9-27
 - SDO, 9-46
 - Skanowanie konfiguracji modułu, 9-57
 - Slave DP, 9-36
 - Tryb online, 9-56
 - Wczytanie statusu modułu, 9-57
 - Wyświetlanie komunikatów diagnostycznych, 9-57
 - Zamienianie lub przełączanie elementów, 9-26
 - Zaznaczanie elementów, 9-25
- Konfiguracja sterownika, 13-105
- Konfiguracja sterownika wersja 2.1, 9-23
- Konfiguracja symboli, 6-16, 13-106
- Konfiguracja symboli z pliku INI, 6-17
- Konfiguracja zadań
 - Biblioteki, 9-62
 - Dodaj wywołanie programu, 9-61
 - Kolejność przetwarzania, 9-64
 - Przebieg czasowy, 9-63
 - Status zadania, 9-62
 - W trybie online, 9-62
 - Zdarzenia systemowe, 9-61
- Konfiguracja zadań, 2-7, 9-57
- Konfiguracja zmiennych
 - Wstaw ścieżki instancji, 9-9
- Konfiguracja zmiennych, 8-17, 9-7
- Konfigurowanie ENI, 6-19
- Konflikt dostępu, 6-14
- Konflikt dostępu, 6-50
- Konkatenacja, 13-50
- Kontrola czasu w zadaniach, 9-60
- Kontrola Gateway, 6-89
- Kontrola przebiegu
 - Edytor sieci, 8-31
 - FBD, 8-38
- Kontrola przebiegu, 6-88, 8-27
- Kontrola watchdog w zadaniach, 9-60
- Kontrola dla urządzenia DeviceNet Slave, 9-53
- Konwersja BCD, 13-60
- Konwersja pliku S5, 13-90
- Konwersje BOOL_TO, 13-16, 13-17
- Konwersje DATE_TO, 13-19
- Konwersje DT_TO, 13-19
- Konwersje LREAL_TO, 13-18
- Konwersje REAL_TO, 13-18
- Konwersje STRING_TO, 13-20
- Konwersje TIME_TO, 13-18
- Konwersje TOD_TO, 13-18
- Konwersje typów, 13-15
- Konwersje w zakresie typów liczb całkowitych, 13-18
- Konwertowanie obiektu, 6-63
- Konwertuj, 6-63
- Kopia zapasowa
 - Automatycznie, 6-4
- Kopia zapasowa, 6-4
- Kopia zapasowa przed kompilacją, 6-33
- Kopiowanie elementów w CFC, 8-61
- Kopiowanie w CFC, 8-61
- Kopiowanie w FBD, 8-37
- Kopiuj, 6-46, 6-63, 6-69
- Kopiuj obiekt, 6-63
- Krawędź opadająca, 13-54
- Krawędź wyzwalająca, 9-70

Krawędź wzrastająca, 13-54, 13-67
 Krok, 2-18
 Krok i przejście w SFC – usuwanie, 8-47
 Krok IEC, 2-19, 8-52
 Krok Init, 2-19
 Krok pojedynczy, 2-26, 6-83
 Krzyżyk, 6-51
 Kwalifikator, 2-19, 2-20
 Kwalifikatory w krokach SFC, 2-20

L

LD
 Cewka, 2-25
 Komentarze, 8-29
 Pozycja kursora, 8-39
 Set/Reset, 2-25
 Styk, 2-24
 Styki równoległe, 2-24
 Wejście EN, 2-25
 Wstawianie komentarza, 8-30
 LD, 2-24
 LD, 8-39
 LE, 13-11
 Lecsf.lib, 2-19
 LEFT, 13-49
 LEN, 13-49
 Library, 9-18
 Licencjonowanie, 6-47, 12-1
 Liczba segmentów danych, 6-14
 Licznik w dół, 13-56
 Licznik w górę, 13-55
 Licznik w górę/w dół, 13-56
 LIMIT, 13-10
 Limit czasu komunikacji, 6-9
 Limit czasu komunikacji dla pobierania, 6-9
 LIMITALARM, 13-70
 LIN_TRAFO, 13-62
 Linia poleceń, 13-79
 Lista instrukcji, 2-9, 8-25, 8-26
 Lista kontrolna, 9-64
 Lista odsyłaczy, 6-33
 Lista parametrów
 Czynności synchroniczne, 9-84
 Eksport, 9-85
 Import, 9-85
 Instancja, 8-17
 Mapowanie, 9-79
 Parametry, 9-79
 Pobieranie, 9-84
 Przesyłanie, 9-84
 Rozmieść, 9-82
 Sortowanie, 9-84
 Szablon, 8-17
 Typy, 9-79
 Usuń, 9-82
 Wczytaj z programem, 9-83
 Wklej, 9-81
 Wpisy przy użyciu pragmy, 8-16
 Zmień nazwę, 9-82
 Zmienne, 8-16
 Lista poleceń, 9-88
 Lista zmiennych globalnych
 Edycja, 9-6
 Tworzenie, 9-3
 Lista zmiennych globalnych, 6-65
 Listy parametrów
 Edycja, 9-83

Listy parametrów w projekt bootowalny, 9-85
 LN, 13-22
 LOG, 13-22
 Logarytm, 13-22
 Login, 6-75, 13-80
 Logout, 13-80
 Logowanie do bazy danych ENI, 6-58
 Logowanie online, 6-75
 LREAL, 13-38
 LT, 13-11

M

MAINTARGETVISU_INPUT_CODE, 13-110
 MAINTARGETVISU_PAINT_CODE, 13-110
 Makra, 6-75
 Makro
 Makro po kompilacji, 6-14
 Makro przed kompilacją, 6-14
 Opcje, 6-22
 Makro, 6-22
 Makro w CFC, 8-67
 Makro w przeglądarce PLC, 9-89
 Maks. ilość modułów, 13-104
 Maks. liczba segmentów danych globalnych, 13-104
 Maksymalny rozmiar komentarza, 8-30
 Mapa bitowa w widoku strony, 6-32
 Maping PDO, 9-45
 Mapowanie, 9-50, 9-79, 13-107
 Mapowanie PDO, 9-79
 Mapy bitowe modułów, 6-7
 Master CAN
 Parametry CAN, 9-41
 Parametry modułu, 9-43
 Parametry podstawowe, 9-41
 Uruchom automatycznie, 9-41
 Master DP
 Parametry DP, 9-33
 Parametry magistrali, 9-34
 Parametry modułu, 9-33
 Właściwości grupy, 9-34
 Master DP parametry podstawowe, 9-33
 MAX, 13-9
 Menedżer bibliotek
 Biblioteka standardowa, 9-19, 13-49
 Korzystanie, 9-18
 Usuwanie biblioteki, 9-20, 9-21
 Wstaw bibliotekę, 9-20
 Menedżer bibliotek, 9-18
 Menedżer parametrów
 Aktywuj, 13-107
 Atrybuty, 9-76
 Dla mapowania PDO, 9-50
 Eksport, 9-85
 Import, 9-85
 Instancja, 9-79
 Lista parametrów, 9-79
 Lista parametrów, 9-76
 Pobieranie i przesyłanie, 9-84
 Pragmy, 8-16
 Szablon, 9-79
 Wczytaj z programem, 9-84
 Menedżer parametrów w trybie online, 9-84
 Menedżer podglądu zmiennych i receptur
 Monitoring, 9-67
 Nowa lista kontrolna, 9-66
 Odczytaj instrukcję, 9-68
 Offline, 9-65

- Tryb online, 9-66
- Wczytaj listę kontrolną, 9-66
- Wymuszanie i zapisywanie wartości, 9-68
- Zapisz instrukcję, 9-68
- Zapisz listę kontrolną, 9-66
- Zmień nazwę listy kontrolnej, 9-66
- Menu Dołącz, 9-59
- Menu Dziennik, 9-22
- Menu Edycja, 6-68
- Menu Gateway, 6-89
- Menu Plik, 6-24
- Menu podręczne, 6-3
- Menu Pomoc, 6-95
- Menu Projekt, 6-24
- MID, 13-50
- MIN, 13-10
- Minimalny rozmiar komentarza, 8-30
- MIPS III ISA, 13-100
- MOD, 13-3
- Moduł, 1-1, 2-1, 2-5, 6-2
- Moduł CAN
 - Mapowanie PDO, 9-45
 - Parametry CAN, 9-43
 - Parametry podstawowe, 9-43
- Moduł CANopen, 9-41
- Moduł EN, 2-25
- Moduł EN w LD, 8-43
- Moduł funkcji
 - Instancja, 2-3
 - Wywołanie, 2-3
- Moduł funkcji, 2-2
- Moduł główny (root), 9-25
- Moduł Profibus, 9-37
- Moduł w CFC, 8-56
- Moduł w FBD, 8-34
- Moduł we/wy
 - Parametry modułu, 9-31
- Modularne urządzenia slave, 9-44
- Moduły biblioteki
 - Przegląd, 13-73
- Moduły regulatora, 13-63
- Moduły standardowe, 2-1
- Modyfikator, 2-10
- Modyfikator C w IL, 2-10
- Modyfikatory i operatory w IL, 2-10
- Monit podczas bootowania, 9-5
- Monitor sterownika, 9-87
- Monitoring, 2-26, 8-12, 8-14, 8-23
- Monitorowanie czasu w edytorze SFC, 8-50
- Monitorowanie typów złożonych, 6-7
- Motorola 68K, 13-97
- MOVE, 13-3
- MUL, 13-2
- MUX, 13-10

N

- Nadawanie identyfikatorom nazw, 13-117
- Nadawanie nazwy wizualizacji, 13-120
- Nakładające się obszary pamięci, 6-14, 6-49
- Narzędzia
 - DefaultDisplayName, 9-92
 - Następna różnica, 6-45
 - Negacja w FBD, 8-35
 - Poprzednia różnica, 6-45
 - Wklej nad, 8-44
 - Wklej pod w LD, 8-44
 - Wklej za, 8-44

- Właściwości obiektu, 9-91
- Właściwości skrótów, 9-91
- Zastosuj pojedynczą zmianę, 6-46
- Zastosuj prawa dostępu, 6-46
- Zastosuj właściwości, 6-46
- Zastosuj zmianę, 6-46
- Narzędzia, 9-90
- Następna różnica, 6-45
- Następny błąd, 6-74
- Nazwa bloku funkcji, 13-119
- Nazwa funkcji, 13-119
- Nazwa instancji, 2-2
- Nazwa modułu, 13-119
- Nazwa programu, 13-119
- Nazwa typu danych, 13-119
- Nazwa wizualizacji, 13-120
- Nazwa zmiennej, 8-7
- Nazwy POU, 13-119
- Nazwy zmiennych, 13-117
- NE, 13-14
- Negacja w FBD, 8-35
- Negacja w LD, 8-44
- Negowanie w CFC, 8-58
- Nie inicjuj, 9-44
- Nie zapisuj danych komunikacji w projekcie, 6-8
- Nie zmieniaj adresów automatycznie, 9-29
- Nie używane zmienne, 6-14, 6-49
- Node ID
 - DeviceNet Master, 9-51
- Node ID, 9-43
- Nodeguarding, 9-44
- Noinfo (linia poleceń), 13-79
- Nonpersistent
 - Pragma, 8-21
- Norma, 2-28
- NOT, 13-6
- Notargetchange (linia poleceń), 13-79
- Nowa deklaracja, 8-12
- Nowy folder, 6-61
- Nowy plik, 13-81
- Nowy z szablonu, 6-24
- Numer listy zmiennych, 9-5
- Numer sieci, 8-29
- Numer węzła, 9-29
- Numery wierszy, 8-11
- Numery wierszy edytora tekstowego, 8-25
- Numery wierszy w edytorze deklaracji, 8-11

O

- Obiekt, 2-1, 6-60
- Obiekt wizualizacji|właściwości, 6-65
- Oblicz adresy, 9-26
- Obsługa alarmów, 13-108
- Obsługa tabeli za pomocą klawiatury, 13-109
- Obsługa wprowadzania danych w zakresie wizualizacji, 13-109
- Obsługiwane fonty, 13-109
- Obszar roboczy, 6-2, 6-8
- Ochrona projektu, 6-15, 6-26, 6-51
- Odczytaj instrukcję, 9-68
- Odpowiedź na monit bootowania, 9-6
- Odstęp czasu w zadaniach, 9-60
- Odśwież status, 6-58
- OF, 2-14
- Offline mode, 13-86
- Ogólne funkcje online, 6-75
- Ogólne ustawienia, 9-27

- Ograniczenie zakresu typów danych, 13-44
 - Okno
 - Dziennik, 6-95
 - Jedno pod drugim, 6-95
 - Kaskada, 6-95
 - Komunikaty, 6-95
 - Menedżer bibliotek, 6-95
 - Rozmieść symbole, 6-95
 - Sąsiadująco, 6-95
 - Zamknij wszystkie, 6-95
 - Zarządzanie biblioteką, 6-95, 9-18
 - Okno dialog. Zapisywanie/wymuszanie, 6-86
 - Okno dialogowe parametrów specyficzne dla aplikacji, 9-28
 - Okno dialogowe punktu wstrzymania, 6-82
 - Okno Dziennik, 6-95, 9-21
 - Okno główne, 6-1
 - Okno komunikatów, 6-2, 6-33, 6-48
 - Okno tematów pomocy, 6-95
 - Oneror, 13-80
 - Online
 - Anuluj wymuszanie, 6-85
 - cykl pojedynczy, 6-83
 - Cykl pojedynczy, 2-26
 - Hierarchia wywołań, 6-87
 - Kontrola przebiegu, 6-88
 - Krok pojedynczy, 2-26
 - Krok pojedynczy przez, 6-83
 - Krok pojedynczy w, 6-83
 - Okno dialog. Zapisywanie/wymuszanie, 6-86
 - Okno dialogowe punktu wstrzymania, 6-82
 - Parametry komunikacji, 6-88, 6-90
 - Punkt wstrzymania, 2-26
 - Reset, 6-81
 - Reset zimny, 6-81
 - Reset źródła, 6-81
 - Start, 6-81
 - Stop, 6-81
 - Symulacja, 6-88
 - Utwórz projekt bootowalny, 6-94
 - Wczytaj, 6-80
 - Wczytaj kod źródłowy, 6-94
 - Wczytaj plik ze sterownika, 6-94
 - Włącz/wyłącz punkt wstrzymania, 6-82
 - Wyloguj, 6-80
 - Wymuszanie wartości, 6-85
 - Zapisz plik w sterowniku, 6-94
 - Zapisz wartości, 6-83
 - zmienianie wartości, 2-26
 - Online, 1-1, 1-2
 - Online (linia poleceń), 13-79
 - Online mode, 13-86
 - Online Wczytaj, 6-80
 - Opcje
 - Baza danych projektu, 6-18
 - Dziennik, 6-12
 - Edytor, 6-6
 - Informacje o użytkowniku, 6-5
 - Katalogi, 6-11
 - Kolory, 6-10
 - Konfiguracja symboli, 6-16
 - Makra, 6-22
 - Obszar roboczy, 6-8
 - Pobieranie źródła, 6-16
 - Wczytywanie i zapisywanie, 6-4
 - Opcje, 9-76
 - Opcje edytora, 6-6
 - Opcje kompilacji, 6-12
 - Opcje projektu, 9-76
 - Openfromplc (linia poleceń), 13-79
 - Operator ADD w IL, 2-10
 - Operator AND w IL, 2-10
 - Operator CAL w IL, 2-11
 - Operator DIV w IL, 2-10
 - Operator EQ w IL, 2-10
 - Operator GE w IL, 2-10
 - Operator GT w IL, 2-10
 - Operator IL, 2-10
 - Operator INI, 13-26
 - Operator inicjalizacji, 13-26
 - Operator JMP w IL, 2-11
 - Operator LD w IL, 2-10
 - Operator LE w IL, 2-10
 - Operator LT w IL, 2-10
 - Operator MUL w IL, 2-10
 - Operator NE w IL, 2-10
 - Operator OR w IL, 2-10
 - Operator przyporządkowania, 2-13
 - Operator R w IL, 2-10
 - Operator RET w IL, 2-11
 - Operator S w IL, 2-10
 - Operator ST, 2-12
 - Operator ST w IL, 2-10
 - Operator SUB w IL, 2-10
 - Operator XOR w IL, 2-10
 - Operator zawartości, 13-15, **13-41**
 - Operatory
 - Przegląd, 13-73
 - Operatory, 8-22, 13-1
 - Opóźnienie włączenia, 13-58
 - Opóźnienie wyłączenia, 13-58
 - Optymalizacja, 13-96
 - OR, 13-5
 - Organizer obiektów, 6-2
 - Ostrzeżenia, 13-121
 - Otwieranie modułu, 6-66
 - Otwórz instancję, 6-66, 8-2
 - Otwórz plik, 13-81
 - Otwórz projekt z bazy danych projektów, 6-25
 - Otwórz projekt ze sterownika, 6-25
 - Out (linia poleceń), 13-79
- P**
- PACK, 13-61
 - Pakiet Target Support Package, 9-86
 - Pakuj zmienne, 9-5
 - Pamięć danych, 13-104
 - Parameter
 - DeviceNet-Slave, 9-56
 - Parametry
 - DeviceNet Master, 9-52
 - Parametry CAN w module CAN, 9-43
 - Parametry CAN w urządzeniu Master CAN, 9-41
 - Parametry DeviceNet
 - DeviceNet Slave, 9-52
 - Parametry DP w urządzeniu master DP, 9-33
 - Parametry DP w urządzeniu slave DP, 9-36
 - Parametry kanału, 9-32
 - Parametry komunikacji
 - Konfigurowanie serwera Gateway, 6-92
 - Porady dotyczące edycji, 6-93
 - Skrócona kontrola, 6-93
 - wybieranie serwera Gateway, 6-91
 - Parametry komunikacji, 6-90
 - Parametry komunikacji, 6-93

- Parametry komunikacji Gateway, 6-88
- Parametry magistrali master DP, 9-34
- Parametry modułu
 - DeviceNet Master, 9-52
 - DeviceNet Slave, 9-56
- Parametry modułu Master CAN, 9-43
- Parametry modułu master DP, 9-33
- Parametry modułu slave DP, 9-40
- Parametry modułu w module we/wy, 9-31
- Parametry podstawowe
 - DeviceNet Master, 9-51
 - DeviceNet Slave, 9-52
- Parametry podstawowe kanału, 9-31
- Parametry podstawowe w urządzeniu Master CAN, 9-41, 9-43
- Parametry podstawowe w urządzeniu master DP, 9-33
- Parametry podstawowe w urządzeniu slave DP, 9-36
- Parametry użytkownika, 9-28
- Parametry użytkownika urządzenia slave DP, 9-39
- Pasek funkcji, 6-2, 6-8
- Pasek menu, 6-1
- Pasek stanu, 6-3, 6-8
- Password (linia poleceń), 13-79
- PD, 13-63
- PDO-Mapping
 - CAN-Device, 9-50
- Persist.dat, 8-21
- Persistent, 8-21
- PERSISTENT, 8-6
- Pętla, 2-11
- Pętla FOR, 2-13, 2-15
- Pętla REPEAT, 2-16
- Pętla WHILE, 2-13, 2-16
- PID, 13-64
- PID_FIXCYCLE, 13-66
- Pierwiastek kwadratowy, 13-22
- Pierwotna kolejność, 8-12
- Pin wejścia, 8-58
- Pin wyjścia, 8-58
- Platforma docelowa, 13-95
- PLC_PRG, 2-7
- Plik
 - Dokumentowanie, 6-31
 - Drukuj, 6-30
 - Nowy, 6-24
 - Nowy z szablonu, 6-24
 - Otwórz, 6-24
 - Ustawienia drukowania, 6-31
 - Zakończ, 6-33
 - Zamknij, 6-25
 - Zapisz, 6-26
 - Zapisz jako, 6-26
 - Zapisz/wyślij archiwum, 6-27
- Plik, 6-24
- Plik asd, 6-4
- Plik asl, 6-5
- Plik bak, 6-4
- Plik cfg, 9-26, 9-27
- plik ci, 6-28
- Plik con, 9-27
- Plik dcf, 9-41
- Plik DCF, 9-3
- Plik docelowy, 9-86
- Plik eds, 9-48
- Plik eksportu, 9-3
- Plik esd, 9-41
- Plik gsd, 9-32
- Plik Info, 9-86
- Plik języka, 6-36
- Plik kompilacji, 6-35, 6-36, 6-38
- Plik komunikatów, 13-82
- Plik konfiguracyjny, 9-23, 9-27, 9-32, 9-41
- Plik mon, 9-75, 9-76
- Plik poleceń, 13-80
- Plik prn, 9-85
- plik ri, 6-35, 6-79
- Plik ri, 6-80, 6-94
- Plik symboli, 6-16, 8-14
- Plik symboli SEQ, 13-89
- Plik tcf, 9-68
- Plik tlt, 6-36
- Plik tnf, 9-85
- Plik trc, 9-68
- Plik vis, 6-36
- Plik zapisu, 9-15
- Plik zapisu dla alarmów, 9-17
- Pliki wizualizacji, 13-109
- Pliki zip do wizualizacji sieciowej, 13-109
- Pobieranie do CoDeSys HMI, 13-79
- Pobieranie kodu źródłowego, 6-16, 6-94
- Pobieranie konfiguracji sterownika, 13-105
- Pobieranie list parametrów, 9-84
- Pobieranie pliku ze sterownika, 6-94
- Pobieranie źródła, 6-16
- Pobierz, 6-78
- Pobierz jako plik, 13-105
- Podział pamięci, 13-95, 13-103
- POINTER, 13-33, **13-41**
- Pokaż kolejność, 8-64
- Pokaż różnice, 6-54
- Pokaż skompilowany projekt, 6-39
- Pokazywanie obszarów wydruku, 6-8
- Pokazywanie symboli modułów, 6-7
- Pola, 2-1, 13-40
- Pole numeru sieci, 6-82, 6-88
- Pole numeru wierszy, 6-82, 6-88, 8-24
- Polecenia batch, 13-79
- Polecenia standardowe, 9-87
- Pomiar częstotliwości, 13-67
- Pomijanie generowania SDO, 9-44
- Pomoc
 - Pomoc kontekstowa, 6-96
 - Zawartość oraz indeks, 6-95
- Pomoc kontekstowa, 6-96
- Pomoc, Zawartość i wyszukiwanie, 6-95
- Poprzedni błąd, 6-75
- Poprzednia różnica, 6-45
- Porównaj, 6-42
- Porównaj zmiany, 6-43
- Porównanie projektów, 6-42
- porównywanie z projektem ENI, 6-43
- Potęgowanie, 13-26
- Potwierdzanie alarmów, 9-11
- Power PC, 13-99
- Powiązanie akcji, 8-52
- Powiązanie akcji w SFC, 2-20
- Powiększ, 8-70
- Powiększ akcję, 8-49
- Powiększ przejście, 8-49
- Powiększanie wywołanego modułu, 8-2, 8-37
- Powiększenie
 - CFC, 8-70
- Powiększenie w edytorach graficznych, 8-28
- Powrót o jeden poziom makr, 8-68
- Powrót w CFC, 8-57
- Powrót w FBD, 8-34

- Powrót w LD, 8-44
- Poziom makra w CFC, 8-68
- Poziom wyzwalania, 9-70
- Pozycja punktu wstrzymania, 6-82
- Pozycja wyzwalania, 9-70
- Pozycje kursora w CFC, 8-55
- Pozycje kursora w edytorze LD, 8-39
- Pozycje kursora w FBD, 8-33
- Pozycje punktów wstrzymania w edytorze tekstowym, 8-24
- Pragma
 - Nonpersistent, 8-21
- Pragma, 8-14, 8-16
- Pragma, 9-76
- Pragma {nonpersistent}, 8-21
- pragma wyświetlająca deklaracje biblioteki, 8-20
- Pragmy listy parametrów, 8-16
- Prawa dostępu, 6-66
- Prawa dostępu do obiektu, 6-66
- Printersetup, 13-81
- Priorytety operatorów ST, 2-12
- Procesor zmiennoprzecinkowy, 13-96, 13-101
- Profibus, 9-32
- Profibus Master, 9-32
- Profibus Slave, 9-32
- Program, 2-5
- PROGRAM, 2-5
- Program główny, 2-7
- Projekt
 - Baza danych projektu, 6-51
 - Dodaj czynność, 6-66
 - Dokumentowanie, 6-40
 - Edytuj obiekt, 6-64
 - Eksportuj, 6-41
 - Hasła dla grupy roboczej, 6-51
 - Import Siemens, 6-42
 - Importuj, 6-41
 - Kompilacja na inny język, 6-35
 - Kompiluj, 6-33
 - Kompiluj wszystko, 6-34
 - Konwertuj obiekt, 6-63
 - Kopiuj, 6-46
 - Kopiuj obiekt, 6-63
 - Opcje, 6-3
 - Otwórz instancję, 6-66
 - Pokaż skompilowany projekt, 6-39
 - Porównaj, 6-42
 - Prawa dostępu do obiektu, 6-66
 - Przełącz kompilację, 6-40
 - Sprawdź, 6-49
 - Szukaj globalnie, 6-48
 - Usuń obiekt, 6-61
 - Wczytaj informacje o pobieraniu, 6-34
 - Właściwości obiektu, 6-64
 - Wstaw obiekt, 6-61
 - Wyczyść wszystko, 6-34
 - Wyświetl drzewo wywołań, 6-68
 - Wyświetl listę odsyłaczy, 6-67
 - Zastąp globalnie, 6-49
 - Zmień nazwę obiektu, 6-63
- Projekt, 1-1, 2-1
- Projekt bootowalny, 6-5, 6-13, 9-85, 13-80, 13-106
- Projekt dziennika, 9-23
- Projekt porównawczy, 6-42
- Projekt Sprawdź
 - Konflikt dostępu, 6-50
 - Nakładające się obszary pamięci, 6-49
 - Nie używane zmienne, 6-49
 - Wielokrotny zapis na wyjściu, 6-50
 - Przeciagnij i upuść, 6-60
 - Przedstawienie graficzne zapisu śledzenia, 9-71
 - Przeglądarka PLC, 9-87, 13-106
 - Przejdź do makra, 8-68
 - Przejście kroku (po), 8-46
 - Przejście kroku (przed), 8-46
 - Przełącz kompilację, 6-40
 - Przeprowadzanie porównania projektów, 6-42
 - Przesuwanie elementów w graficznym edytorze schematów funkcji, 8-61
 - Przesuwanie w CFC, 8-61
 - Przesyłanie list parametrów, 9-84
 - Przesyłanie sumy kontrolnej, 9-5
 - Przetwarzane w SFC, 8-52
 - Przetwarzanie krokowe
 - SFC, 8-52
 - Przetwarzanie krokowe, 8-23, 8-31
 - Przetwornik, 13-60
 - Przypomnij o proj. bootowalnym przed zakończeniem, 6-5
 - Przyporządkowanie, 2-12
 - Przyporządkowanie parametrów przy wywołaniu programu, 2-6
 - Przyporządkowanie urządzenia slave DP do grupy, 9-40
 - Przyporządkowanie w FBD, 8-34
 - Przywołaj, 6-53
 - Przywołaj wszystko, 6-56
 - Przywoływanie, 6-20
 - Przywracanie, 6-69
 - Przywróć, 6-81
 - Pulsator, 13-57
 - Punkt wstrzymania, 1-1, 2-26, 6-82, 8-23, 8-25, 8-31
 - PUTBIT, 13-61

Q

Query, 13-83

R

R_TRIG, 13-54

Ramię przyporządkowujące, 8-35

RAMP_INT, 13-69

RAMP_REAL, 13-69

REAL, 13-38

Referencje, 13-44

Regulator P, 13-64

Regulator PI, 13-66

Rejestrowanie, 6-21

Rejestrowanie wielokrotne, 6-56

REPEAT, 2-13, 2-16

REPLACE, 13-51

Reset źródła, 6-81

Resetuj węzeł, 9-44

RETAIN, 8-6

RETURN, 2-13, 2-14

RIGHT, 13-50

Rodzaj czcionki, 6-7

ROL, 13-7

ROR, 13-8

Rotacja, 13-7

Rozciąganie, 9-73

Rozmieść symbole, 6-95

Rozmieszczanie okien, 6-95

Rozszerz makro, 8-68

Rozwiń węzły, 6-61

RS, 13-53

RTC, 13-59
Run (linia poleceń), 13-79

S

S5 Import, 13-89
Schemat funkcji, 2-23, 8-32
Schemat styków
Tryb online, 8-45
Schemat styków, 2-24, 8-39
Schemat styków (LD) jako schemat funkcji (FBD), 2-25
Schowek, 6-69
Ścieżki biblioteki, 9-20
Ścieżki instancji, 9-9
SEL, 13-9
SEMA, 13-53
Separator ekranu, 6-2
Serwer ENI, 10-1
Serwer Gateway, 6-88, 6-91
Setreadonly, 13-85
SFC
Akcja wyjściowa, 8-48
Akcje wejściowe, 8-48
Atrybuty kroków, 8-50
Gałąź alternatywna, 8-47
Gałąź równoległa, 8-47, 8-48
Kolejka do przetwarzania, 8-52
Krok IEC, 8-52
Opcje w SFC, 8-51
Powiązanie akcji, 8-52
Powiększ akcję/przejście, 8-49
Przegląd czasów, 8-50
Przejście-skok, 8-48
skok, 8-47
Usuwanie akcji i przejścia, 8-49
Usuwanie kroku i przejścia, 8-47
Wstaw przejście kroku, 8-46
Zaznacz bloki, 8-46
Znacznik, 8-48
Znacznik skoku, 8-49
SFCCurrentStep, 2-22
SFCEnableLimit, 2-21
SFCErrror, 2-22
SFCErrrorAnalyzationTable, 2-22
SFCErrrorPOU, 2-22
SFCErrrorStep, 2-22
SFCInIt, 2-22
SFCPause, 2-22
SFCQuitError, 2-22
SFCReset, 2-22
SFCTip, 2-22
SFCTipMode, 2-22
SFCTrans, 2-22
Shift, 13-6
SHL, 13-6
Show... (linia poleceń), 13-79
SHR, 13-7
Siatka współrzędnych, 9-72
Sieć, 8-29, 8-32
Sieć (przed), 8-40
Sieć (za), 8-40
Sieć w FBD, 2-23
Sieć w schemacie styków, 2-24
Sieć w SFC, 2-17
Siemens, importowanie, 6-42, 13-89
SIN, 13-23
SINT, 13-38
Sinus, 13-23

SIZEOF, 13-4
Skalowanie Y, 9-72
Skanowanie bieżącego urządzenia, 9-56, 9-57
Skanowanie urządzenia, 9-57
Skanuj konfigurację modułu, 9-57
Składnia dostępu w przypadku zmiennych, 13-32
Składnik struktury, 8-14
Skok przejścia, 8-48
Skok w AS, 2-23
Skok w CFC, 8-57
Skok w FBD, 8-34
Skok w LD, 8-44
Skok w SFC, 8-47
Skróty za pomocą narzędzi, 9-90
Slajd Master, 6-65
Slave DP
Parametry DP, 9-36
Parametry modułu, 9-40
Parametry użytkownika, 9-39
Wejścia/wyjścia, 9-37
Slave DP parametry podstawowe, 9-36
Slave DP w trybie slave, 9-40
Słowa kluczowe, 8-7
Softmotion, 13-106
Sortowanie w edytorze tabeli, 8-12
Sortuj wg adresów, 8-12
Sortuj wg nazwy, 8-12
Sortuj wg typu, 8-12
Sourcecodedownload, 13-80
Sprawdź, 6-49
Sprawdź skrzyżowania adresów, 9-27
Sprawdzanie automatyczne, 6-14
Sprawdzanie ID dostawcy dla urządzenia DeviceNet
Slave, 9-53
Sprawdzanie kodu produktu dla urządzenia DeviceNet
Slave, 9-53
Sprawdzanie typu urządzenia dla urządzenia DeviceNet
Slave, 9-53
Sprawdzanie wersji produktu dla urządzenia DeviceNet
Slave, 9-53
Sprzężenie zwrotne w CFC, 8-69
SQRT, 13-22
SR, 13-52
ST, 2-11, 8-27
Stałe
BOOL, 13-29
DATE, 13-29
Globalne
REAL \ LREAL, 13-30
Globalne, 8-7
Stałe liczbowe, 13-30
STRING, 13-30
TIME, 13-29
TIME_OF_DAY, 13-29
Typed Literals, 8-7
Typowane, 8-7, 13-31
Stałe, 8-7
Stałe BYTE, 13-30
Stałe DATE, 13-29
Stałe DATE_AND_TIME, 13-30
Stałe DINT, 13-30
Stałe DWORD, 13-30
Stałe globalne, 8-7, 9-7
Stałe INT, 13-30
Stałe liczbowe, 13-30
Stałe LREAL, 13-30
Stałe REAL, 13-30
Stałe SINT, 13-30

- Stałe STRING, 13-30
 - Stałe TIME, 13-29
 - Stałe TIME_OF_DAY, 13-29
 - Stałe typowane, 13-31
 - Stałe typu BOOL, 13-29
 - Stałe UDINT, 13-30
 - Stałe UINT, 13-30
 - Stałe USINT, 13-30
 - Stałe WORD, 13-30
 - Standard.lib, 9-19, 13-49
 - Start, 6-81
 - STATISTICS_INT, 13-63
 - STATISTICS_REAL, 13-63
 - Status sterownika, 6-94
 - Statystyka, 6-47
 - Sterownik, status, 6-94
 - Stop, 6-81
 - STRING, 13-38
 - StrongARM, 13-99
 - STRUCT, 13-43
 - Struktury, 2-1, 13-43
 - Strzałka, 6-33
 - Styk, 2-24, 8-41
 - Styk (zanegowany), 8-41
 - Styk równoległy, 8-41
 - Styk równoległy (zanegowany), 8-41
 - SUB, 13-2
 - Suchen
 - Bibliothek, 9-20
 - Suma kontrolna, 6-94
 - Symbol zastępczy w widoku strony, 6-32
 - Symbole w Organizерze obiektów, 6-51
 - Symulacja, 2-27, 6-75, 6-88, 13-80
 - Sync.COB-Id, 9-42
 - Syntaxcoloring, 8-4, 8-8
 - SysLibAlarmTrend.lib, 13-108
 - SysTaskInfo.lib, 9-62
 - System alarmów, 9-11
 - System docelowy, 9-85
 - System docelowy 8051, 13-102
 - System docelowy Hitachi SH, 13-101
 - System docelowy Infineon C16x, 13-98
 - System docelowy kompatybilny z Intel 386, 13-96
 - System docelowy MIPS III ISA, 13-100
 - System docelowy Motorola 68K, 13-97
 - System docelowy Power PC, 13-99
 - System docelowy StrongARM, 13-99
 - System docelowy TriCore, 13-102
 - System Gateway, 6-89
 - SysTime.lib, 9-62
 - Szablon, 6-24
 - Szablon dokumentacji
 - utwórz, 9-9
 - wybierz, 9-10
 - Szablon dokumentacji, 9-10
 - Szablon dokumentu, 9-9
 - Szablon obiektów, 6-61, 6-62
 - Szablon obiektu, 6-61, 6-62
 - Szablon pliku eds, 9-48
 - Szablon projektu, 6-24
 - Szablon wydruku, 8-1
 - Szerokość tabulatora, 6-7
 - Szerokość/wysokość widoku, 13-108
 - Szkielet, 8-1
 - Szyfrowanie, 6-15, 6-26, 6-51
 - Szyfrowanie biblioteki, 6-27
- ## T
- Tabela deklaracji, 8-11
 - Tabela symboli w formacie XML, 6-17
 - Tablica
 - Dostęp, 13-41
 - Inicjalizacja, 13-40
 - Tablice w menedżerze parametrów, 9-79
 - TAN, 13-24
 - Tangens, 13-24
 - Tekst komunikatu, 9-14
 - Tekst pomocy, 9-88
 - Tekst w wizualizacji, 6-36
 - Tekst zhierarchizowany, 2-11, 8-27
 - Telegram alarmowy, 9-44
 - THEN, 2-14
 - TIME, 13-39
 - TIME_OF_DAY, 13-39
 - Timer, 13-57
 - Timer (TON) w LD, 8-44
 - TO, 2-15
 - TOD, 13-39
 - TOF, 13-58
 - TON, 13-58
 - Tooltip for identifiers, 8-2
 - TP, 13-57
 - Transfer potwierdzony, 9-5
 - Transmisja cykliczna, 9-6
 - Transmisja sterowana zdarzeniem, 9-6
 - Transmisja w przypadku zmiany, 9-6
 - Tranzycja, 2-19
 - TREND_TASK, 13-109
 - TriCore Zielsystem, 13-102
 - TRUNC, 13-21
 - Tryb bezpieczeństwa, 6-9
 - Tryb demo, 12-1
 - Tryb online
 - CFC, 8-69
 - Edytor deklaracji, 8-12
 - Edytor schematu styków, 8-45
 - Edytor sieci, 8-31
 - Edytor tekstowy, 8-23
 - FBD, 8-38
 - konfiguracja zadań, 9-62
 - Lista instrukcji, 8-27
 - Menedżer podglądu zmiennych i receptur, 9-66
 - SFC, 8-52
 - Tryb online, 6-75
 - Tryb online z zabezpieczeniami, 6-8
 - Tryb porównywania, 6-42
 - Tryb skróconego wprowadzania, 8-9
 - Tryb synchroniczny, 9-40
 - Tryb wprowadzania, 6-3, 8-22
 - Tryb wstrzymania, 9-40
 - Tryb z wieloma użytkownikami, 10-1
 - Tryb zastępowania, 6-3, 8-22
 - TSP, 9-85
 - Tworzenie binarnych informacji o symbolu (.sdb), 6-22
 - Tworzenie informacji o symbolu ASCII (.sym), **6-22**
 - Tworzenie kopii zapasowej, 6-4
 - Tworzenie listy zmiennych sieciowych, 9-4
 - Tworzenie pliku binarnego, 6-13
 - Tworzenie projektu bootowalnego (.sym), 6-22
 - Tworzenie składników pól, 6-18
 - Tworzenie składników struktury, 6-18
 - Tworzenie symboli, 8-14
 - Tworzenie wpisów symboli, 6-17
 - Tworzenie wpisów zbiorowych, 6-17

- Tworzenie zmiennych obiektu, 6-17
- Tylko na żądanie, 6-16
- Typ danych
 - Deklaracja, 13-119
 - Enumeracja, 13-43
 - String, 13-38
- Typ wyliczania, 13-43
- TYPE, 13-43, 13-44
- Typed Literals, 8-7, 13-31
- Typy, 8-8
- Typy alarmów, 9-16
- Typy danych
 - Całkowite, 13-38
 - Referencje, 13-44
 - Struktury, 13-43
 - Tablica, 13-40
 - Typ wyliczania, 13-43
 - Typy danych czasu, 13-39
 - Typy podzakresów, 13-44
 - Wskaźnik
 - Real \ LReal, 13-38
- Wskaźnik, 13-41**
- Typy danych, 13-38
- Typy danych czasu, 13-39
- Typy podzakresów, 13-44

U

- UCMM, 9-53
- UDINT, 13-38
- UINT, 13-38
- UNPACK, 13-61
- UNTIL, 2-16
- Uproszczona obsługa wprowadzania danych, 13-109
- Uruchom śledzenie, 9-71
- Uruchomienie programu, 13-80
- Urządzenie aktywne w konfiguracji, 9-52
- Urządzenie opcjonalne, 9-44
- Userlevel (linia poleceń), 13-79
- USINT, 13-38
- Ustal, 6-53
- Ustal wielokrotnie, 6-55
- Ustalanie debugowania zadania, 9-64
- Ustaw/Przywróć w CFC, 8-58
- Ustaw/Przywróć w LD, 8-45
- Ustawianie katalogów, 13-83
- Ustawianie wyjścia, 8-45
- Ustawienia CAN, 9-5
- Ustawienia CAN urządzenia CanDevice, 9-49
- Ustawienia docelowe, 13-95
- Ustawienia Ekspert dla urządzenia DeviceNet Slave, 9-53
- Ustawienia komunikacji
 - Sprawdzenie przed zalogowaniem, 6-8
- Ustawienia podstawowe, 9-48
- Ustawienia rozszerzone dla urządzenia DeviceNet Slave, 9-53
- Ustawienia systemu docelowego
 - Domyślnie, 13-95
 - Funkcje sieciowe, 13-95
 - Ogólne, 13-95
 - Okno dialogowe, 9-86
 - Pakiet Target Support Package, 9-85
 - Plik docelowy, 9-85
 - Plik tnf, 9-85
 - Podział pamięci, 13-95
- Ustawienia systemu docelowego, 9-85
- Ustawienia UDP, 9-4

- Ustawienie domyślne, 13-95
- Usuń, 6-70
- Usuń akcję/przejsięcie, 8-49
- Usuń obiekt, 6-61
- Usuń znacznik skoku, 8-49
- Usuwanie akcji, 8-49
- Usuwanie kroku i przejścia w SFC, 8-47
- Usuwanie połączeń w CFC, 8-62
- Usuwanie w FBD, 8-37
- Util.lib, 13-60
- Utwórz plik kompilacji, 6-35
- Utwórz projekt bootowalny, 6-94
- Użycie kodu, 6-33
- Użycie pamięci, 6-33
- Użycie pamięci kodu, 6-33

V

- VAR, 8-5, 8-9
- VAR PERSISTENT, 9-7
- VAR RETAIN, 9-7
- VAR_CONFIG, 8-17, 9-2, 9-7
- VAR_CONSTANT, 8-7, 9-7
- VAR_EXTERNAL, 8-7
- VAR_GLOBAL, 8-9, 9-2, 9-6
- VAR_IN_OUT, 8-5, 8-9
- VAR_INPUT, 8-4, 8-9
- VAR_INPUT CONSTANT w CFC, 8-59
- VAR_OUTPUT, 8-5, 8-9
- VARIANCE, 13-63
- Visudownload (linia poleceń), 13-79

W

- Wartość bezwzględna, 13-21
- Wartości bitów
 - Wyświetlanie, 6-7
- Wartości bitów, 6-7
- Wartości śledzenia w pliku ASCII, 9-74
- Warunek tranzycji, 2-19, 8-49
- Watchlist, 13-83
- Wczytaj, 6-80
- Wczytaj informacje o pobieraniu, 6-34
- Wczytaj listę kontrolną, 9-66
- Wczytaj opis modułu, 9-29
- Wczytaj śledzenie, 9-71
- Wczytaj śledzenie automatycznie, 9-71
- Wczytaj status modułu, 9-57
- Wczytaj wartości, 9-74
- Wczytywanie
 - Automatyczne, 6-5
- Wczytywanie automatyczne, 6-5
- Wczytywanie i zapisywanie, 6-4
- Wczytywanie projektu ze sterownika, 13-79
- Wczytywanie z pliku, 9-75
- Wczytywanie zapisu śledzenia, 9-74
- Wczytywanie ze sterownika, 9-76
- Wejścia/wyjścia, 8-64
- Wejścia/wyjścia urządzenia slave DP, 9-37
- Wejście do FBD, 8-35
- Wejście EN, 2-25, 8-42
- Wejście modułu w CFC, 8-57
- Wejście w CFC, 8-57
- Wejście w FBD, 8-35
- Wersja etykiet, 6-57
- Wersja kompilatora, 6-14
- Wersja projektu 1.5, 6-26
- Wersja projektu 2.0, 6-26

- Wersja projektu 2.1, 6-26
- Wersja projektu 2.2, 6-26
- Wewnętrznie podczas wczytywania, 6-16
- Wewnętrznie przy projekcie bootowalnym, 6-16
- WHILE, 2-16
- Wiadomość, 9-13**
- Wielkość nastawcza, 13-64, 13-65
- Wielkość regulowana, 13-64, 13-65
- Wielokrotny zapis na wyjściu, 6-14, 6-50
- Wizualizacja
 - Slajd Master, 6-65
- Wizualizacja, 2-9, 6-2
- Wizualizacja, 13-108
- Wizualizacja elementu docelowego
 - Właściwości obiektu, 6-65
- Wizualizacja elementu docelowego, 13-110
- Wizualizacja sieciowa
 - Właściwości obiektu, 6-65
- Wizualizacja sieciowa, 13-109
- Wklej
 - w FBD, 8-37
- Wklej, 6-70
- Wklej nad w LD, 8-44
- Wklej po, 8-49
- Wklej pod w LD, 8-44
- Wklej przy modułach w LD, 8-43
- Wklej za w LD, 8-44
- Włącz generowanie Heartbeat, 9-44
- Włącz zużycie Heartbeat, 9-44
- Włączanie/wyłączanie zadania, 9-64
- Właściwości
 - Biblioteka, 9-21
- Właściwości, 9-21
- Właściwości grupy master DP, 9-34
- Właściwości obiektu, 6-64
- Właściwości PDO, 9-46
- Właściwości w CFC, 8-59
- WORD, 13-38
- Wpisywanie poleceń, 9-87, 9-89
- Wpisz polecenie standardowe, 9-88
- Wprowadź klucz, 6-27
- Wprowadzanie zmiennych, 8-2
- Wskaźnik
 - Monitoring, 2-26
 - Sprawdzanie adresu, 13-42
 - Sprawdzanie alokacji, 13-42
- Wskaźnik, 13-41**
- Wskazówka podczas wczytywania, 6-16
- Wstaw
 - Argument, 8-22
 - Argument w edytorach tekstowych, 8-22
 - Blok funkcji, 8-22
 - Blok funkcji w edytorach tekstowych, 8-22
 - Blok funkcji w LD, 8-42
 - Cewka, 8-42
 - Cewka Reset, 8-42
 - Dodaj akcję wejściową, 8-48
 - Dodaj akcję wyjściową, 8-48
 - Dodaj bibliotekę, 9-20
 - Dodaj gałąź równoległą (z prawej), 8-48
 - Dodaj wywołanie programu, 9-61
 - Dodaj zadanie, 9-59
 - Dodaj znacznik do gałęzi równoległej, 8-48
 - Funkcja, 8-22
 - Funkcja w edytorach tekstowych, 8-22
 - Gałąź alternatywna (w prawo), 8-47
 - Gałąź alternatywna (z lewej), 8-47
 - Gałąź równoległa (z lewej), 8-47
 - Gałąź równoległa (z prawej), 8-47
 - Komentarz, 8-29, 8-30
 - Komentarz w CFC, 8-57
 - Mapa bitowa w widoku strony, 6-32
 - Moduł w CFC, 8-56
 - Moduł w FBD, 8-34
 - Moduł z EN w LD, 8-43
 - Nowa deklaracja, 8-12
 - Nowa lista kontrolna, 9-66
 - operator w edytorach tekstowych, 8-22
 - Pin wejścia, 8-58
 - Pin wyjścia, 8-58
 - Powrót w CFC, 8-57
 - Powrót w FBD, 8-34
 - Powrót w LD, 8-44
 - Przejście kroku (po), 8-46
 - przejście kroku (przed), 8-46
 - Przejście-skok, 8-48
 - Przyporządkowanie w FBD, 8-34
 - Sieć (przed), 8-31
 - Sieć (za), 8-31, 8-40
 - Skok, 8-34
 - Skok w CFC, 8-57
 - Skok w LD, 8-44
 - Skok w SFC, 8-47
 - Słowa kluczowe deklaracji, 8-8
 - Styk (zanegowany) w LD, 8-41
 - Styk równoległy (zanegowany) w LD, 8-41
 - Styk równoległy w LD, 8-41
 - Styk w LD, 8-41
 - Symbol zastępczy w widoku strony, 6-32
 - Timer (TON) w LD, 8-44
 - Typy, 8-8
 - W SFC, 8-46
 - Wejście, 8-35
 - Wejście modułu w CFC, 8-57
 - Wejście w CFC, 8-57
 - Wklej przy modułach, 8-42
 - Wklej przy modułach w LD, 8-43
 - Wszystkie ścieżki instancji, 9-9
 - Wyjście, 8-35
 - Wyjście w CFC, 8-57
 - Wykrywanie narastającego zbocza w LD, 8-43
 - Wykrywanie opadającego zbocza w LD, 8-43
 - Znacznik w CFC, 8-57
 - Znacznik w CFC, 8-57
 - Wstaw bibliotekę, 9-20, 13-85
 - Wstaw obiekt, 6-61, 6-62
 - Wstaw wspólne obiekty, 6-57
 - Wstawianie nowej deklaracji, 8-12
 - Wstawianie sieci, 8-31
 - Wszystkie poziomy makr wstecz, 8-68
 - Wybieranie pliku eksportu, 9-27
 - Wybór obiektu, 6-64
 - Wybór systemu docelowego, 13-86
 - Wycinanie w FBD, 8-37
 - Wyczyść wszystko, 6-34
 - Wydawanie komunikatów w pliku, 13-79
 - Wydruk tekstowy, 9-14
 - Wyjście Ustaw/Przywróć w FBD, 8-37
 - Wyjście w CFC, 8-57
 - Wyjście w FBD, 8-35
 - Wyklucz obiekty, 6-33
 - Wykluczanie obiektów z kompilacji, 6-14
 - Wykluczanie z kompilacji, 6-14
 - Wykluczenie z kompilacji, 6-33
 - Wykrywanie narastającego zbocza w LD, 8-43
 - Wykrywanie opadającego zbocza w LD, 8-43

- Wyłącz tworzenie zadań, 13-110
- Wyloguj, 6-80
- Wymuszanie, 6-85, 8-12
- Wymuszanie wartości
 - Menedżer podglądu zmiennych i receptur, 9-68
- Wymuszanie wartości, 6-85, 8-12
- Wyrażenie, 2-12
- Wyrejestrowanie wielokrotne, 6-56
- Wyrejestrowywanie, 6-21
- Wyrejestruj, 6-54
- Wyślij plik symboli, 13-106
- Wyświetl drzewo wywołań, 6-68
- Wyświetl komunikaty diagnostyczne, 9-57
- Wyświetl kursor, 9-72
- Wyświetl listę odsyłaczy, 6-67
- Wyświetlanie listy składników, 6-7, 8-2
- Wyświetlanie MDI, 6-9
- Wyświetlanie okna, 13-79
- Wyświetlanie wyniku porównania, 6-44
- Wytnij, 6-69
- Wywołanie bloków funkcji w języku ST, 2-13
- Wywołanie bloku funkcji, 2-3, 2-12, 2-13
- Wywołanie funkcji, 2-1, 13-34
- Wywołanie modułu, 8-23
- Wywołanie pliku poleceń, 13-80
- Wywołanie programu, 2-5

X

XOR, 13-5

Z

- Z argumentami, 2-6
- Zabezpieczenie dostępu, 6-15, 6-51
- Zabezpieczenie pliku, 6-15
- Zabezpieczenie przed zapisem, 6-51
- Żądaj informacji o projekcie, 6-5
- Zadanie
 - Maksymalna liczba, 9-59
- Zadanie cykliczne, 9-59
- Zadanie sterowane zdarzeniem, 9-60
- Zadanie sterowane zdarzeniem zewnętrznym, 9-60
- Zainicjuj wejścia, 13-106
- Zakończ, 6-33
- Zakończ plik, 13-81
- Zakresy indeksu, 13-107
- Zaloguj, 6-75, 6-78
- Zamień, 6-71, 13-83
- Zamknij plik, 13-81
- Zapis DCF, 9-43
- Zapis pliku w sterowniku, 6-94
- Zapis śledzenia
 - Format XML, 9-75, 9-76
 - Kanał wielokrotny, 9-72
 - Kompresuj, 9-74
 - Konfiguracja projektu, 9-76
 - Plik *.mon, 9-75, 9-76
 - Przedstawienie graficzne, 9-71
 - Rozciągnij, 9-73
 - Siatka współrzędnych, 9-72
 - Skalowanie Y, 9-72
 - Wczytaj śledzenie, 9-71
 - Wczytaj śledzenie automatycznie, 9-71
 - Wczytaj z pliku, 9-75
 - Wczytaj ze sterownika, 9-76
 - Wybór zmiennych, 9-70
 - Wyświetl kursor, 9-72

- Zapisz do pliku, 9-75, 9-76
- Zapisz śledzenie, 9-74
- Zapis śledzenia, 9-68
- Zapis śledzenia w pliku ASCII, 9-74
- Zapisywanie, 6-4
- Zapisywanie alarmów, 9-17
- Zapisywanie do pliku, 9-75, 9-76
- Zapisywanie i wczytywanie zapisu śledzenia, 9-74
- Zapisywanie plików konfiguracyjnych w projekcie, 9-27
- Zapisywanie wartości
 - Menedżer podglądu zmiennych i receptur, 9-68
- Zapisywanie wartości, 9-74
- Zapisywanie wartości śledzenia
 - Wartości w pliku ASCII, 9-74
- Zapisz, 6-83, 8-12
- Zapisz dane dostępu do bazy projektu, 6-5
- Zapisz historię, 9-90
- Zapisz instrukcję, 9-64, 9-65, 9-68
- Zapisz jako szablon, 6-62
- Zapisz plik, 13-81
- Zapisz pliki konfiguracyjne w projekcie, 9-27
- Zapisz wartości, 6-83, 8-12
- Zapobieganie pobieraniu plików wizualizacji, 13-109
- Zapytanie o tryb, 13-86
- Zarejestruj, 6-54
- Zarządzanie projektami, 6-24
- Zarządzanie wersjami, 10-1
- Zarządzanie zadaniami, 9-57
- Zasoby
 - Dziennik, 9-21
 - Konfiguracja zadań, 9-57
 - Listy zmiennych globalnych, 9-2
 - Menedżer bibliotek, 9-18
 - Obszar roboczy, 9-76
 - Ustawienia systemu docelowego, 9-85
 - Zapis śledzenia, 9-68
- Zasoby, 6-2, 9-1
- Zastąp globalnie, 6-49
- Zastępowanie stałych, 6-13
- Zastosuj jako konfigurację projektu, 9-76
- Zastosuj kroki IEC, 8-52
- Zastosuj pojedynczą zmianę, 6-46
- Zastosuj prawa dostępu, 6-46
- Zastosuj VIS_INPUT_TASK, 13-110
- Zastosuj właściwości, 6-46
- Zastosuj zmianę, 6-46
- Zaszyfrowana biblioteka wewnętrzna, 6-26
- Zaszyfrowana biblioteka zewnętrzna, 6-26
- Zaszyfrowany projekt CoDeSys, 6-26
- Zatrzymaj śledzenie, 9-71
- Zatrzymanie programu, 13-80
- Zaznaczanie elementów, 9-25
- Zaznaczanie elementów w CFC, 8-61
- Zaznaczanie w CFC, 8-61
- Zaznaczanie w edytorach graficznych, 6-7
- Zdarzenia systemowe w konfiguracji zadań, 9-61
- Zdarzenia w zadaniach, 9-60
- Zdarzenie, 9-60
- Zdarzenie alarmowe, 9-11, 9-13
- Zegar czasu przebiegu (runtime clock), 13-59
- Zewnętrzne konfiguracje śledzenia
 - Wczytaj z pliku, 9-75
 - Wczytaj ze sterownika, 9-76
 - Zapisz do pliku, 9-75, 9-76
 - Zastosuj jako konfigurację projektu, 9-76
- Zewnętrzne konfiguracje śledzenia, 9-75
- Zimny reset, 6-81
- Zimny start, 6-81

- Zmiana celu, 9-28
- Zmiana języka, 6-9, 6-36
- Zmiana online, 6-33, 6-34, 6-75, 6-78, 6-79, 6-80, 13-33, 13-105
- Zmiana referencji, 13-15, **13-42**
- Zmiana systemu docelowego, 9-28, 13-79
- Zmień nazwę, 6-63
- Zmianie połączeń w CFC, 8-62
- Zmianie wartości (online), 2-26
- Zmienna dezaktywująca, 9-16, 9-18
- Zmienna monitoringu, 8-12, 8-38
- Zmienna retain
 - W blokach funkcji, 2-3
 - W funkcjach, 2-2
- Zmienna retain, 2-2
- Zmienna retain, 2-3
- Zmienna retain, 6-79
- Zmienna retain, 8-6
- Zmienna retain, 9-7
- Zmienna systemowa, 13-32
- Zmienna trwała, 8-6
- Zmienna wyzwalająca, 9-70
- Zmienne
 - Składnia dostępu, 13-32
- Zmienne, 13-31
- Zmienne adresowane bitowo, 8-23, 8-31
- Zmienne bezpośrednie, 13-32
- Zmienne globalne
 - Obiekty, 9-2
 - Zmienne remanentne, 9-7
 - Zmienne sieciowe, 9-6, 9-7
 - Zmienne trwałe, 9-7
 - Zmienne zapisu (retain), 9-7
- Zmienne globalne, 9-3
- Zmienne globalne remanentne, 9-7
- Zmienne globalne sieciowe, 9-6
- Zmienne globalne trwałe, 9-7
- Zmienne lokalne, 8-5
- Zmienne remanentne, 6-81, 8-6
- Zmienne sieciowe, 9-3, 9-6, 13-107
- Zmienne trwałe, 9-7
- Zmienne wejściowe, 8-4
- Zmienne wejściowe/wyjściowe, 8-5
- Zmienne wyjściowe, 8-5
- Zmienne zapisu śledzenia, 9-70
- Zmienne zewnętrzne, 8-7
- Znacznik, 6-51, 8-14, 13-34
- Znacznik połączenia w CFC, 8-62
- Znacznik w AS, 2-21
- Znaczniki AS, 2-21
- Znaczniki skoku, 8-29, 8-48
- Znajdź, 6-71
- Znajdź następny, 6-71
- Zwiń węzły, 6-61