

SIEMENS
Ingenuity for life



Podręcznik pierwsze kroki z SIMATIC S7-1200

Easybook

siemens.pl/s7-1200

SIEMENS

SIMATIC

S7-1200

Easy Book

Manual

Przedmowa

Charakterystyka zaawansowanego sterownika S7-1200

1

STEP 7 ułatwia pracę

2

Rozpoczęcie pracy

3

Prosty w obsłudze sterownik PLC

4

Konfiguracja sterownika

5

Koncepcja programowania

6

Komunikacja pomiędzy urządzeniami – standard PROFINET

7

Łatwy w użyciu regulator PID

8

Wbudowany serwer WWW ułatwiający komunikację z Internetem

9

Łatwe sterowanie napędami

10

Łatwe w użyciu narzędzia online

11

IO-Link

12

Dane techniczne

A

Wymiana CPU V3.0 na CPU V4.0

B


03/2014


A5E02486774-AF


Informacje prawne

System ostrzeżeń

Uwagi pojawiające się w tym podręczniku służą zachowaniu bezpieczeństwa ludzi i uniknięcia szkód wynikłych z niewłaściwego użytkowania urządzenia. Wskazówki te podzielono i oznaczono zależnie od stopnia zagrożenia w następujący sposób:

	ZAGROŻENIE
oznacza, że w przypadku nie zachowania odpowiednich środków bezpieczeństwa występuje zagrożenie śmiercią lub ciężkimi obrażeniami ciała.	

	OSTRZEŻENIE
Oznacza, że w przypadku nie zachowania odpowiednich środków bezpieczeństwa może wystąpić zagrożenie śmiercią lub ciężkimi obrażeniami ciała.	

	OSTROŻNIE
Oznacza, że w przypadku nie zachowania odpowiednich środków bezpieczeństwa mogą wystąpić lekkie obrażenia ciała.	

OSTROŻNIE	
Bez znaku ostrzegawczego oznacza, że w przypadku nie zachowania odpowiednich środków bezpieczeństwa mogą wystąpić szkody materialne.	

UWAGA	
Oznacza, że w przypadku nie zachowania odpowiednich środków bezpieczeństwa mogą wystąpić szkody materialne.	


W przypadku gdy występuje kilka niebezpieczeństw o różnym stopniu narażenia, to wszystkie są sygnalizowane jednym ostrzeżeniem odpowiadającym najwyższemu zagrożeniu. Ostrzeżenie o możliwości wystąpienia obrażeń ciała z odpowiednim symbolem, obejmuje również możliwość uszkodzenia mienia.

Kwalifikacje personelu

Urządzenia/system mogą być konfigurowane i używane wyłącznie na podstawie niniejszej dokumentacji. Do uruchamiania i obsługi urządzeń/systemu upoważniony jest tylko wykwalifikowany personel. Jako personel wykwalifikowany, w rozumieniu uwag zawartych w niniejszym opisie, rozumie się osoby, które mają uprawnienia do uruchamiania, dozoru, uziemiania i oznaczania urządzeń, systemów i obwodów zgodnie ze standardami i praktyką bezpieczeństwa.

Właściwe użycie wyrobów firmy Siemens

Prosimy o przestrzeganie następujących uwag:

	OSTRZEŻENIE
Wyroby firmy Siemens mogą być używane wyłącznie w aplikacjach opisanych w katalogu i dokumentacji technicznej. Jeżeli wykorzystuje się produkty i podzespoły pochodzące od innych producentów, to muszą być one rekomendowane lub zatwierdzone przez firmę Siemens. Dla zapewnienia bezpiecznej pracy i uniknięcia problemów niezbędne są odpowiednie: transport, przechowywanie, instalacja, montaż, uruchamianie, obsługa i konserwacja. Należy zapewnić dozwolone warunki zewnętrzne. Należy stosować się do informacji podanych w dokumentacji technicznej.	

Znaki zastrzeżenia

Wszystkie nazwy identyfikowane znakiem © są zarejestrowanymi znakami towarowymi Siemens AG. Inne oznaczenia występujące w niniejszym podręczniku mogą być znakami towarowymi, których wykorzystanie dla własnych celów przez osoby trzecie może naruszyć prawa właścicieli.

Zrzeczenie się odpowiedzialności

Treść niniejszej publikacji sprawdzona została pod kątem zgodności opisanego sprzętu i oprogramowania ze stanem faktycznym. Niemniej jednak nie można założyć braku jakichkolwiek nieprawidłowości. Wyklucza się wszelką odpowiedzialność i gwarancję całkowitej prawdziwości zawartych informacji. Treść podręcznika poddana jest okresowo uzupełnieniom i poprawkom. Wszelkie konieczne korekty wprowadza się w kolejnych wydaniach.

Wprowadzenie

Zapraszamy do zapoznania się z systemem S7-1200. System S7-1200 jest jednym z najnowszych opracowań firmy Siemens w rodzinie sterowników swobodnie programowalnych. SIMATIC S7-1200 jest modułowym systemem automatyki, charakteryzującym się niewielkimi wymiarami, przeznaczonym do stosowania w aplikacjach wymagających realizacji zarówno prostych jak i rozbudowanych algorytmów logicznych, wygodnej obsługi paneli operatorskich i interfejsów HMI, a także przystosowanym do współpracy w systemach sieciowych. Kompaktowa budowa, niska cena i jednocześnie wysoka wydajność powodują, że S7-1200 jest idealnym systemem dla wielu aplikacji.

System S7-1200 spełnia założenia nowatorskiej idei „totalnie zintegrowanej automatyki” (TIA – *Totally Integrated Automation*), w ramach której urządzenia z rodziny S7-1200 wraz z oprogramowaniem programistycznym TIA Portal zapewniają niespotykaną elastyczność i wygodę projektowania, pozwalającą rozwiązać wszelkie problemy aplikacyjne.

S7-1200 powoduje, że najtrudniejsze zadania stają się łatwe!

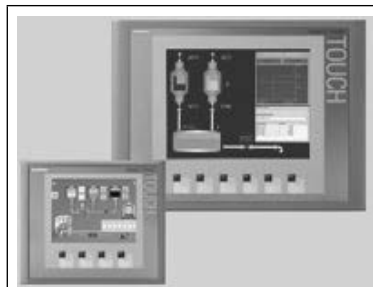
System SIMATIC S7-1200 należy do klasy rozwiązań kompaktowych, w jego skład wchodzi sterowniki SIMATIC S7-1200 oraz panele operatorskie SIMATIC HMI Basic. Obydwa rodzaje urządzeń są programowane za pomocą pakietu narzędziowego TIA Portal. Możliwość przygotowania programów dla całego systemu sterowania za pomocą jednego narzędzia radykalnie skraca czas ich przygotowywania i zmniejsza koszty. TIA Portal zawiera program narzędziowy STEP 7 do programowania sterownika S7-1200 oraz program narzędziowy WinCC do obsługi projektów dla paneli operatorskich Basic.



Sterowniki S7-1200 są wyposażone w:

- Interfejs PROFINET
- Szybkie I/O przystosowane do współpracy z systemami nadzoru ruchu, wbudowane wejścia analogowe minimalizują konieczność stosowania dodatkowych modułów rozszerzeń, 4 generatory impulsowe mogące generować przebiegi PWM (*Pulse Width Modulation*) i PTO (*Pulse Train Output*) (strona 64) i do 6 szybkich liczników.

Wbudowane interfejsy linii I/O zapewniają dostęp do 6...14 wejść i 4...10 wyjść.



Dodatkowe moduły wejść/wyjść (przełącznikowe, tranzystorowe, analogowe) rozszerzają dostępną liczbę punktów wej./wyj. Innowacyjne płytki sygnałowe montowane na przedniej ścianie sterownika pozwalają go elastycznie dopasować do wielu aplikacji (strona 18).

Panele operatorskie SIMATIC HMI Basic (strona 20) zaprojektowano specjalnie do współpracy ze sterownikami S7-1200.

Niniejszy podręcznik jest wprowadzeniem w obsługę S7-1200. Na kolejnych stronach przedstawiamy cechy i możliwości poszczególnych elementów tego systemu.

Dodatkowe informacje są dostępne w podręczniku użytkownika sterownika S7-1200 (S7-1200 System manual).

Dodatkowe informacje o certyfikatach UL, FM, oznakowaniu CE i C-Tick, a także innych standardach można znaleźć w specyfikacji technicznej (strona 277).

W podręczniku opisano następujące wyroby:

- Pakiety narzędziowe STEP 7 V13 SP1 Basic i Professional
- Sterownik S7-1200 CPU z firmware wersja V4.0

Dodatkowe dokumentacje i informacje

Z myślą o systemie S7-1200 i pakiecie oprogramowania STEP 7 firma Siemens przygotowała wiele różnorodnej dokumentacji i innych zasobów szkoleniowych i informacyjnych, dzięki którym możliwe jest dotarcie do informacji niezbędnych podczas realizacji projektów.

- Podręcznik systemowy S7-1200 zawiera informacje o sposobie działania sterownika, jego programowaniu, a także komplet specyfikacji urządzeń tworzących system. W niniejszym podręczniku uzupełniono informacje opublikowane w podręczniku systemowym o przegląd możliwości urządzeń z rodziny S7-1200. Obydwa podręczniki są dostępne w wersji cyfrowej (PDF). Wersje cyfrowe można bezpłatnie pobrać z działu wsparcia technicznego na stronie supportowej SIMATIC, są one dostarczane także z dokumentacjami na płytach dołączonych do każdego kupionego sterownika S7-1200.
- Zaawansowany system informacji online wbudowany w pakiet STEP 7 zapewnia natychmiastowy dostęp do wszelkich informacji i dokumentacji z opisami sposobu działania i funkcjonalności jednostek SIMATIC CPU.
- Internetowy interfejs My Documentation Manager zapewnia wygodny dostęp do dokumentacji w wersji cyfrowej (PDF) urządzeń i oprogramowania tworzących system SIMATIC. Za pomocą My Documentation Managera użytkownik może tworzyć własne wersje podręczników, w czym pomocny jest mechanizm drag-and-drop służący do wybierania interesujących tematów i składania ich w jeden, dedykowany dokument. Dostęp do My Documentation Managera jest możliwy w dziale mySupport dostępnym pod adresem <http://support.automation.siemens.com>.

- Internetowy dział wsparcia technicznego oferuje użytkownikom także dostęp do podcastów, działu FAQ oraz wielu innych dokumentów dotyczących rodziny S7-1200 i oprogramowania STEP 7. W podcastach udostępniono wiele krótkich filmów i prezentacji pokazujących obsługę i praktyczne zalety korzystania z pakietu STEP 7. Pod podanymi poniżej adresami dostępne są kolekcje podcastów dotyczących:
 - Pakietu STEP 7 Basic (<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-basic/Pages/Default.aspx>)
 - Pakietu STEP 7 Professional (<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-professional/Pages/Default.aspx>)
- Zachęcamy także do uczestnictwa w dyskusjach na naszym internetowym forum wsparcia technicznego, które jest dostępne pod adresem <https://www.automation.siemens.com/WW/forum/guests/Conferences.aspx?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid0=34612486>
 - Forum użytkowników S7-1200 jest dostępne pod adresem: <https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=258&Language=en&onlyInternet=False>
 - Forum użytkowników pakietu STEP 7 Basic jest dostępne pod adresem: <https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=265&Language=en&onlyInternet=False>

Serwis i wsparcie techniczne

W uzupełnieniu dokumentacji, firma Siemens oferuje także wsparcie techniczne przez Internet, dostępne poprzez stronę (<http://www.siemens.com/automation/>).

W sprawie wsparcia technicznego sugerujemy także bezpośredni kontakt z naszymi lokalnymi biurami sprzedaży i dystrybutorami, których przedstawiciele udzielą wszelkich porad i informacji dotyczących także szkoleń oraz sprzedaży urządzeń S7. Nasi dystrybutorzy i przedstawiciele handlowi są doskonale przeszkoleni i dysponują głęboką wiedzą na temat lokalnych rynków, ich charakterystyki i potrzeb, wymogów przemysłu, znają także cechy i możliwości urządzeń i oprogramowania SIMATIC, co pozwala im kompetentnie wesprzeć działania klientów, zapewniając szybkie i niezawodne rozwiązanie możliwych problemów.

Zapewnienie bezpieczeństwa

Firma Siemens oferuje produkty automatyki i napędy z przemysłowymi funkcjami bezpieczeństwa, które zapewniają bezpieczne działanie instalacji lub maszyn. Są to ważne elementy kompleksowej koncepcji bezpieczeństwa przemysłowego. Mając to na uwadze, nasze produkty są stale rozwijane. Dlatego zalecamy zasięganie na bieżąco informacji o aktualizacjach naszych produktów. Więcej infor-

Wprowadzenie	3
1. Charakterystyka zaawansowanego sterownika S7-1200	15
1.1. Wprowadzenie do sterownika PLC S7-1200	15
1.2. Możliwość rozbudowy CPU	18
1.3. Moduły S7-1200	20
1.4. Panele operatorskie Basic HMI.....	20
1.5. Wymiary montażowe i wymagane tolerancje pasowania	22
1.6. Nowe możliwości	26
2. STEP 7 ułatwia pracę	28
2.1. Łatwe wstawianie instrukcji do programu użytkownika	29
2.2. Łatwy dostęp do ulubionych instrukcji z paska narzędzi	29
2.3. Łatwe dodawanie wejść i wyjść do instrukcji w drabince lub bloku funkcyjnym.....	30
2.4. Rozwijane instrukcje.....	30
2.5. Łatwa zmiana trybu pracy CPU	30
2.6. Łatwa zmiana wyglądu i konfiguracji dla STEP 7.....	31
2.7. Ułatwienia dostępu dla projektu i globalnych bibliotek	31
2.8. Łatwy dobór wersji instrukcji	32
2.9. Łatwe przeciąganie elementów pomiędzy edytorami.....	33
2.10. Zmiana typu wywołania DB	34
2.11. Tymczasowe odłączenie urządzeń z sieci.....	34
2.12. Wirtualne „rozłączanie” modułów bez straty konfiguracji	36
3. Rozpoczęcie pracy	37
3.1. Tworzenie projektu	37
3.2. Tworzenie zmiennych (tagów) dla obszaru I/O CPU.....	38
3.3. Tworzenie prostej sieci w programie użytkownika.....	39
3.4. Użycie zmiennych PLC z tablicy zmiennych do adresowania instrukcji.....	41
3.5. Dodawanie pola instrukcji.....	42
3.6. Użycie instrukcji CALCULATE dla złożonych równań matematycznych	43

3.7.	Dodawanie urządzenia HMI do projektu	45
3.8.	Tworzenie połączenia pomiędzy CPU i urządzeniem HMI	46
3.9.	Tworzenie połączenia urządzeń HMI do współdzielenia zmiennych.....	46
3.10.	Tworzenie ekranu urządzenia HMI	47
3.11.	Wybór zmiennych PLC dla elementów HMI	48
4.	Prosty w obsłudze sterownik PLC	49
4.1.	Zadania wykonywane w każdym cyklu	49
4.2.	Tryby pracy CPU.....	50
4.3.	Wykonanie programu użytkownika	52
4.3.1.	Wykonywanie cyklu programu w trybie RUN	52
4.3.2.	Zastosowanie bloków organizacyjnych OB porządkuje program użytkownika	53
4.3.3.	Kolejkowanie i priorytety wykonania obsługi zdarzeń.....	53
4.4.	Obszary pamięci, adresowanie i typy danych.....	57
4.4.1.	Typy danych obsługiwane przez S7-1200	58
4.4.2.	Adresowanie obszarów pamięci	60
4.4.3.	Uzyskiwanie dostępu do „przedziału danych” o typie danych określonym tagiem	63
4.4.4.	Uzyskiwanie dostępu do tagu za pomocą nakładki typu AT	64
4.5.	Wyjścia impulsowe.....	65
5.	Konfiguracja sterownika.....	68
5.1.	Rozpoznawanie konfiguracji dla nieokreślonego CPU	69
5.2.	Dołączanie modułów	70
5.3.	Zmiana urządzenia.....	71
5.4.	Dodawanie modułów do systemu	71
5.5.	Konfiguracja parametrów pracy CPU i modułów	72
5.5.1.	Pamięć systemu i zegara jako standardowa funkcjonalność.....	74
5.6.	Konfiguracja adresu IP dla CPU.....	77
5.7.	Ochrona dostępu do CPU i kodu programu	78
5.7.1.	Zabezpieczenie „know-how”	81
5.7.2.	Zabezpieczenie przed kopiowaniem	82

6.	Koncepcja programowania	84
6.1.	Projektowanie programu użytkownika	84
6.1.1.	Użycie bloków OB do organizacji programu użytkownika.....	86
6.1.2.	FB i FC upraszczają programowanie zadań modułarnych	87
6.1.3.	Bloki danych (DB) ułatwiają przechowywanie danych programu.....	89
6.1.5.	Tworzenie bloków kodu do wielokrotnego wykorzystania	90
6.1.6.	Wywoływanie bloku kodu z innego bloku kodu	90
6.2.	Języki programowania	91
6.2.1.	Język drabinkowy (LAD)	91
6.2.2.	Język programowania Function Block Diagram (FBD).....	92
6.2.3.	Język SCL	92
6.2.4.	Edytor programów SCL.....	93
6.3.	Instrukcje programowania	95
6.3.1.	Podstawowe instrukcje	95
6.3.2.	Instrukcje porównania oraz move	97
6.3.3.	Instrukcje konwersji.....	98
6.3.4.	Operacje arytmetyczne z instrukcją Calculate	100
6.3.5.	Układy czasowe – timery.....	101
6.3.6.	Liczniki	106
6.3.7.	Instrukcja modulacji szerokości impulsu (PWM).....	108
6.4.	Tworzenie dzienników zdarzeń w plikach .csv.....	109
6.5.	Monitorowanie i testowanie programu.....	111
6.5.1.	Tablice monitorujące i tablice wymuszeń.....	111
6.5.2.	Cross-reference lista użycia zmiennych w projekcie.....	111
6.5.3.	Struktura wywołania do badania hierarchii połączeń	112
6.5.4.	Instrukcje diagnostyczne do monitorowania sprzętu.....	113
6.5.4.1.	Odczytywanie stanu diod LED na sterowniku	113
6.5.4.2.	Instrukcje odczytujące stan diagnostyczny urządzeń.....	114
6.6.	Szybkie liczniki (HSC).....	114
6.6.1.	Działanie szybkiego licznika.....	116
6.6.2.	Konfiguracja HSC.....	122
7.	Komunikacja pomiędzy urządzeniami – standard PROFINET	123
7.1.	Tworzenie połączenia sieciowego.....	123
7.2.	Opcje komunikacji	125

7.3.	Liczba asynchronicznych połączeń komunikacyjnych.....	126
7.4.	Instrukcje do obsługi PROFINET i PROFIBUS	127
7.5.	PROFINET	128
7.5.1.	System otwartej komunikacji użytkowników.....	128
7.5.1.1.	Tryb ad hoc.....	129
7.5.1.2.	Identyfikatory ID połączeń dla systemu otwartej komunikacji użytkowników	30
7.5.1.3.	Parametry połączenia PROFINET	133
7.5.2.	Konfigurowanie ścieżki połączenia	135
7.6.	PROFIBUS	138
7.6.1.	Usługi komunikacyjne modułów CM PROFIBUS.....	139
7.6.2.	Podręczniki modułów komunikacyjnych CM PROFIBUS.....	140
7.6.3.	Dodawanie modułów CM 1243-5 (DP master) oraz DP slave.....	140
7.6.4.	Przypisanie adresów PROFIBUS do modułu CM 1243-5 i urządzeń DP slave	141
7.7.	AS-i.....	143
7.7.1.	Dodanie modułu CM 1243-2 AS-i Master oraz AS-i Slave	143
7.7.2.	Nadawanie adresu AS-i urządzeniu AS-i Slave.....	144
7.8.	Komunikacja S7	146
7.8.1.	Instrukcje GET i PUT	146
7.8.2.	Tworzenie połączenia S7.....	147
7.8.3.	Przypisanie parametrów połączenia dla instrukcji GET/PUT.....	148
7.9.	GPRS	150
7.9.1.	Połączenie z siecią GSM.....	150
7.10.	Protokoły komunikacyjne PtP, USS oraz Modbus	156
7.10.1	Komunikacja PtP (<i>Point-to-Point</i>).....	156
7.10.2.	Korzystanie z szeregowych interfejsów komunikacyjnych	157
7.10.3.	Instrukcje PtP	158
7.10.4.	Instrukcje USS	159
7.10.5.	Instrukcje obsługujące protokół Modbus	160
8.	Łatwy w użyciu regulator PID.....	163
8.1.	Wstawianie instrukcji PID i obiektów technologicznych	165
8.2.	Instrukcja PID_Compact	167
8.3.	Parametry ErrorBit instrukcji PID_Compact.....	170

8.4.	Instrukcja PID_3Step.....	172
8.5.	Parametry ErrorBits instrukcji PID_3STEP.....	179
8.6.	Konfigurowanie regulatora PID	181
8.7.	Uruchomienie regulatora PID.....	184
9.	Wbudowany serwer WWW ułatwiający komunikację z Internetem.....	186
9.1.	Łatwość użycia standardowych stron Web	187
9.2.	Ograniczenia, które mogą mieć wpływ na korzystanie z serwera sieci Web	189
9.2.1.	Ograniczenia funkcji związane z wyłączeniem obsługi JavaScript w opcjach internetowych.....	190
9.2.2.	Funkcje ograniczone poprzez wyłączenie obsługi „ciasteczek”	191
9.3.	Uprozczone tworzenie stron definiowanych przez użytkownika.....	192
9.3.1.	Łatwe tworzenie dowolnych, zdefiniowanych przez użytkownika stron sieci Web.....	192
9.3.2.	Ograniczenia związane ze stronami definiowanymi przez użytkownika	193
9.3.3.	Konfiguracja zdefiniowanych przez użytkownika stron Web	194
9.3.4.	Używanie instrukcji WWW	195
10.	Łatwe sterowanie napędami.....	196
10.1.	Sterowanie fazowe.....	201
10.2.	Konfiguracja generatora impulsów.....	203
10.3.	Konfiguracja osi napędu	204
10.4.	Konfiguracja obiektu technologicznego TO_CommandTable_PTO	207
10.5.	Instrukcje sterowania ruchem <i>Motion Control</i>	211
10.5.1.	Przegląd instrukcji sterowania ruchem.....	211
10.5.2.	Instrukcja MC_Power (zwolnienie/zablokowanie osi).....	212
10.5.3.	Instrukcja MC_Reset (potwierdzanie błędów).....	215
10.5.4.	Instrukcja MC_Home (bazowanie osi).....	216
10.5.5.	Instrukcja MC_Halt (zatrzymanie osi).....	218
10.5.6.	Instrukcja MC_MoveAbsolute (ustawienie osi w pozycji bezwzględnej).....	220

10.5.7.	Instrukcja MC_MoveRelative (ustawienie osi względem pozycji początkowej).....	222
10.5.8.	Instrukcja MC_MoveVelocity	224
10.5.9.	Instrukcja MC_MoveJog.....	226
10.5.10.	MC_CommandTable (wykonywanie poleceń ruchu osi jako sekwencji ruchów).....	228
10.5.11.	Instrukcja MC_ChangeDynamic (zmiana ustawień dynamicznych osi).....	231
10.5.12.	Instrukcja MC_WriteParam (zapisywanie parametrów obiektu technologicznego).....	233
10.5.13.	Instrukcja MC_ReadParam (odczytywanie parametrów obiektu technologicznego).....	234
10.6.	Sterowanie ruchem przez sterownik S7-1200.....	236
10.6.1.	Wyjścia CPU wykorzystywane do sterowania ruchem	236
10.6.2.	Sprzętowe i programowe wyłączniki krańcowe do sterowania ruchem	238
10.6.3.	Homing.....	241
10.6.3.1.	Przejście osi do pozycji spoczynkowej	241
10.6.3.2.	Konfiguracja parametrów homingu	243
10.6.3.3.	Sekwencja homingu aktywnego.....	245
10.7.	Uruchomienie	247
11.	Łatwe w użyciu narzędzia online	253
11.1.	Tryb online i połączenie z CPU.....	253
11.2.	Interakcja z połączonym CPU	254
11.3.	Połączenie online w celu monitorowania wartości w CPU.....	254
11.4.	Wyświetlanie stanu programu użytkownika.....	255
11.5.	Tablice monitorujące do monitorowania CPU.....	256
11.6.	Użycie tablic wymuszeń.....	257
11.7.	Przechwytywanie wartości online bloków DB do skasowania wartości początkowych.....	260
11.8.	Kopiowanie elementów projektu	261
11.9.	Porównywanie CPU w trybach offline i online	262
11.10.	Wyświetlanie zdarzeń diagnostycznych.....	263
11.11.	Ustawianie adresu IP i czasu dnia	263
11.12.	Przywracanie ustawień fabrycznych	264

11.13.	Aktualizacja oprogramowania sprzętowego	265
11.14.	Załadowanie adresu IP do połączonego online CPU.....	266
11.15.	Użycie „nieznanego CPU” do automatycznej detekcji	267
11.16.	Ładowanie w trybie RUN.....	268
11.16.1.	Zmiana programu w trybie RUN.....	269
11.17.	Śledzenie i rejestrowanie zmiennych CPU wyzwalane zdarzeniowo	270
12.	IO-Link.....	272
12.1.	Przegląd technologii IO-Link	272
12.2.	Elementy systemu IO-Link.....	272
12.3.	Po włączeniu zasilania	272
12.4.	Protokół IO-Link.....	273
12.5.	Konfiguracja Fieldbus.....	273
12.6.	IO-Link w programie STEP 7	273
12.7.	SM 1278 4xIO-Link Master.....	274
A.	Dane techniczne.....	277
A.1.	Dane techniczne ogólne	277
A.2.	Specyfikacja CPU.....	286
A.3.	Moduły cyfrowych I/O.....	290
A.3.1.	Płytki sygnałowe SB 1221, SB 1222, oraz SB 1223 – cyfrowe wejścia/wyjścia (DI, DQ, oraz DI/DQ).....	290
A.3.2.	Moduł SM 1221 – cyfrowe wejścia (DI).....	293
A.3.3.	Moduł SM 1222 – cyfrowe wyjścia (DQ).....	294
A.3.4.	Moduł SM 1223 cyfrowe wejścia/wyjścia VDC (DI/DQ).....	295
A.3.5.	Moduł SM 1223 120/Wejścia 230 VAC/Wyjścia przekaźnikowe	297
A.4.	Specyfikacja cyfrowych wejść i wyjść.....	298
A.4.1.	Cyfrowe wyjścia 24 VDC (DI).....	298
A.4.2.	Cyfrowe wyjścia 120/230 VAC	299
A.4.3.	Cyfrowe wyjścia (DQ)	300
A.5.	Moduły analogowych I/O	302
A.5.1.	Płytki sygnałowe analogowych wejść (AI) – SB 1231 i analogowych wyjść (AQ) – SB 1232.....	302

A.5.2.	SM 1231 – analogowe wejścia (AI).....	303
A.5.3.	SM 1232 – analogowe wyjścia (AQ).....	303
A.5.4.	SM 1234 – analogowe wejścia/wyjścia (AI/AQ).....	303
A.5.5.	Schematy połączeń dla SM 1231 (AI), SM 1232 (AQ) oraz SM 1234 (AI/AQ).....	304
A.6.	Płytką bateryjna BB 1297.....	305
A.7.	Specyfikacja analogowych wejść/wyjść.....	306
A.7.1.	Specyfikacja analogowych wejść (CPU, SM oraz SB).....	306
A.7.2.	Napięciowe i prądowe zakresy pomiarowe dla analogowych wejść (AI).....	307
A.7.3.	Odpowiedź skokowa analogowych wejść (AI).....	308
A.7.4.	Czas próbkowania i czas odświeżania dla analogowych wyjść.....	309
A.7.5.	Specyfikacja analogowych wyjść (SM oraz SB).....	309
A.7.6.	Napięciowe i prądowe zakresy pomiarowe dla analogowych wyjść (AQ).....	310
A.8.	Moduły termopar i RTD.....	311
A.8.1.	Specyfikacja modułów SB 1231 RTD oraz SB 1231 TC.....	312
A.8.2.	Specyfikacja modułu SM 1231 RTD.....	314
A.8.3.	Specyfikacja modułu SM 1231 TC.....	315
A.8.4.	Specyfikacja analogowego wejścia dla RTD oraz TC (SM i SB).....	316
A.8.5.	Typy termopar.....	317
A.8.6.	Wybór filtra termopary i czasy odświeżania.....	317
A.8.7.	Tabela wyboru typu czujnika RTD.....	318
A.8.8.	Wybór filtra RTD i czasy odświeżania.....	319
A.9.	Interfejsy komunikacyjne.....	319
A.9.1.	PROFIBUS master/slave.....	320
A.9.1.1.	Moduł CM 1242-5 PROFIBUS slave.....	320
A.9.1.2.	Moduł CM 1243-5 PROFIBUS master.....	321
A.9.2.	Procesor komunikacyjny GPRS.....	322
A.9.2.1.	Specyfikacja techniczna CP 1242-7.....	323
A.9.3.	TeleService (TS).....	325
A.9.4.	Komunikacja poprzez RS485 i RS232.....	325
A.9.4.1.	Specyfikacja CB 1241 RS485.....	325
A.9.4.2.	Specyfikacja modułów CM 1241 RS422/RS485.....	328
A.9.4.3.	Specyfikacja CM 1241 RS232.....	329

A.10.	Moduły technologiczne	330
A.10.1.	SM 1278 4xIO-Link Master.....	330
A.10.1.1.	Specyfikacja modułu SM 1278 4xIO-Link Master.....	330
A.10.1.2.	Schematy połączeń modułu SM 1278 4xIO-Link Master.....	333
A.11.	Urządzenia towarzyszące.....	333
A.11.1.	Zasilacz PM1207	333
A.11.2.	Kompaktowy switch CSM 1277	334
A.11.3.	Moduł CM CANopen	334
B	Wymiana CPU V3.0 na CPU V4.0	336
B.1.	Wymiana CPU V3.0 na CPU V4.0	336

Charakterystyka zaawansowanego sterownika S7-1200

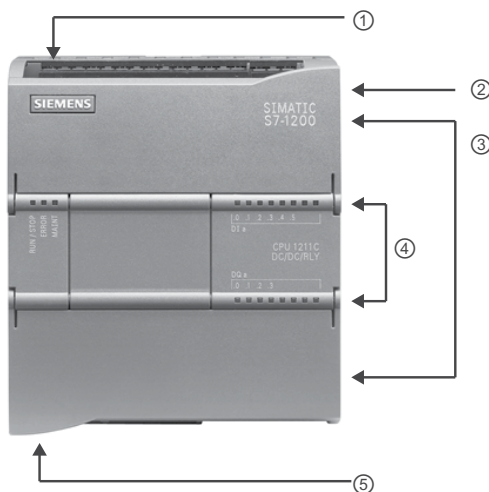
1

1.1. Wprowadzenie do sterownika PLC S7-1200

Sterownik S7-1200 charakteryzuje się dużą elastycznością i mocą obliczeniową potrzebną do sterowania wieloma różnymi urządzeniami w aplikacjach automatyki. Kompaktowa konstrukcja, elastyczna budowa, łatwa konfiguracja i rozbudowany zestaw instrukcji czynią S7-1200 idealnym rozwiązaniem dla szerokiej gamy aplikacji.

Jednostka centralna CPU sterownika zawiera mikroprocesor, wbudowany zasilacz, obwody wejściowe i wyjściowe, zintegrowany port PROFINET oraz szybkie wejścia/wyjścia do sterowania napędami. Sterownik wyposażono także w wejścia analogowe. Wszystko to czyni z S7-1200 sterownik o potężnych możliwościach. Po załadowaniu programu jednostka CPU będzie zawierać logikę do monitorowania i sterowania urządzeniami w Twojej aplikacji. CPU nadzoruje wejścia i ustawia wyjścia zgodnie z logiką stworzoną w programie użytkownika. Logika ta może zawierać logikę boole'owską, operacje czasowe, zliczanie, złożone operacje matematyczne oraz komunikację z innymi inteligentnymi urządzeniami.

CPU wyposażono w interfejs PROFINET do komunikacji w sieci PROFINET. Dostępne są dodatkowe moduły umożliwiające komunikację z wykorzystaniem innych interfejsów i protokołów jak: PROFIBUS, GPRS, RS485 czy też RS232.



- ① Złącze zasilające
- ② Gniazdo karty pamięci pod górną klapką
- ③ Rozpinane złącza na przewody (pod klapkami)
- ④ Sygnalizacyjne diody LED dla zintegrowanych WE/WY
- ⑤ Złącze PROFINET (w dolnej części CPU)

Kilka wbudowanych w sterownik funkcji zabezpieczeń pomaga chronić dostęp zarówno do procesora jak i programu sterującego:

- Każde CPU jest chronione hasłem (strona 78), które umożliwia konfigurację dostępu do funkcji CPU.
- Możesz użyć ochrony typu „know-how protection” (strona 81) do ochrony kodu w określonym bloku.
- Możesz użyć ochrony przed kopiowaniem 77 w celu powiązania twojego programu z konkretną kartą pamięci lub jednostką CPU.

Tabela 1.1. Porównanie modeli CPU

Cecha		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Wymiary fizyczne [mm]		90 × 100 × 75	90 × 100 × 75	110 × 100 × 75	130 × 100 × 75	150 × 100 × 75
Pamięć użytkownika	Robocza	30 kb	50 kb	75 kb	100 kb	125 kb
	Ładowania	1 Mb	1 Mb	4 Mb	4 Mb	4 Mb
	Nieulotna	10 kb	10 kb	10 kb	10 kb	10 kb
Lokalne porty I/O	Cyfrowe	6 wejść/ 4 wyjść	8 wejść/ 6 wyjść	14 wejść/ 10 wyjść	14 wejść/ 10 wyjść	14 wejść/ 10 wyjść
	Analogowe	2 wejścia	2 wejścia	2 wejścia	2 wejścia/ 2 wyjścia	2 wejścia/ 2 wyjścia
Rozmiar obrazu procesu	Wejścia (I)	1024 bitów	1024 bitów	1024 bitów	1024 bitów	1024 bitów
	Wyjścia (Q)	1024 bitów	1024 bitów	1024 bitów	1024 bitów	1024 bitów
Pamięć bitów (M)		4096 bitów	4096 bitów	8192 bitów	8192 bitów	8192 bitów
Moduł rozszerzeń sygnałów (SM)		Brak	2	8	8	8
Płytki sygnałowa (SB), płytki bateryjna (BB), płytki komunikacyjna (CB)		1	1	1	1	1
Moduł komunikacyjny (CM) (lewostronne rozszerzenie)		3	3	3	3	3
Szybkie liczniki	Ogółem	Do 6 konfigurowanych do użycia dowolnych wejść wbudowanych lub SB				
	1 MHz	–	–	–	–	–
	100/180 kHz	Ia.0 do Ia.5	Ia.0 do Ia.5	Ia.0 do Ia.5	Ia.0 do Ia.5	Ia.0 do Ia.5
	30/120 kHz	–	Ia.6 do Ia.7	Ia.6 do Ib.5	Ia.6 do Ib.5	Ia.6 do Ib.1
Wyjścia impulsowe ²	Ogółem	Do 4 konfigurowanych do użycia dowolnych wyjść wbudowanych lub SB				
	1 MHz	–	–	–	–	Qa.0 do Qa.3
	100 kHz	Qa.0 do Qa.3	Qa.0 do Qa.3	Qa.0 do Qa.3	Qa.0 do Qa.3	Qa.4 do Qb.1
	20 kHz	–	Qa.4 do Qa.5	Qa.4 do Qb.1	Qa.4 do Qb.1	–
Karta pamięci		Karta pamięci SIMATIC (opcjonalnie)				
Czas podtrzymania zegara czasu rzeczywistego		20 dni typowo/12 dni minimalnie przy 40°C (bezbosługowy superkondensator)				
PROFINET Port komunikacyjny Ethernet		1	1	1	2	2
Rzeczywisty czas wykonania instrukcji matematycznych		2,3 μs/instrukcja				
Szybkość wykonania operacji Boolowskiej		0,08 μs/instrukcja				

¹ Ma zastosowanie prędkość wolniejsza, gdy szybki licznik (HSC) jest skonfigurowany w kwadratowym trybie pracy.

² Dla modeli CPU z wyjściami przekaźnikowymi należy zainstalować płytkę sygnałów cyfrowych (SB), aby móc skorzystać z wyjść impulsowych.

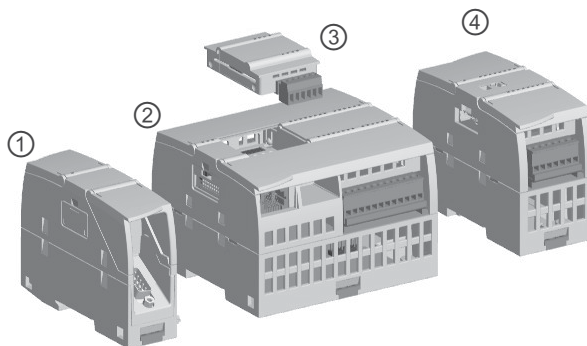
Poszczególne modele CPU charakteryzują się zróżnicowanymi właściwościami i możliwościami, co pomaga w stworzeniu efektywnych rozwiązań dla konkretnych aplikacji. Szczegółowe specyfikacje techniczne odnośnie danego CPU znajdują się w dalszej części przewodnika (strona 277).

Tabela 1.2. Bloki, timery i liczniki obsługiwane przez S7-1200

	Element	Opis
Blok	Typ	OB, FB, FC, DB
	Rozmiar	30 kb (CPU 1211C) 50 kb (CPU 1212C) 64 kb (CPU 1214C, CPU 1215C, oraz CPU 1217C)
	Liczba	do 1024 bloków razem (OB + FB + FC + DB)
	Głębokość zagnieżdżenia	16 od OB cyklu programu lub startowego; 6 od każdego OB przerwania
	Monitorowanie	Status bloków dwustanowych może być monitorowany w trybie ciągłym
Blok organizacyjny OB	Cykl programu	Wiele
	Startowy	Wiele
	Przerwania opóźnienia	4 (1 na zdarzenie)
	Przerwania cykliczne	4 (1 na zdarzenie)
	Przerwania sprzętowe	50 (1 na zdarzenie)
	Przerwania od błędu czasowego	1
	Przerwania od błędu diagnostycznego	1
	Wyciąganie lub wstawianie modułu	1
	Usterka listwy montażowej lub stacji	2
	Czas dnia	Wiele
	Status	1
	Aktualizacja	1
	Profil	1
Timery	Typ	IEC
	Ilość	Ograniczona tylko przez rozmiar pamięci
	Magazynowanie	Struktura w DB, 16 bitów na timer
Liczniki	Typ	IEC
	Ilość	Ograniczona tylko przez rozmiar pamięci
	Magazynowanie	Struktura w DB, rozmiar zależny od rodzaju zliczania SInt, USInt: 3 bity Int, UInt: 6 bitów DInt, UDInt: 12 bitów

1.2. Możliwość rozbudowy CPU

Rodzina sterowników S7-1200 zawiera wiele typów modułów rozszerzeń i płytek sygnałowych służących do rozszerzania możliwości CPU. Możliwe jest również instalowanie modułów komunikacyjnych obsługujących inne protokoły komunikacyjne. Szczegółowe informacje o poszczególnych modułach podano w dodatku „Dane techniczne” (strona 277).



- ① Moduł komunikacyjny (CM) lub procesor komunikacyjny (CP)
- ② CPU
- ③ Płytkę sygnałową (SB), płytkę komunikacyjną (CB), lub płytkę bateryjną (BB)
- ④ Moduł rozszerzeń (SM)

Tabela 1.3. Cyfrowe moduły rozszerzeń oraz płytki sygnałowe

Typ	Tylko wejścia	Tylko wyjścia	Kombinacja WE/WY
③ cyfrowe SB	4 × 24 V DC, 200 kHz 4 × 5 V DC, 200 kHz	4 × 24 V DC, 200 kHz 4 × 5 V DC, 200 kHz	2 × 24 V DC wejście/2 × 24 V DC wyjście 2 × 24 V DC wejście/2 × 24 V DC wyjście 200 kHz 2 × 5 V DC wejście/2 × 5 V DC wyjście, 200 kHz
④ cyfrowe SM	8 × 24 V DC	8 × 24 V DC 8 × wyjście przełącznikowe 8 × wyjście przełącznikowe (styk przelącz- ny)	8 × 24 V DC wejście/8 × 24 V DC wyjście 8 × 24 V DC wejście/8 × wyjście przełącznikowe 8 × 120/230 V AC wejście/8 × wyjście przełącznikowe
	16 × 24 V DC	16 × 24 V DC wyjście 16 × wyjście przełącznikowe	16 × 24 V DC wejście/16 × 24VDC wyjście 16 × 24 V DC wejście/16 × wyjście przełącznikowe

Tabela 1.4. Analogowe moduły rozszerzeń oraz płytki sygnałowe

Typ	Tylko wejścia	Tylko wyjścia	Kombinacja WE/WY
③ analogowe SB	1 × 12-bitowe analogowe wejście 1 × 16-bitowe RTD 1 × 16-bitowa termopara	1 × analogowe wyjście	–
④ analogowe SM	4 × analogowe wejście 4 × 16-bitowe analogowe wejście 8 × analogowe wejście Termopara: 4 × 16-bitowa TC 8 × 16-bitowa TC RTD: 4 × 16-bitowe RTD 8 × 16-bitowe RTD	2 × analogowe wyjście 4 × analogowe wyjście	4 × analogowe wejście/2 × analogowe wyjście

Tabela 1.5. Interfejsy komunikacyjne

Moduł	Typ	Opis
① Moduły komunikacyjne (CM)	RS232	Pełny duplex
	RS422/485	Pełny duplex (RS422) Pół-duplex (RS485)
	PROFIBUS Master	DPV1
	PROFIBUS Slave	DPV1
	AS-i Master (CM 1243-2)	Interfejs AS
① Procesor komunikacyjny (CP)	Łączność przez modem	GPRS
① Płytki komunikacyjna (CB)	RS485	Pół-duplex
① Zdalne połączenie (TeleService) ¹	TS Adapter IE Basic ¹	Połączenie z CPU
	TS Adapter GSM	GSM/GPRS
	TS Modem	Modem
	TS Adapter ISDN	ISDN
	TS Adapter RS232	RS232

¹ TS Adapter IE Basic pozwala na połączenie różnych interfejsów komunikacyjnych z portem PROFINET jednostki centralnej CPU używając kabla ethernetowego. Do TS Adapteru IE Basic można podłączyć do trzech różnych modułów TS Adaptera.

Tab. 1.6. Moduły technologiczne

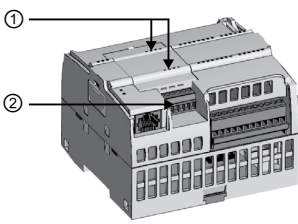
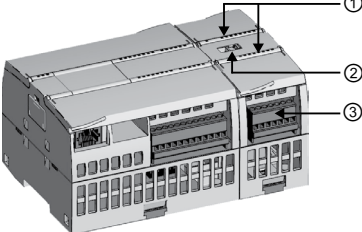
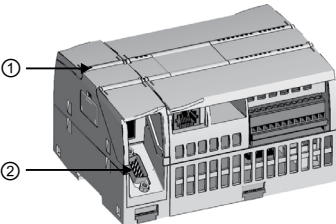
Moduł	Typ	Opis
④ IO Link	SM 1278 4xIO-Link Master	Moduł montowany w gnieździe rozszerzeń znajdującym się na płycie czołowej sterownika. Zapewnia długoterminowe podtrzymanie pracy zegara RTC

Tab. 1.7. Inne płytki

Moduł	Opis
③ Płytki bateryjna	Włączana do interfejsu płytki rozszerzeń od strony frontowej CPU. Zapewnia długotrwałe podtrzymanie pracy zegara czasu rzeczywistego

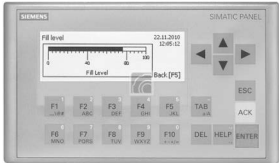
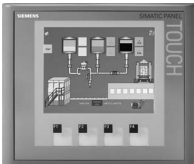
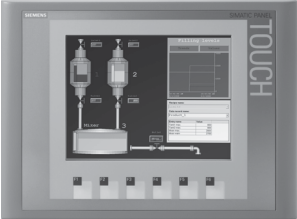
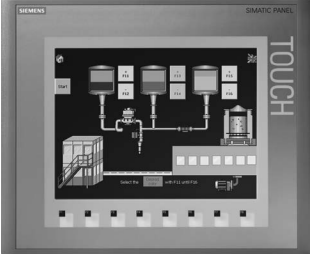
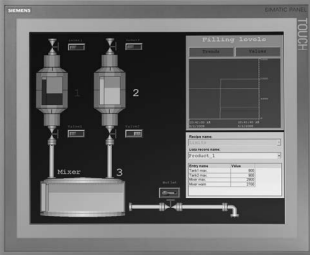
1.3. Moduły S7-1200

Tabela 1.8. Moduły rozszerzeń S7-1200

Typ modułu	Opis	
<p>CPU obsługuje jedną płytke rozszerzającą:</p> <ul style="list-style-type: none"> ● Płytki sygnałowe (SB) pozwalają dodawać do CPU porty I/O. SB jest dołączana od strony frontowej CPU ● Płytki komunikacyjne (CB) pozwalają na dodanie następnego portu komunikacyjnego do CPU ● Płytką bateryjną (BB) umożliwia długotrwale podtrzymanie pracy zegara czasu rzeczywistego. 		<p>① Diody LED statusu SB</p>
<p>Moduły rozszerzeń (SM) służą do zwiększania funkcjonalności CPU. Są dołączane z prawej strony CPU.</p> <ul style="list-style-type: none"> ● Cyfrowe I/O ● Analogowe I/O ● RTD i termopary ● SM 1278 IO-Link Master 		
<p>Moduły komunikacyjne (CM) oraz procesory komunikacyjne (CP) zwiększają możliwości komunikacyjne CPU przez dodanie portów w standardzie PROFIBUS lub RS232/RS485 (dla PtP, Modbus lub USS), lub AS-i Master. CP zapewnia kompatybilność z innymi typami standardów połączeń, takimi jak sieć GPRS.</p> <ul style="list-style-type: none"> ● CPU obsługuje do 3 modułów komunikacyjnych (CM lub CP) ● Każdy moduł komunikacyjny jest dołączany z lewej strony CPU (lub z lewej strony innego CM lub CP, który jest już dołączony do CPU) 	<p>② Gniazdo magistrali</p>	
<p>Moduły komunikacyjne (CM) oraz procesory komunikacyjne (CP) zwiększają możliwości komunikacyjne CPU przez dodanie portów w standardzie PROFIBUS lub RS232/RS485 (dla PtP, Modbus lub USS), lub AS-i Master. CP zapewnia kompatybilność z innymi typami standardów połączeń, takimi jak sieć GPRS.</p> <ul style="list-style-type: none"> ● CPU obsługuje do 3 modułów komunikacyjnych (CM lub CP) ● Każdy moduł komunikacyjny jest dołączany z lewej strony CPU (lub z lewej strony innego CM lub CP, który jest już dołączony do CPU) 	<p>③ Wyjmowalna listwa zaciskowa</p>	
<p>Moduły komunikacyjne (CM) oraz procesory komunikacyjne (CP) zwiększają możliwości komunikacyjne CPU przez dodanie portów w standardzie PROFIBUS lub RS232/RS485 (dla PtP, Modbus lub USS), lub AS-i Master. CP zapewnia kompatybilność z innymi typami standardów połączeń, takimi jak sieć GPRS.</p> <ul style="list-style-type: none"> ● CPU obsługuje do 3 modułów komunikacyjnych (CM lub CP) ● Każdy moduł komunikacyjny jest dołączany z lewej strony CPU (lub z lewej strony innego CM lub CP, który jest już dołączony do CPU) 		<p>① Diody LED statusu modułu komunikacyjnego</p>
<p>Moduły komunikacyjne (CM) oraz procesory komunikacyjne (CP) zwiększają możliwości komunikacyjne CPU przez dodanie portów w standardzie PROFIBUS lub RS232/RS485 (dla PtP, Modbus lub USS), lub AS-i Master. CP zapewnia kompatybilność z innymi typami standardów połączeń, takimi jak sieć GPRS.</p> <ul style="list-style-type: none"> ● CPU obsługuje do 3 modułów komunikacyjnych (CM lub CP) ● Każdy moduł komunikacyjny jest dołączany z lewej strony CPU (lub z lewej strony innego CM lub CP, który jest już dołączony do CPU) 		<p>② Złącze komunikacyjne</p>

1.4. Panele operatorskie Basic HMI

Panele operatorskie SIMATIC HMI Basic oferują urządzenia z ekranami dotykowymi, które umożliwiają podstawową obsługę operatorską i monitorowanie zadań. Wszystkie panele zapewniają stopień ochrony do poziomu IP65 i posiadają certyfikaty CE, UL, cULus oraz NEMA 4x.

Panel Basic HMI	Opis	Dane techniczne
 <p>KP 300 Basic PN</p>	<p>Klawiatura membranowa o przekątnej 3,6 cala z 10 swobodnie konfigurowalnymi przyciskami.</p> <ul style="list-style-type: none"> • Mono (STN, czarno-biały) 87 mm x 31 mm (3,6") • Programowany kolor podświetlenia (biały, zielony, żółty lub czerwony) • Rozdzielczość: 240 x 80 	<ul style="list-style-type: none"> • 250 tagów • 50 ekranów procesów • 200 alarmów • 25 wykresów • Pamięć receptury 40 kB • 5 receptur, 20 rekordów danych, 20 zmiennych
 <p>KTP 400 Basic PN</p>	<p>4" ekran dotykowy z 4 dotykowymi przyciskami funkcyjnymi</p> <ul style="list-style-type: none"> • Wyświetlacz monochromatyczny (STN, skala szarości) • Wymiary 6,79 mm x 57,59 mm (przekątna 3,8") • Wertykalnie lub horyzontalnie • Rozdzielczość: 320 x 240 	<ul style="list-style-type: none"> • 250 tagów • 50 ekranów procesu • 200 alarmów • 25 krzywych trendu • Pamięć receptury 40 kB • 5 receptur, 20 rekordów danych, 20 zmiennych
 <p>KTP 600 Basic PN</p>	<p>6" ekran dotykowy z 6 dotykowymi przyciskami funkcyjnymi</p> <ul style="list-style-type: none"> • Wyświetlacz kolorowy (TFT, 256 kolorów) lub monochromatyczny (STN, skala szarości) • Wymiary 115,2 mm x 86,4 mm (przekątna 5,7") • Wertykalnie lub horyzontalnie • Rozdzielczość: 320 x 240 	<ul style="list-style-type: none"> • 500 tagów • 50 ekranów procesu • 200 alarmów • 25 krzywych trendu • Pamięć receptury 40 kB • 5 receptur, 20 rekordów danych, 20 zmiennych
 <p>KTP 1000 Basic PN</p>	<p>10" ekran dotykowy z 8 dotykowymi przyciskami funkcyjnymi</p> <ul style="list-style-type: none"> • Wyświetlacz kolorowy (TFT, 256 kolorów) • Wymiary 211,2 mm x 158,4 mm (przekątna 10,4") • Rozdzielczość: 640 x 480 	<ul style="list-style-type: none"> • 500 tagów • 50 ekranów procesu • 200 alarmów • 25 krzywych trendu • Pamięć receptury 40 kB • 5 receptur, 20 rekordów danych, 20 zmiennych
 <p>TP 1500 Basic PN</p>	<p>15" ekran dotykowy</p> <ul style="list-style-type: none"> • Wyświetlacz kolorowy (TFT, 256 kolorów) • Wymiary 304,1 mm x 228,1 mm (przekątna 15,1") • Rozdzielczość: 1024 x 768 	<ul style="list-style-type: none"> • 500 tagów • 50 ekranów procesu • 200 alarmów • 25 krzywych trendu • Pamięć receptury 40 kB (zintegrowana pamięć flash) • 5 receptur, 20 rekordów danych, 20 zmiennych

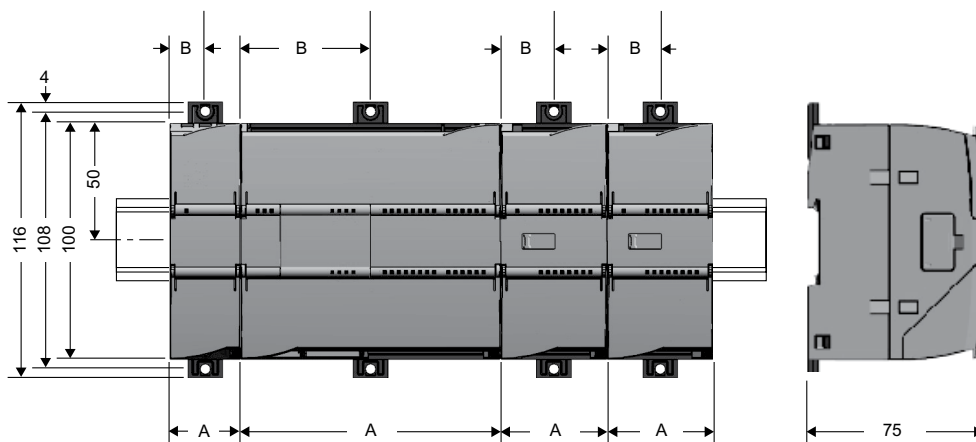
1.5. Wymiary montażowe i wymagane tolerancje pasowania

Urządzenia z serii S7-1200 zostały zaprojektowane tak, by były łatwe do instalacji. Zarówno przy instalacji na panelu montażowym jak i na standardowej szynie montażowej kompaktowe wymiary S7-1200 pozwalają efektywnie wykorzystywać dostępne miejsce.

Dokładne wytyczne i wymagania dla instalacji znajdują się w podręczniku systemu S7-1200.

CPU 1211C, CPU 1212C, CPU 1214C

(wymiary w mm)



CPU 1215C, CPU 1217C

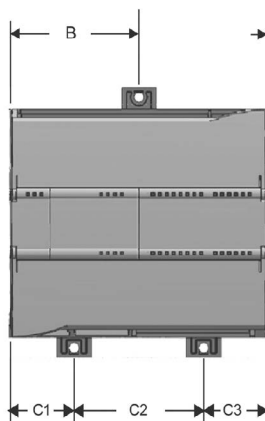


Tabela 1.9. Wymiary montażowe [mm]

Urządzenia S7-1200		Szerokość A [mm]	Szerokość B [mm]	Szerokość C [mm]
CPU	CPU 1211C i CPU 1212C	90	45	–
	CPU 1214C	110	55	–
	CPU 1215C	130	65 (górze)	Dół: C1: 32,5, C2: 65, 33: 32,5
	CPU 1217C	150	75	Dół: C1: 37,5 C2: 75 C3: 37,5
Moduły rozszerzeń	Cyfrowe: 8 i 16 punktów; analogowe: 2, 4 i 8 punktów, termopary: 4 i 8 punktów, RTD: 4 punkty SM 1278 IO Link-Master	45	22,5	–
	Cyfrowe wyjścia przekaźnikowe (8 × DQ, przełączane)	70	35	–
	Analogowe: 16 punktów, RTD: 8 punktów	70	35	–
Moduły komunikacyjne	CM 1241 RS232 i CM 1241 RS422/485 CM 1243-5 PROFIBUS dla urządzenia master, CM 1242-5 PROFIBUS dla urządzenia slave CM 1242-2 AS-i Master CP 1242-7 GPRS	30	15	–
	Teleservice Adapter IE Basic ¹			
	TS Adapter	30	15	–
	TS Moduł	30	15	–

¹ Przed zainstalowaniem TS (Teleservice) Adaptera IE Basic, należy najpierw podłączyć TS Adapter i TS Moduł. Całkowita szerokość („szerokość A”) wynosi 60 mm.

Każdy CPU, SM, CM, oraz CP jest dostosowany do montażu na szynie DIN lub płycie montażowej. W celu montażu na szynie należy wykorzystać zaczepek montażowe znajdujące się na module. Te zaczepek można także zatrzasknąć w pozycjach rozsuniętych, co pozwala bezpośrednio zamocować moduł za pomocą śrub na ścianie, drzwiach szafy sterującej lub płycie montażowej. Wewnętrzny wymiar otworu dla zaczepek DIN modułu wynosi 4,3 mm.

Nad i pod modulem należy pozostawić wolną przestrzeń o szerokości co najmniej 25 mm dla umożliwienia swobodnego przepływu powietrza chłodzącego.

Urządzenia z serii S7-1200 zostały zaprojektowane tak, by były łatwe do instalacji. Można je instalować na panelu montażowym lub na standardowej szynie montażowej i mogą być ułożone poziomo lub pionowo. Małe wymiary S7-1200 pozwalają efektywnie wykorzystywać dostępne miejsce.

 **OSTRZEŻENIE**

Wymagania instalacyjne dla sterowników PLC S7-1200

Sterowniki PLC rodziny SIMATIC S7-1200 są sterownikami kompaktowymi. Wymaga się by były instalowane w obudowach, szafkach lub sterowniach elektrycznych. Dostęp do tych obudów, szafek i sterowni elektrycznych powinien mieć wyłącznie upoważniony personel.

Nieprzestrzeżenie podczas instalacji podanych wymagań może spowodować śmierć, poważne obrażenia ciała i/lub uszkodzenie mienia.

Podczas instalacji sterowników PLC S7-1200 zawsze należy przestrzegać podanych wymagań

Separacja urządzeń S7-1200 od źródeł ciepła, wysokiego napięcia oraz zakłóceń elektrycznych

Generalną zasadą jest, aby zawsze rozdzielać urządzenia generujące wysokie napięcia i zakłócenia od cyfrowych urządzeń niskonapięciowych, takich jak np. sterowniki S7-1200.

Podczas projektowania zabudowy S7-1200 na panelu, należy uwzględnić elementy wytwarzające ciepło i umieścić urządzenia elektroniczne w obszarach chłodzonych szafy montażowej. Unikanie podwyższonej temperatury przedłuża czas pracy wszystkich urządzeń elektronicznych.

Należy również rozważyć prowadzenie okablowania urządzeń na panelu. Trzeba unikać prowadzenia niskonapięciowych przewodów sygnałowych i kabli przesyłu danych obok kabli zasilających i wysokoenergetycznych, a także generujących zakłócenia impulsowe.

Zapewnienie właściwej przestrzeni dla chłodzenia i okablowania

Urządzenia S7-1200 są zaprojektowane w taki sposób, że wystarczające jest naturalne chłodzenie konwekcyjne. Należy jedynie zapewnić wolną przestrzeń wynoszącą co najmniej 25 mm nad i pod sterownikiem. Trzeba także zapewnić wolną przestrzeń montażową o głębokości co najmniej 25 mm.

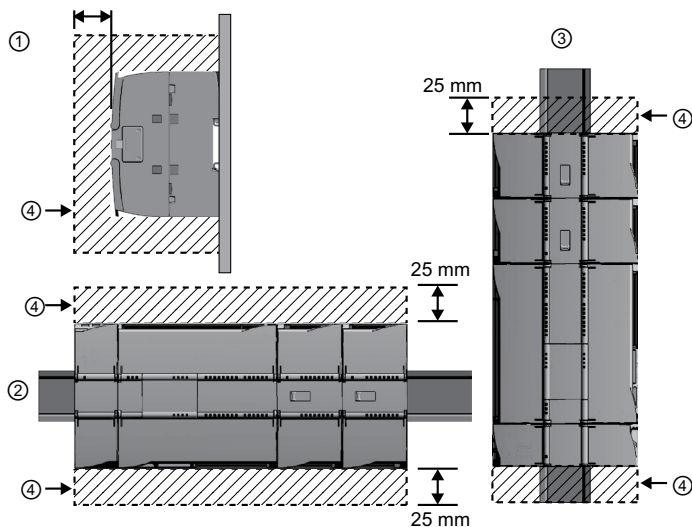
 **OSTROŻNIE**

Przy montażu pionowym, maksymalna dopuszczalna temperatura otoczenia jest mniejsza o 10°C.

Należy tak montować system S7-1200, jak pokazano na rysunku poniżej.

Należy się upewnić, że system S7-1200 został zamontowany poprawnie.

Podczas planowania zabudowy systemu S7-1200, należy zapewnić wystarczającą przestrzeń dla wykonania okablowania oraz poprowadzenia kabli przesyłu danych.



- ① Widok z boku
- ② Montaż poziomy
- ③ Montaż pionowy
- ④ Niezbędny odstęp

Należy jedynie zapewnić wolną przestrzeń wynoszącą co najmniej 25 mm nad i pod sterownikiem dla swobodnej cyrkulacji powietrza.

⚠ OSTRZEŻENIE

Próba instalowania lub demontażu S7-1200 lub współpracujących urządzeń z załączonym napięciem zasilania może doprowadzić do porażenia elektrycznego lub niewłaściwego zadziałania urządzenia.

Nie wyłączenie zasilania S7-1200 i współpracujących urządzeń podczas procedury instalacji lub deinstalacji może spowodować śmierć lub poważne zranienie personelu i/lub zniszczenie mienia.

Należy zawsze stosować odpowiednie procedury bezpiecznej pracy oraz sprawdzić czy napięcie zasilania S7-1200 i współpracujących urządzeń jest wyłączone przed instalacją lub demontażem CPU lub jakiegokolwiek modułu rozszerzeń.

Zawsze należy się upewnić czy do instalacji lub wymiany urządzenia S7-1200 jest używany właściwy moduł lub kompatybilne urządzenie.

 **OSTRZEŻENIE**

Poprawna instalacja modułów S7-1200

Niewłaściwe zainstalowanie dowolnego modułu S7-1200 może spowodować nieprzewidywalne działanie programu w S7-1200.

Wymiana urządzenia rodziny S7-1200 na niewłaściwy model lub niewłaściwie przeprowadzona procedura wymiany (błędna orientacja lub kolejność modułów) może spowodować, poprzez nieprzewidywalne działanie sprzętu, śmierć lub poważne zranienie personelu i/lub zniszczenie mienia.

Należy zawsze wymieniać urządzenia systemu S7-1200 na taki sam model oraz zapewnić właściwą orientację i położenie.

1.6. Nowe możliwości

W tym wydaniu wprowadzono następujące nowe możliwości:

- S7-1200 obsługuje nowe bloki organizacyjne (OB) (strona 86) z różnicami poziomów priorytetów i przerw.
- Web Serwer (strona 186) obsługuje teraz wyświetlanie standardowych stron internetowych oraz definiowanych przez użytkownika stron internetowych przez urządzenie mobilne, a także komputer PC. W tym wydaniu standardowe strony internetowe są dostępne w języku angielskim, niemieckim, francuskim, hiszpańskim, włoskim oraz uproszczonym chińskim.
- Funkcja *Download in RUN* (Ładowanie w trybie pracy) (strona 268) obsługuje teraz maksymalnie dwadzieścia bloków, które można pobierać w trybie RUN. Można też dodawać i zmieniać tagi w istniejących blokach danych i blokach funkcyjnych oraz pobierać zmodyfikowane bloki danych w trybie RUN.
- Narzędzia online i diagnostyczne STEP 7 zapewniają środki do wykonywania aktualizacji firmware (strona 265) dla CPU, modułów rozszerzeń (sygnalowych), modułów komunikacyjnych, oraz dołączonej płytki sygnałowej lub komunikacyjnej.
- STEP 7 posiada funkcję śledzenia (trace) oraz analizatora stanów logicznych (*logic analyzer*) (strona 270), które można wykorzystać dla CPU S7-1200 w wersji 4.0. Dzięki tej funkcji, można skonfigurować określone dane, które mają być śledzone i rejestrowane, kiedy procesor spełni warunek wyzwalania, który można zdefiniować. CPU przechowuje zarejestrowane dane i STEP 7 dostarcza narzędzia do pobierania i analizowania zarejestrowanych danych.
- Nowe instrukcje programowania:
 - Ustaw tag na zbocze sygnału: R_TRIG, F_TRIG
 - Zapisz czas lokalny: WR_LOC_T
 - Maksymalna długość ciągu znaków: MAX_LEN
 - Przerwania dzienne: SET_TINTL, CAN_TINT, ACT_TINT, QRY_TINT
 - Receptury procesowe: RecipeExport, RecipeImport
 - Obsługa adresów: LOG2GEO, RD_ADDR

- Sterowanie ruchem: MC_WriteParam, MC_ReadParam
- Włączenie/wyłączenie hasła: ENDIS_PW
- Ulepszenia HSC, umożliwiające do każdego wejścia lub wyjścia instrukcji HSC przypisanie każdego wejścia wbudowanego lub cyfrowej płytki sygnałowej (SB)
- Ulepszenia PTO/PWM, umożliwiające do każdego wejścia lub wyjścia instrukcji PTO/PWM przypisanie każdego wyjścia wbudowanego lub cyfrowej płytki sygnałowej (SB)
- Zaawansowane funkcje biblioteczne, w tym kontrola wersji

Nowe moduły sterownika S7-1200

Nowe moduły rozszerzają możliwości sterownika S7-1200 w zakresie zaspokojenia potrzeb automatyzacji:

- Nowe CPU 1217C DC/DC/DC z szybkimi wejściami/wyjściami różnicowymi
- Nowe i ulepszone moduły rozszerzeń S7-1200. Nowe moduły rozszerzeń (6ES7 2xx-xxx32-0XB0) zastępują istniejące moduły rozszerzeń (6ES7 2xx-xxx30-0XB0). Nowe moduły zapewniają:
 - zakres prądowy 4...20 mA, dodany dla modułów wejść/wyjść analogowych,
 - wykrywanie przerwania przewodu przy korzystaniu z zakresu 4...20 mA, dodane dla modułów wejść analogowych,
 - kodowanie złączy w celu uniknięcia błędów podczas podłączania na obiekcie złączy przewodów, dodane dla modułów z wyjściami przekaźnikowymi,
 - kompatybilność części zamiennych: można bez zmian zastosować moduł skorygowany w miejsce istniejącego
- Nowe części zamienne dla CPU S7-1200
- Nowy symulator wejść CPU 1217C Input Simulator (6ES7 274-1XK30-0XA0)
- Nowy moduł sygnałowy SM 1278 4xIO-Link Master (6ES7 278-4BD32-0XB0) funkcjonuje zarówno jako moduł rozszerzeń, jak i moduł komunikacyjny oraz umożliwia podłączenie do 4 urządzeń IO-Link Slave (połączenie 3-przewodowe) lub 4 standardowych urządzeń wykonawczych lub standardowych koderów
- Nowy moduł potencjometryczny S7-1200 Potentiometer Module (6ES7 274-1XA30-0XA0)
- Nowy moduł komunikacyjny CM CANopen dla S7-1200 to wkładany moduł, który umożliwia podłączenie urządzeń CANopen do sterownika PLC S7-1200. Może być skonfigurowany jako urządzenie master lub slave.

Wymiana CPU w wersji 3.0 na CPU w wersji 4.0

W przypadku wymiany CPU S7-1200 w wersji 3.0 na CPU S7-1200 w wersji 4.0, należy zwrócić uwagę na udokumentowane różnice między wersjami.

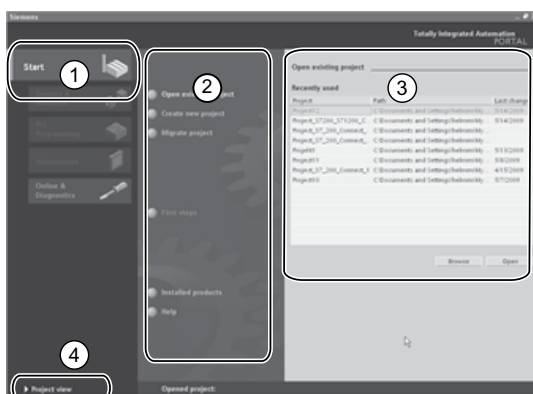
Zobacz także

Podstawowe panele HMI (strona 20)

STEP 7 ułatwia pracę

2

STEP 7 jest przyjaznym dla użytkownika środowiskiem do projektowania logiki niezbędnej do sterowania aplikacją, konfiguracji wizualizacji interfejsów HMI i komunikacji. W celu zwiększenia wydajności projektowania STEP 7 wyposażono w dwa różne widoki projektu: zorientowany zadaniowo (*Portal view* – widok Portalu), lub zorientowany projektowo i oparty na widoku elementów projektu (*Project view* – widok Projektu). Użytkownik decyduje, który widok pozwala mu na lepsze wykorzystanie możliwości systemu. Pojedyncze kliknięcie umożliwia przełączanie się pomiędzy obydwoma typami widoku.



Widok Portalu

- ① Portale dla różnych zadań
- ② Zadania dla wybranego portalu
- ③ Paner wyboru dla wybranych akcji
- ④ Zmiana na widok Projektu



Widok Projektu

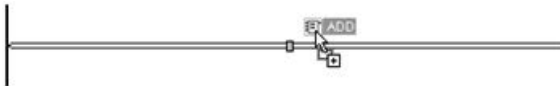
- ① Menu i paski narzędzi
- ② Nawigator projektu
- ③ Obszar roboczy
- ④ Karty zadań
- ⑤ Okno nadzoru
- ⑥ Zmiana na widok Portalu
- ⑦ Pasek edycji

Wszystkie te składniki umieszczone w jednym miejscu pozwalają na łatwy dostęp do każdego elementu projektu. Przykładowo, okno nadzoru pokazuje właściwości i informacje dla obiektu, który został zaznaczony w obszarze roboczym. Wybranie innego obiektu powoduje wyświetlanie przez okno nadzoru właściwości, które mogą być dla niego konfigurowane. Okno nadzoru zawiera karty, w których widoczne są informacje diagnostyczne i inne wiadomości.

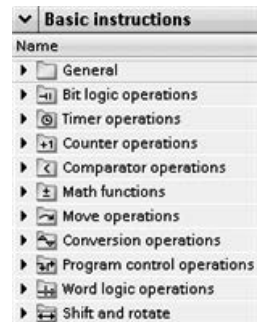
Pokazując wszystkie otwarte edytory, pasek edytora pozwala na szybszą i wydajniejszą pracę. Wystarczy kliknąć na dany edytor, aby przełączyć się pomiędzy poszczególnymi otwartymi edytorami. Istnieje także możliwość równoczesnego wyświetlania dwóch edytorów w pozycji poziomej lub pionowej. Pozwala to na przenoszenie danych za pomocą mechanizmu przeciągnąć i upuścić pomiędzy edytorami.

2.1. Łatwe wstawianie instrukcji do programu użytkownika

STEP 7 wyświetla karty zadań, które zawierają instrukcje do programu. Instrukcje są pogrupowane zgodnie z ich funkcjami.



Dla stworzenia programu należy tylko przeciągnąć daną instrukcję z karty zadań do programu.



2.2. Łatwy dostęp do ulubionych instrukcji z paska narzędzi

STEP 7 zapewnia pasek *Favorites* (Ulubione) w celu uzyskania szybkiego dostępu do najczęściej używanych instrukcji. Jedno kliknięcie na ikonie instrukcji wprowadza ją do programu!



(*Favorites* w drzewie instrukcji uruchamiane jest poprzez podwójne kliknięcie ikony).




Favorites można łatwo dostosowywać poprzez dodawanie nowych instrukcji.

Wystarczy przeciągnąć i upuścić instrukcję na *Favorites*.

Instrukcja jest teraz dostępna przez jedno kliknięcie!



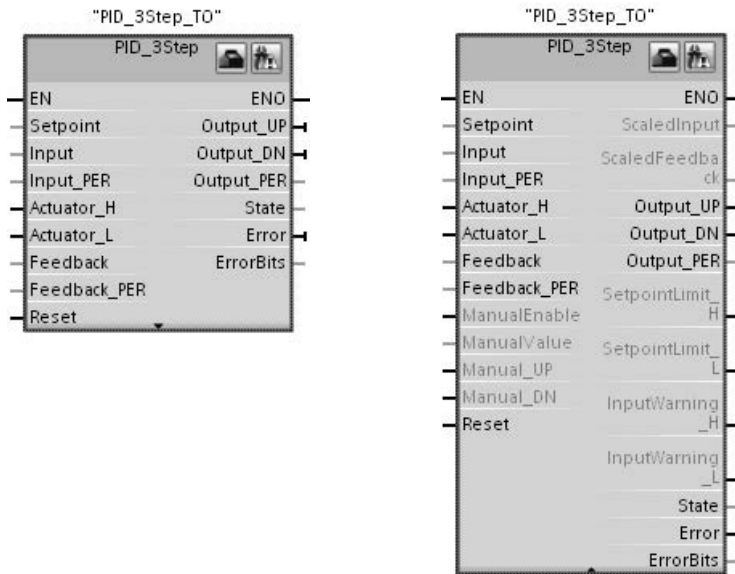
2.3. Łatwe dodawanie wejść i wyjść do instrukcji w drabince lub bloku funkcyjnym

 Niektóre instrukcje pozwalają na tworzenie dodatkowych wejść i wyjść.

- Aby dodać wejście i wyjście wystarczy kliknąć ikonę utwórz *Create* lub PPM (prawy przycisk myszy) kliknąć na wejście i zamienić z istniejącym parametrem wejściowym lub wyjściowym oraz wybrać komendę *Insert input* lub *Insert output*.
- Do usunięcia wejścia lub wyjścia należy kliknąć PPM na parametry istniejącego wejścia lub wyjścia (jeśli istnieje więcej niż dwa oryginalne wejścia) i wybrać komendę *Delete* (Kasuj).


2.4. Rozwijane instrukcje

Niektóre bardziej złożone instrukcje wyświetlają tylko kluczowe wejścia i wyjścia. Aby wyświetlić wszystkie wejścia i wyjścia należy nacisnąć strzałkę na dole instrukcji.



2.5. Łatwa zmiana trybu pracy CPU

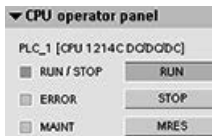
CPU nie ma fizycznego przycisku do zmiany trybu pracy (STOP lub RUN).

Należy użyć przycisków *Start CPU* i *Stop CPU* z paska narzędzi dla  zmiany trybu pracy CPU.

Podczas konfiguracji CPU należy ustawić domyślne ustawienia startowe we właściwościach CPU (strona 72).

Zakładka diagnostyki *Online and diagnostic* zawiera panel operacyjny pozwalający na zmianę trybu pracy podłączonego CPU. Do użycia tego panelu należy

połączyć online jednostkę centralną. Pasek zadań narzędzia online *Online Tools* wyświetla panel operacyjny, który pokazuje tryb pracy podłączonego CPU. Panel operacyjny pozwala także na zmianę trybu pracy podłączonego CPU.

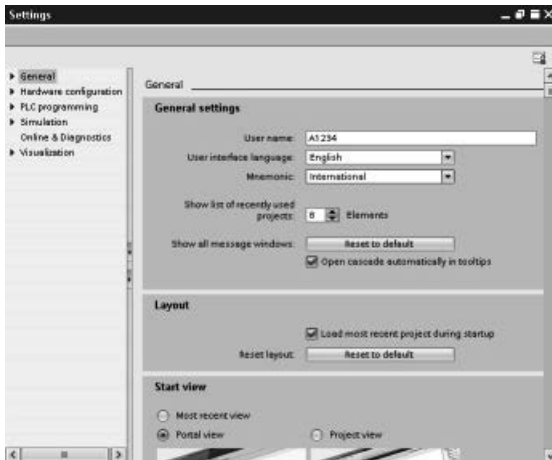


Użyj przycisku w panelu operacyjnym do zmiany trybu pracy (STOP lub RUN). Panel operacyjny ma także przycisk MRES do czyszczenia pamięci.

Kolor wskaźnika RUN/STOP pokazuje bieżący stan trybu pracy CPU. Kolor żółty oznacza tryb STOP, kolor zielony oznacza tryb RUN.

Informacje na temat ustawienia domyślnego trybu pracy po włączeniu zasilania znajdują się w rozdziale Tryby pracy CPU publikacji „Podręcznik systemu S7-1200”.

2.6. Łatwa zmiana wyglądu i konfiguracji dla STEP 7



Dostępne jest szeroka gama ustawień, takich jak wygląd interfejsu, język lub też zmiana folderu do zapisu projektu użytkownika.

Użyj komendy *Settings* (Ustawienia) z menu rozwijalnego *Options* (Opcje) dla zmiany tych ustawień.

2.7. Ułatwienia dostępu dla projektu i globalnych bibliotek

Biblioteki globalne i biblioteki projektowe pozwalają na ponowne użycie zapisanych obiektów przez projekt lub grupę projektów. Przykładowo, istnieje możliwość stworzenia bloku szablonów, który można używać w różnych projektach lub łatwo modyfikować do konkretnego zadania automatyzacji. Biblioteki przechowują różne obiekty, takie jak bloki FC, FB, DB, konfiguracje urządzenia, typy danych, tabele nadzoru oraz obrazy procesów. Biblioteki umożliwiają także możliwość zapisu komponentów urządzeń HMI w projekcie.

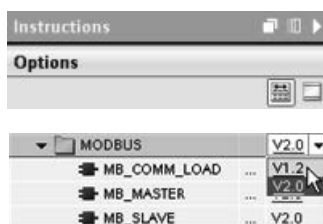


Każdy projekt ma bibliotekę przechowującą obiekty, które używane są w projekcie więcej niż jeden raz. Biblioteka ta jest integralną częścią projektu. Otwieranie lub zamykanie projektu otwiera lub zamyka równocześnie jego bibliotekę, a zapisywanie projektu zapisuje także wszelkie zmiany w jego bibliotece.

Możesz utworzyć globalną bibliotekę przechowującą obiekty dostępne do użycia w innych projektach. Po stworzeniu globalnej biblioteki należy ją zapisać na komputerze lub w sieci.

2.8. Łatwy dobór wersji instrukcji

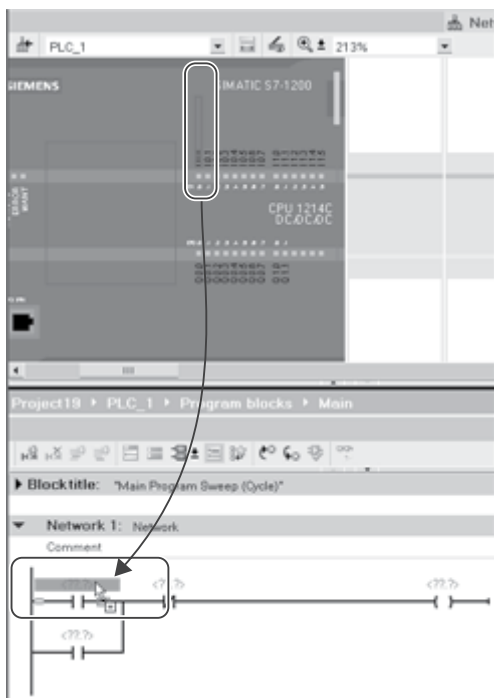
Rozwój i ciągle tworzenie oraz doskonalenie niektórych instrukcji (takich jak Modbus, PID i instrukcje sterujące napędami) spowodował powstanie wielu wersji tych instrukcji. Dla zapewnienia kompatybilności ze starszymi projektami, STEP 7 pozwala użytkownikowi na wybór wersji instrukcji użytej w programie.



Kliknij na ikonę drzewie instrukcji karty zadań, żeby uaktywnić nagłówki i kolumny drzewa instrukcji.

W celu zmiany wersji instrukcji wybierz odpowiednią wersję z listy rozwijalnej.

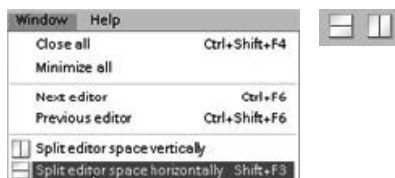
2.9. Łatwe przeciąganie elementów pomiędzy edytorami



Do wyświetlenia dwóch edytorów jednocześnie użyj komendy *Split editor* z menu wybieralnego lub przycisków na pasku narzędzi.

W celu przyspieszenia realizacji zadań STEP 7 pozwala na przeciągnięcie i uproszczenie elementów pomiędzy poszczególnymi edytorami. Przykładowo można chwycić adres wejścia z CPU i przenieść go do instrukcji w programie użytkownika. Aby zaznaczyć wejścia lub wyjścia CPU należy użyć powiększenia co najmniej 200%.

Tagi nazw nie tylko są wyświetlane w tablicy tagów sterownika, ale także na samym sterowniku.



Dla przełączenia się pomiędzy otwartymi edytorami kliknij ikony w pasku edytora.



2.10. Zmiana typu wywołania DB



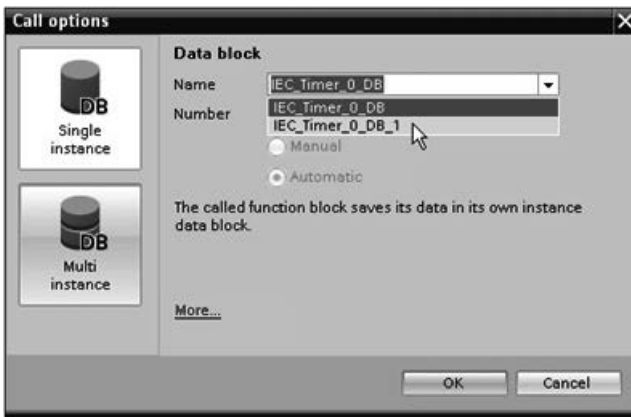
STEP 7 pozwala na łatwą zmianę lub stworzenie powiązania DB (bloku danych) dla instrukcji lub FB (bloku funkcji) znajdującym się w FB.

Możesz przełączać powiązania pomiędzy różnymi DB.

Możesz zmieniać powiązania pomiędzy pojedynczą instancją DB i DB z wieloma instancjami.

Możesz stworzyć instancję DB (jeśli została ona zgubiona lub nie jest dostępna).

Możesz uzyskać dostęp do komendy *Change call type* (Zmiana typu wywołania) PPM na instrukcji lub FB w edytorze programu lub wybierając komendę *Block call* (Wywołanie bloku) z menu *Options* (Opcje).

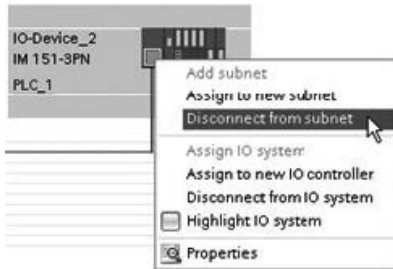


Call options (Opcje wywołania) pozwalają na wybranie DB z pojedynczą lub wieloma instancjami. Można także wybrać określony DB z rozwijalnej listy DB.

2.11. Tymczasowe odłączenie urządzeń z sieci

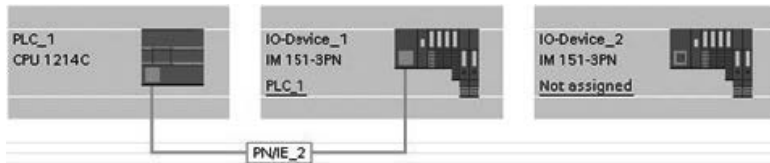
Istnieje możliwość odłączenia pojedynczego urządzenia z podsieci. Ponieważ konfiguracja tego urządzenia nie jest usuwana z projektu, można łatwo przywrócić połączenie z tym urządzeniem.





Kliknij PPM na porcie interfejsu sieciowego i wybierz komendę *Disconnect from subnet* (Odłącz z podsieci) z menu kontekstowego.

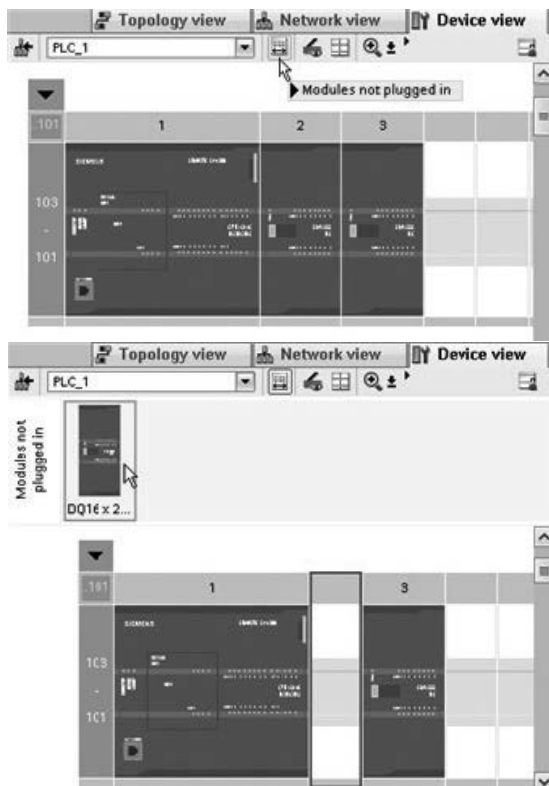
STEP 7 rekonfiguruje połączenia sieciowe, ale nie usuwa rozłączonego urządzenia z projektu. Podczas gdy połączenie sieciowe danego urządzenia jest usunięte, adresy jego interfejsu nie są zmienione.



Podczas ładowania nowych połączeń sieciowych, CPU musi być ustawiony w trybie pracy STOP.

Dla powtórnego połączenia urządzenia stwórz nowe połączenie sieciowe z portem danego urządzenia.

2.12. Wirtualne „rozłączenie” modułów bez straty konfiguracji



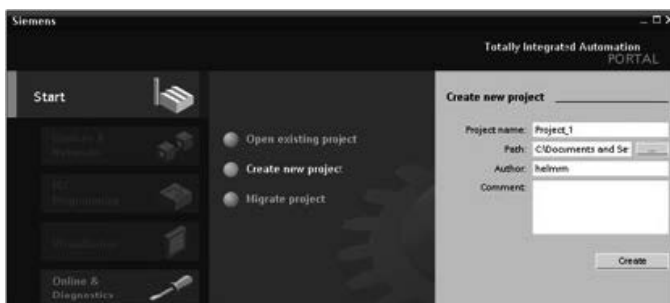
STEP 7 ma obszar magazynowania dla rozłączonych modułów. Możesz łatwo przemieścić moduł z wirtualnej listwy montażowej, aby zapisać konfigurację tego modułu. Te rozłączone moduły są zapisywane razem z projektem pozwalając na powtórne podłączenie modułu w przyszłości bez konieczności rekonfiguracji jego parametrów.

Podczas czasowych konserwacji sprzętu ta zdolność może być wykorzystana. Wyobraź sobie następujący scenariusz, w którym możesz czekać na moduł zastępczy i planujesz tymczasowo używać innego modułu jako krótkotrwale zastępstwo. Możesz przenieść skonfigurowany moduł z wirtualnej listwy do *Unplugged modules (Odłączonych modułów)* i później włożyć w jego miejsce tymczasowy zamiennik.

Rozpoczęcie pracy

3.1. Tworzenie projektu

Praca ze STEP 7 jest łatwa. Poniżej przedstawiono przykład szybkiego tworzenia projektu.



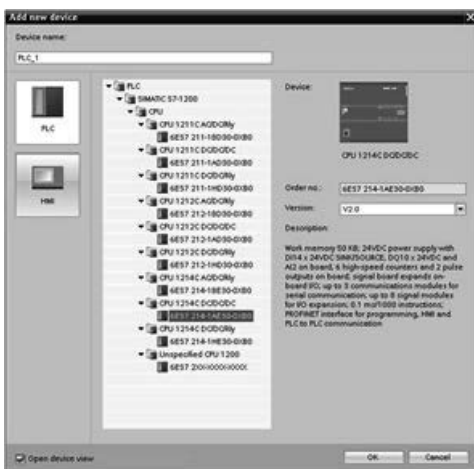
W portalu startowym kliknij *Create new project* (Stwórz nowy projekt).

Wprowadź nazwę projektu i kliknij przycisk *Create*.



Po stworzeniu projektu użyj portalu *Devices & Networks* (Urządzenia i Sieci).

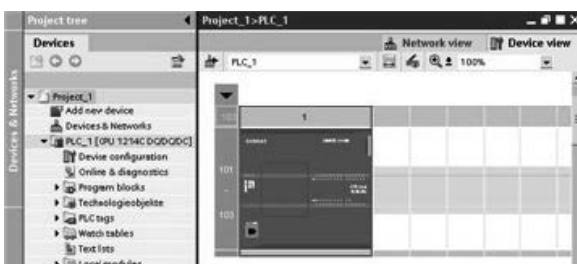
Kliknij przycisk *Add new device* (Dodaj nowe urządzenie).



Wybierz CPU, aby dodać go do projektu:

1. W oknie dialogowym *Add new device* (Dodaj nowe urządzenie) naciśnij przycisk *SIMATIC PLC*.
2. Wybierz CPU z listy.
3. Aby dodać wybrane CPU do projektu naciśnij przycisk *Add* (Dodaj).

Zauważ, że zaznaczona jest opcja *Open device view* (Otwórz widok urządzenia). Klikając *Dodaj* z tą opcją otwierasz *Device configuration* (Konfigurację urządzenia) w widoku Projektu.



Widok Urządzenia wyświetla dodane CPU.

3.2. Tworzenie zmiennych (tagów) dla obszaru I/O CPU

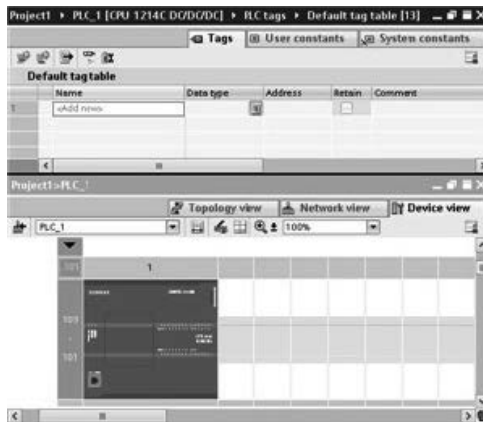
PLC tags (zmiennie PLC) są nazwami symbolicznymi dla wejść/wyjść i adresów. Po stworzeniu zmiennej PLC, STEP 7 magazynuje ją w tablicy zmiennych. Wszystkie edytory użyte w projekcie (takie jak edytor programu, edytor urządzeń, edytor wizualizacji i edytor tablic monitorujących) mają dostęp do tablicy zmiennych.



Otwórz tablicę zmiennych, gdy otwarty jest już edytor urządzeń. Możesz zobaczyć otwarte edytory w pasku edytorów.



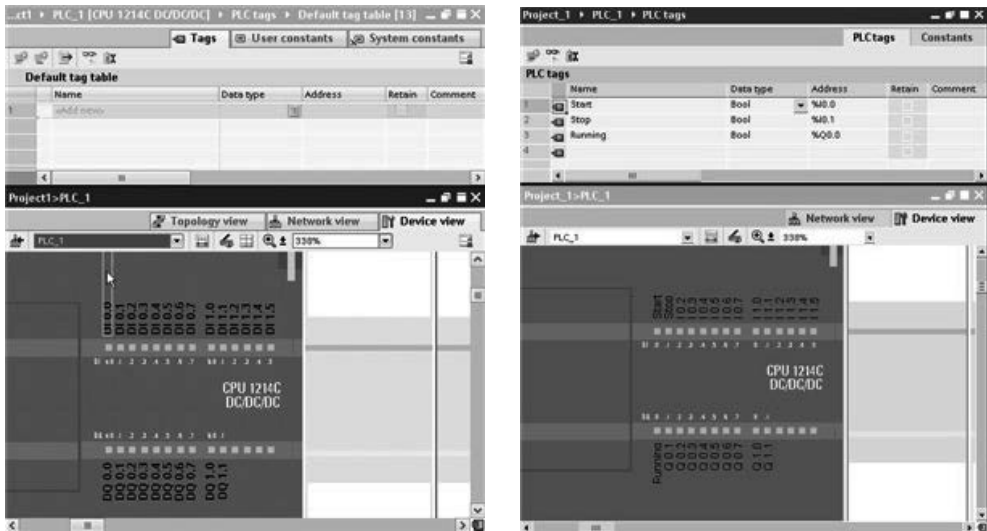
Na pasku narzędzi kliknij przycisk *Split editor space horizontally* (Rozłóż obszar edytorów horyzontalnie).



STEP 7 wyświetla obydwie edytory na jednym ekranie.

Powiększ konfigurację urządzenia ponad 200%, aby punkty I/O CPU były widoczne i wybieralne. Przeciągnij wejścia i wyjścia z CPU do tablicy zmiennych:

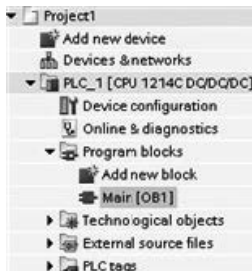
1. Wybierz I0.0 i przeciągnij je do pierwszej wiersza tablicy zmiennych.
2. Zmień nazwę zmiennej z „I0.0” na „Start”.
3. Przeciągnij I0.1 do tablicy zmiennych i zmień nazwę na „Stop”.
4. Przeciągnij Q0.0 (dół CPU) do tablicy zmiennych i zmień jego nazwę na *Running* (Uruchomiony).



Po wprowadzeniu nazw zmiennych do tablicy są one dostępne w programie użytkownika.

3.3. Tworzenie prostej sieci w programie użytkownika

Kod programu użytkownika składa się z instrukcji, które CPU wykonuje w sekwencji. W tym przykładzie użyj języka drabinkowego (LAD) do stworzenia kodu programu. Program w LAD jest sekwencją sieci, która przypomina szczeble drabiny.



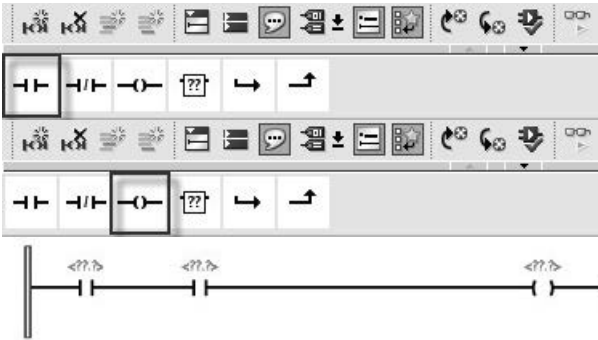
Do otwarcia programu należy:

1. Rozwinąć folder *Program blocks* w drzewie projektu, aby wyświetlić blok „Main [OB1]“.
2. Kliknąć dwukrotnie blok „Main [OB1]“.

Edytor programu otworzy blok programu (OB1).

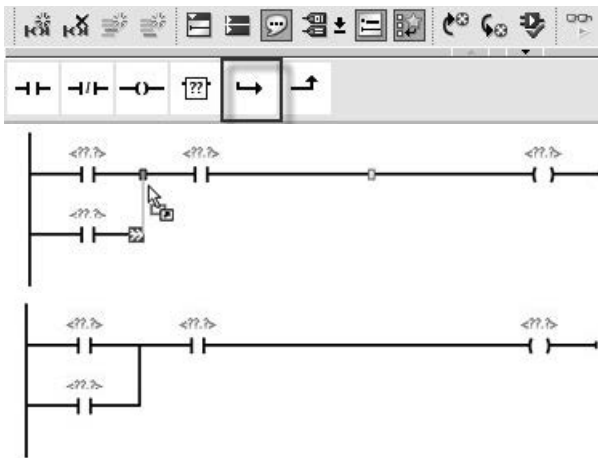
Użyj przycisków z *Favorites* (Ulubione) do dodania styków i cewek w linii programu.

3. Rozpoczęcie pracy



1. Kliknij *Normally open contact* (Styk normalnie otwarty) w *Ulubione*, aby dodać styk do programu
2. Dodaj drugi styk
3. Kliknij *Output coil* (Cewka wyjściowa), aby dodać do programu cewkę.

Zakładka *Ulubione* posiada także przycisk tworzenia gałęzi.

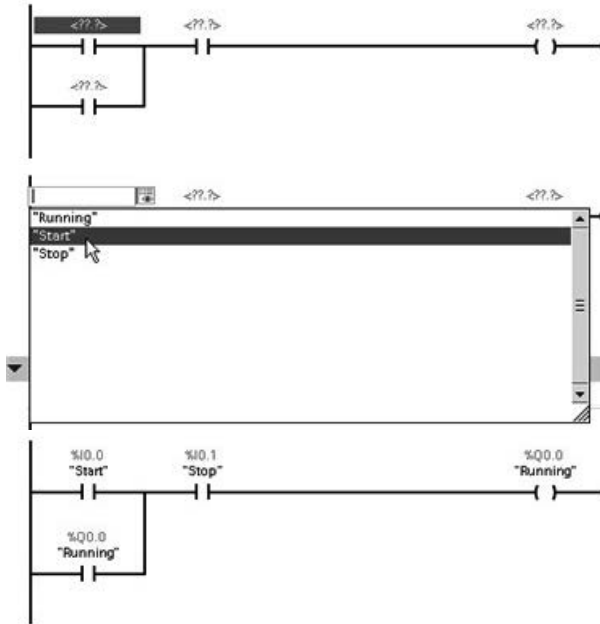


1. Wybierz lewą szynę do połączenia rozgałęzienia.
2. Kliknij ikonę *Open branch* (Otwórz rozgałęzienie), aby dodać gałąź do szyny sieci.
3. Włóż następny normalnie otwarty styk do otwartego rozgałęzienia.
4. Przeciągnij strzałkę do punktu połączenia (zielony kwadrat na szczęblu) pomiędzy dwoma stykami w pierwszej linii.

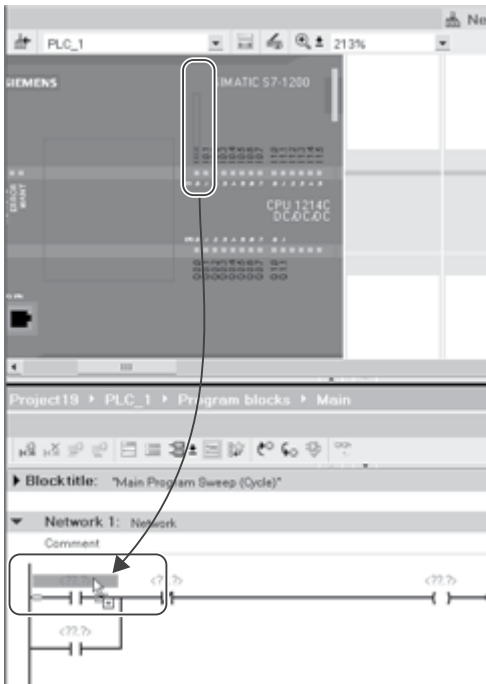
Kliknij przycisk *Save project* na pasku narzędzi, aby zapisać projekt. Zauważ, że edycja szczębla nie musi być zakończona przed zapisaniem. Możesz teraz powiązać nazwy zmiennych z tymi instrukcjami.

3.4. Użycie zmiennych PLC z tablicy zmiennych do adresowania instrukcji

Używając tablicy zmiennych (tagów) możesz szybko wprowadzić zmienne do adresowania styków i cewek.



1. Dwukrotnie kliknij adres domyślny <???.?> nad pierwszym normalnie otwartym stykiem.
2. Naciśnij ikonę selektora znajdującą się na prawo od adresu, aby otworzyć zmiennę w tablicy zmiennych.
3. Z listy rozwijalnej wybierz *Start* dla pierwszego styku.
4. Dla drugiego styku powtórz procedurę i wybierz zmienną *Stop*.
5. Dla cewki i styku podtrzymującego wybierz *Running*.



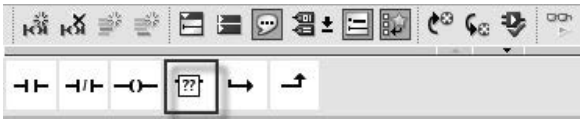
Możesz także przeciągnąć adresy I/O bezpośrednio z CPU. Podziel obszar roboczy w Widoku projektu (strona 33).

Musisz powiększyć widok CPU przynajmniej do 200% jeśli chcesz zaznaczyć punkty I/O.

Możesz przeciągnąć I/O CPU w *Device configuration* (Ustawieniach urządzenia) do instrukcji LAD w edytorze programu nie tylko dla stworzenia adresu instrukcji, ale także dla stworzenia wpisu w tablicy zmiennych.

3.5. Dodawanie pola instrukcji

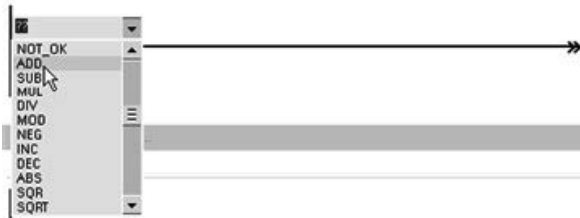
Edytor programu umożliwi stworzenia ogólnego pola instrukcji. Po włożeniu tego pola do programu istnieje możliwość wybrania typu instrukcji, takich jak np. instrukcja ADD, z listy rozwijanej.



Kliknij box (ogólne pole instrukcji) w pasku narzędzi *Ulubione*.



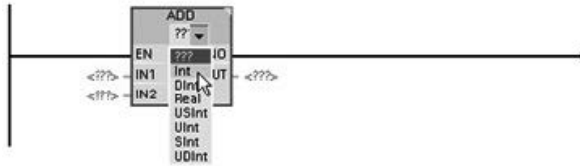
Ogólne pole instrukcji posiada różne instrukcje. W tym przypadku użyj instrukcji ADD:



1. Naciśnij żółty narożnik pola instrukcji dla wyświetlenia listy rozwijalnej dostępnych instrukcji.

2. Przewiń w dół listę i zaznacz instrukcję ADD.

3. Kliknij w żółtym narożniku „?” aby wybrać typ danych dla wejść i wyjść.



Możesz teraz wpisać zmienne (lub adresy pamięci) dla wartości używanych przez instrukcję ADD.



Możesz stworzyć także dodatkowe wejścia dla pewnych instrukcji:

1. Naciśnij jedno z wejść znajdujących się wewnątrz instrukcji w polu instrukcji.

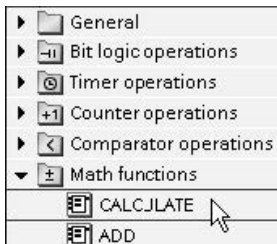
2. PPM dla wyświetlenia menu kontekstowego i zaznaczenia komendy *Insert input* (Dodaj wejście).



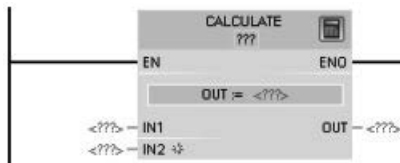
Instrukcja ADD używa teraz trzech wejść.

3.6. Użycie instrukcji CALCULATE dla złożonych równań matematycznych

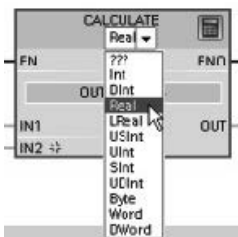
Instrukcja CALCULATE pozwala na stworzenie matematycznej funkcji, która operuje na wielu parametrach wejściowych i generuje wyniki zgodne z zdefiniowanym równaniem.



W drzewie *Basic instruction* (Instrukcje podstawowe) rozwiń folder *Math functions*. Kliknij dwukrotnie instrukcję *Calculate*, żeby wprowadzić instrukcje do programu użytkownika.



Nieskonfigurowana instrukcja *Calculate* ma dwa parametry wejściowe i jeden parametr wyjściowy.

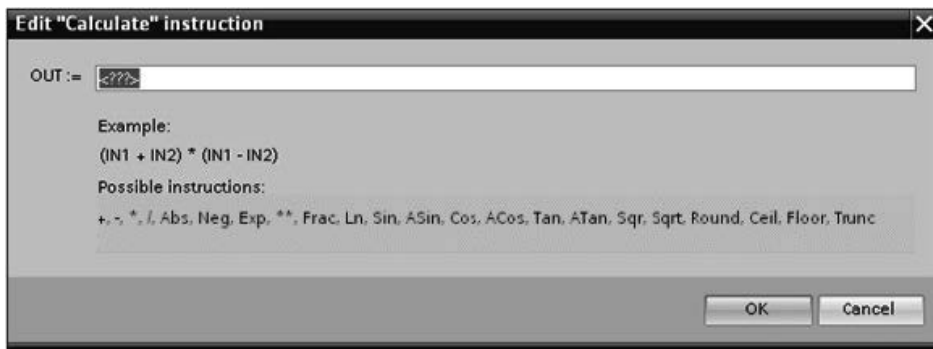


Naciśnij „???” i wybierz typ danych dla parametrów wejściowych i wyjściowych (parametry wejściowe i wyjściowe muszą obsługiwać ten sam typ danych).

Dla tego przypadku kliknij typ danych *Real*.



Naciśnij ikonę *Edit equation* (Edytuj równanie) dla wprowadzenia równania.



3. Rozpoczęcie pracy

W tym przypadku wprowadź następujące równanie do przeskalowania nieobrobionych wartości analogowych („In“ oraz „Out“ są oznaczeniami parametrów związanych z instrukcją *Calculate*).

$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) \cdot (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

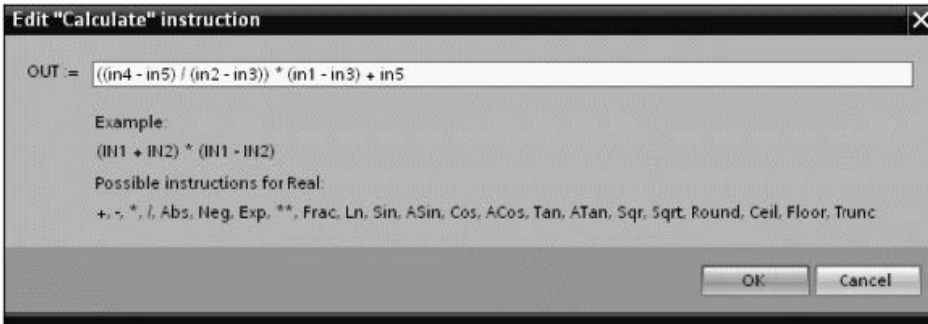
$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) \cdot (\text{in1} - \text{in3}) + \text{in5}$$

gdzie:

Out _{value}	(Out)	Przeskalowane wartości wyjściowe
In _{value}	(in1)	Wartości analogowego wejścia
In _{high}	(in2)	Górna granica dla przeskalowanej wartości wejściowej
In _{low}	(in3)	Dolna granica dla przeskalowanej wartości wejściowej
Out _{high}	(in4)	Górna granica dla przeskalowanej wartości wyjściowej
Out _{low}	(in5)	Dolna granica dla przeskalowanej wartości wyjściowej

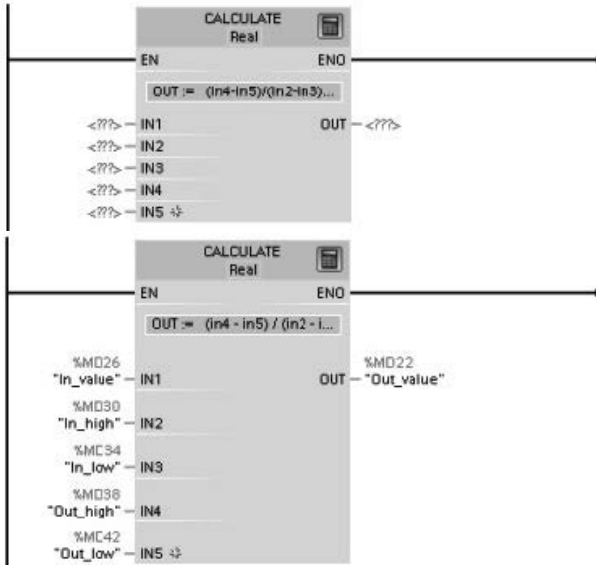
W polu *Edit Calculate* wpisz równanie z nazwami parametrów:

$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) \cdot (\text{in1} - \text{in3}) + \text{in5}$$

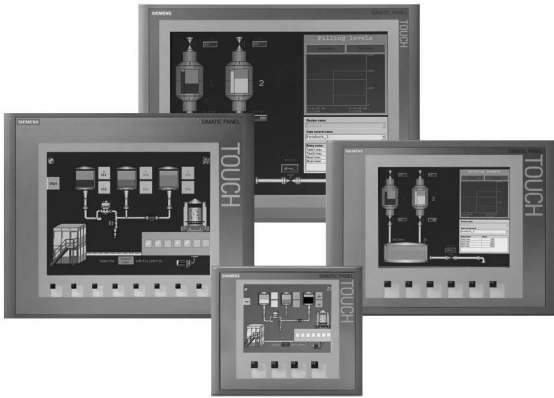


Po naciśnięciu OK, instrukcja *Calculate* tworzy wejścia wymagane do jej obsługi.

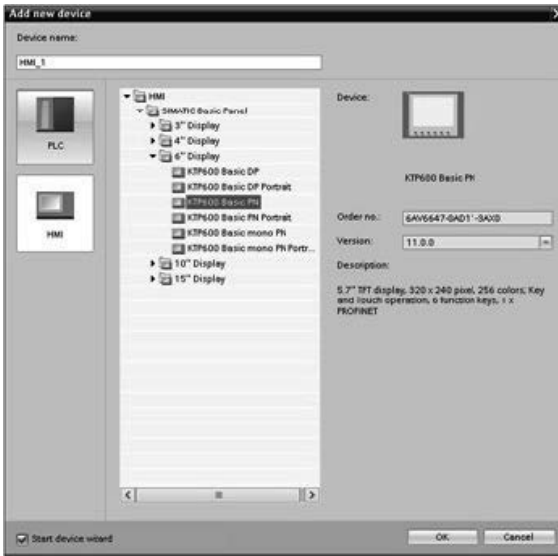
Wprowadź nazwy zmiennych dla wartości poszczególnych parametrów.



3.7. Dodawanie urządzenia HMI do projektu



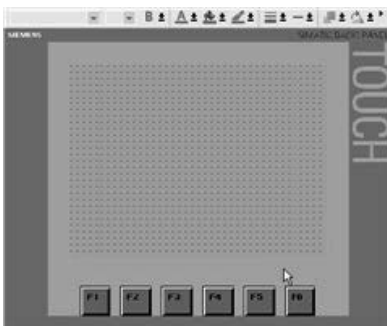
Dodawanie urządzenia HMI do projektu jest bardzo łatwe!



1. Kliknij dwukrotnie na ikonę *Add new device* (Dodaj nowe urządzenie).
2. Naciśnij przycisk *SIMATIC HMI* w oknie dialogowym *Add new device*.
3. Wybierz podłączone urządzenie HMI z listy.

Możesz uruchomić kreatora HMI (HMI wizard), który pomoże Ci w skonfigurowaniu ekranów urządzenia HMI.

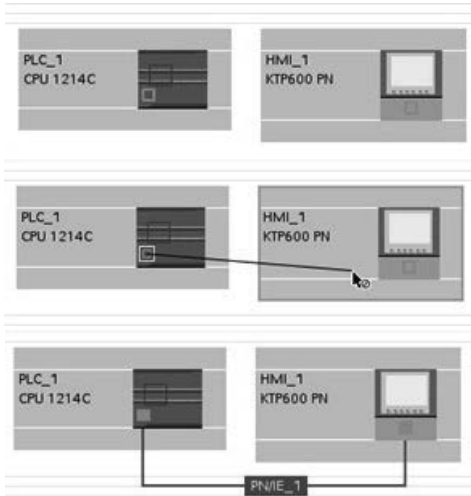
4. Naciśnij *OK* dla dodania urządzenia HMI do twojego projektu.



TIA Portal dodaje urządzenie HMI do projektu. The TIA Portal posiada kreator HMI, który pomaga w konfiguracji wszystkich ekranów i struktur urządzenia HMI.

Jeśli kreator HMI nie zostanie uruchomiony, TIA Portal stworzy prosty, domyślny ekran HMI. Później można dodać dodatkowe ekrany lub obiekty na ekranach.

3.8. Tworzenie połączenia pomiędzy CPU i urządzeniem HMI

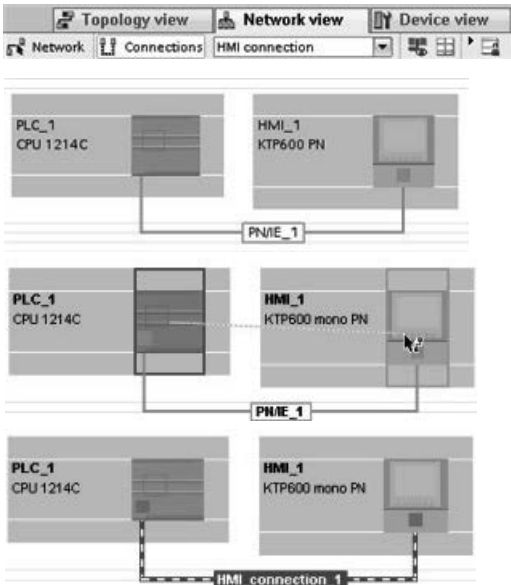


Tworzenie sieci jest proste!

- Idź do *Devices and Networks* (Urządzenia i sieci), a następnie wybierz *Network view* (Widok sieci), aby wyświetlić CPU i urządzenia HMI.
- Aby utworzyć sieć PROFINET, przeciągnij linię z zielonego pola (port Ethernet) danego urządzenia do zielonego pola urządzenia, które chcesz przyłączyć.

Zostało utworzone połączenie pomiędzy dwoma urządzeniami.

3.9. Tworzenie połączenia urządzeń HMI do współdzielenia zmiennych



Tworząc połączenie HMI pomiędzy dwoma urządzeniami możesz łatwo współdzielić zmienne pomiędzy nimi.

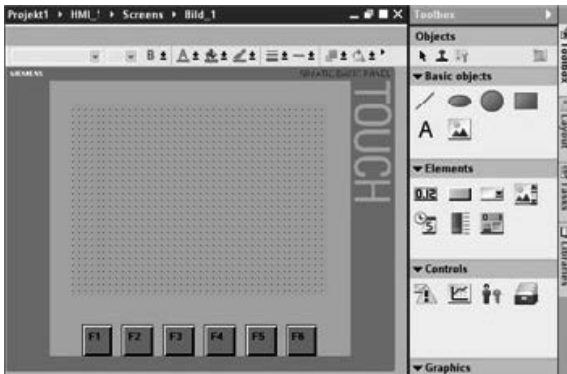
- Po zaznaczeniu połączenia sieciowego naciśnij przycisk *Connections* i wybierz z menu rozwijalnego *HMI connection*.
- Połączenie HMI zmieni kolor obydwu urządzeń na niebieski.
- Zaznacz urządzenie CPU i przeciągnij linię do urządzenia HMI.
- Połączenie HMI pozwoli na konfigurację zmiennych HMI poprzez wybranie listy zmiennych PLC.

Możesz skorzystać także z innej możliwości stworzenia połączenia HMI:

- Przeciągnij zmienną PLC z tablicy zmiennych PLC. Edytor programu lub edytor konfiguracji urządzenia obsługujący ekran urządzenia HMI automatycznie stworzy połączenie HMI.
- Użycie kreatora HMI do wyboru PLC automatycznie stworzy połączenie HMI.

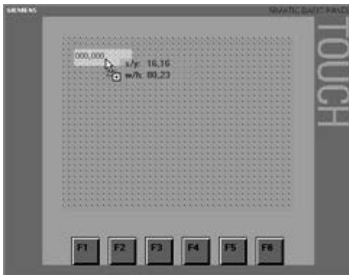
3.10. Tworzenie ekranu urządzenia HMI

Nawet nie korzystając z kreatora HMI, konfigurowanie ekranu HMI jest łatwe!



STEP 7 dostarcza standardowy zestaw bibliotek do wstawiania podstawowych kształtów, elementów interaktywnych jak i standardowych grafik.

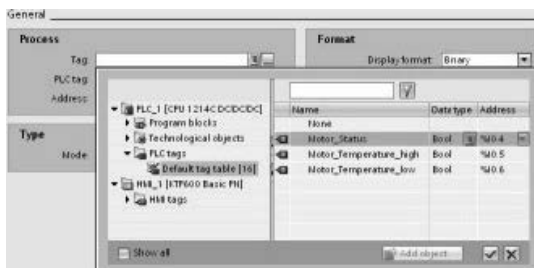
Aby dodać element, wystarczy przeciągnąć i upuścić jeden z elementów na ekranie. Użyj właściwości elementu (w oknie *Inspector*), aby skonfigurować wygląd i zachowanie elementu.



Można również tworzyć elementy na ekranie przez przeciągnięcie i upuszczenie zmienna PLC z drzewa projektu lub edytora programu do ekranu HMI. Zmienia PLC staje się elementem na ekranie. Następnie można użyć właściwości do zmiany parametrów dla tego elementu.

3.11. Wybór zmiennych PLC dla elementów HMI

Po stworzeniu elementów na ekranie użyj właściwości elementu do przypisania zmiennych PLC do danego elementu. Kliknij przycisk wyboru w polu zmienna do wyświetlenia zmiennych PLC dla CPU.



Możesz także przeciągnąć i upuścić zmienne PLC z drzewa projektu do ekranu HMI. Wyświetl zmienne PLC w widoku *Details* (Szczegóły) z drzewa projektu i przeciągnij je na ekran HMI.

Prosty w obsłudze sterownik PLC

4

4.1. Zadania wykonywane w każdym cyklu

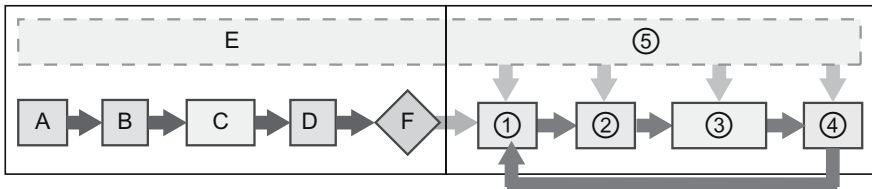
Każdy cykl programu obejmuje ustawienie wyjść, odczytanie stanu wejść, wykonanie instrukcji z programu użytkownika i wykonanie obsługi systemu lub przetwarzanie w tle.



Cykl jest określany jako cykl skanowania lub skanowanie. W warunkach standardowych, wszystkie punkty I/O analogowe i cyfrowe są uaktualniane synchronicznie z cyklem programu wykorzystującym obszar pamięci wewnętrznej zwanym obrazem procesu. Obraz procesu zawiera chwilowy stan fizyczny wejść i wyjść (fizyczne punkty I/O CPU, płytki sygnałowej i modułów rozszerzeń).

- Tuż przed wykonaniem programu użytkownika CPU odczytuje stan fizycznych wejść i zapamiętuje te wartości wejściowe w obszarze wejściowym pamięci obrazu procesu. Dzięki temu uzyskuje się pewność, że te dane pozostają stałe w trakcie wykonywania instrukcji użytkownika.
- CPU wykonuje zadania określone instrukcjami użytkownika i – nie zmieniając stanu fizycznych wyjść – uaktualnia wartości wyjściowe w obszarze wyjściowym pamięci obrazu procesu.
- Po wykonaniu programu użytkownika, CPU przepisuje stany wyjść z obszaru wyjściowego pamięci obrazu procesu do fizycznych wyjść.

Ten proces zapewnia zachowanie spójności logiki poprzez wykonywanie w danym cyklu instrukcji użytkownika i zapobiega zmianom („migotaniu”) stanu fizycznych punktów wyjściowych, w wyniku mogących występować wielokrotnie w danym cyklu zmianom w obszarze wyjściowym pamięci obrazu procesu.



STARTUP	RUN
A Czyści obszar pamięci I.	① Zapisuje zawartość pamięci Q do wyjść fizycznych.
B Inicjalizuje wyjścia z ostatnimi wartościami lub zastępczymi wartościami.	② Kopiuje stan fizycznych wejść do pamięci I.
C Wykonuje startowe OB.	③ Wykonuje cykliczne OB.
D Kopiuje stan fizycznych wejść do pamięci I.	④ Wykonuje autodiagnostykę.
E Zapisuje zdarzenia przerwania do kolejki oczekującej na wykonanie w trybie RUN.	⑤ Obsługuje przerwania w dowolnej części cyklu programu.
F Uaktywnia zapis zawartości pamięci Q do fizycznych wyjść.	

Użytkownik może zmienić standardowe działanie modułu, wyłączając automatyczne uaktualnianie stanu punktów wyjściowych. Można również natychmiastowo odczytywać i zapisywać cyfrowe i analogowe stany I/O modułów podczas wykonywania instrukcji. Natychmiastowy odczyt stanu fizycznych wejść nie uaktualnia obszaru wejściowego pamięci obrazu procesu. Natychmiastowy zapis stanu do fizycznych wyjść uaktualnia zarówno obszar wyjściowy pamięci obrazu procesu, jak i stan fizycznych punktów wyjściowych.

4.2. Tryby pracy CPU

CPU może pracować w jednym z trzech trybów: w trybie STOP, w trybie STARTUP i w trybie RUN. Diody statusu znajdujące się na płycie czołowej CPU wskazują jaki jest aktualny tryb pracy.

- W trybie STOP CPU nie wykonuje programu i użytkownik może wczytać projekt. Dioda LED RUN/STOP pali się na żółto.
- W trybie STARTUP wykonywany jest jednokrotnie startowy OB (jeśli istnieje). W fazie startowej CPU nie obsługuje przerwania. Dioda LED RUN/STOP miga przemiennie na zielono i żółto.
- W trybie RUN regularnie jest powtarzany cykl programu. Mogą się pojawiać przerwania i CPU może przetwarzać je w dowolnym miejscu cyklu programu. W trybie RUN można ładować niektóre części projektu. Dioda LED RUN/STOP pali się na zielono.

Przechodzenie do trybu RUN CPU wykonuje metodą gorącego restartu. Gorący restart nie obejmuje kasowania pamięci, ale pamięć może zostać skasowana odpowiednim rozkazem z programu STEP 7. Kasowanie pamięci czyści całą pamięć roboczą, obszary pamięci trwałej i nietrwałej, kopiuje zawartość pamięci ładowania do pamięci roboczej, oraz ustawia wyjścia według skonfigurowanego parametru *Reaction to CPU STOP* (Reakcja na zatrzymanie CPU). Kasowanie pamięci

nie czyści zawartości bufora diagnostycznego a także na stałe zapisanej wartości adresu IP. Gorący restart inicjalizuje wszystkie nietrwałe dane systemowe oraz dane użytkownika.

Można skonfigurować ustawienie CPU *Startup after POWER ON* (uruchomienie po włączeniu zasilania) wraz z metodą restartu używając programu STEP 7. Ta pozycja konfiguracji jest dostępna w menu *Startup* (Uruchomienie) dla CPU pod nagłówkiem *Device configuration* (Konfiguracja urządzeń). Po włączeniu zasilania CPU wykonuje sekwencję testów diagnostycznych i inicjalizuje system. Podczas inicjalizacji systemu CPU usuwa całą nietrwałą pamięć bitową i resetuje całą nietrwałą zawartość DB do wartości wyjściowych z pamięci ładowania. Następnie CPU przechodzi w odpowiedni tryb włączania zasilania. Wykrycie pewnych błędów uniemożliwia CPU wejście w tryb RUN. CPU może pracować w następujących trybach włączania zasilania: *No restart (stay in STOP mode)* (pozostanie w trybie STOP), *Warm restart – RUN* (przejdzie do trybu RUN po wykonaniu gorącego restartu), oraz *Warm restart – mode prior to POWER OFF* (przejdzie do przedniego trybu pracy przed wyłączeniem zasilania, po wykonaniu gorącego restartu).

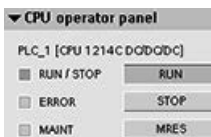
Uwaga

Konfiguracja trybu po gorącym restarcie

CPU może przejść do trybu STOP z powodu naprawialnych błędów, np. awarii wymiennego modułu rozszerzeń, lub tymczasowych błędów, takich jak zakłócenia linii zasilającej lub zmiany napięcia zasilania.

Jeżeli wybrano opcję konfiguracji *Warm restart – mode prior to POWER OFF* (przejdzie do przedniego trybu pracy przed wyłączeniem zasilania, po wykonaniu gorącego restartu), CPU nie powróci do trybu RUN po naprawieniu lub usunięciu awarii dopóty, dopóki nie otrzyma nowego polecenia z programu STEP 7, aby przejść do trybu RUN. Bez nowego polecenia, tryb STOP jest utrzymywany jako poprzedni tryb pracy przed wyłączeniem zasilania (POWER OFF).

CPU, który ma pracować niezależnie od programu STEP 7, zwykle powinien być skonfigurowany z wybraną opcją *Warm restart – RUN* (przejdzie do trybu RUN po wykonaniu gorącego restartu), aby mógł powrócić do trybu RUN po ponownym włączeniu zasilania po usunięciu awarii.



CPU nie ma żadnego fizycznego przełącznika do zmiany trybu pracy. Do zmiany trybu pracy należy użyć następujących narzędzi STEP 7:

- Przycisków *Stop* oraz *Run* w pasku narzędzi STEP 7.
- Panelu operatorskiego CPU w narzędziach online.

Można również wykonać z programu użytkownika instrukcję STP, która ustawia tryb STOP pracy CPU. Pozwala to zatrzymać wykonywanie programu użytkownika zgodnie z jego logiką.

4.3. Wykonanie programu użytkownika

CPU obsługuje następujące typy bloków kodu, które umożliwiają stworzenie wydajnej struktury programu użytkownika:

- Bloki organizacyjne (OB) definiujące strukturę programu. Niektóre OB mają predefiniowane działanie i czynności początkowe, ale użytkownik może również tworzyć OB z czynnościami początkowymi, które sam określi (strona 53).
- Funkcje (FC) i bloki funkcji (FB) zawierające kod programu odpowiadający za wykonanie określonego zadania lub kombinacje parametrów. Każdy FC lub FB ma zbiór parametrów wejściowych i wyjściowych służących do wymiany danych z blokiem wywołującym. Ponadto FB wykorzystuje skojarzone bloki danych (zwane blokami instance), które przechowują dane wykorzystywane w aktualnie przetwarzanym bloku FB lub w innych blokach programu. Poprawny zakres numerów bloków FC i FB wynosi od 1 do 65 535.
- Bloki danych (DB) przechowujące dane, które mogą być używane przez wszystkie bloki programu. Poprawny zakres numerów bloków DB wynosi od 1 do 65 535.

Rozmiar programu użytkownika, danych i konfiguracji jest ograniczony przez ilość załadowanej i pracującej pamięci w CPU (strona 15). Nie ma określonego limitu dla liczby danego bloku OB, FC, FB lub DB. Jakkolwiek sumaryczna liczba bloków nie może przekroczyć 1024.

4.3.1. Wykonywanie cyklu programu w trybie RUN

W każdym cyklu programu CPU zapisuje wyjścia, odczytuje wejścia, uaktualnia moduły komunikacyjne, wykonuje zadania na własne potrzeby i odpowiada na przerwania wynikające z warunków ustalonych przez użytkownika.

Te działania (z wyłączeniem zdarzeń ustalonych przez użytkownika) są wykonywane regularnie i w porządku sekwencyjnym. Te zdarzenia użytkownika, które są odblokowane, są obsługiwane zgodnie ze swoimi priorytetami w takiej kolejności, w jakiej się pojawiają. Dla zdarzeń przerwań, procesor odczytuje wejścia, wykonuje OB, a następnie zapisuje wyjścia, używając odpowiednich partycji obrazu procesu (PIP – *Process Image Partitions*), o ile to możliwe.

System gwarantuje wykonanie kompletnego cyklu programu w czasie nazywanym maksymalnym czasem cyklu; w przeciwnym wypadku generowane jest zdarzenie błędu czasowego.

- Każdy cykl programu rozpoczyna się od pobrania z obrazu procesu bieżących wartości wyjść cyfrowych, analogowych i zapisaniu ich do fizycznych wyjść CPU, SB i SM skonfigurowanych tak, by były synchronicznie uaktualniane (konfiguracja domyślna). Kiedy dostęp do wyjścia fizycznego odbywa się za pomocą instrukcji, to zarówno wyjściowy obszar obrazu procesu jak i stan tego wyjścia jest uaktualniony.
- Cykl programu jest kontynuowany poprzez odczytanie bieżących wartości cyfrowych i analogowych wejść z CPU, SB oraz SM skonfigurowanych do automatycznego uaktualniania I/O (konfiguracja domyślna). Następuje wówczas

przepisanie ich wartości do obrazu procesu. Kiedy dostęp do wejścia fizycznego odbywa się za pomocą instrukcji, to stan tego wejścia można odczytać, ale wejściowy obszar obrazu procesu nie jest uaktualniany.

- Po odczytaniu stanu wejść, jest wykonywany program użytkownika począwszy od pierwszej instrukcji po ostatnią. Wykonywane są więc wszystkie OB cyklu programu wraz z powiązanymi z nimi FC i FB. OB cyklu programu są wykonywane w kolejności posiadanych numerów, przy czym jako pierwszy jest wykonywany OB z najniższym numerem.

Prowadzenie komunikacji odbywa się okresowo podczas cyklu programu i może być przerwane przez wykonanie programu użytkownika.

Testy autodiagnostyki obejmują okresowe sprawdzanie systemu i sprawdzanie stanu modułów I/O.

Przerwania mogą występować w dowolnej części cyklu programu i są generowane zdarzeniami. Kiedy zachodzi zdarzenie, CPU przerywa cykl programu i wywołuje OB przygotowany do obsługi tego zdarzenia. Kiedy OB zakończy obsługę zdarzenia, wtedy CPU podejmuje wykonywanie programu użytkownika od miejsca, w którym został przerwany.

4.3.2. Zastosowanie bloków organizacyjnych OB porządkuje program użytkownika

OB sterują wykonywaniem programu użytkownika. Każdy OB musi mieć unikalny numer. Domyślnie numery OB są zarezerwowane poniżej wartości 200, inne OB muszą mieć numer 200 lub większy.

Określone zdarzenia w CPU uruchamiają wykonywanie bloku organizacyjnego. OB nie może wywołać innego OB lub być wywołany z FC lub FB. Jedynie zdarzenie, takie jak przerwanie diagnostyczne lub interwał czasowy może uruchomić wykonywanie OB. CPU obsługuje OB zgodnie z posiadaną przez nie klasą priorytetu; OB o wyższym priorytecie są wykonywane przed OB o niższym priorytecie. Najniższa klasa priorytetu to klasa 1 (dla cyklu programu głównego), a najwyższa klasa priorytetu to klasa 24.

4.3.3. Kolejowanie i priorytety wykonania obsługi zdarzeń

Praca CPU jest sterowana zdarzeniami. Zdarzenie powoduje uruchomienie przerwania OB. Istnieje możliwość określenia przerwania OB dla danego zdarzenia podczas tworzenia bloku, podczas konfiguracji urządzenia lub przy pomocy instrukcji ATTACH i DETACH. Niektóre zdarzenia, takie jak cykl programu czy też przerwania cykliczne, występują regularnie. Inne zdarzenia obsługiwane są jednokrotnie, tak jak zdarzenie rozruchowe lub zdarzenie opóźnienia czasowego. Niektóre zdarzenia wykonywane są przez uruchomienie sprzętowe, takie jak wykrycie zbocza na punkcie wejściowym lub zadziałanie wejścia szybkiego licznika. Zdarzenia, takie jak błąd diagnostyczny lub błąd czasowy pojawiają się tylko wtedy, gdy wystąpi błąd. W celu określenia kolejności przetwarzania bloków OB przerwania zdarzenia używa się priorytetu i kolejowanie zdarzeń.

CPU przetwarza zdarzenia w kolejności priorytetów, gdzie 1 oznacza najniższy priorytet, a 26 najwyższy priorytet. Dla CPU S7-1200 w wersjach wcześniejszych niż V4.0, każdy rodzaj OB należał do ustalonej klasy priorytetu (od 1 do 26). W przypadku wersji V4.0, można przypisać klasę priorytetu do każdego skonfigurowanego OB. Numer priorytetu konfigurowany jest w atrybutach właściwości OB.

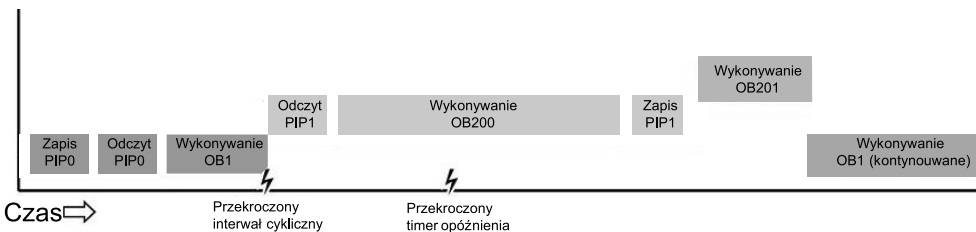
Przerywalne i nieprzerywalne tryby wykonywania

Bloki OB są wykonywane w kolejności priorytetów zdarzeń, które je wywołują. Dla CPU S7-1200 w wersji V4.0 można skonfigurować wykonywanie OB, aby było przerywalne lub nieprzerywalne. Należy pamiętać, że bloki OB cyklu programu są zawsze przerywalne, ale wszystkie inne bloki OB można skonfigurować, aby były przerywalne lub nieprzerywalne.

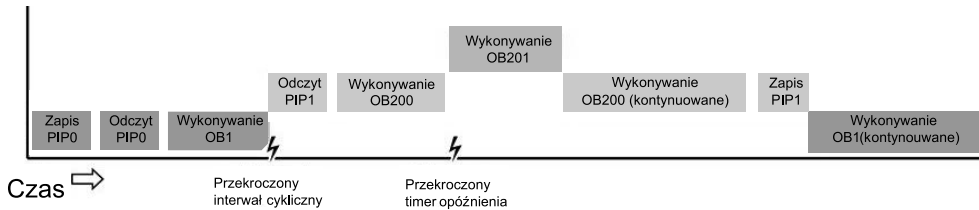
Jeśli zostanie ustawiony tryb przerywalny, a następnie jest wykonywany blok OB i wystąpi zdarzenie o wyższym priorytecie przed zakończeniem wykonywania tego OB, to wykonywanie OB zostanie przerwane w celu umożliwienia uruchomienia OB zdarzenia o wyższym priorytecie. Wykonywane jest OB zdarzenia o wyższym priorytecie i po jego zakończeniu kontynuowane jest wykonywanie OB, które zostało przerwane. Gdy występuje wiele zdarzeń podczas wykonywania przerywalnego OB, to CPU przetwarza te zdarzenia w kolejności priorytetów.

Jeśli nie zostanie ustawiony tryb wykonywania przerywalny, to OB jest wykonywane, aż do jego zakończenia, niezależnie od wszelkich innych zdarzeń, które wystąpią podczas jego wykonywania.

Rozważmy następujące dwa przypadki, w których zdarzenia przerwań wywołują OB cykliczne i OB opóźnienia. W obu przypadkach, OB opóźnienia (OB201) nie ma przypisanej partycji obrazu procesu i jest wykonywany w kolejności priorytetu 4. OB cykliczny (OB200) ma przypisaną partycję obrazu procesu PIP1 i jest wykonywany w kolejności priorytetu 2. Na rysunkach poniżej przedstawiono różnicę trybów wykonywania przerywalnego i nieprzerywalnego:



Rysunek 4.1. Przypadek 1: Nieprzerywalne wykonywanie bloków OB



Rysunek 4.2. Przypadek 2: Przerwalne wykonywanie bloków OB

Uwaga

Jeśli tryb wykonywania bloków OB zostanie skonfigurowany jako nieprzerwalny, to blok OB błędu czasu nie może przerwać wykonywania bloków OB z wyjątkiem bloków OB cyklu programu. Dla CPU S7-1200 w wersjach wcześniejszych niż V4.0, blok OB błędu czasu mógł przerwać wykonywanie dowolnego bloku OB. W wersji V4.0, aby blok OB błędu czasu (lub inny blok OB o wyższym priorytecie) mógł przerywać wykonywanie bloków OB innych niż bloki OB cyklu programu, to należy skonfigurować wykonywanie bloków OB jako przerywalne.

Kolejkowanie i priorytety wykonania obsługi zdarzeń

Liczba pochodzących z jednego źródła zdarzeń oczekujących w kolejce na obsługę jest ograniczona przez stosowanie różnych kolejek dla zdarzeń różnego typu. Po osiągnięciu limitu oczekujących zdarzeń określonego typu, kolejne zdarzenia są tracone. Więcej informacji na temat przepelnienia kolejek jest podanych przy omawianiu bloków OB błędów czasu.

Każde zdarzenie CPU ma przyznany priorytet. Ogólnie, CPU obsługuje zdarzenia zgodnie z ich priorytetem (najpierw najwyższy priorytet). CPU obsługuje zdarzenia mające ten sam priorytet zgodnie z zasadą „pierwszy się pojawił, pierwszy jest obsługany”.

Tabela 4.1. Zdarzenia OB

Zdarzenie	Dozwolona liczba zdarzeń	Domyślny priorytet OB
Cykl programu	1 zdarzenie cykl programu Dozwolonych wiele OB	1 ⁴
Rozruch	1 zdarzenie rozruchu ¹ Dozwolonych wiele OB	1 ⁴
Czas opóźnienia	Do 4 zdarzeń 1 OB na zdarzenie	3
Przerwanie cykliczne	Do 4 zdarzeń 1 OB na zdarzenie	8
Przerwanie sprzętowe	Do 50 zdarzeń przerwania sprzętowego	18
	1 OB na zdarzenie, ale można użyć tego samego OB dla wielu zdarzeń	18

Zdarzenie	Dozwolona liczba zdarzeń	Domyślny priorytet OB
Błąd czasu	1 zdarzenie (tylko jeśli skonfigurowany) ³	22 lub 26 ⁴
Błąd diagnostyki	1 zdarzenie (tylko jeśli skonfigurowany)	5
Wpięcie lub wypięcie modułów	1 zdarzenie	6
Usterka listwy montażowej lub stacji	1 zdarzenie	6
Czas dnia	Do 2 zdarzeń	2
Status	1 zdarzenie	4
Aktualizacja	1 zdarzenie	4
Profil	1 zdarzenie	4

- 1 Zdarzenie rozruchowe i zdarzenie cyklu programu nigdy nie występują w tym samym czasie, ponieważ zdarzenie rozruchu kończy się zanim rozpocznie się zdarzenie cyklu programu.
- 2 Można użyć więcej niż 50 bloków OB zdarzeń przerwania sprzętowego, jeśli zostały użyte instrukcje ATTACH oraz DETACH.
- 3 Można skonfigurować CPU, aby pozostał w trybie RUN, nawet jeśli maksymalny czas cyklu programu zostanie przekroczony, lub można skorzystać z instrukcji RE_TRIGR, aby zresetować czas cyklu. CPU przejdzie jednak w tryb STOP, jeśli maksymalny czas wykonania cyklu zostanie przekroczony po raz drugi w jednym cyklu skanowania.
- 4 Priorytetem dla CPU w nowej wersji V4.0 jest 22. W przypadku wymiany CPU w wersji V3.0 na CPU w wersji V4.0, priorytetem jest 26 – priorytet, który obowiązywał w wersji V3.0. W każdym razie, priorytet można zmienić i ustawić na dowolną wartość z zakresu od 22 do 26.

Więcej informacji można znaleźć w części „Wymiana CPU V.3 na CPU V.4”.

Ponadto CPU rozpoznaje inne zdarzenia, które nie są związane z blokami OB. W tabeli poniżej przedstawiono te zdarzenia i odpowiadające im działania CPU:

Tabela 4.2. Dodatkowe zdarzenia

Zdarzenie	Opis	Działanie CPU
Błąd dostępu do wejścia/wyjścia (I/O)	Błąd bezpośredniego odczytu/zapisu I/O	CPU rejestruje pierwsze wystąpienie w buforze diagnostycznym i pozostaje w trybie RUN
Błąd maksymalnego czasu cyklu	CPU przekracza ustawiony czas cyklu po raz drugi	CPU rejestruje błąd w buforze diagnostycznym i przechodzi w tryb STOP
Błąd dostępu do urządzeń peryferyjnych	Błąd I/O podczas aktualizacji obrazu procesu	CPU rejestruje pierwsze wystąpienie w buforze diagnostycznym i pozostaje w trybie RUN
Błąd programowy	Błąd podczas wykonywania programu	Jeśli blok z błędem zapewnia obsługę błędów, to aktualizuje strukturę błędów; jeśli nie, to CPU rejestruje błąd w buforze diagnostycznym i pozostaje w trybie RUN

Opóźnienie przerwania

Opóźnienie przerwania (czas od momentu, gdy CPU zostało powiadomione o zdarzeniu do chwili wykonania pierwszej instrukcji OB obsługującego to zdarzenie) wynosi około 175 μs, pod warunkiem, że OB cyklu programu jest w trakcie pojawienia się tego przerwania jedynym aktywnym programem obsługi zdarzenia.

Zobacz również

Rozdział „Przejdźcie do trybu online w celu monitorowania wartości w CPU” (strona 254)

4.4. Obszary pamięci, adresowanie i typy danych

CPU dzieli pamięć na następujące obszary służące do przechowywania programu użytkownika, danych i konfiguracji:

- Pamięć ładowania jest pamięcią nieulotną, w której przechowywane są program użytkownika, dane i konfiguracja. Kiedy projekt jest wczytany do CPU, to najpierw trafia do obszaru pamięci ładowania. Ten obszar jest ulokowany albo na karcie pamięci (jeśli jest) albo w CPU. Ten nieulotny obszar pamięci jest zachowany w przypadku utraty zasilania. Możesz zwiększyć wielkość pamięci ładowania dostępnej dla danych przez instalację dodatkowej karty pamięci.
- Pamięć robocza jest pamięcią ulotną, służącą do przechowywania pewnych ograniczonych elementów projektu użytkownika podczas wykonywania programu. CPU kopiuje pewne elementy projektu z pamięci ładowania do pamięci roboczej. Zawartość tej pamięci jest tracona przy odłączaniu zasilania i CPU odtwarza ją po ponownym włączeniu zasilania.
- Pamięć trwała jest pamięcią nieulotną przechowującą ograniczoną liczbę wartości z pamięci roboczej. Pamięć trwała jest stosowana do przechowywania wartości z wybranych miejsc pamięci użytkownika na wypadek utraty zasilania. W przypadku utraty zasilania, CPU odtwarza te trwałe wartości po włączeniu zasilania.



Opcjonalna karta pamięci SIMATIC zapewnia dodatkową pamięć do zapisywania programu użytkownika lub transferu do innych sterowników. Jeśli karta jest używana, to CPU wykonuje program na niej zapisany, a nie ten który jest zapisany w pamięci stałej CPU.

Karta może być zabezpieczona przed zapisem. W tym celu należy przesunąć przycisk ochrony na pozycję LOCK.

Opcjonalna karta pamięci SIMATIC może być użyta jako karta programu lub karta transferu do gromadzenia plików dzienników danych, lub do wykonania aktualizacji oprogramowania:

- Użyj kartę transferu do skopiowania twojego projektu do wielu CPU bez konieczności użycia STEP 7. Karta transferu kopiuje zapisany na niej projekt do pamięci CPU, następnie karta musi być usunięta.
- Karta programu pełni funkcję pamięci CPU; wszystkie funkcje CPU są sterowane przez kartę. Włożenie karty programowej powoduje wymazanie całej załadowanej pamięci wewnętrznej CPU (włączając w to program użytkownika oraz wymuszone I/O). Po tym CPU wykonuje program użytkownika z karty programu.

- Karta może zostać także użyta do zapisu plików CSV rejestratora uruchomionego w sterowniku S7-1200 (strona 109). Karta programu zapewnia więcej pamięci niż w wewnętrznej pamięci CPU. Funkcja Web server (strona 186) pozwala na ściąganie plików dzienników zdarzeń do komputera.
- Kartę pamięci można również użyć do wykonania aktualizacji oprogramowania sprzętowego (*firmware*). Więcej informacji na ten temat znajduje się w podręczniku systemu S7-1200.

Uwaga

Karta programowa musi pozostać w sterowniku. Jeśli zostanie ona usunięta, CPU przejdzie w tryb STOP.

4.4.1. Typy danych obsługiwane przez S7-1200

Stosuje się różne typy danych, które cechują się z jednej strony rozmiarem, a z drugiej sposobem interpretacji danych. Każdy parametr instrukcji jest daną co najmniej jednego typu, a niektóre parametry mogą przyjmować jeden z kilku typów danych. Jeżeli kursor zostanie przytrzymany nad polem parametru instrukcji, to zostanie wyświetlona informacja jaki typ danych może przyjmować ten parametr.

Tabela 4.3. Typy danych obsługiwane przez S7-1200

Typy danych	Opis
Bit i sekwencja bitów	<ul style="list-style-type: none"> • Bool jest wartością logiczną lub wartością bitu. • Byte jest 8-bitową wartością bajtu. • Word (Słowo) jest wartością 16-bitową. • DWord jest 32-bitową wartością podwójnego słowa.
Integer	<ul style="list-style-type: none"> • USInt (8-bitowa liczba całkowita bez znaku) oraz SInt (8-bitowa liczba całkowita ze znakiem) są „krótkimi” liczbami całkowitymi (8 bitów lub jeden bajt pamięci), które mogą posiadać lub nie posiadać znaku. • UInt (16-bitowa liczba całkowita bez znaku) oraz Int (16-bitowa liczba całkowita ze znakiem) są liczbami całkowitymi (16 bitów lub jedno słowo pamięci), które mogą posiadać lub nie posiadać znaku. • UDInt (32-bitowa liczba całkowita bez znaku) oraz DInt (32-bitowa liczba całkowita ze znakiem) są liczbami całkowitymi w formacie double (32 bity lub jedno podwójne słowo pamięci), które mogą posiadać lub nie posiadać znaku.
Real	<ul style="list-style-type: none"> • Real – liczby rzeczywiste (lub zmiennoprzecinkowe) są reprezentowane przez 32 bity. • LReal – liczby rzeczywiste (lub zmiennoprzecinkowe) są reprezentowane przez 64 bity.

Typy danych	Opis
Data i czas	<ul style="list-style-type: none"> • Date jest 16-bitową wartością daty (podobną do UInt), która zawiera liczbę dni od 1 stycznia 1990. Maksymalna wartość daty to 65 378 (16#FF62), co oznacza 31 grudnia 2169. • DTL (<i>date and time long</i>) jest 12-bitową strukturą, która zapisuje informacje o dacie i czasie w predefiniowanej strukturze: <ul style="list-style-type: none"> – Rok (UInt): 1970 do 2554, – Miesiąc (USInt): 1 do 12, – Dzień (USInt): 1 do 31 – Godziny (USInt): 0 do 23, – Minuty (USInt): 0 do 59, – Sekundy (USInt): 0 do 59, – Nanosekundy (UDInt): od 0 do 999 999 999. • Time jest 32-bitową reprezentacją wartości czasu IEC (podobnie do DInt), która zbiera liczbę milisekund (od 0 do 24 dni 20 godzin 31 minut 23 sekund i 674 ms). Wszystkie możliwe wartości czasu są poprawne. Wartości te mogą być użyte w obliczeniach i mogą przyjmować wartości ujemne. • TOD (<i>time of day</i>) jest 32-bitową reprezentacją wartości czasu (podobnie do DInt), która zbiera liczbę milisekund od północy (od 0 do 86 399 999).
Char oraz string	<ul style="list-style-type: none"> • Char jest 8-bitowym pojedynczym znakiem. • String – typ danych łańcuchowych, zmienna obsługująca do 254 znaków.
Tablice i struktury danych	<ul style="list-style-type: none"> • Tablice zawierają wiele elementów tego samego typu danych. Tablice mogą być tworzone w edytorze interfejsu bloków OB, FC, FB i DB. Nie można natomiast tworzyć tablicy w edytorze zmiennych PLC. • Struct definiuje strukturę danych zawierającą inne typy danych. Ten typ danych może być użyty do obsługi grupy danych zależnych od siebie procesów jako pojedynczej jednostki danych. Użytkownik może zdefiniować własną strukturę składającą się z wybranych typów danych. <p>Tablice i struktury mogą być łączone w większe struktury. Struktura może być zagnieżdżona do 8 poziomów. Przykładowo można stworzyć strukturę ze struktur, które zawierają tablice.</p>
PLC	<p>PLC – typ danych zdefiniowanych przez strukturę utworzoną przez użytkownika, która definiuje często używaną strukturę danych, którą można użyć wielokrotnie w programie. Podczas tworzenia tego typu danych pojawia się nowa pozycja w liście rozwijalnej wyboru typu danych, jak również w edytorze DB i edytorze interfejsu kodu bloku.</p> <p>Ten typ danych może być użyty jako szablon do wielokrotnego tworzenia globalnych bloków danych, które używają tych samych struktur danych.</p>

Typy danych	Opis
Pointer	<ul style="list-style-type: none"> • Pointer – zapewnia pośrednie odniesienie do adresu zmiennej. Zajmuje 6 bajtów (48 bitów) pamięci i może zawierać informacje o następujących zmiennych: numer DB (0 jeśli dane nie są przechowywane w DB), obszar pamięci CPU i adresy pamięci. • Pointery zapewniają pośredni dostęp do początku obszaru pamięci i rozpoznają jego długość. Każdy Pointer używa 10 bajtów pamięci i może zawierać następujące informacje: typ danych w elementach danych, liczbę elementów danych, obszar pamięci lub numer DB i dostęp do bitu początkowych adresów danych (zwany również adresowaniem byte.bit). • Zmienna pozwala na pośredni dostęp do zmiennych danych różnego typu lub parametrów. Zmienna rozpoznaje strukturę i pojedyncze jej składniki. Zmienna nie zajmuje miejsca w pamięci.

Mimo, że nie jest dostępny jako osobny typ danych, następujący format numeryczny BCD jest akceptowany przez instrukcje konwersji:

- BCD16 jest reprezentowany na 16 bitach (od –999 to 999).
- BCD32 jest reprezentowany na 32 bitach (od –9 999 999 to 9 999 999).

4.4.2. Adresowanie obszarów pamięci

Step 7 ułatwia symboliczne programowanie. Można stworzyć symboliczne nazwy lub „zmiennie” dla adresów danych, niezależnie czy zmiennie (tagi) są związane z adresami pamięci i punktami I/O, czy też z lokalnymi zmiennymi użytymi w kodzie bloku. Zmiennie można wykorzystywać podczas programowania, podając nazwę zmiennej jako parametr instrukcji. Aby lepiej zrozumieć jak CPU zarządza obszarami pamięci, w kolejnych częściach przedstawiono w jaki sposób podawać argument bezwzględny, który odnosi się do zmiennych CPU. Sterownik dysponuje kilkoma sposobami przechowywania danych podczas wykonywania programu użytkownika:

- Pamięć globalna: CPU dzieli pamięć na różne specjalizowane obszary – wejścia (I), wyjścia (Q), pamięci bitowej (M). Program użytkownika ma nieograniczony dostęp do danych przechowywanych w tych obszarach.
- Bloki danych (DB): DB mogą być wykorzystywane w programie użytkownika do przechowywania danych. Dane przechowywane w DB nie są wymazywane, gdy blok danych zostaje zamknięty, lub gdy blok kodu skojarzony z tym blokiem danych zostanie zakończony. Globalne DB przechowują dane, które mogą być wykorzystywane przez wszystkie inne bloki. Bloki danych instance DB przechowują dane dla określonych FB i ich struktura jest zgodna z parametrami używanymi przez FB.
- Pamięć tymczasowa: za każdym razem gdy wywoływany jest blok kodu, system operacyjny CPU alokuje tymczasową – lokalną – pamięć (L), która jest wykorzystywana podczas wykonywania tego bloku. Po zakończeniu wykonywania tego bloku kodu, CPU realokuje pamięć lokalną na potrzeby innych bloków.

Różne obszary pamięci mają swoje unikalne adresy. Program użytkownika wykorzystuje te adresy w celu uzyskania dostępu do informacji przechowywanych w tych miejscach pamięci.

Odwołania do obszaru wejść (I) lub wyjść (Q), takie jak I0.3 i Q1.7, realizują dostęp do obrazu procesu. W celu dostępu do fizycznego wejścia lub wyjścia do odwołania należy dodać „:P” (na przykład: I0.3:P, Q1.7:P, lub „Stop:P”).

Wymuszanie nadpisania stałej wartości jest możliwe tylko dla fizycznego wejścia (Ix.y:P) lub fizycznego wyjścia (Qx.y:P). Aby wymusić stan wejścia lub wyjścia należy dołączyć „:P” do znacznika PLC lub do adresu. Więcej informacji znajduje się w dziale „Wymuszanie stanu zmiennych w CPU” (strona 257).

Tabela 4.4. Obszary pamięci

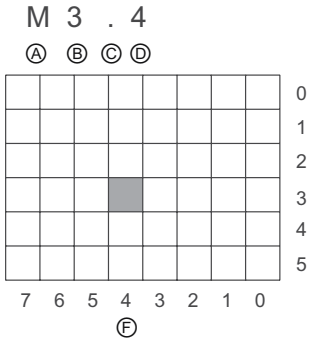
Obszar pamięci	Opis	Wymuszony	Trwały
I obraz procesu – wejście I_:P ¹ (fizyczne wejście)	Skopiowany na początku cyklu programu stan wejść fizycznych.	Nie	Nie
	Bezpośredni odczyt wejściowych punktów fizycznych CPU, SB, SM.	Tak	Nie
Q obraz procesu – wyjście Q_:P ¹ (fizyczne wyjście)	Stan skopiowany na początku cyklu programu do wyjść fizycznych.	Nie	Nie
	Bezpośredni zapis do wyjściowych punktów fizycznych CPU, SB, SM.	Tak	Nie
M pamięć bitowa	Pamięć sterująca i danych.	Nie	Tak (opcjonalnie)
L pamięć chwilowa	Chwilowe dane dla bloku, lokalne dla tego bloku.	Nie	Nie
DB blok danych	Pamięć danych, jak również parametrów dla FB.	Nie	Tak (opcjonalnie)

¹ W celu dostępu (lub wymuszenia) do fizycznego wejścia lub wyjścia do adresu lub nazwy zmiennej należy dodać „:P” (na przykład: I0.3:P, Q1.7:P, lub „Stop:P”).

Różne obszary pamięci mają swoje unikalne adresy. Program użytkownika wykorzystuje te adresy w celu uzyskania dostępu do informacji przechowywanych w tych miejscach pamięci. Bezwzględny adres składa się z:

- Identyfikatora obszaru pamięci (takich jak I, Q, M).
- Rozmiaru danych dostępowych („B” dla bajtów, W dla Słów (Word)).
- Adresu danych (na przykład Byte 3, Word 3).

Aby uzyskać dostęp do pojedynczego bitu w obszarze pamięci należy podać jego adres, który składa się z identyfikatora obszaru pamięci, adresu bajtu i numeru bitu (przykładowo I0.0, Q0.1, lub M3.4).

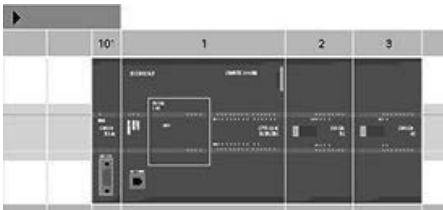


Bezwzględny adres obszaru pamięci:

- A Identyfikator obszaru pamięci
- B Adres bajtu: bajt 3
- C Kropka rozdzielająca adres bajtu od numeru bitu
- D Położenie bitu w bajcie (bit 4 z 8)
- E Bajty obszaru pamięci
- F Bity wybranego bajtu

W tym przykładzie po identyfikatorze obszaru pamięci i adresie bajtu (M = bit obszaru pamięci M; cyfra 3 = bajt 3) podany jest, oddzielony kropką („.”), adres bitu (bit 4).

Konfigurowanie I/O w CPU i modułach I/O



Kiedy na ekranie konfiguracyjnym są dodawane CPU i moduły I/O, to automatycznie są alokowane adresy I oraz Q. Użytkownik może zmienić domyślne adresowanie wybierając w konfiguracji urządzeń pole adresu i wpisując tam nowe liczby.

- Wejściom i wyjściom cyfrowym przypisuje się do grup 8-punktowych (1 bajt), niezależnie od tego, czy moduł ma wszystkie punkty, czy nie.
- Wejścia i wyjścia analogowe tworzą grupy po dwa punkty (4 bajty).

III

Device overview

Module	Slot	I address	Q addr.	Type	Order
	103				
	102				
RS485_1	101			CM 1241 (RS485)	6ES7
PLC_1	1			CPU 1214C DCDCU	6ES7
DI14/DO10	1.1	0...1	0...1	DI14/DO10	
AI2	1.2	64...67		AI2	
AO1 x 12bit	1.3		80...81	AO1 signal board	6ES7
HSC_1	1.16	1000....		High speed counts	
HSC_2	1.17			High speed counts	
HSC_3	1.18			High speed counts	
HSC_4	1.19			High speed counts	
HSC_5	1.20			High speed counts	
HSC_6	1.21			High speed counts	
Pulse_1	1.32			Pulse generator (P	
Pulse_2	1.33			Pulse generator (P	
PROFINET L_X1				PROFINET interface	
DI8 x 24VDC	2	8		SM 1221 DI8 x 24	6ES7

Na rysunku przedstawiono przykładowy CPU 1214C z dwoma SM i jednym SB. W podanym przykładzie, użytkownik może zmienić adres DI8 z 8 na 2. Program asystuje użytkownikowi i zmienia zakres adresów, które mają niewłaściwy rozmiar lub wchodzą w konflikt z innymi adresami.

4.4.3. Uzyskiwanie dostępu do „przedziału danych” o typie danych określonym tagiem

Tagi PLC oraz tagi bloków danych mogą być udostępniane na poziomie bitu, bajtu, lub słowa w zależności od ich wielkości. Przykłady składni kodu uzyskiwania dostępu do takiego przedziału danych są podane poniżej:

- "<nazwa tagu PLC>".xn (dostęp do bitu)
- "<nazwa tagu PLC>".bn (dostęp do bajtu)
- "<nazwa tagu PLC>".wn (dostęp do słowa)
- "<nazwa bloku danych>".<nazwa tagu>.xn (dostęp do bitu)
- "<nazwa bloku danych>".<nazwa tagu>.bn (dostęp do bajtu)
- "<nazwa bloku danych>".<nazwa tagu>.wn (dostęp do słowa)

Tag określający wielkość podwójnego słowa może być udostępniony przez bity z zakresu od 0 do 31, bajty z zakresu od 0 do 3, lub słowa z zakresu od 0 do 1. Tag określający wielkość słowa może być udostępniony przez bity z zakresu od 0 do 15, bajty z zakresu od 0 do 1, lub słowo 0. Tag określający wielkość bajtu może być udostępniony przez bity z zakresu od 0 do 7, lub bajt 0. Przedziały bitów, bajtów, oraz słów mogą być używane wszędzie tam, gdzie bity, bajty lub słowa są oczekiwanymi argumentami.

																BYTE															
																WORD															
DWORD																															
x31	x30	x29	x28	x27	x26	x25	x24	x23	x22	x21	x20	x19	x18	x17	x16	x15	x14	x13	x12	x11	x10	x9	x8	x7	x6	x5	x4	x3	x2	x1	x0
b3								b2								b1								b0							
w1																w0															

Uwaga

Poprawnymi typami danych, które mogą być udostępnione przez przedział danych (slice) są Byte, Char, Conn_Any, Date, DInt, DWord, Event_Any, Event_Att, Hw_Any, Hw_Device, HW_Interface, Hw_Io, Hw_Pwm, Hw_SubModule, Int, OB_Any, OB_Att, OB_Cyclic, OB_Delay, OB_WHINT, OB_PCYCLE, OB_STARTUP, OB_TIMEERROR, OB_Tod, Port, Rtm, SInt, Time, Time_Of_Day, UDIInt, UInt USINT oraz Word. Przez przedział danych mogą być udostępnione Tagi PLC typu Real, ale nie tagi bloku danych typu Real.

Przykłady

W tablicy tagów PLC, „DW” jest zadeklarowanym tagiem typu DWORD. Przykłady pokazują, dostęp do przedziału bitu, bajtu, i słowa:

	LAD	FBD	SCL
Dostęp do bitu			<pre>IF "DW".x11 THEN ... END_IF;</pre>
Dostęp do bajtu			<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
Dostęp do słowa			<pre>out:= "DW".w0 AND "DW".w1;</pre>

4.4.4. Uzyskiwanie dostępu do tagu za pomocą nakładki typu AT

Nakładka tagu typu AT pozwala na dostęp do zadeklarowanego już tagu bloku danych o standardowym dostępie z nałożoną deklaracją innego rodzaju typu danych. Można, na przykład, adresować poszczególne bity tagu typu Byte, Word, lub DWord za pomocą tablicy danych typu boolowskiego.

Deklaracje

Aby nałożyć parametr, należy zadeklarować dodatkowy parametr bezpośrednio po parametrze, który ma być nałożony i wybrać typ danych „AT”. Edytor utworzy nakładkę, i następnie można wybrać typ danych, Struct, lub Array, który ma być użyty do nakładki.

Przykład

Ten przykład pokazuje parametry wejściowe FB o standardowym dostępie. Tag B1 typu bajt (Byte) jest nałożony tablicą danych typu boolowskiego:

■	B1	Byte
▼	AT	AT "B1" Array [0..7] of Bool
■	AT[0]	Bool
■	AT[1]	Bool
■	AT[2]	Bool
■	AT[3]	Bool
■	AT[4]	Bool
■	AT[5]	Bool
■	AT[6]	Bool
■	AT[7]	Bool


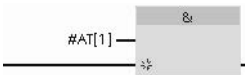
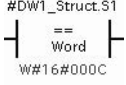
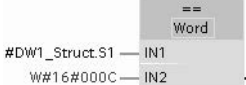

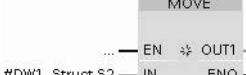
Tabela 4.5. Przykład zastąpienia bajtu tablicą Bool

7	6	5	4	3	2	1	0
AT[0]	AT[1]	AT[2]	AT[3]	AT[4]	AT[5]	AT[6]	AT[7]

Innym przykładem jest tag typu DWord nałożony za pomocą typu Struct:

■ DW1	DWord
▼ DW1_Struct	AT "DW1" Struct
■ S1	Word
■ S2	Byte
■ S3	Byte

Typy nakładek mogą być adresowane bezpośrednio w programie:

LAD	FBD	SCL
		<pre>IF #AT[1] THEN ... END_IF;</pre>
		<pre>IF (#DW1_Struct.S1 = W#16#000C) THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.S2;</pre>

Zasady

- Nakładanie tagów jest możliwe tylko w blokach FB i FC o standardowym dostępie.
- Parametry można nałożyć dla wszystkich typów bloków i we wszystkich sekcjach deklaracji.
- Nałożony parametr może być używany jak każdy inny parametr bloku.
- Nie można nakładać parametrów typu VARIANT.
- Wielkość nakładającego parametru musi być mniejsza lub równa wielkości parametru nakładanego.
- Zmienna nakładająca musi być zadeklarowana bezpośrednio po zmiennej, na której jest nakładana i oznaczona za pomocą słowa kluczowego „AT”.

4.5. Wyjścia impulsowe

W CPU bądź w płytce sygnałowej (SB) dostępne są cztery generatory impulsów pozwalające realizować funkcje wymagające ciągu szybkich impulsów: PWM (modulacja szerokości impulsu) i PTO (wyjście ciągu impulsów). PTO jest wykorzystywane przez podstawowe instrukcje sterowania napędami. Użytkownik może przypisać każdy z generatorów impulsów albo do PWM albo do PTO, ale nie do obu jednocześnie.



Wyjścia impulsowe nie mogą być używane przez inne instrukcje w programie. Podczas konfiguracji wyjść na generatory impulsów dla CPU lub SB, odpowiednie adresy wyjściowe są usuwane z obszaru pamięci Q i nie mogą już zostać użyte w programie użytkownika. Jeśli program użytkownika nadpisze te wyjścia, CPU nie przepisze tych wartości do odpowiadających wyjść fizycznych.

Uwaga

Nie przekraczaj maksymalnej częstotliwości impulsów

Maksymalna częstotliwość impulsów generatorów wyjść impulsowych wynosi 1 MHz dla CPU 1217C i 100 kHz dla CPU 1211C, 1212C, 1214C i 1215C; 20 kHz (dla standardowej płytki sygnałowej (SB)); lub 200 kHz (dla szybkiej płytki sygnałowej (SB)).

Cztery generatory impulsowe mają domyślne przypisania I/O; jednakże, mogą być skonfigurowane dla każdego wyjścia cyfrowego CPU lub SB. Generatory impulsów na CPU nie mogą być przypisane do rozproszonych I/O.

Podczas konfiguracji podstawowych instrukcji ruchu, STEP 7 **nie informuje** jeśli któraś z osi z maksymalną prędkością lub częstotliwością przekracza to ograniczenie sprzętowe. Może to powodować problemy z aplikacją użytkownika, więc należy zawsze upewnić się czy nie przekroczono maksymalnej częstotliwości dla danego urządzenia.

Oba generatory impulsów są mapowane do określonych wyjść cyfrowych, tak jak to pokazano w poniższej tabeli. Można wykorzystać w tym celu zarówno wewnętrzne wyjścia CPU, jak i opcjonalne wyjścia płytki sygnałowej. Numery punktów wyjściowych są pokazane w poniższej tabeli (przy założeniu domyślnej konfiguracji wyjść). Jeżeli użytkownik zmieni numerację wyjść, to numerami punktów wyjściowych będą te, które ustalił użytkownik. Niezależnie od tego PTO1/PWM1 wykorzystują dwa pierwsze wyjścia cyfrowe, a wyjścia PTO2/PWM2 wykorzystują kolejne dwa wyjścia cyfrowe CPU albo dołączonej płytki sygnałowej. Zwracamy uwagę na to, że PWM wymaga tylko jednego wyjścia, podczas gdy PTO może opcjonalnie wykorzystywać dwa wyjścia na kanał. Jeżeli wyjście nie jest potrzebne dla impulsów, to jest dostępne dla innych celów.

Cztery generatory impulsowe mają domyślne przypisania I/O; jednakże, mogą być skonfigurowane dla każdego wyjścia cyfrowego CPU lub SB. Generatory impulsów na CPU nie mogą być przypisane do modułów rozszerzeń (SM) lub do rozproszonych I/O.

Tabela 4.6. Domyślne przypisanie wyjść dla generatorów impulsów

Opis	Impuls	Kierunek
PTO1		
Wejścia/wyjścia wbudowane	Q0.0	Q0.1
Wejścia/wyjścia płytki sygnałowej	Q4.0	Q4.1
PWM1		
Wyjścia wbudowane	Q0.0	–
Wyjścia płytki sygnałowej	Q4.0	–
PTO2		
Wejścia/wyjścia wbudowane	Q0.2	Q0.3
Wejścia/wyjścia płytki sygnałowej	Q4.2	Q4.3
PWM2		
Wyjścia wbudowane	Q0.2	–
Wyjścia płytki sygnałowej	Q4.2	–
PTO3		
Wejścia/wyjścia wbudowane	Q0.4 ¹	Q0.5 ¹
Wejścia/wyjścia płytki sygnałowej	Q4.0	Q4.1
PWM3		
Wyjścia wbudowane	Q0.4 ¹	–
Wyjścia płytki sygnałowej	Q4.1	–
PTO4		
Wejścia/wyjścia wbudowane	Q0.6 ²	Q0.7 ²
Wejścia/wyjścia płytki sygnałowej	Q4.2	Q4.3
PWM4		
Wyjścia wbudowane	Q0.6 ²	–
Wyjścia płytki sygnałowej	Q4.3	–

¹ CPU 1211C nie ma wyjść Q0.4, Q0.5, Q0.6 lub Q0.7. Dlatego wyjścia te nie mogą być używane w CPU 1211C.

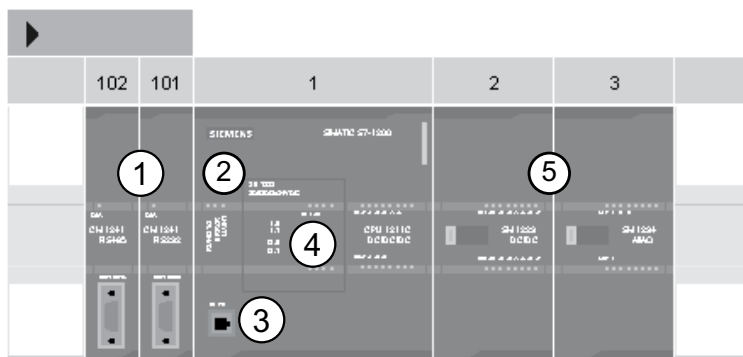
² CPU 1212C nie ma wyjść Q0.6 lub Q0.7. Dlatego wyjścia te nie mogą być używane w CPU 1212C.

³ Ta tabela dotyczy funkcji PTO/PWM CPU 1211C, CPU 1212C, CPU 1214C, oraz CPU 1215C.

Konfiguracja sterownika

5

Konfiguracji systemu PLC dokonuje się dodając do projektu CPU i dodatkowe moduły.



- ① Moduł komunikacyjny (CM) lub processor komunikacyjny (CP): do 3 modułów, instalowane na pozycjach 101,102 i 103.
- ② CPU: na pozycji 1.
- ③ Port ethernetowy CPU.
- ④ Płytki sygnałowa (SB), płytki komunikacyjna (CB), lub płytki baterijna (BB): maksymalnie 1 sztuka, instalowana w CPU.
- ⑤ Moduł rozszerzeń (SM) dla cyfrowych lub analogowych I/O: do 8 modułów, instalowane na pozycjach 2...9 (CPU 1214C, CPU 1215C, oraz CPU 1217C obsługują 8 SM, CPU 1212C obsługuje 2 SM, CPU 1211C nie obsługuje SM).

W celu dokonania konfiguracji urządzenia, należy dodać urządzenie do projektu:

- W widoku portalu należy wybrać *Devices & Networks* i kliknąć *Add device*.
- W widoku projektu należy, pod nazwą projektu, podwójnie kliknąć *Add new device*.



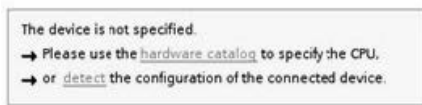
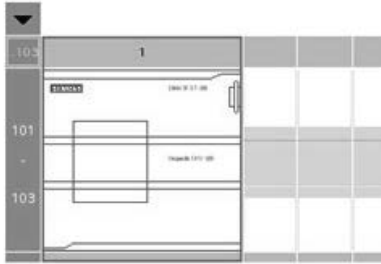
5.1. Rozpoznawanie konfiguracji dla nieokreślonego CPU



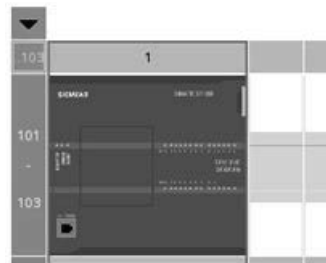
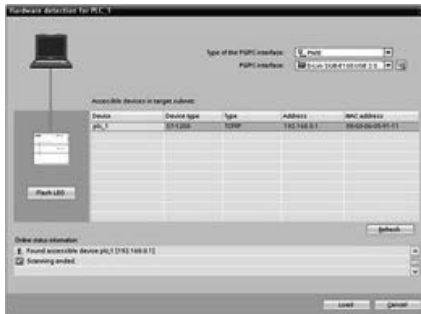
Jeśli uzyskano połączenie z CPU, możliwe jest przesłanie konfiguracji tego CPU do programu użytkownika, włączając w to każdy dołączony moduł. Po stworzeniu nowego projektu należy wybrać opcję *unspecified CPU* (Nieokreślone CPU) zamiast wyboru ściśle określonego CPU. Można także ominąć ten krok w całości wybierając opcję *Create a PLC program* z zakładki *First steps*. W tym wypadku STEP 7 automatycznie wybierze opcję nieokreślonego CPU.

Z poziomu edytora programu należy wybrać komendę *Hardware detection* z menu *Online*.

Z poziomu edytora konfiguracji należy wybrać opcję wykrywania konfiguracji podłączonego urządzenia.

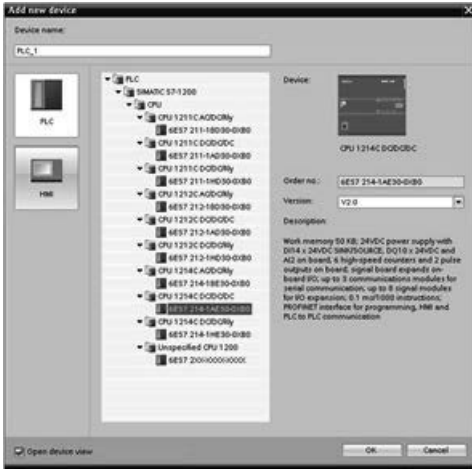


Po wybraniu CPU z okna dialogowego należy nacisnąć przycisk *Load*. STEP 7 prześle sprzętową konfigurację urządzenia z CPU, łącznie z wszelkimi podłączonymi modułami (SM, SB lub CM). Istnieje możliwość konfiguracji parametrów CPU i podłączonych modułów (strona 72).

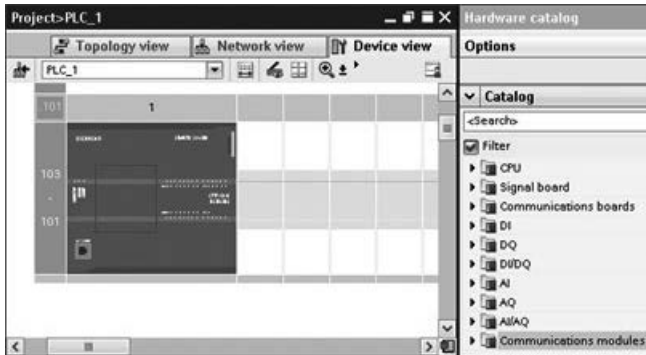


5.2. Dołączanie modułów

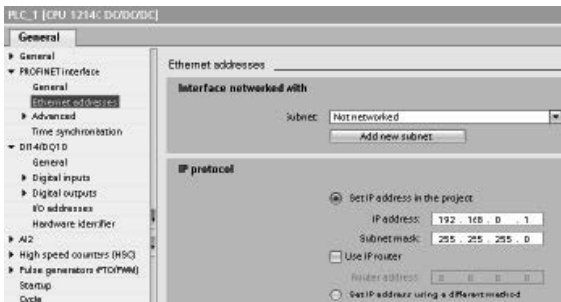
W celu dokonania konfiguracji urządzenia, należy dodać CPU do projektu. Wybranie CPU w oknie dialogowym *Add a new device* powoduje utworzenie wirtualnej listwy montażowej i dodanie na niej CPU.



Okno *Device view* konfiguracji sprzętowej i wirtualnej listwy.



Wybór CPU w oknie *Device view* powoduje wyświetlenie w oknie inspekcyjnym właściwości CPU. Użyj tych właściwości do konfiguracji parametrów pracy CPU (strona 72).



Uwaga

CPU nie ma wstępnie ustawionego adresu IP. Użytkownik musi w trakcie konfiguracji CPU ręcznie wpisać adres IP urządzenia. Jeżeli CPU jest połączone z routerem sieciowym, to należy również wpisać adres IP routera. Adres IP zostanie automatycznie nadany po pierwszym załadowaniu programu do sterownika.

5.3. Zmiana urządzenia

Można zmienić typ urządzenia – skonfigurowanego CPU lub modułu. W konfiguracji urządzeń, należy kliknąć prawym przyciskiem myszy na urządzenie i z menu kontekstowego wybrać polecenie *Change device* (Zmień urządzenie). W oknie dialogowym należy wybrać CPU lub moduł, który ma być zastąpiony. W oknie dialogowym *Change device* (Zmień urządzenie) zostanie wyświetlona informacja na temat kompatybilności dwóch urządzeń.

Uwaga

Wymiana urządzenia: zastąpienie CPU V3.0 przez CPU V4.0

Można otworzyć projekt programu STEP 7 V12 w programie STEP 7 V13 i zastąpić CPU V3.0 przez CPU V4.0. Nie można zastąpić jednostek CPU, które są w wersji wcześniejszej niż V3.0. Po zastąpieniu CPU V3.0 przez CPU V4.0, należy rozważyć różnice (strona 336) funkcji i działania pomiędzy dwoma wersjami oraz czynności, które należy wykonać.

W przypadku projektu zastępowanego CPU w wersji wcześniejszej niż V3.0, najpierw można zaktualizować ten projekt do wersji V3.0, a następnie zaktualizować projekt z wersji V3.0 do wersji V4.0.


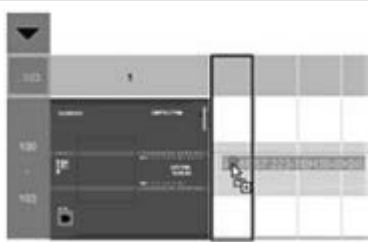


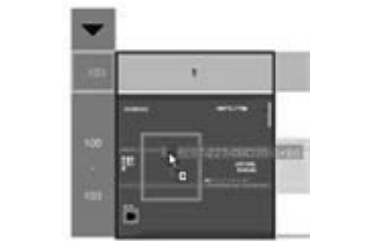


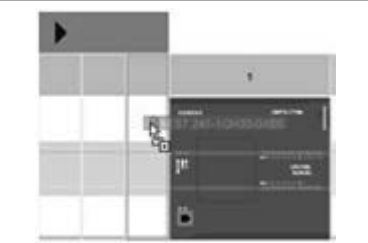
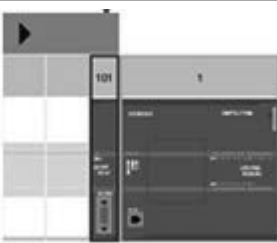
5.4. Dodawanie modułów do systemu

W celu dołączenia modułów do CPU należy się posłużyć katalogiem sprzętu (*hardware catalog*):

- Moduły rozszerzeń (SM) pozwalają zwiększyć liczbę cyfrowych lub analogowych punktów I/O. Są dołączane z prawej strony CPU.
- Płytki sygnałowe (SB) pozwalają dodawać do CPU porty I/O. SB jest dołączana od strony frontowej CPU.
- Płytką bateryjna (BB) 1297 zapewnia długotrwałe podtrzymanie pracy zegara czasu rzeczywistego. Płytką bateryjna jest dołączana od strony frontowej CPU.
- Płytki komunikacyjne (CB) pozwalają dodać do CPU port komunikacyjny (jak np. RS485). CB jest dołączana od strony frontowej CPU.
- Moduły komunikacyjne (CM) i procesory komunikacyjne (CP) zapewniają dodatkowe porty komunikacyjne, jak np. PROFIBUS lub GPRS. Moduły te podłączane są z lewej strony CPU.

W celu dodania modułu do konfiguracji urządzeń należy wybrać moduł z *Hardware catalog* i albo go podwójnie kliknąć, albo przeciągnąć na podświetloną pozycję. Aby moduły funkcjonowały konieczne jest dodanie modułów do konfiguracji urządzeń i załadowanie konfiguracji sprzętowej do CPU.

Tabela 5.1. Dodawanie modułu do projektu

Moduł	Wybór modułu	Instalacja modułu	Rezultat
SM			
SB, BB, lub CB			
CM lub CP			

5.5. Konfiguracja parametrów pracy CPU i modułów

W celu skonfigurowania parametrów operacyjnych CPU, należy w oknie *Device view* wybrać CPU, a następnie zakładkę *Properties* (właściwości) okna inspekcyjnego.



Można skonfigurować następujące właściwości procesora:

- **PROFINET interface:** Ustawienie adresu IP CPU i synchronizacji czasu.
- **Startup:** Ustalanie sposobu działania CPU po przejściu ze stanu wyłączenia do stanu włączenia.
- Konfiguracja lokalnych (wbudowanych do CPU) cyfrowych i analogowych I/O, szybkich liczników (HSC) oraz generatorów impulsowych
- **System clock:** Ustawianie czasu, strefy czasowej i czasu letniego.
- Ustawianie zabezpieczenia przed odczytem/zapisem oraz hasła dostępu do CPU.
- Określanie maksymalnego czasu cyklu lub ustalonego minimalnego czasu cyklu i przypisywanie procentowej części czasu cyklu zadaniom komunikacyjnym.
- Właściwości web serwera.

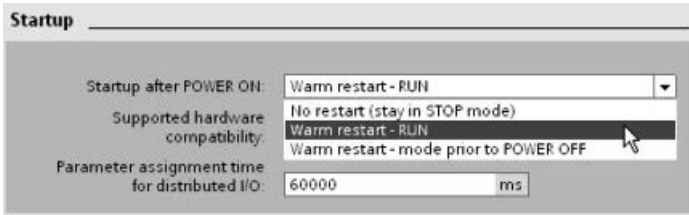
Czynności wykonywane przez CPU przy przejściu z trybu pracy STOP do RUN

Zawsze jak tryb pracy zmienia się ze STOP na RUN, CPU kasuje obszar wejściowy obrazu procesu, inicjalizuje obszar wyjściowy obrazu procesu i wykonuje startowe OB. (Dlatego też dowolna operacja odczytu obszaru wejściowego obrazu procesu wykonana instrukcjami zawartymi w startowym OB da w wyniku same zera, a nie rzeczywiste stany na wejściach fizycznych). W celu odczytania bieżących stanów wejść fizycznych podczas uruchomienia, należy wykonać odczyt bezpośredni. Startowe OB i dowolne powiązane FC i FB są wykonywane w następnej kolejności. Jeżeli istnieje więcej niż jeden startowy OB, to wszystkie są wykonywane zgodnie ze swoimi numerami, przy czym jako pierwszy jest wykonywany OB z najniższym numerem. W trybie STARTUP CPU wykonuje również następujące zadania:

- Przerwania są ustawiane w kolejce, ale w fazie rozruchowej nie są wykonywane.
- W fazie rozruchowej nie jest monitorowany czas cyklu.
- Podczas startu mogą być wykonywane zmiany konfiguracji modułów HSC (szybki licznik), PWM (modulator szerokości impulsów) i PtP (komunikacji punkt-punkt).
- Moduły HSC, PWM i komunikacji PtP mogą pracować tylko w trybie RUN.

Po zakończeniu startowych OB, CPU przechodzi w tryb RUN i wykonuje zadania sterowania w ciągłym cyklu programu.

Konfigurowanie sposobu działania CPU w trakcie startu po podłączeniu zasilania jest dokonywane za pomocą atrybutów CPU.



- W trybie STOP
- W trybie RUN
- W poprzednim trybie (przed wyłączeniem zasilania)

Przed wejściem w tryb RUN, CPU wykonuje gorący restart. W trakcie gorącego restartu kasowane są do wartości domyślnych wszystkie pamięci, które nie są trwałe, ale zachowywane są bieżące wartości przechowywane w pamięciach trwałych.

Uwaga

CPU przeprowadza restart po każdym załadowaniu

Przy każdym załadowaniu jakiegokolwiek elementu (takiego jak blok programu, blok danych lub konfiguracja sprzętowa), CPU przeprowadza restart przy następnym przejściu w tryb RUN. Dodatkowo, w celu wyczyszczenia wejść, inicjalizacji wyjść i pamięci nietrwałych, restart inicjuje obszar pamięci trwałych.

Po restarcie wywołanym załadowaniem nowego elementu kolejne przejście z trybu STOP do RUN powoduje gorący restart, który nie inicjuje pamięci trwałej.

5.5.1. Pamięć systemu i zegara jako standardowa funkcjonalność

Bajty „pamięci systemu” i „pamięci zegara” uaktywnia się we właściwościach CPU. Program użytkownika ma dostęp do pojedynczych bitów tych funkcji za pomocą ich nazw tagów.

- Użytkownik może przeznaczyć jeden bajt pamięci M na pamięć systemu. Bajt pamięci systemu zawiera cztery bity, do których ma dostęp program użytkownika za pomocą wymienionych poniżej nazw tagów:
 - Bit *First scan* (nazwa tagu „FirstScan”) jest ustawiony na 1 przez czas trwania pierwszego cyklu programu po zakończeniu rozruchowego OB. (Po wykonaniu pierwszego cyklu programu bit „first scan” jest ustawiany na 0.)
 - Bit *Diagnostic graph changed* (nazwa tagu: „DiagStatusUpdate”) jest ustawiony na 1 przez jeden cykl programu po tym, jak CPU zarejestruje zdarzenie diagnostyczne. Ponieważ procesor nie ustawia bitu *diagnostic graph changed* do końca pierwszego wykonania OB cyklu programu, program użytkownika nie może wykryć, czy doszło do zmiany diagnostyki w trakcie wykonywania rozruchowego OB, czy w trakcie wykonywania OB cyklu programu.

- Bit *Always 1 (high)* (nazwa tagu „AlwaysTRUE”) jest zawsze ustawiony na 1.
- Bit *Always 0 (low)* (nazwa tagu „AlwaysFALSE”) jest zawsze ustawiony na 0.
- Użytkownik może przeznaczyć jeden bajt pamięci M na pamięć zegara. Każdy bit bajtu skonfigurowanego jako pamięć zegara generuje ciąg impulsów prostokątnych. Jest dostępnych osiem częstotliwości fali prostokątnej generowanej przez bajt pamięci zegara; od 0,5 Hz (wolno) do 10 Hz (szybko). Użytkownik może wykorzystywać te bity jako bity sterujące, zwłaszcza z instrukcjami dotyczącymi zbroczy, w celu cyklicznego wyzwalania akcji w programie użytkownika.

CPU inicjalizuje te bajty podczas przełączenia z trybu STOP w tryb STARTUP. Bity pamięci zegara zmieniają swoje stany synchronicznie z sygnałem zegarowym CPU w trybach STARTUP i RUN.



Ostrzeżenie

Zagrożenia wynikające z nadpisania bitów pamięci systemu lub pamięci zegara

Nadpisanie bitów pamięci systemu lub pamięci zegara może uszkodzić dane w tych funkcjach i spowodować, że program użytkownika będzie działał nieprawidłowo, co z kolei może doprowadzić do zniszczenia sprzętu i obrażeń ciała personelu.

Ponieważ zarówno pamięć zegara, jak i pamięć systemu nie są zarezerwowanymi obszarami pamięci M, więc instrukcje i komunikaty mogą wpisywać dane do tych miejsc i zniszczyć istniejące tam dane.

Aby zapewnić poprawne działanie tych funkcji, należy unikać zapisywania danych do tych miejsc i zawsze instalować wyłącznik bezpieczeństwa procesu lub maszyny.

System memory bits

Enable the use of system memory byte

Address of system memory byte (MBx):

First cycle:

Diagnostics status changed:

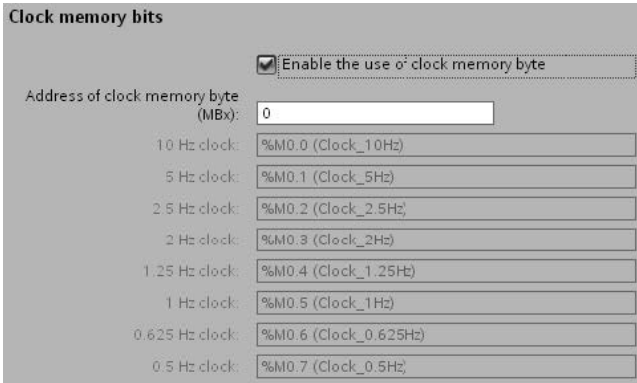
Always 1 (high):

Always 0 (low):

Pamięć systemu konfiguruje jeden bajt, włączając (tj. ustawiając wartość 1) jego bity dla określonego zdarzenia.

Tab. 5.2. Pamięć systemowa

7	6	5	4	3	2	1	0
Zarezerwowane Wartość 0				Zawsze wyłączony Wartość 0	Zawsze włączony Wartość 1	Diagnostyczny wskaźnik stanu • 1: Zmiana • 0: Nie ma zmiany	Wskaźnik pierwszego cyklu programu • 1: Pierwszy cykl programu po rozruchu • 0: Nie pierwszy cykl programu



Pamięć zegara ma tak skonfigurowany bajt, że pojedyncze bity na przemian włączają się i wyłączają z ustaloną częstotliwością. Każdy bit zegara generuje falę prostokątną na odpowiedniej pozycji pamięci M. Te bity mogą być wykorzystane jako bity sterujące, zwłaszcza z instrukcjami dotyczącymi zbcocy, w celu cyklicznego wyzwalania akcji w kodzie użytkownika.

Tab. 5.3. Pamięć zegarowa

Numer bitu	7	6	5	4	3	2	1	0
Nazwa tagu								
Okres [s]	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Częstotliwość [Hz]	0,5	0,625	1	1,25	2	2,5	5	10

Ponieważ pamięć zegara działa asynchronicznie w stosunku do cyklu CPU, stan pamięci zegara może zmieniać się kilka razy w ciągu długiego cyklu.

Konfiguracja pracy modułów I/O oraz modułów komunikacyjnych

W celu skonfigurowania parametrów operacyjnych modułów sygnałowych (SM), płytek sygnałowych (SB) lub modułów komunikacyjnych (CM), należy w oknie *Device view* wybrać moduł, a następnie zakładkę *Properties* (Właściwości) okna inspekcyjnego.



Moduły sygnałowe (SM) i płytki sygnałowe (SB)

- Digital I/O: Konfigurowanie indywidualnych wejść, takich jak wykrywanie zbcocy i „pulse catch“ (decyduje o pozostaniu w stanie niskim lub wysokim na jeden cykl po chwilowym niskim lub wysokim impulsie). Konfigurowanie wyjść w celu uaktywnienia funkcji zachowania lub podstawienia wartości wyjściowej indywidualnego wyjścia, przy zmianie trybu z RUN na STOP.
- Analog I/O: Konfigurowanie parametrów indywidualnych wyjść (takich jak sygnał napięciowy/prądowy, zakres i wygładzenie) i uruchomienie diagnostyki przekroczenia zakresu od góry i od dołu. Konfigurowanie parametrów indywidualnych wyjść a także uaktywnianie diagnostyki, takiej jak badanie zwarcia (przy wyjściu napięciowym) lub przekroczenia zakresu.
- Adresowanie I/O: Konfigurowanie adresu początkowego wejść i wyjść modułu.



Moduły komunikacyjne (CM) i płytki komunikacyjne (CB)

- *Port configuration*: Konfigurowanie parametrów komunikacyjnych, takich jak szybkość transmisji, parzystość, bity danych i czas oczekiwania.

- *Transmit and receive message*: konfigurowanie opcji związanych z nadawaniem i odbieraniem danych (takich jak parametry początku wiadomości i końca wiadomości).

Wszystkie opisane tutaj parametry konfiguracyjne mogą być zmieniane również w programie użytkownika.

5.6. Konfiguracja adresu IP dla CPU

Ponieważ CPU nie ma wstępnie ustawionego adresu IP, to użytkownik musi w trakcie konfiguracji CPU ręcznie wpisać ten adres.

- Każde urządzenie znajdujące się w sieci PROFINET ma nadany przez producenta adres MAC (*Media Access Control*) pozwalający je zidentyfikować. Każde urządzenie musi również mieć adres protokołu internetowego (IP).
- Podsieć (subnet) tworzą logicznie pogrupowane urządzenia podłączone do sieci. Maskę (mask), zwana również maską podsieci lub maską sieci, określa granice IP podsieci. Jedyne połączenie pomiędzy różnymi podsieciami możliwe jest poprzez router. Routery są łączami pomiędzy sieciami lokalnymi LAN. Podczas odbierania i wysyłania pakietów danych routery posługują się adresami IP.

Przed załadowaniem adresu IP do CPU należy upewnić się, czy adres IP dla CPU jest kompatybilny z tym wpisanym do urządzenia programującego.

W celu nadania adresu IP można użyć STEP 7 i postępować zgodnie z następującą procedurą:

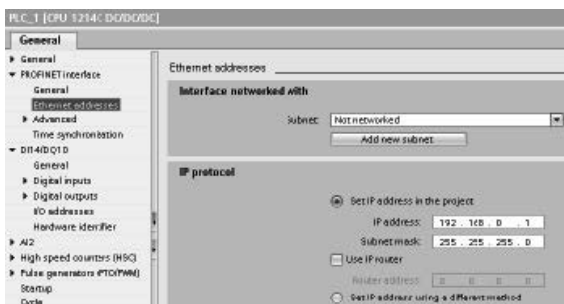
1. Rozwinąć folder *Online access* w drzewie projektu, aby wyświetlić sieci użytkownika.
2. Wybrać sieć, w której znajduje się sterownik.
3. Kliknąć PPM na wybranej sieci dla wyświetlenia menu kontekstowego.
4. Wybrać komendę *Properties*.

Uwaga

Adres IP dla fizycznego CPU musi odpowiadać adresowi IP i masce podsieci urządzenia programującego. Należy skonsultować się z specjalistą ds. sieci w celu wybrania odpowiedniego adresu IP i maski podsieci dla danego CPU.



Okno *Properties* wyświetla ustawienia urządzenia programującego.



Po ustaleniu adresu IP i maski podсети należy wpisać go dla CPU i dla routera (jeśli jest). Więcej informacji znajduje się w *S7-1200 System Manual*.



Po skończeniu konfiguracji należy załadować projekt do CPU.

Adresy IP dla CPU i dla routera (jeśli jest) są skonfigurowane po załadowaniu projektu

5.7. Ochrona dostępu do CPU i kodu programu

CPU zapewnia 4-poziomą ochronę przed niepowołanym dostępem do pewnych funkcji. Użytkownik podczas konfigurowania poziomu bezpieczeństwa i hasła CPU, ogranicza funkcje i obszary pamięci, do których można mieć dostęp bez podania hasła.

Na każdym poziomie zabezpieczenia można ustalić funkcje, które są dostępne bez podania hasła. Warunkami domyślnymi jest brak ograniczeń i zabezpieczenia

hasłem dostępu do CPU. Aby wprowadzić zabezpieczenia należy skonfigurować własności CPU i zdefiniować hasło.

Wprowadzenie hasła przez sieć nie zagraża ochronie CPU hasłem. Ochrona hasłem nie obejmuje wykonywanie instrukcji programu użytkownika oraz funkcje komunikacyjne. Wprowadzenie poprawnego hasła umożliwia dostęp do wszystkich funkcji na danym poziomie.

Komunikacja PLC-PLC (za pomocą instrukcji komunikacyjnych w bloku kodu) nie jest ograniczona przez poziom zabezpieczenia CPU. Nie jest również ograniczona funkcjonalność HMI.

Tabela 5.4. Poziomy zabezpieczeń CPU

Poziom bezpieczeństwa	Ograniczenia dostępu
Pełen dostęp (brak zabezpieczenia)	Bez podania hasła dozwolony jest pełen dostęp.
Dostęp do odczytu	Bez podania hasła dozwolony dostęp do HMI oraz komunikacja PLC-PLC. Hasło jest wymagane dla modyfikacji (zapisu do) CPU i zmiany trybu pracy CPU (RUN/STOP).
Dostęp do HMI	Bez podania hasła dozwolony jest dostęp do HMI oraz pełna komunikacja PLC-PLC. Hasło jest wymagane dla odczytu danych z CPU, modyfikacji (zapisu do) CPU i zmiany trybu pracy CPU (RUN/STOP).
Brak dostępu (pełne zabezpieczenie)	Bez podania hasła nie ma dostępu. Hasło jest wymagane dla dostępu do HMI, odczytu danych z CPU, modyfikacji (zapisu do) CPU i zmiany trybu pracy CPU (RUN/STOP).

Podczas wpisywania hasła jest różniana wielkość liter. Aby skonfigurować hasło należy wykonać następujące kroki:

1. W oknie *Device configuration* należy wybrać CPU.
2. W oknie *Inspector* należy wybrać zakładkę *Properties*.
3. Wybrać właściwość *Protection*, aby skonfigurować poziom zabezpieczeń i wprowadzić hasło.



Po załadowaniu tej konfiguracji do CPU, użytkownik uzyskuje dostęp do HMI i może mieć dostęp do funkcji HMI bez podania hasła. Dla odczytu danych, użytkownik musi wprowadzić skonfigurowane hasło dla poziomu dostępu *Read Access* (Dostęp do odczytu) lub hasło dla poziomu dostępu *Full access (no protection)* (Pełen dostęp (brak zabezpieczenia)). Dla zapisu danych, użytkownik musi wprowadzić skonfigurowane hasło dla poziomu dostępu *Full access (no protection)* (Pełen dostęp (brak zabezpieczenia)).

Ostrzeżenie

Nieautoryzowany dostęp do zabezpieczonego CPU

Użytkownicy z uprawnieniami pełnego dostępu do CPU mają uprawnienia do odczytu i zapisu zmiennych PLC. Niezależnie od poziomu dostępu do CPU, użytkownicy web serwera mogą mieć uprawnienia do odczytu i zapisu zmiennych PLC. Nieautoryzowany dostęp do CPU lub zmiana wartości zmiennych PLC na nieprawidłowe może zakłócić działanie procesu, i może spowodować śmierć, poważne obrażenia ciała i/lub uszkodzenie mienia.

Autoryzowani użytkownicy mogą wykonywać zmiany w trybie pracy, wprowadzać zapisy do danych PLC i aktualizować firmware. Firma Siemens zaleca przestrzegać następujących zasad bezpieczeństwa:

- Zabezpieczyć poziomy dostępu do CPU oraz identyfikatory ID użytkowników web serwera za pomocą silnych haseł. Silne hasła, czyli trudne do odgadnięcia przez inne osoby, składają się z co najmniej ośmiu znaków, w tym pomieszczone litery, cyfry i znaki specjalne nie tworzące słów, które można znaleźć w słowniku, ani nazw i identyfikatorów, które mogą być uzyskane z informacji osobistych. Hasła należy utrzymywać w tajemnicy i często je zmieniać.
- Umożliwić dostęp do web serwera tylko za pośrednictwem protokołu HTTPS.
- Nie rozszerzać domyślnych, minimalnych uprawnień użytkownika *Everybody* (Każdy) web serwera.
- Wykonywać testy na występowanie błędów oraz sprawdzać poprawność zakresów wartości dla zmiennych w programie, ponieważ użytkownicy stron internetowych mogą zmienić wartości zmiennych PLC na niepoprawne.

Mechanizmy połączeń

Aby uzyskać dostęp do zdalnych partnerów połączenia za pomocą instrukcji PUT/GET, użytkownik musi mieć uprawnienia.

Domyślnie opcja *Permit access with PUT/GET communication* (Pozwolenie na dostęp do komunikacji za pomocą instrukcji PUT/GET) jest wyłączona. W tym przypadku, dostęp do odczytu i zapisu danych CPU jest możliwy tylko dla połączeń komunikacyjnych, które wymagają konfiguracji lub programowania, zarówno lokalnego (Local) CPU, jak i partnerskiego (Partner) CPU. Dostęp, na przykład, za pomocą instrukcji BSEND/BRCV jest możliwy.

Połączenia, w których lokalny CPU jest tylko serwerem (co oznacza, że dla lokalnego CPU nie ma możliwości konfiguracji/programowania komunikacji z partnerem komunikacyjnym), nie są możliwe podczas pracy CPU. Dotyczy to na przykład:

- dostępu za pomocą instrukcji PUT/GET, FETCH/WRITE lub FTP poprzez moduły komunikacyjne,
- dostępu za pomocą instrukcji PUT/GET z innych jednostek CPU S7,
- dostępu do HMI poprzez komunikację za pomocą instrukcji PUT/GET.

Aby umożliwić dostęp do danych CPU od strony klienta i tym samym nie ograniczać usług komunikacyjnych CPU, należy wykonać następujące kroki:

1. Skonfigurować poziom zabezpieczenia dostępu tak, aby był inny niż *No access (complete protection)* (Brak dostępu (pełne zabezpieczenie)),
2. Zaznaczyć pole wyboru *Permit access with PUT/GET communication* (Pozwolenie na dostęp do komunikacji za pomocą instrukcji PUT/GET).



Po załadowaniu tej konfiguracji do CPU, CPU umożliwia komunikację ze zdalnymi partnerami za pomocą instrukcji PUT/GET.

5.7.1. Zabezpieczenie „know-how”

Możliwość zabezpieczenia wiedzy przed skopiowaniem pozwala użytkownikowi ograniczyć nieautoryzowany dostęp do jednego lub wielu bloków kodu (OB, FB lub FC) lub bloków danych (DB) w swoim programie. W celu ograniczenia dostępu do bloku kodu użytkownik ustala hasło. Ochrona hasłem zabezpiecza przed nieautoryzowanym odczytem lub modyfikowaniem bloku kodu. Bez podania hasła można odczytać wyłącznie następujące informacje dotyczące bloku kodu:

- Nazwę bloku, komentarz i właściwości bloku.
- Parametry transferowe (IN, OUT, IN_OUT, Return).
- Strukturę wywołującą program.
- Globalne zmienne w odsyłaczach (bez informacji gdzie są używane); zmienne lokalne są ukryte.

Jeśli blok został zabezpieczony przed niepowołanym dostępem („know-how” protection), to zawartość bloku można odczytać tylko po podaniu hasła.

Należy użyć zakładki *Properties* dla skonfigurowania ochrony „know-how” danego bloku. Po otwarciu bloku należy wybrać pole *Protection* we właściwościach.



1. Naciśnij przycisk *Protection* we właściwościach bloku kodu, aby uzyskać dostęp do okna dialogowego *Know-how protection*.
2. Naciśnij przycisk *Define* w celu wprowadzenia hasła.

Po wprowadzeniu i potwierdzeniu hasła naciśnij *OK*.



5.7.2. Zabezpieczenie przed kopiowaniem

Dodatkowe zabezpieczenia umożliwiają powiązanie bloków programu z określonym CPU lub kartą pamięci. Ta możliwość jest szczególnie użyteczna w ochronie własności intelektualnej użytkownika. Po związaniu bloku programu z określonym urządzeniem, możliwe jest jego uruchomienie tylko na określonym urządzeniu lub karcie pamięci. Pozwala to na elektroniczną dystrybucję programu lub bloku kodu (przez Internet lub e-mail) lub też poprzez przesłanie karty pamięci. Zabezpieczenie przed kopiowaniem jest dostępne dla bloków organizacyjnych (OB), bloków funkcyjnych (FB), and funkcji (FC).

Należy użyć zakładki *Properties* w bloku kodu, aby powiązać ten blok z określonym CPU lub kartą pamięci.

1. Po otwarciu bloku kodu wybierz *Protection*.



2. Z listy rozwijalnej pod zakładką *Copy protection* wybierz opcje powiązania kodu z kartą pamięci lub określonym CPU.

The screenshot shows a configuration window with two sections. The top section, titled "Know-how protection", contains a text box with the message "The block is not protected." and a button labeled "Protection". The bottom section, titled "Copy protection", features a dropdown menu currently set to "Bind to serial number of the CPU". Below the dropdown are two radio button options: "Serial number inserted when downloading to a device or a memory card." (which is selected) and "Enter serial number." followed by an empty text input field.

3. Wybierz rodzaj zabezpieczenia przed kopiowaniem i wprowadź numer seryjny dla karty pamięci lub CPU.

Uwaga

W numerze seryjnym ważna jest wielkość liter!

Koncepcja programowania

6.1. Projektowanie programu użytkownika

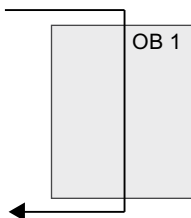
Podczas tworzenia programu użytkownika dla sterownika S7-1200 używa się instrukcji zamieszczanych w blokach kodu (OB., FB lub FC).

Wybór typu struktury programu użytkownika

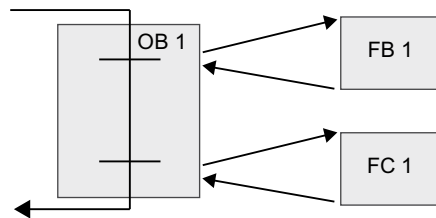
W oparciu o wymagania aplikacji użytkownik podczas tworzenia swojego programu może wybrać dla niego strukturę liniową lub modułową.

- Program liniowy wykonuje wszystkie instrukcje zadania automatyzacji po kolei – jedną po drugiej. Zwykle, program liniowy umieszcza wszystkie instrukcje w OB przeznaczonym do cyklicznego wykonywania (OB 1).
- Program modułowy wywołuje określone bloki kodu do wykonania specyficznych zadań. W celu stworzenia struktury modularnej, użytkownik musi podzielić złożone zadania automatyzacji na mniejsze podzadania odpowiadające funkcjom procesu. Każdy blok kodu zapewnia segment programu dla wykonania podzadania. Użytkownik określa strukturę programu poprzez wywołanie jednego bloku kodu z innego bloku.

Struktura liniowa:

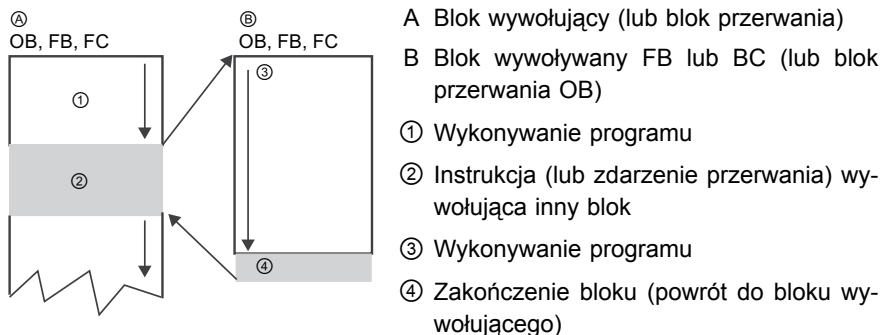


Struktura modułowa:

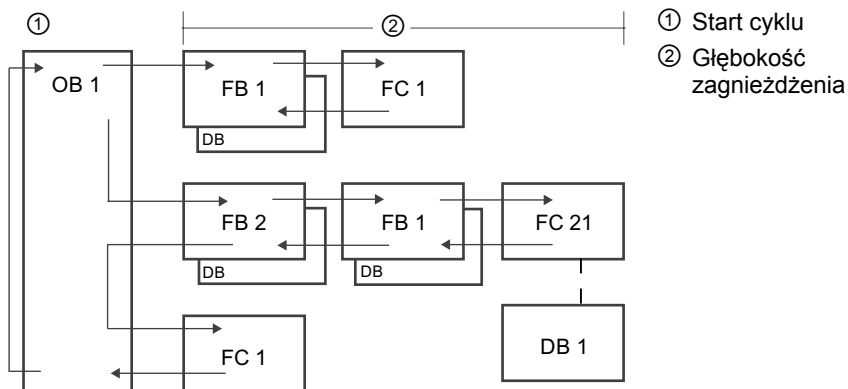


Poprzez utworzenie FB i FC służących do wykonywania ogólnych zadań użytkownik stwarza modułarne bloki kodu. Jeżeli inne bloki kodu mogą wywoływać te moduły przewidziane do wielokrotnego użycia, to charakter programu staje się strukturalny. Bloki wywołujące przekazują do bloków wywoływanych parametry związane z określonymi urządzeniami. Kiedy blok kodu wywoła inny blok kodu, wtedy CPU wykonuje program zawarty w bloku wywołanym. Po zakończeniu wykonania programu bloku wywołanego, CPU powraca do wykonywania programu bloku wywołującego. Wykonywanie programu jest kontynuowane od instrukcji następującej po tej instrukcji przy wykonywaniu której nastąpiło wywołanie bloku.

OB może zostać także wykorzystany do obsługi zdarzenia przerwania. Jeśli zajdzie takie zdarzenie, to CPU przerywa cykl programu użytkownika i wywołuje OB skonfigurowany do obsługi tego zdarzenia. Po zakończeniu działania przez OB obsługujący przerwanie, CPU podejmuje wykonywanie programu użytkownika od miejsca, w którym wystąpiło przerwanie.



W celu uzyskania modularnej struktury programu, wywołania bloków mogą być zagnieżdżone. W zamieszczonym przykładzie głębokość zagnieżdżenia wynosi 3: cykliczny OB plus 3 warstwy wywołań bloków kodu.



Poprzez stworzenie ogólnego bloku kodu, który może być wykorzystywany w programie użytkownika można uprościć projektowanie i implementację programu użytkownika.

- Do wykonywani standardowych zadań, takich jak sterowanie pompami lub silnikami można stworzyć bloki kodu wielokrotnego użytku. Te ogólne bloki kodu można przechowywać w bibliotece z możliwością wykorzystania w różnych aplikacjach lub rozwiązaniach.
- Jeżeli program użytkownika ma strukturę modularną zgodną z zadaniami funkcjonalnymi, to realizacja programu użytkownika jest łatwiejsza do zrozumienia i zarządzania. Składniki modularne nie tylko pomagają standaryzować konstrukcję programu, ale również sprawiają, że wprowadzanie uaktualnień i modyfikacji kodu programu jest łatwiejsze i szybsze.
- Tworzenie składników modularnych upraszcza debugowanie programu. Jeżeli cały program ma strukturę złożoną ze zbioru modułów programowych, to funkcjonalność każdego bloku kodu można testować zaraz po jego opracowaniu.
- Tworzenie składników modułowych powiązanych z określonymi zadaniami funkcyjnymi upraszcza i przyspiesza wdrażanie całej aplikacji.

6.1.1. Użycie bloków OB do organizacji programu użytkownika

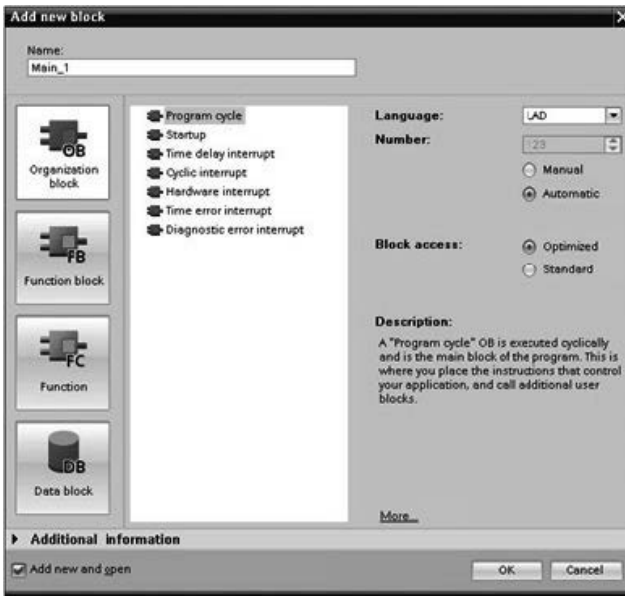
Bloki organizacyjne wprowadzają w programie pewną strukturę. Służą jako interfejs między systemem operacyjnym i programem użytkownika. OB są sterowane zdarzeniami. Zdarzenie, takie jak przerwanie diagnostyczne lub interwału czasowego, powoduje, że CPU wykonuje OB. Niektóre OB mają predefiniowane zdarzenia startowe i działanie.

Cykliczny OB zawiera główny program. Użytkownik może włączyć więcej niż jeden cykliczny OB w swój program użytkownika. W trybie RUN, cykliczne OB działają z najniższym priorytetem i mogą być przerwane przez wszystkie inne typy programów. Rozruchowy OB nie przerywa działania cyklicznego OB, ponieważ CPU wykonuje rozruchowy OB przed wejściem do trybu RUN.

Po zakończeniu wykonywania cyklicznego OB, CPU natychmiast zaczyna ponownie wykonywać cykliczny OB. To działanie cykliczne jest normalnym sposobem pracy sterowników PLC. W wielu aplikacjach cały program użytkownika jest zlokalizowany w jednym cyklicznym OB.

Użytkownik może utworzyć inne OB przewidziane do wykonywania specyficznych zadań, takich jak obsługa przerw i błędów lub wykonywanie określonego kodu programu w stałych odstępach czasu. Te OB przerywają wykonywanie cyklicznych OB.

W celu utworzenia nowego OB w programie użytkownika należy skorzystać z okna dialogowego Add new block.



Obsługa przerwania jest zawsze sterowana zdarzeniami. Jeśli zajdzie takie zdarzenie, to CPU przerywa wykonywanie programu użytkownika i wywołuje OB skonfigurowany do obsługi tego zdarzenia. Po zakończeniu działania przez OB obsługujący przerwanie, CPU podejmuje wykonywanie programu użytkownika od miejsca, w którym wystąpiło przerwanie.

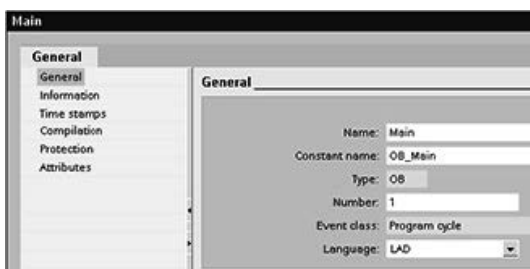
CPU określa kolejność obsługi przerw na podstawie priorytetu. Można przypisać wiele przerw do tej samej klasy priorytetu. Więcej informacji można znaleźć w częściach opisu bloków organizacyjnych (strona 53) i wykonywania programu użytkownika (strona 52).

Tworzenie dodatkowych bloków OB

Użytkownik może utworzyć wiele OB do wykorzystania w swoim programie, nawet w OB cyklicznym i rozruchowym. W celu utworzenia nowego OB należy skorzystać z okna dialogowego „Add new block” i wprowadzić nazwę OB.

Jeśli w programie użytkownika zostanie utworzonych wiele cyklicznych OB, to CPU wykonuje wszystkie cykliczne OB w kolejności zgodnej z ich numerami, począwszy od OB o najmniejszym numerze (takiego jak OB 1). Na przykład: po zakończeniu pierwszego cyklicznego OB (takiego jak OB 1), CPU wykonuje cykliczny OB o kolejnym wyższym numerze.

Konfiguracja właściwości OB



Można modyfikować właściwości OB. Na przykład, można skonfigurować numer OB lub język programowania.

Uwaga

Należy pamiętać, że do OB można przypisać numer partycji obrazu procesu, jak PIP0, PIP1, PIP2, PIP3 lub PIP4. Jeśli zostanie wprowadzony numer partycji obrazu procesu, to CPU utworzy partycję obrazu procesu. Więcej informacji na temat partycji obrazu procesu znajduje się w części „Wykonanie programu użytkownika” (strona 52).

6.1.2. FB i FC upraszczają programowanie zadań modułarnych

Funkcja (FC) działa jak podprogram. FC to blok kodu, który zazwyczaj wykonuje określone działania na zbiorze wartości wejściowych. FC przechowuje wyniki operacji w komórkach pamięci i stosuje się do wykonywania następujących zadań:

- Standardowych i powtarzalnych działań, jak na przykład obliczeń arytmetycznych.
- Zadania funkcyjne, takie jak indywidualne sterowanie za pomocą działań logicznych.

FC może być wywoływana wielokrotnie w różnych miejscach programu. Ta możliwość wielokrotnego użycia FC upraszcza programowanie często występujących zadań.

Przeciwnie niż blok funkcji (FB), FC nie jest skojarzona z żadnym blokiem danych instance (DB). Dla danych tymczasowych (obszar pamięci L) występujących podczas przeprowadzania obliczeń, FC wykorzystuje lokalny stos danych. Dane tymczasowe nie są zapamiętywane. W celu zapamiętania danych należy przypisać wartości wyjściowej miejsce w pamięci, na przykład w pamięci M lub w globalnym DB.

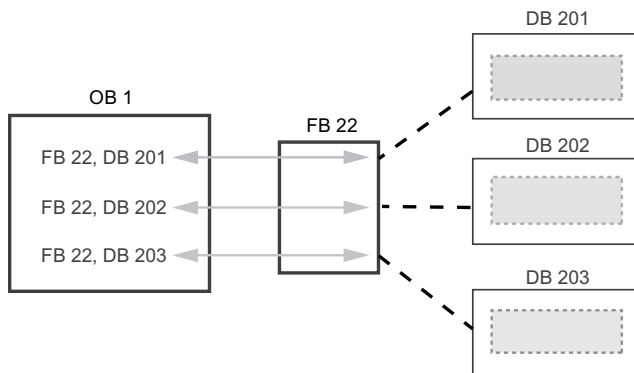
Blok funkcji (FB) działa jak podprogram z pamięcią. Jego wywołanie może być programowane za pomocą parametrów bloku. FB przechowuje wejściowe (IN), wyjściowe (OUT) i wejściowo/wyjściowe parametry w zmiennej pamięci zlokalizowanej w bloku danych (DB) lub instancji DB. Instancja DB zapewnia blok pamięci skojarzonej z „wywołaniem” FB i przechowuje dane po zakończeniu działania FB.

FB są zwykle używane do sterowania działaniem zadań lub urządzeń, które nie kończą swojej pracy w jednym cyklu programu. W celu przechowywania parametrów operacyjnych w taki sposób, by były szybko dostępne w kolejnych cyklach programu, każda FB w programie użytkownika ma jedną lub więcej instancji DB. Kiedy FB jest wywoływana, wtedy również jest otwierany DB przechowujący parametry bloku i statyczne dane lokalne dla danego wywołania lub instancji FB. Instancja DB pamięta te wartości po zakończeniu działania FB.

Użytkownik może nadać wartości początkowe parametrom interfejsu FB. Te wartości zostaną przeniesione do skojarzonej instancji DB. Jeśli parametrom nie zostaną nadane wartości, to będą wykorzystane wartości pamiętane w instancji DB. W niektórych przypadkach istnieje konieczność nadania parametrów początkowych.

Można skojarzyć różne bloki danych instancji DB z różnymi wywołaniami FB. Bloki DB pozwalają na użycie tego samego FB do sterowania wielu urządzeń. Tworzenie struktury programu polega na tym, że jeden blok kodu wywołuje inny FB oraz instancję DB. CPU wykonuje wówczas program zawarty w FB i zapamiętuje parametry bloku oraz statyczne dane lokalne w danej instancji DB. Gdy wykonanie FB jest zakończone, wtedy CPU powraca do bloku kodu, z którego FB został wywołany. Instancja DB zachowuje wartości wpisane podczas tego wykonania FB. Projektując FB do wykonywania ogólnych zadań sterowania, użytkownik może wykorzystać ten FB z wieloma urządzeniami wybierając różne instancje DB do różnych wywołań FB.

Na poniższym rysunku przedstawiono jeden OB, który trzykrotnie wywołuje jeden FB, za każdym razem z innym blokiem danych. Ta struktura pozwala wykorzystać jeden ogólny FB do sterowania kilku podobnych urządzeń, takich jak silniki, poprzez przypisanie FB w każdym wywołaniu różnych instancji bloku danych DB dla różnych urządzeń.



Każda instancja DB przechowuje dane (jak szybkość, czas rozpędzania i całkowity czas pracy) dla indywidualnego urządzenia. W podanym przykładzie FB 22 steruje trzema oddzielnymi urządzeniami, przy czym DB 201 pamięta dane dla pierwszego, DB 202 dla drugiego, a DB 203 dla trzeciego urządzenia.

6.1.3. Bloki danych (DB) ułatwiają przechowywanie danych programu

Bloki danych (DB) są umieszczane w programie użytkownika po to, by przechowywały dane dla bloków kodu. Wszystkie bloki kodu w programie użytkownika mają dostęp do globalnego DB, ale poszczególne instancje DB przechowują dane dla określonych bloków funkcji (FB).

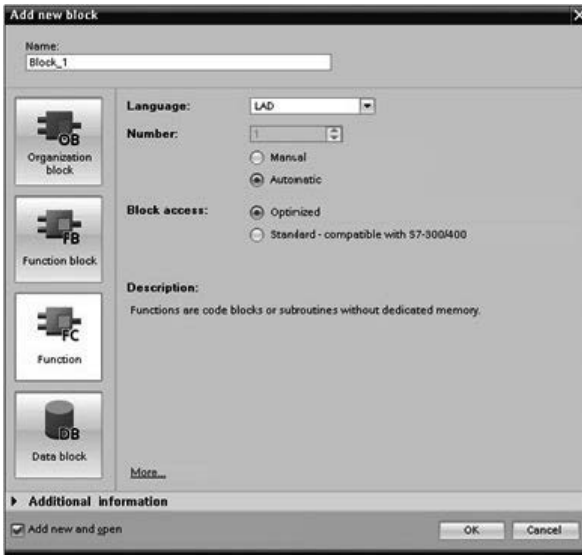
Program użytkownika może przechowywać dane w specjalizowanych obszarach pamięci CPU, przeznaczonych dla wejścia (I), wyjścia (Q) i pamięci bitowej (M). Ponadto użytkownik może wykorzystywać bloki danych (DB) dla uzyskania szybkiego dostępu do danych przechowywanych w samym programie. Użytkownik może nadać DB status „tylko do odczytu”.

Dane pamiętane w DB nie są usuwane po zamknięciu bloku danych lub po zakończeniu wykonywania korzystającego z nich bloku kodu. Są dwa typy DB:

- Globalny DB przechowuje dane dla bloków kodu programu użytkownika. Dostęp do danych zawartych w globalnym DB ma dowolny OB, FB i FC.
- Instancja DB przechowuje dane dla określonych FB. Struktura danych w instancji DB odzwierciedla parametry (wejściowe, wyjściowe i wejściowo/wyjściowe) oraz dane statyczne FB. Pamięć Temp FB nie jest przechowywana w instancji DB.

Mimo, że instancja DB przechowuje dane dla konkretnego FB, to dostęp do tych danych ma dowolny blok kodu.

6.1.4. Tworzenie bloków kodu do wielokrotnego wykorzystania



W celu utworzenia OB, FB, FC i globalnego DB, należy w *Project navigator* wybrać okno dialogowe *Add new block* z menu *Program blocks*.

Po utworzeniu bloku kodu, należy wybrać język programowania dla tego bloku. Dla DB nie wybiera się języka programowania ponieważ ten blok przechowuje tylko dane.

Wybierając opcję *Add new and open* (domyślna) otwierasz blok kodu w widoku projektu.

Obiekty do ponownego użycia można przechowywać w bibliotekach. W każdym projekcie jest biblioteka projektu, która jest związana z projektem. Oprócz biblioteki projektu, można utworzyć dowolną liczbę bibliotek globalnych, które można wykorzystywać w innych projektach. Elementy biblioteki mogą być kopiowane i przenoszone z jednej biblioteki do drugiej, ponieważ biblioteki są ze sobą kompatybilne.

Biblioteki służą, na przykład, do tworzenia szablonów dla bloków kodu, które najpierw są wklejane do biblioteki projektu, a następnie dalej tam rozwijane. Na koniec, można kopiować bloki kodu z biblioteki projektu do biblioteki globalnej. Biblioteka globalna jest dostępna dla innych kolegów pracujących nad projektem. Oni mogą używać bloki kodu i w razie potrzeby dostosowywać je do indywidualnych wymagań.

Szczegółowe informacje na temat funkcjonowania bibliotek, można znaleźć w opisach bibliotek w pomocy online programu STEP 7.

6.1.5. Wywoływanie bloku kodu z innego bloku kodu



Użytkownik może użyć każdego bloku kodu (OB, FB, lub FC), żeby wywołać z niego blok FB lub FC w CPU.

1. Otwórz blok kodu, który będzie wywoływał inny blok kodu.
2. W drzewie projektu zaznacz blok kodu, który ma zostać wywołany.
3. Przeciągnij blok do wybranej sieci, aby utworzyć wywołanie do bloku kodu.

Uwaga

OB nie może zostać wywołane przez program użytkownika, ponieważ OB jest sterowane zdarzeniami (strona 53). CPU rozpoczyna wykonanie OB w odpowiedzi na zdarzenie.

6.2. Języki programowania

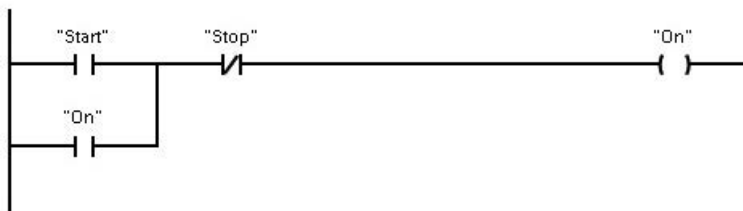
STEP 7 umożliwia korzystanie z następujących standardów języków programowania:

- LAD jest graficznym językiem programowania. Reprezentacja jest oparta na schematach obwodów.
- FBD (*Function Block Diagram*) jest językiem programowania, w którym reprezentacja logiki jest w nim oparta na graficznych symbolach logicznych stosowanych w algebrze Boole'a.
- SCL (*Structured Control Language*) jest językiem programowania wysokiego poziomu bazującym na tekście strukturalnym.

Podczas tworzenia bloku kodu zostaje wybrany język programowania, który będzie używany przez ten blok. Program użytkownika może korzystać z kodu stworzonego w dowolnym lub we wszystkich językach programowania.

6.2.1. Język drabinkowy (LAD)

Elementy obwodu, takie jak styki normalnie zwarte lub normalnie rozwarne i cewki, są ze sobą łączone w celu utworzenia sieci.



W celu zaprojektowania logiki dla złożonych operacji, na schemacie można umieszczać gałęzie dla utworzenia logiki obwodów równoległych. Gałęzie równoległe są na początku rozwarne lub bezpośrednio łączone do szyny zasilania. Od strony końcowej gałęzi występują połączenia zakańczające.

LAD umożliwia stosowanie instrukcji dla różnych funkcji, takich jak arytmetyczne, czasowe, zliczające oraz związane z napędami.

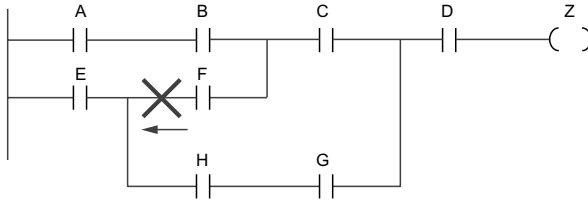
STEP 7 nie ma ograniczenia co do liczby instrukcji (wierszy i kolumn) w sieci programowanej w LAD.

Uwaga

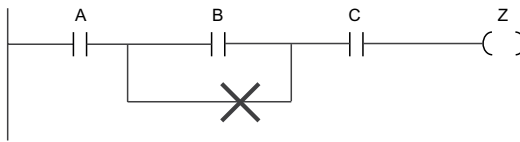
Każda sieć LAD musi kończyć się cewką lub instrukcją.

Podczas tworzenia schematu LAD, należy kierować się następującymi zasadami:

- Nie wolno tworzyć gałęzi, w której może nastąpić przepływ mocy w odwrotnym kierunku.

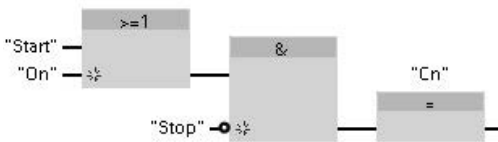


- Nie wolno tworzyć gałęzi, która mogłaby spowodować zwarcie.



6.2.2. Język programowania Function Block Diagram (FBD)

Podobnie jak LAD, również FBD jest graficznym językiem programowania. Reprezentacja logiki jest w nim oparta na graficznych symbolach logicznych stosowanych w algebrze Boole'a.



W celu stworzenia logiki złożonych operacji wystarczy połączyć symbole logiczne równoległymi gałęziami.

Działania arytmetyczne i inne złożone funkcje mogą być reprezentowane bezpośrednio razem z symbolami logicznymi.

STEP 7 nie ma ograniczenia co do ilości instrukcji (wierszy i kolumn) w sieci programowanej w FBD.

6.2.3. Język SCL

Język wysokiego poziomu SCL (Structured Control Language), bazujący na języku programowania PASCAL, jest używany do tworzenia programów dla jednostek CPU z rodziny SIMATIC S7. SCL obsługuje blokową strukturę programu STEP 7. Możliwe jest także łączenie bloków programowych napisanych w języku SCL z blokami programowymi napisanymi w językach LAD oraz FBD.

Instrukcje SCL wykorzystują standardowe operatory programowe, takie jak operator przypisania (:=), czy operatory arytmetyczne (+ dla dodawania, – dla odejmowania, * dla mnożenia oraz / dla dzielenia). SCL używa również standardowe w języku PASCAL, instrukcje sterujące wykonywaniem programu, takie jak IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO oraz RETURN. Można użyć dowolnej referencji języka PASCAL dla elementów składniowych języka programowania

SCL. Wiele innych instrukcji SCL, jak te związane z obsługą timerów i liczników, ma swoje odpowiedniki w językach LAD i FBD.

Ponieważ język SCL, podobnie jak Pascal, zapewnia struktury sterowania do warunkowego wykonywania bloków kodu, wykonywania pętli programowych i zagnieźdzenia instrukcji, realizacja złożonych algorytmów w języku SCL może być łatwiejsza niż za pomocą języków LAD lub FBD.

W poniższych przykładach pokazano różne wyrażenia dla różnych zastosowań:

"C" := #A+#B;	Przypisuje sumę dwóch zmiennych lokalnych do tagu
"Data_block_1".Tag := #A;	Przypisanie do tagu bloku danych
IF #A > #B THEN "C" := #A;	Warunek instrukcji IF-THEN
"C" := SQRT (SQR (#A) + SQR (#B));	Parametry instrukcji SQRT

Jako język programowania wysokiego poziomu, SCL do podstawowych zadań wykorzystuje standardowe instrukcje:

- Instrukcja przypisania: :=
- Operatory arytmetyczne: +, -, *, and /
- Adresowanie zmiennych globalnych (tagów): "<nazwa tagu>" (nazwa tagu lub nazwa bloku danych ujęte w podwójny cudzysłów)
- Adresowanie zmiennych lokalnych: #<nazwa zmiennej> (nazwa zmiennej poprzedzona symbolem „#”)
- Adresowanie bezwzględne: %<adres bezwzględny>, na przykład %I0.0 lub %MW10

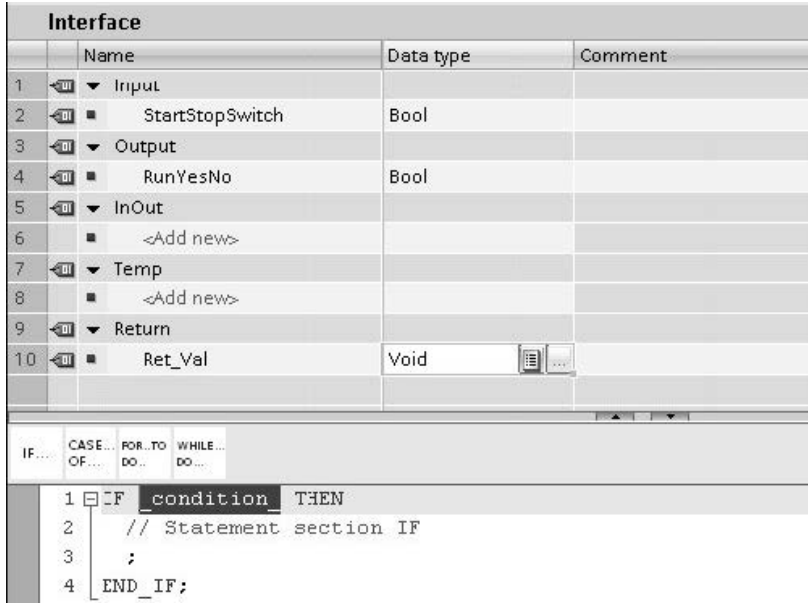
Operatory arytmetyczne mogą przetwarzać różne typy danych liczbowych. Typ danych wyniku zależy od typu danych argumentów najbardziej znaczących. Na przykład, operacja mnożenia argumentu typu INT przez argument typu REAL daje w wyniku wartość typu REAL.

6.2.4. Edytor programów SCL

Można wyznaczyć dowolny typ bloku (OB, FB, lub FC), który będzie korzystał z języka programowania SCL, podczas tworzenia bloku. Program STEP 7 dostarcza edytor programów SCL zawierający następujące elementy:

- sekcję interfejsu do definiowania parametrów bloku kodu,
- sekcję kodu dla kodu programu,
- drzewo instrukcji zawierające instrukcje SCL obsługiwane przez CPU.

Kod SCL określonych instrukcji wprowadza się bezpośrednio w sekcji kodu. Edytor zawiera przyciski typowych konstrukcji kodu i komentarzy. W przypadku złożonych instrukcji, można przeciągać instrukcje SCL z drzewa instrukcji do kodu programu w sekcji kodu. Można również utworzyć kod programu w języku SCL za pomocą dowolnego edytora tekstu, a następnie zaimportować go do pakietu STEP 7.



W sekcji interfejsu bloku kodu SCL można zadeklarować następujące typy parametrów:

- Input, Output, InOut oraz Ret_Val: Te parametry określają tagi wejściowe, tagi wyjściowe, a także wartość zwracaną przez blok kodu. Nazwa tagu, którą użytkownik tutaj wprowadza, jest używana lokalnie podczas wykonywania bloku kodu. Zazwyczaj nie używa się globalnej nazwy tagu w tablicy tagów.
- Static (tylko w FB; ilustracja powyżej dotyczy FC): Blok kodu używa tagów statycznych do przechowywania pośrednich wyników statycznych w bloku danych instancji. Blok przechowuje dane statyczne do czasu ich nadpisania, co może się zdarzyć po kilku cyklach programu. Nazwy bloków, które ten blok wywołuje jako bloki wielu instancji, są również przechowywane w statycznych danych lokalnych.
- Temp: Te parametry są tymczasowymi tagami, które są używane podczas wykonywania bloku kodu.

Jeśli blok kodu SCL jest wywoływany z innego bloku kodu, to parametry bloku kodu SCL są pokazywane jako wejściowe lub wyjściowe.



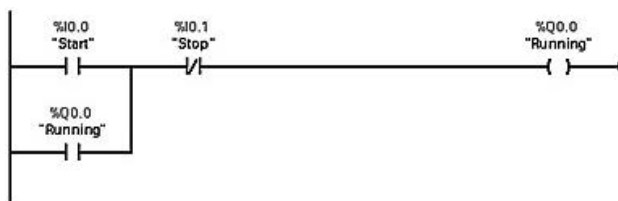
W tym przykładzie, tagi „Start” oraz „On” (z tablicy tagów projektu) odpowiadają tagom „StartStopSwitch” oraz „RunYesNo” w tabeli deklaracji programu SCL.

6.3. Instrukcje programowania

6.3.1. Podstawowe instrukcje

Instrukcje logiki bitowej

Podstawowymi elementami instrukcji logicznych są styki i cewki. Styki odczytują stan bitu, a cewki nadpisują stan operacji do bitu.



Styki testują stan logiczny przypisanego bitu. Styki są zwarte dla 1 i rozwarne dla 0.

Stan cewek jest zgodny z przetworzona logiką układu.

Jeśli cewka o tym samym adresie jest użyta w więcej niż jednym miejscu programu, to podczas odświeżania fizycznych wyjść zostanie ustawiony stan odpowiadający stanowi ostatniej cewki.

Styki normalnie
rozwarne



Styki normalnie
zwarte



Styki normalnie rozwarne (*normally open*) są zwarte (ON) wtedy, kiedy wartość przypisanego bitu jest równa 1.

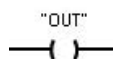
Styki normalnie zwarte (*normally closed*) są zwarte (ON) wtedy, kiedy wartość przypisanego bitu jest równa 0.

Podstawowymi elementami operacji logicznych są bramki AND i OR. Styki połączone szeregowo realizują sieć logiczną opartą na AND, natomiast równoległe połączenie styków tworzy sieć OR.

Styki można łączyć z innymi stykami i tworzyć w ten sposób własną logikę kombinacyjną. Jeżeli wyspecyfikowany przez użytkownika bit wejściowy używa identyfikatora pamięci I (wejście) lub Q (wyjście), to wartość bitu jest czytana z rejestru obrazu procesu. Sygnały fizyczne styków są w systemie sterowania procesem połączone przewodami do wyprowadzeń wejściowych w PLC. System PLC skanuje okablowane sygnały wejściowe i w sposób ciągły uaktualnia odpowiednie stany rejestru wejściowego obrazu procesu.

Użytkownik może spowodować bezpośredni odczyt wejścia fizycznego komendą „:P” występującą zaraz po adresie wejścia (na przykład: „Motor_Start:P” lub „I3.4:P”). Odczyt bezpośredni polega na tym, że wartości bitów danych są czytane bezpośrednio z wejścia fizycznego, a nie z obrazu procesu. Odczyt bezpośredni nie uaktualnia obrazu procesu.

Cewka wyjściowa



Zanegowana cewka wyjściowa



Należy zwrócić uwagę na następujące wartości bitów wyjściowych zależne od zasilania cewek wyjściowych oraz zanegowanych cewek wyjściowych:

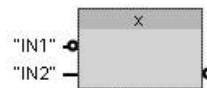
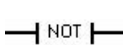
- Jeżeli cewka wyjściowa jest zasilana, to wartość bitu wyjściowego jest ustalana na 1.
- Jeżeli cewka wyjściowa nie jest zasilana, to wartość bitu wyjściowego jest ustalana na 0.
- Jeżeli zanegowana cewka wyjściowa jest zasilana, to wartość bitu wyjściowego jest ustalana na 0.
- Jeżeli zanegowana cewka wyjściowa nie jest zasilana, to wartość bitu wyjściowego jest ustalana na 1.

Instrukcja wyjściowa sterowania cewką ustala wartość bitu wyjściowego. Jeżeli wyspecyfikowany przez użytkownika bit wyjściowy ma identyfikator pamięci Q, to CPU włącza lub wyłącza ten bit w rejestrze obrazu procesu tak, by był zgodny ze stanem zasilania. Wyjściowe sygnały sterujące urządzeniami wykonawczymi są podłączone do zacisków wyjściowych sterownika PLC. W trybie RUN, system CPU w sposób ciągły skanuje sygnały wejściowe, przetwarza te sygnały zgodnie z logiką programu i w rezultacie ustala nowe wartości stanów wyjściowych w rejestrze wyjściowym obrazu procesu. Po zakończeniu każdego cyklu programu, CPU przesyła te nowe wartości stanów wyjściowych zapamiętane w rejestrze obrazu procesu do okablowanych zacisków wyjściowych.

Użytkownik może spowodować bezpośredni zapis stanu do wyjścia fizycznego komendą „P” występującą zaraz po adresie wyjściowym (na przykład: „Motor_On:P” lub „%Q3.4:P”). Zapis bezpośredni polega na tym, że wartości bitów danych są zapisywane do obszaru wyjściowego obrazu procesu i jednocześnie bezpośrednio do wyjścia fizycznego.

Pozycje cewki nie muszą być ograniczone do końca sieci. Można umieścić cewkę w środku szczebla sieci LAD, pomiędzy stykami i innymi instrukcjami.

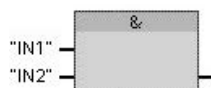
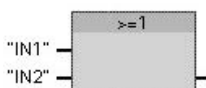
Inwerter stykowy NOT (LAD) Ramka AND z jednym wejściem zanegowanym (FBD) Ramka AND z zanegowanymi wejściem i wyjściem (FBD)



Inwerter stykowy NOT języka LAD neguje wejściowy stan logiczny zasilania.

- Jeżeli na wejściu styku NOT nie ma zasilania, to na wyjściu zasilanie występuje.
- Jeżeli na wejściu styku NOT jest zasilanie, to na wyjściu zasilanie nie występuje.

W języku programowania FBD z menu narzędziowego *Favorites* lub drzewa z instrukcjami można przeciągnąć narzędzie *Negate binary input* i upuścić je na wejście lub wyjście bloku, aby umieścić na nim inwerter logiczny.

Funkcja logiczna AND
(FBD)Funkcja logiczna OR
(FBD)Funkcja logiczna XOR
(FBD)

- Aby stan wyjścia bloku AND był TRUE, wszystkie wejścia muszą być w stanie TRUE.
- Aby stan wyjścia bloku OR był TRUE, dowolne wejście musi być w stanie TRUE.
- Aby stan wyjścia bloku XOR był TRUE, nieparzysta liczba wejść musi być w stanie TRUE.

W języku programowania FBD, stykowe sieci LAD są reprezentowane przez operatory logiczne AND (&), OR (>=1) i XOR (x), w których użytkownik może wyspecyfikować wartości bitów wejściowych i wyjściowych bloków. Może również wykonać połączenia z innymi blokami logicznymi, co pozwala utworzyć własną logikę kombinacyjną. W celu zwiększenia liczby wejść, po umieszczeniu bloków w sieci, z menu narzędziowego *Favorites* lub drzewa z instrukcjami można przeciągnąć narzędzie *Insert binary input* i upuścić je po stronie wejściowej bloków. Można również kliknąć prawym klawiszem myszy na złącze wejściowe bloku wybrać *Insert inputs*.

Wejścia i wyjście bloku można połączyć z innym blokiem logicznym lub podać na niepodłączone wejście adres bitu lub symboliczną nazwę bitu. Kiedy wykonywana jest instrukcja przypisana blokowi, wtedy na jego wejścia są podawane bieżące stany logiczne bitów i jeśli wynik działania jest prawdziwy, to na wyjściu pojawia się stan TRUE (prawda).

6.3.2. Instrukcje porównania oraz move

Instrukcje porównania służą do porównania dwóch wartości danych tego samego typu.

Tabela 6.1. Instrukcje porównania

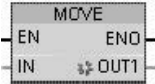
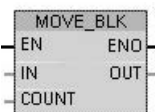
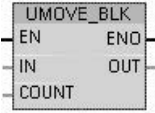
Instrukcja	SCL	Opis
LAD: 	<pre> out := in1 == in2; out := in1 <> in2; out := in1 >= in2; out := in1 <= in2; out := in1 > in2; out := in1 < in2; </pre>	<ul style="list-style-type: none"> • Equals (==): Porównanie jest prawdą, jeśli IN1 jest równe IN2. • Not equal (<>): Porównanie jest prawdą, jeśli IN1 nie jest równe IN2. • Greater than or equal to (>=): Porównanie jest prawdą, jeśli IN1 jest większe od lub równe IN2. • Less than or equal to (<=): Porównanie jest prawdą, jeśli IN1 jest mniejsze od lub równe IN2. • Greater than (>): Porównanie jest prawdą, jeśli IN1 jest większe od IN2. • Less than (<): Porównanie jest prawdą, jeśli IN1 jest mniejsze od IN2.
FBD: 		

¹ LAD i FBD: styk porównania LAD lub wynik porównania FBD ma wartość TRUE, gdy porównanie jest prawdą.

Instrukcja MOVE jest stosowana do kopiowania elementów danych do miejsca pamięci o nowym adresie oraz konwersji jednego typu danych na inny. Dane wejściowe instrukcji MOVE nie ulegają zmianie.

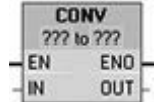
- MOVE: kopiuje elementy danych pamiętane pod określonym adresem w miejsce pamięci o nowym adresie. Aby dodać kolejne wyjście należy nacisnąć ikonę obok parametru OUT1.
- MOVE_BLK (przerywalna instrukcja MOVE) i UMOVE_BLK (nieprzerywalna instrukcja MOVE) kopiuje blok elementów danych pamiętany pod określonym adresem w miejsce pamięci o nowym adresie. Dodatkowy parametr COUNT określa ile elementów danych ma zostać skopiowanych. Liczba bajtów przypadająca na kopiowany element zależy od typu danych przypisanych parametrom IN i OUT, co określone jest w tabelicy zmiennych PLC.

Tabela 6.2. Instrukcje MOVE, MOVE_BLK i UMOVE_BLK

LAD/FBD	SCL	Opis
	<pre>out1 := in;</pre>	<p>Kopiuje elementy przechowujące dane pod określonym adresem do nowego adresu lub wielu adresów. Aby dodać kolejne wyjście w LAD lub FBD należy kliknąć ikonę przy wyjściowym parametrze. Dla SCL należy użyć instrukcji wielokrotnego przypisania. Można także użyć jedną z instrukcji pętli.</p>
	<pre>MOVE_BLK(in:=_variant_in, count:=_uint_in, out=>_variant_out);</pre>	<p>Przerywalna instrukcja MOVE, która kopiuje blok elementów danych pod nowy adres.</p>
	<pre>UMOVE_BLK(in:=_variant_in, count:=_uint_in out=>_variant_out);</pre>	<p>Nieprzerywalna instrukcja MOVE, która kopiuje blok elementów danych pod nowy adres.</p>

6.3.3. Instrukcje konwersji

Tabela 6.3. Instrukcje teorii

LAD/FBD	SCL	Opis
	<pre>out := <data type in>_TO_<data type out>(in);</pre>	<p>Konwertuje elementy danych z jednego typu na inny.</p>

¹ LAD i FBD: Kliknięcie poniżej nazwy bloku pozwala wybrać z rozwijanego menu typ danych. Po dokonaniu wyboru typu danych źródłowych (*convert from*) jest rozwijana lista możliwych konwersji (*convert to*).

² Dla SCL: Należy utworzyć instrukcję konwersji poprzez określenie typu danych parametru wejściowego (in) i parametru wyjściowego (out). Na przykład, DWORD_TO_REAL konwertuje wartość typu DWORD na wartość typu REAL.

Tabela 6.4. Instrukcje zaokrąglania i skrócenia


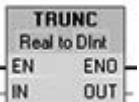
LAD/FBD	SCL	Opis
	<code>out := ROUND (in);</code>	<p>ROUND zamienia liczbę rzeczywistą (Real lub LReal) na całkowitą. Część ułamkowa liczby rzeczywistej jest zaokrąglana do najbliższej wartości całkowitej (IEEE – round to nearest). Jeśli wartość liczby jest dokładnie pomiędzy dwoma całkowitymi (np. 10,5), to liczba jest zaokrąglana do najbliższej całkowitej liczby parzystej. Przykładowo:</p> <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12 <p>Dla LAD/FBD, w bloku instrukcji należy kliknąć „???", aby wybrać typ danych wyjściowych, na przykład, „DInt”. Dla SCL, domyślnym typem danych wyjściowych jest DINT. Aby zaokrąglić wartości innego typu danych wyjściowych, należy wprowadzić nazwę instrukcji z jawną nazwą typu danych, na przykład, ROUND_REAL lub ROUND_LREAL.</p>
	<code>out := TRUNC (in);</code>	<p>TRUNC zamienia liczbę rzeczywistą (Real lub LReal) na całkowitą. Część ułamkowa liczby rzeczywistej jest skrócona do zera.(IEEE – round to zero).</p>

Tabela 6.5. Instrukcje wyznaczania najbliższych liczb całkowitych

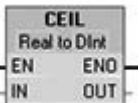
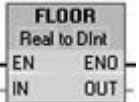


LAD/FBD	SCL	Opis
	<code>out := CEIL(in);</code>	<p>CEIL zamienia liczbę rzeczywistą (Real lub LReal) na najmniejszą liczbę całkowitą większą lub równą liczbie rzeczywistej (IEEE – round to +infinity).</p>
	<code>out := FLOOR(in);</code>	<p>FLOOR zamienia liczbę rzeczywistą na największą liczbę całkowitą mniejszą lub równą liczbie rzeczywistej (IEEE – round to +infinity).</p>

Tabela 6.6. Instrukcje skalowania i normalizacji

LAD/FBD	SCL	Opis
	<code>out := SCALE_X(min:=_undef_in_ value:=_real_in_, max:=_undef_in_);</code>	<p>SCALE_X skaluje znormalizowany parametr VALUE (gdzie $0.0 \leq \text{VALUE} \leq 1.0$) do typu danej i zakresu wartości wyspecyfikowanych przez parametry MIN i MAX¹: $\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$.</p>
	<code>out := NORM_X(min:=_undef_in_ value:=_undef_in_, max:=_undef_in_);</code>	<p>NORM_X normalizuje parametr VALUE wewnątrz zakresu wartości wyspecyfikowanych przez parametry MIN i MAX²: $\text{OUT} = (\text{VALUE} - \text{MIN})/(\text{MAX} - \text{MIN})$, gdzie $0.0 \leq \text{OUT} \leq 1.0$.</p>

¹ Ekwiwalent SCL: $\text{out} := \text{value} (\text{max} - \text{min}) + \text{min}$.

² Ekwiwalent SCL: $\text{out} := (\text{value} - \text{min})/(\text{max} - \text{min})$.

6.3.4. Operacje arytmetyczne z instrukcją Calculate

Tabela 6.7. Instrukcja CALCULATE

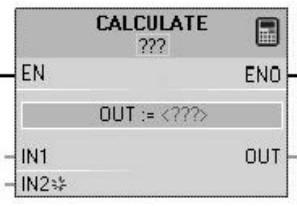
LAD/FBD	SCL	Opis
	<p>Do utworzenia równania należy użyć standardowych wyrażeń matematycznych SCL.</p>	<p>Instrukcja CALCULATE pozwala na utworzenie matematycznych funkcji, które operują na wejściach (IN1, IN2, INn) i tworzą wynik OUT zgodny ze zdefiniowanym przez użytkownika równaniem.</p> <ul style="list-style-type: none"> • Najpierw należy wybrać typ obsługiwanych danych. Wszystkie wejścia i wyjście muszą być tego samego typu. • Aby dodać kolejne wyjście należy nacisnąć ikonę przy ostatnim wejściu.

Tabela 6.8. Typy danych dla parametrów

Parametr	Obsługiwany typ danych ¹
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord

¹ Parametry IN oraz OUT muszą być tego samego typu (łącznie z niejawnymi konwersjami parametrów). Przykładowo wartość: A SINT dla wejścia zostanie przekonwertowana na INT lub REAL jeśli wartość OUT będzie INT lub REAL.

Należy nacisnąć ikonę kalkulatora dla otwarcia okna dialogowego i zdefiniowania funkcji matematycznej. W równaniach należy podawać wejścia (jak IN1 i IN2) i operacje. Po kliknięciu OK następuje zapisanie funkcji i automatyczne stworzenie wymaganych przez instrukcję wejść.

Przykład i lista możliwych operacji matematycznych pokazana jest na dole edytora.



Uwaga

Dla stałych należy także stworzyć wejście w funkcji obliczającej. Stała wartość będzie wtedy wprowadzona z powiązanego z instrukcją CALCULATE wejścia.

Przez wprowadzenie stałych jako wejść można skopiować instrukcję CALCULATE w inne miejsce w programie użytkownika bez zmiany samej funkcji. Dzięki temu można zmienić wartości lub zmienne na wejściach instrukcji bez potrzeby zmiany samego równania.

Po pomyślnym wykonaniu instrukcji CALCULATE i wszystkich indywidualnych operacji obliczeniowych instrukcja ustawia ENO = 1. W przypadku błędu ENO = 0.

6.3.5. Układy czasowe – timery

S7-1200 obsługuje następujące timery:

- TP: Układ czasowy *Pulse timer* generuje impuls o ustalonym czasie trwania.
- TON: Układ czasowy *ON-delay timer* ustawia stan swojego wyjścia Q na ON (włączony) po upływie zadanego czasu opóźnienia.
- TOF: Układ czasowy *OFF-delay timer* kasuje stan swojego wyjścia Q na OFF (wyłączony) po upływie zadanego czasu opóźnienia.
- TONR: Układ czasowy *ON-delay Retentive timer* ustawia stan swojego wyjścia na ON (włączony) po upływie zadanego czasu opóźnienia. Upływający czas jest naliczany przez wiele okresów, aż do chwili, gdy zliczany upływ czasu zostanie wyzerowany za pomocą wejścia R.

Dla LAD i FBD timery te są dostępne jako blok instrukcji lub jako wyjściowa cewka. STEP 7 zapewnia także następujące cewki czasowe dla LAD i FBD:

- PT (*preset timer*) – cewka ładująca nowe nastawy czasu do określonego timera.
- RT (*reset timer*) – cewka kasująca określony timer.

Liczba timerów użytych w programie użytkownika jest ograniczona tylko rozmiarem pamięci CPU. Każdy timer używa 16 bajtów pamięci.

Każdy timer używa struktury przechowywanej w bloku danych. Dla SCL należy najpierw stworzyć DB dla instrukcji pojedynczego timera, aby móc się do niego odwoływać. Dla LAD i FBD tworzone są automatycznie DB, gdy tylko tworzona jest instrukcja.

Kiedy instrukcja dotycząca timera zostaje umieszczona w bloku funkcji, to można wybrać opcję wielokrotnej instancji bloku danych. Ponieważ dane timera są zawarte w jednym bloku danych i nie są wymagane oddzielne bloki danych dla każdego timera, to redukuje czas przetwarzania. Nie występuje żadna interakcja pomiędzy strukturami danych timerów we współdzielonych wielokrotnych instancjach bloków danych.

Tabela 6.9. TP (timer impulsowy)

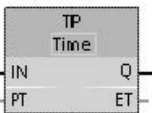
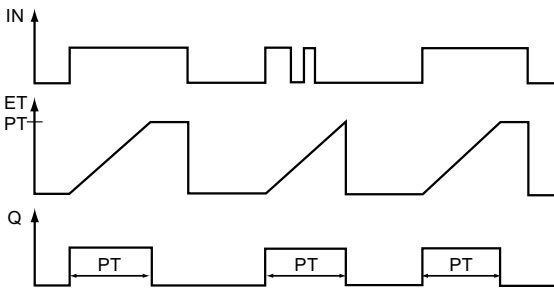
LAD/FBD	SCL	Przebieg czasowy
<p>IEC_Timer_0</p>  <p>TP Time</p> <p>IN Q</p> <p>PT ET</p> <p>TP_DB</p> <p>(TP)</p> <p>"PRESET_Tag"</p>	<pre> „timer_db”.TP(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_); </pre>	 <p>The timing diagram shows three input pulses (IN). The first pulse is long, the second is short, and the third is long. The ET signal (top) shows a sawtooth ramp starting at the beginning of each pulse and reaching a peak at the end of the pulse. The PT signal (middle) shows a rectangular pulse that starts at the beginning of each pulse and ends at the end of the pulse. The Q signal (bottom) shows a rectangular pulse that starts at the beginning of each pulse and ends at the end of the pulse. The duration of each Q pulse is labeled as PT.</p>

Tabela 6.10. TON (ON-delay timer)

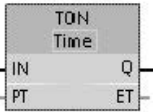
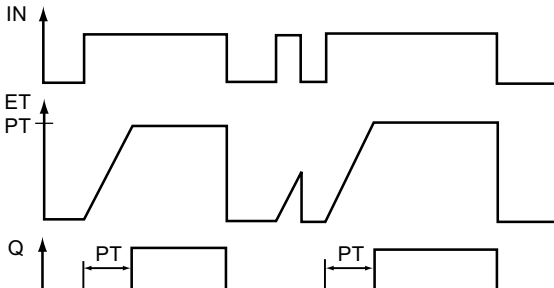
LAD/FBD	SCL	Przebieg czasowy
<p>IEC_Timer_1</p>  <p>TON Time</p> <p>IN Q</p> <p>PT ET</p> <p>TON_DB</p> <p>(TON)</p> <p>"PRESET_Tag"</p>	<pre> „timer_db”.TON(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_); </pre>	 <p>The timing diagram shows three input pulses (IN). The first pulse is long, the second is short, and the third is long. The ET signal (top) shows a sawtooth ramp starting at the beginning of each pulse and reaching a peak at the end of the pulse. The PT signal (middle) shows a rectangular pulse that starts at the beginning of each pulse and ends at the end of the pulse. The Q signal (bottom) shows a rectangular pulse that starts at the beginning of each pulse and ends at the end of the pulse. The duration of each Q pulse is labeled as PT.</p>

Tabela 6.11. TOF (OFF-delay timer)

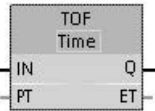
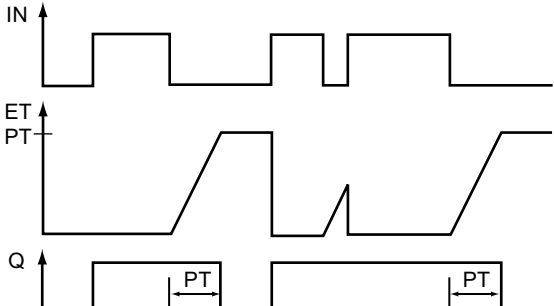
LAD/FBD	SCL	Przebieg czasowy
<p>IEC_Timer_2</p>  <p>TOF Time</p> <p>IN Q</p> <p>PT ET</p> <p>TOF_DB</p> <p>(TOF)</p> <p>"PRESET_Tag"</p>	<pre> „timer_db”.TOF(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_); </pre>	 <p>The timing diagram shows three input pulses (IN). The first pulse is long, the second is short, and the third is long. The ET signal (top) shows a sawtooth ramp starting at the beginning of each pulse and reaching a peak at the end of the pulse. The PT signal (middle) shows a rectangular pulse that starts at the beginning of each pulse and ends at the end of the pulse. The Q signal (bottom) shows a rectangular pulse that starts at the beginning of each pulse and ends at the end of the pulse. The duration of each Q pulse is labeled as PT.</p>

Tabela 6.12. TONR (ON-delay Retentive timer)

LAD/FBD	SCL	Przebieg czasowy
	<pre>"timer_db".TONR(IN:=_bool_in_, R:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	

Tabela 6.13. Instrukcje PT (preset timer – czas nastawiony) i RT (reset timer)

LAD	Opis
	<p>Instrukcje-PT oraz RT jako cewki lub bloki. Cewki instrukcji mogą zostać wstawione w pozycji środkowej. Stan zasilania wyjścia cewki jest zawsze taki sam jak stan zasilania jej wejścia.</p> <ul style="list-style-type: none"> Kiedy cewka -(PT)- jest aktywna, wartość elementu PRESET time w określonym polu IEC_Timer bloku DB jest ustawiana na czas trwania PRESET_Tag. Kiedy cewka RT jest aktywna, wartość elementu Elapsed time w określonym polu IEC_Timer bloku DB jest ustawiana na 0.

Tabela 6.14. Typy danych dla parametrów

Parametr	Typ danych	Opis
Blok: IN Cewka: Power flow (zasilanie)	Bool	TP, TON, i TONR: Blok: 0 = zatrzymanie timera, 1 = uruchomienie timera Cewka: Brak zasilania=zatrzymanie timera, zasilanie = uruchomienie timera TOF: Blok: 0 = uruchomienie timera, 1 = zatrzymanie timera Cewka: brak zasilania=uruchomienie timera, zasilanie = zatrzymanie timera
R	Bool	Tylko blok TONR: 0 = bez zerowania 1 = zerowanie upływu czasu i bitu Q na 0
Blok: PT Cewka: "PRESET_Tag"	Time	Blok lub cewka timera: nastawiony czas na wejściu
Blok: Q Cewka: DBdata.Q	Bool	Blok timera: wyjście bloku Q lub bit Q w danych DB timera Cewka timera: można tylko adresować bit Q w danych DB timera
Blok: ET Cewka: DBdata.ET	Time	Blok timera: wyjście bloku ET (upływu czasu) lub wartość czasu ET w danych DB timera Cewka timera: można tylko adresować wartość czasu ET w danych DB timera

Tabela 6.15. Wynik zmiany wartości w parametrach PT oraz IN

Timer	Zmiany w blokach PT oraz bloku parametrów IN i odpowiadające im parametry cewek
TP	<ul style="list-style-type: none"> • Zmiana PT nie ma żadnego efektu podczas pracy timera. • Zmiana IN nie ma żadnego efektu podczas pracy timera.
TON	<ul style="list-style-type: none"> • Zmiana PT nie ma żadnego efektu podczas pracy timera. • Zmiana IN na FALSE, podczas gdy timer pracuje, kasuje i zatrzymuje timer.
TOF	<ul style="list-style-type: none"> • Zmiana PT nie ma żadnego efektu podczas pracy timera. • Zmiana IN na TRUE, podczas gdy timer pracuje, kasuje i zatrzymuje timer.
TONR	<ul style="list-style-type: none"> • Zmiana PT nie ma żadnego efektu podczas pracy timera, ale zadziała podczas ponownego uruchomienia timera. • Zmiana IN na FALSE, podczas gdy timer pracuje, zatrzymuje timer, ale go nie resetuje. Zmiana IN z powrotem na TRUE powoduje, że timer rozpoczyna pracę od zliczonej wartości czasu.

Wartości PT (*preset time* – czas nastawiony) i ET (*elapsed time* – upływający czas) są przechowywane w pamięci w określonym bloku danych IEC_TIMER DB jako liczby całkowite o podwójnej długości ze znakiem i wyrażają czas w milisekundach. Dana TIME wykorzystuje identyfikator T# i może być wprowadzana jako prosta jednostka czasu „T#200ms” lub 200 oraz w postaci złożonej „T#2s_200ms”.

Tabela 6.16. Rozmiar i zakres dla danych typu TIME

Typ danych	Rozmiar	Zakres poprawnych wartości ¹
TIME	32 bity Przechowywany jako DInt	T#-24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms Przechowywane jako -2,147,483,648 ms do +2,147,483,647 ms

¹ Ujemne wartości dla danych typu TIME nie mogą być używane z instrukcjami timerów. Ujemne wartości PT zostają wyzerowane podczas wykonania instrukcji danego timera. ET ma zawsze wartość dodatnią.

Programowanie timera

Podczas planowania i tworzenia programu użytkownika należy rozważyć następujące konsekwencje działania timera:

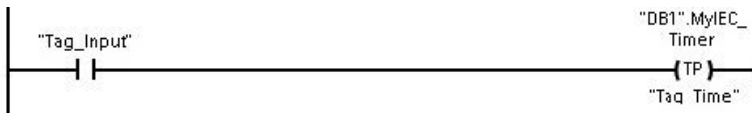
- Może być wiele ukatualnień timera w tym samym cyklu programu. Timer jest aktualizowany za każdym razem, gdy jest wykonywana instrukcja timera (TP, TON, TOF, TONR) oraz gdy element ELAPSED lub Q struktury timera jest używany jako parametr innej wykonywanej instrukcji. Korzystne jest posiadanie aktualnych danych czasowych (zwłaszcza bezpośredni odczyt timera). Jednak, jeśli chcemy mieć spójne wartości w całym cyklu programu, to należy umieścić instrukcje timera przed wszystkimi innymi instrukcjami, które potrzebują tych wartości, i używać tagów z wyjść Q i ET timera zamiast elementów ELAPSED i Q struktury DB timera.
- Mogą być cykle programu, w czasie których aktualizacja timera nie nastąpi. Możliwe jest uruchomienie timera w pewnej funkcji, a następnie nie wywołanie tej funkcji w jednym cyklu lub w wielu cyklach programu. Jeśli nie ma innych wykonywanych instrukcji, które odwołują się do elementów ELAPSED

lub Q struktury timera, wtedy timer nie będzie aktualizowany. Nowa aktualizacja nie nastąpi dopóty, dopóki ponownie albo nie zostanie wykonana instrukcja timera albo nie będą wykonane inne instrukcje wykorzystujące jako parametr element ELAPSED lub Q struktury timera.

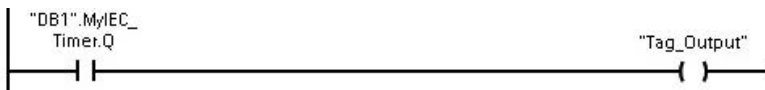
- Choć jest to nietypowe, można przypisać taką samą strukturę timera DB do wielu instrukcji timera. Ogólnie rzecz biorąc, aby uniknąć nieoczekiwanych interakcji, należy tylko stosować instrukcję jednego timera (TP, TON, TOF, TONR) na strukturę timera DB.

Samoresetujące się timery są przydatne do uruchomienia działań, które muszą występować okresowo. Zazwyczaj samoresetujące timery są tworzone przez umieszczenie styku normalnie zwartego, który odwołuje się do bitu timera przed instrukcją timera. Ta sieć timera zazwyczaj znajduje się nad jedną siecią lub wieloma podległymi sieciami, które używają bitu timera do uruchomienia działań. Gdy działanie timera kończy się (upływa czas określony przez wstępnie zadaną wartość), bit timera jest ustawiony na ON (włączenie) na okres jednego cyklu programu, co pozwala na wykonanie operacji logicznych podległej sieci sterowanej przez bit timera. Po kolejnym wykonaniu działań w sieci timera, styk normalnie zwarty zostanie ustawiony na OFF (wyłączenie), i w ten sposób nastąpi resetowanie timera i kasowanie jego bitu. Podczas tworzenia samoresetującego timera, takiego jak ten, przed instrukcją timera nie należy używać elementu "Q" struktury DB timera jako parametru dla styku normalnie zwartego. Zamiast tego należy użyć tagu podłączonego do wyjścia "Q" z instrukcji timera przeznaczonej na ten cel. Należy unikać dostępu do elementu Q struktury DB timera, ponieważ powoduje to aktualizację timera i jeśli timer zostanie zaktualizowany ze względu na styk normalnie zwarty, wtedy styk natychmiast zresetuje instrukcję timera. Wyjście Q instrukcji timera nie będzie ustawione na ON (włączenie) dla jednego cyklu programu i działania w podległej sieci nie będą wykonywane.

Cewki timerów -(TP)-, -(TON)-, -(TOF)-, oraz-(TONR)- muszą być zawsze ostatnią instrukcją w sieci. Jak zostało to pokazane w przykładzie timera, styk w podprogramie sprawdza bit Q w polu IEC_Timer bloku DB dla cewki timera. Podobnie należy adresować element ELAPSED w polu IEC_Timer bloku DB, jeśli wykorzystuje się wartość upłyniętego czasu w programie.



Timery impulsowe startują przy zmianie z 0 na 1 bitu Tag_Input. Ich praca trwa przez czas określony w polu Tag_Time.



Tak długo jak timer pracuje, stan bitów w DB1.MyIEC_Timer.Q oraz Tag_Output będzie równy 1. Kiedy wartość z Tag_Time zostanie odliczona, DB1.MyIEC_Timer.Q oraz Tag_Output przyjmą wartość 0.

6.3.6. Liczniki

Instrukcje dotyczące liczników są stosowane do zliczania wewnętrznych zdarzeń w programie i zewnętrznych zdarzeń procesu:

- CTU jest to licznik zliczający w górę o wartość 1, gdy wartość parametru wejściowego CU zmienia się z 0 na 1.
- CTD jest to licznik zliczający w dół o wartość 1, gdy wartość parametru wejściowego CD zmienia się z 0 na 1.
- CTUD jest to licznik zliczający w górę lub w dół o wartość 1, gdy wartość parametru wejściowego CU lub CD zmienia się z 0 na 1.

S7-1200 ma także szybkie liczniki (HSC) (strona 114) do zliczania zdarzeń, które występują szybciej niż cykl wykonania OB.

Szybkość zliczania za pomocą instrukcji CTU, CTD i CTUD jest ograniczona przez częstotliwość cyklu programu wykonywanego przez CPU.

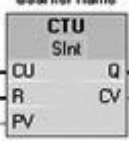
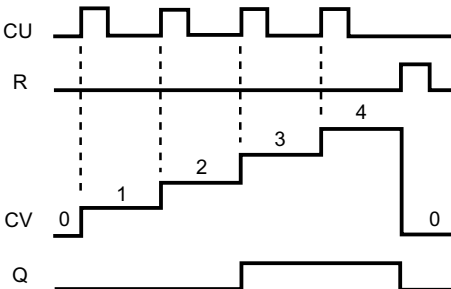
Uwaga

Jeśli zliczane zdarzenia występują wolniej niż częstotliwość wykonywania OB, należy użyć liczników CTU, CTD, lub CTUD. Jeśli zdarzenia występują częściej niż wynosi częstotliwość wywoływania cyklu programu wykonywanego przez CPU, należy użyć HSC.

Każdy licznik wykorzystuje do pamiętania danych licznika strukturę przechowywaną w bloku danych. Dla SCL należy najpierw stworzyć DB dla instrukcji pojedynczego licznika, aby móc się do niego odwoływać. Dla LAD i FBD tworzone są automatycznie DB, gdy tylko tworzona jest instrukcja licznika.

Liczba liczników użytych w programie użytkownika jest ograniczona tylko rozmiarem pamięci CPU. Poszczególne liczniki używają 3 bajtów (dla Sint lub USInt), 6 bajtów (dla Int lub UInt) lub 12 bajtów (dla DInt lub UInt).


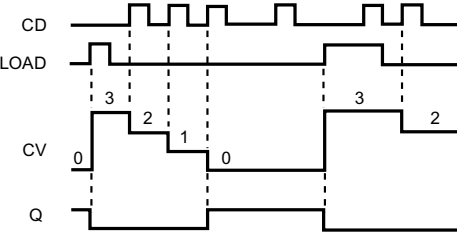
Tabela 6.17. Licznik CTU zliczający w górę

LAD/FBD	SCL	Cykl pracy
	<pre>"ctu_db".CTU(CU:=_bool_in, R:=_bool_in, PV:=_undef_in, Q=>_bool_out, CV=>_undef_out);</pre>	

Na powyższym rysunku przedstawiono przebieg czasowy w przypadku licznika CTU zliczającego liczby całkowite bez znaku (dla PV = 3).

- Jeżeli wartość parametru CV (*current count value* – bieżąca wartość zliczeń) jest większa lub równa wartości parametru PV (*preset count value* – ustalona wartość zliczeń), to parametr wyjściowy licznika Q = 1.
- Jeżeli wartość parametru kasującego R zmienia się z 0 na 1, to bieżąca wartość zliczeń zostaje skasowana do 0.

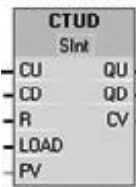
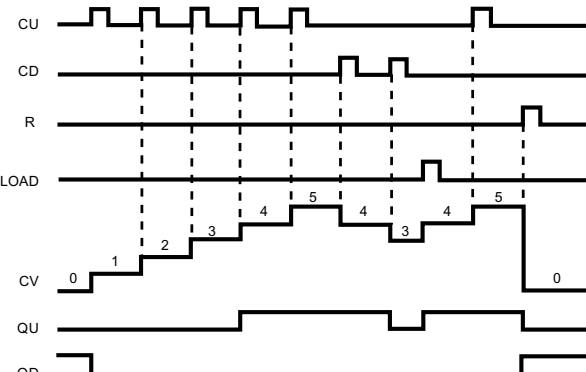
Tabela 6.18. Licznik CTD zliczający w dół

LAD/FBD	SCL	Cykl pracy
	<pre>"ctd_db".CTD(CD:=_bool_in, LOAD:=_bool_in, PV:=_undef_in, Q=>_bool_out, CV=>_undef_out);</pre>	

Na powyższym rysunku przedstawiono przebieg czasowy w przypadku licznika CTD zliczającego liczby całkowite bez znaku (dla PV = 3).

- Jeżeli wartość parametru CV (*current count value* – bieżąca wartość zliczeń) jest mniejsza lub równa 0, to parametr wyjściowy licznika Q = 1.
- Jeżeli wartość parametru LOAD zmienia się z 0 na 1, to wartość parametru PV (*preset count value* – ustalona wartość zliczeń) jest wpisywana do licznika jako nowa wartość CV (*current count value* – bieżąca wartość zliczeń).

Tabela 6.19. Licznik CTUD zliczający w górę i w dół

LAD/FBD	SCL	Cykl pracy
	<pre>"ctud_db".CTUD(CU:=_bool_in, CD:=_bool_in, R:=_bool_in, LOAD:=_bool_in, PV:=_undef_in, QU=>_bool_out, QD=>_bool_out, CV=>_undef_out);</pre>	

Na powyższym rysunku przedstawiono przebieg czasowy w przypadku licznika CTUD zliczającego liczby całkowite bez znaku (dla PV = 4).

- Jeżeli wartość parametru CV (*current count value* – bieżąca wartość zliczeń) jest równa lub większa od wartości parametru PV (*preset value* – ustalona wartość), to parametr wyjściowy licznika QU = 1.

- Jeżeli wartość parametru CV jest mniejsza lub równa 0, to parametr wyjściowy licznika QD = 1.
- Jeżeli wartość parametru LOAD zmienia się z 0 na 1, to wartość parametru PV jest wpisywana do licznika jako nowa wartość CV.
- Jeżeli wartość parametru kasującego R zmienia się z 0 na 1, to bieżąca wartość zliczeń zostaje skasowana do 0.

6.3.7. Instrukcja modulacji szerokości impulsu (PWM)

Tabela 6.20. Instrukcja CTRL_PWM

LAD/FBD	SCL	Opis
	<pre>"ctrl_pwm_db" (PWM:=_hw_pwm_in_, enable:=_bool_in_, busy=>_bool_out_, status=>_word_out_);</pre>	<p>Instrukcja CTRL_PWM dotycząca modulacji szerokości impulsów PWM (<i>Pulse Width Modulation</i>) zapewnia stały okres przebiegu wyjściowego ze zmiennym współczynnikiem wypełnienia. Wyjście PWM działa przez cały czas od chwili uruchomienia generując impulsy o określonej częstotliwości (okresie). Szerokość impulsów jest zmieniana zgodnie z potrzebami, tak by uzyskać pożądany efekt sterowania.</p>

Instrukcja CTRL_PWM wykorzystuje blok danych (DB) do przechowywania informacji o parametrach. Dla SCL należy najpierw stworzyć DB dla instrukcji pojedynczej instrukcji, aby móc się do niego odwoływać. STEP 7 dla LAD i FBD tworzy automatycznie DB, gdy tylko tworzona jest instrukcja. Bloki danych parametrów są sterowane za pomocą instrukcji CTRL_PWM.

Kiedy PLC po raz pierwszy wchodzi w tryb RUN, wtedy wartość szerokości impulsu przyjmuje wartość początkową określoną w konfiguracji urządzenia. Jeżeli trzeba zmienić szerokość impulsu, to użytkownik wpisuje wartości pod adres Q-word określony w konfiguracji urządzenia (*Output addresses/Start address*). W celu wpisania pożądanej szerokości impulsu pod adres Q-word korzysta się z instrukcji przesunięcia, konwersji, arytmetycznych lub bloku PID. Należy stosować właściwy zakres dla wartości Q-word (procenty, tysięczne, dziesięciotysięczne lub format analogowy S7).



Współczynnik wypełnienia można określić w procentach jako część okresu (0–100%), w tysięcznych (0–1000), dziesięciotysięcznych (1–10 000). Szerokość impulsu może się zmieniać od 0 (brak impulsu, zawsze OFF) do pełnej skali (brak impulsu, zawsze ON).

- ① Okres cyklu
- ② Szerokość impulsu

Ponieważ sygnał wyjściowy PWM może się zmieniać od zera do pełnego zakresu, więc jest to sygnał cyfrowy bardzo zbliżony do sygnału analogowego. Na przykład, sygnał wyjściowy PWM może być zastosowany do sterowania szybkością silnika od zatrzymania do pełnej prędkości albo może być użyty do sterowania położeniem zaworu od całkowitego zamknięcia do pełnego otwarcia.

6.4. Tworzenie dzienników zdarzeń w plikach .csv

Program użytkownika może używać instrukcji dziennika zdarzeń do zbierania wartości danych z czasu pracy w trwałych plikach dziennika. Pliki dziennika są przechowywane w pamięci typu flash (dla CPU lub karty pamięci) w standardowym formacie CSV (*Comma Separated Value*). Rekordy danych są zorganizowane w cyklicznych plikach o predefiniowanych rozmiarach.

Instrukcje dziennika zdarzeń są użyte w programie do tworzenia, otwierania, zapisywania rekordów i zamykania plików dziennika. Użytkownik decyduje o tym, jakie dane mają zostać zapisane tworząc bufor danych, który definiuje pojedynczy rekord. Bufor danych jest używany do tymczasowego przechowania nowego rekordu dziennika. Nowe wartości bieżące muszą zostać przeniesione programowo do bufora podczas pracy. Kiedy wszystkie bieżące wartości danych zostaną odświeżone, można użyć instrukcji DataLogWrite do przeniesienia danych z bufora do dziennika zdarzeń.

Ze strony przeglądarki plików na web serwerze można otwierać, edytować, zapisywać, zmieniać nazwy i usuwać pliki dziennika danych. Aby przeglądać przeglądarkę plików potrzebne są uprawnienia do odczytu, natomiast do edycji, modyfikacji, usuwania lub zmiany nazw plików dziennika danych trzeba mieć uprawnienia do modyfikacji.

Należy użyć instrukcji DataLog do programowego przechowania danych procesu podczas pracy w pamięci flash CPU. Rekordy danych są zorganizowane w cyklicznych plikach o predefiniowanych rozmiarach. Nowe rekordy są dołączane do dzienników zdarzeń. Po przekroczeniu maksymalnej liczby rekordów, nowe nadpisują najstarsze rekordy. Aby zapobiec nadpisaniu jakiegokolwiek rekordu należy użyć instrukcji DataLogNewFile. Nowe rekordy danych zostaną zapisane w nowym pliku dziennika, podczas gdy stare pliki dziennika pozostaną w CPU.

Tabela 6.21. Instrukcja DataLogWrite

LAD/FBD	SCL	Opis
	<pre>"DataLogWrite_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogWrite zapisuje rekordy danych w określonym dzienniku. Istniejący docelowy dziennik musi być otwarty.</p> <p>Należy załadować programowo bufor rekordów z bieżącymi wartościami danych procesu a następnie uruchomić instrukcję DataLogWrite, aby przenieść rekordy danych do dziennika.</p> <p>Jeżeli nastąpi przerwa w zasilaniu podczas wykonywania operacji DataLogWrite, to rekordy danych, które były w trakcie przenoszenia do dziennika mogą zostać utracone.</p>

Tabela 6.22. Instrukcje DataLogCreate and DataLogNewFile

LAD/FBD	SCL	Opis
<p>DataLogCreate_DB</p>	<pre>"DataLogCreate_DB" (req:=_bool_in_, records:=_uint_in_, format:=_uint_in_, timestamp:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_string_inout_, ID:=_dword_inout_, header:=_variant_inout_, data:=_variant_inout_);</pre>	<p>DataLogCreate¹ tworzy i inicjuje pliki dziennika przechowywane w katalogu DataLogs w CPU. Pliki dziennika tworzone są w predefiniowanym rozmiarze.</p>
<p>DataLogNewFile_DB</p>	<pre>"DataLogNewFile_DB" (req:=_bool_in_, records:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_DataLog_out_, ID:=_dword_inout_);</pre>	<p>DataLogNewFile¹ pozwala na stworzenie nowego pliku dziennika bazując na już istniejącym pliku dziennika. Nowy plik będzie stworzony i otwarty niejawnie bazując na określonej nazwie. Rekord nagłówka zostanie zduplikowany z istniejącego dziennika wraz z pierwotnymi właściwościami dziennika. Pierwotny dziennik zostanie niejawnie zamknięty.</p>

¹ Operacje DataLogCreate i DataLogNewFile obejmują wiele cykli skanu programu. Czas potrzebny do stworzenia pliku dziennika zależy od struktury rekordu i liczby rekordów. Przed użyciem nowego dziennika dla innych operacji, należy wymusić logiką programu przejście bitu DONE w stan wysoki.

Tabela 6.23. Instrukcje DataLogOpen and DataLogClose

LAD/FBD	SCL	Opis
<p>DataLogOpen_DB</p>	<pre>"DataLogOpen_DB" (req:=_bool_in_, mode:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_string_inout_, ID:=_dword_inout_);</pre>	<p>Instrukcja DataLogOpen otwiera istniejący plik dziennika. Dziennik musi być otwarty przed zapisem do niego nowych rekordów. Dzienniki mogą być otwierane i zamykane niezależnie. 8 dzienników może być otwartych równocześnie.</p>
<p>DataLogClose_DB</p>	<pre>"DataLogClose_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>Instrukcja DataLogClose zamyka otwarte pliki dzienników. Po zamknięciu dzienników operacje DataLogWrite będą skutkowały błędem. Żadne operacje zapisu nie mogą zostać przeprowadzone w tym dzienniku, dopóki nie zostanie przeprowadzona kolejna operacja DataLogOpen operation.</p> <p>Przejście sterownika w tryb STOP zamyka wszystkie pliki dziennika.</p>

6.5. Monitorowanie i testowanie programu

6.5.1. Tablice monitorujące i tablice wymuszeń

Do monitorowania i modyfikowania wartości wykonywanego programu użytkownika w podłączonym CPU stosuje się tablice monitorujące (*watch tables*). Użytkownik może w swoim projekcie stworzyć i zapisać tablice monitorujące, dostarczające danych dla wielu środowisk testowych.

Tablica monitorująca pozwala monitorować i interaktywnie współdziałać z CPU w czasie, kiedy wykonuje on program użytkownika. Można wyświetlać i zmieniać wartości nie tylko zmiennych bloków kodu i danych, ale również obszarów pamięci CPU, włączając w to wejścia i wyjścia (I i Q), peryferyjne wejścia i wyjścia (I:P), pamięć bitową (M) i bloki danych (DB). Posługując się tablicą monitorującą można uaktywniać wyjścia fizyczne (Q:P) CPU w trybie STOP. Na przykład, podczas testowania okablowania CPU, można przypisywać wyjściom określone wartości.

Step 7 pozwala również wymuszać lub ustawiać określone wartości zmiennych w tablicy wymuszeń. Więcej informacji na temat wymuszania znajduje się w rozdziale „Narzędzia online i diagnostyczne”, w części dotyczącej wymuszaniu wartości w CPU (strona 257).

Uwaga

Wymuszone wartości są przechowywane w CPU, a nie w tablicy monitorującej.

Nie można wymusić stanu wejścia (lub adresu „I”). Można natomiast wymusić stan peryferyjnego wejścia. W tym celu należy dodać do tego adresu „:P” (przykładowo: „On:P”).

STEP 7 zapewnia również możliwość śledzenia i rejestrowania zmiennych programu za pomocą warunków wyzwiania.

6.5.2. Cross-reference lista użycia zmiennych w projekcie

Okno inspektora wyświetla informacje w *Cross-reference* mówiące gdzie w projekcie używane są określone obiekty, np. w programie użytkownika, w CPU lub w HMI. Zakładka *Cross-reference* wyświetla instancje, w których dany obiekt został użyty i jakie inne obiekty go wykorzystują. Okno inspektora zawiera także bloki, które są dostępne w *Cross-reference* tylko w trybie online. Aby wyświetlić *Cross-reference* należy użyć komendy *Show cross-references* (w widoku projektu w zakładce *Tools*).

Uwaga

Nie trzeba zamykać edytorów, aby zobaczyć informacje o *Cross-reference*.

Można sortować wpisy w *Cross-reference*. Lista *Cross-reference* ma przegląd używanych adresów i znaczników w programie użytkownika.

- Podczas tworzenia i edycji programu użytkownik zachowuje przegląd argumentów, zmiennych i wywołań bloków, które zostały użyte.
- Można przejść z *Cross-reference* bezpośrednio do punktu użycia argumentu lub zmiennej.
- Podczas testowania programu lub w przypadku problemów użytkownik jest informowany, który obszar pamięci jest przetwarzany przez daną komendę, w którym bloku, która zmienna jest używana, w którym ekranie i który blok jest wywoływany przez dany blok.

Tabela 6.24. Elementy *Cross-reference*

Kolumna	Opis
Object	Nazwa obiektu, który używa obiektów niższych poziomów lub jest używany przez obiekty niższego poziomu.
Quantity	Liczba użyczeń
Location	Każda używana lokacja, przykładowo sieć
Property	Specjalne właściwości obiektów odniesienia, np. nazwy zmiennych w wieloinstancyjnych zgłoszeniach.
as	Pokazuje dodatkowe informacje dotyczące obiektu, np. która instancja DB jest używana jako szablon lub wielokrotna instancja.
Access	Typ dostępu, może wystąpić dostęp do odczytu argumentu (R) lub/i dostęp do zapisu (W).
Address	Adres argumentu
Type	Informacja o typie języka użytym przy tworzeniu obiektu.
Path	Ścieżka obiektu w drzewie projektu.

6.5.3. Struktura wywołania do badania hierarchii połączeń

Struktura wywołania opisuje hierarchię połączeń bloków w programie. Zapewnia to podgląd użytych bloków, wywołań bloków, relacji pomiędzy nimi, wymagań dotyczących używanych danych i statusu bloków. Można otworzyć edytor programu i edytować blok z poziomu struktury wywołującej.

Wyświetlanie struktury wywołań pozwala użytkownikowi na podgląd listy bloków użytych w programie. STEP 7 podświetla pierwszy poziom struktury wywołującej i podaje każdy blok, który nie jest wywołany przez inny blok w programie. Pierwszy poziom struktury wywołującej wyświetla OB i każdy FC, FB i DB, który nie jest wywoływany przez OB. Jeśli blok kodu wywołuje inny blok, to wywołany blok jest pokazany jako wcięcie pod blokiem wywołującym. Struktura wywołująca wyświetla tylko te bloki, które są wywoływane przez blok kodu.

Użytkownik może wyświetlać selektywnie tylko te bloki, które powodują konflikty w strukturze wywołania. Następujące warunki powodują konflikty:

- Bloki, który wykonują wywołania ze starszymi i nowszymi znacznikami czasowymi.
- Bloki wywołujące bloki ze zmodyfikowanym interfejsem.
- Bloki, które używają zmiennych ze zmodyfikowanym adresem i/lub typem danych.
- Bloki, które nie są wywoływane pośrednio lub bezpośrednio przez OB.

- Bloki, które wywołują nieistniejące lub brakujące bloki.

Użytkownik może grupować wywołania bloków i bloki danych w jednej grupie. Należy użyć listy rozwijalnej do pokazania połączeń pomiędzy poszczególnymi lokalizacjami wywołań.

Można również przeprowadzić kontrolę spójności, aby pokazać konflikty znaczników czasowych. Zmiana znaczników czasowych wywołań bloków podczas lub po wygenerowaniu programu może prowadzić do konfliktów znaczników czasowych, które wywołane będą brakiem spójności pomiędzy blokami wywołującymi i wywołanymi.

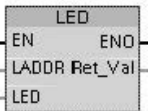
- Większość znaczników czasowych i konfliktów interfejsów może być poprawiona przez ponowną kompilację bloków kodu.
- W przypadku niepowodzenia kompilacji, aby usunąć niespójność należy użyć linku w kolumnie *Details*, aby zobaczyć źródło problemu w edytorze programu. Można wtedy ręcznie wyeliminować wszystkie niespójności.
- Wszystkie bloki zaznaczone na czerwono muszą zostać ponownie skompilowane.

6.5.4. Instrukcje diagnostyczne do monitorowania sprzętu

6.5.4.1. Odczytywanie stanu diod LED na sterowniku

Instrukcje LED pozwalają programowi użytkownika określić stan fizycznych diod LED na CPU. Można użyć tych instrukcji do zaprogramowania zmiennej dla urządzenia HMI.

Tabela 6.25. Instrukcje LED

LAD/FBD	SCL	Opis
	<pre>ret_val := #LED(laddr:=_hw_io_in_, LED:=_uint_in_);</pre>	<p>RET_VAL zwraca następujące stany diod LED z CPU:</p> <ul style="list-style-type: none"> • RUN/STOP: zielona lub żółta • Error(błąd): czerwona • MAINT (podtrzymanie): żółta • Link: zielona • Tx/Rx (transmisja/otrzymywanie): żółta

6.5.4.2. Instrukcje odczytujące stan diagnostyczny urządzeń

STEP 7 zawiera instrukcje do odczytania informacji o stanie urządzeń użytych w sieci.

Tabela 6.26. Instrukcje diagnostyczne

LAD/FBD	SCL	Opis
	<pre>ret_val := #GET_DIAG(mode:=_uint_in_, laddr:=_hw_any_in_, channel:=_uint_in_, cnt_diag=>_uint_out_, diag:=_uint_inout_, detail:=_variant_inout_);</pre>	Instrukcja GET_DIAG odczytuje informacje diagnostyczne z określonego urządzenia w sieci.
	<pre>ret_val := DeviceStates(laddr:=hw_io_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	Instrukcja DeviceStates odczytuje informacje o stanie urządzeń PROFINET lub PROFIBUS.
	<pre>ret_val := ModuleStates(laddr:=_hw_io_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	Instrukcja ModuleStates odczytuje informacje o stanie modułów PROFINET lub PROFIBUS.

6.6. Szybkie liczniki (HSC)

Szybkie liczniki (*high-speed counters* – HSC) używane są do zliczania zdarzeń występujących częściej niż wynosi częstość wywoływania cyklu programu. Instrukcja CTRL_HSC kontroluje szybkie liczniki.

Uwaga

Jeśli zliczane zdarzenia występują wolniej niż częstość wykonywania OB, należy użyć liczników CTU, CTD, lub CTUD. Jeśli zdarzenia występują częściej niż wynosi częstość wywoływania cyklu programu wykonywanego przez CPU, należy użyć HSC.

Należy skonfigurować parametry każdego HSC na etapie definiowania projektu podczas konfiguracji urządzenia PLC. Podczas konfiguracji urządzenia HSC dokonuje się wyboru trybów zliczania, połączeń I/O, przypisuje przerwania, oraz definiuje czy urządzenie ma pracować jako szybki licznik, czy jako miernik częstotliwości impulsów.

Tabela 6.27. Instrukcja CTRL_HSC

LAD/FBD	SCL	Opis
<p>"Counter name"</p>	<pre>"counter_name" (hsc:= _hw_hsc_in_, HSC:= , DIR:= _bool_in_, CV:= _bool_in_, RV:= _bool_in_, Period:= _bool_in_, New_DIR:= _int_in_, New_CV:= _int_in_, New_RV:= _dint_in_, New_Period:= _int_in_, Busy:= _bool_out_, Status:= _word_out_);</pre>	<p>Każda instrukcja CTRL_HSC używa struktury przechowywanej w DB do utrzymania danych. Licznik HSC używa struktury przechowywanej w bloku danych to przechowanie danych licznika. Dla SCL należy najpierw stworzyć DB dla instrukcji pojedynczego licznika, aby móc się do niego odwoływać. Dla LAD i FBD tworzone są automatycznie DB, gdy tylko tworzona jest instrukcja licznika.</p>

Instrukcja CTRL_HSC jest zwykle umieszczana w OB przerwania sprzętowego, który jest wykonywany wtedy, kiedy pojawia się sprzętowe przerwanie pochodzące od licznika. Na przykład, jeśli warunek CV = RV wyzwała przerwanie pochodzące od licznika, to blok kodu OB przerwania sprzętowego wykonuje instrukcję CTRL_HSC, która może zmienić wartość referencyjną poprzez wczytanie nowej wartości NEW_RV.

Uwaga

Bieżąca wartość zliczeń nie jest dostępna jako parametr instrukcji CTRL_HSC. Adres obrazu procesu pod którym jest przechowywana bieżąca wartość zliczeń jest definiowany podczas konfiguracji sprzętowej szybkiego licznika. Użytkownik może w programie zrealizować bezpośredni odczyt wartości zliczeń. Wczytana do programu liczba będzie równa rzeczywistej liczbie zliczeń w chwili dokonywania odczytu – trzeba jednak pamiętać, że licznik kontynuuje zliczanie szybkich zdarzeń. Może się więc zdarzyć, że faktyczna liczba zliczeń zmieni się zanim program zakończy operację korzystając ze starej wartości.

Za pomocą parametrów instrukcji CTRL_HSC można kontrolować proces zliczania z programu:

- Kierunek zliczania jako wartość parametru NEW_DIR.
- Bieżące zliczenia jako wartość parametru NEW_CV.
- Wartość referencyjna jako wartość parametru NEW_RV.
- Wartość okresu jako wartość parametru NEW_PERIOD (tylko w trybie pomiaru częstotliwości).

Odpowiednie wartości parametrów NEW_xxx są wpisywane do licznika wtedy, kiedy podczas wykonywania instrukcji CTRL_HSC następujące znaczniki boolowskie są ustawione na 1. Wielokrotne żądania (w tym samym czasie jest ustawiony więcej niż jeden znacznik) są realizowane w trakcie pojedynczego wykonania in-

strukcji CTRL_HSC. Ustawienie następujących znaczników boolowskich na 0 nie spowoduje żadnej zmiany.

- DIR = 1 jest żądaniem wpisania do licznika wartości NEW_DIR,
- CV = 1 jest żądaniem wpisania do licznika wartości NEW_CV,
- RV = 1 jest żądaniem wpisania do licznika wartości NEW_RV,
- PERIOD = 1 jest żądaniem wpisania do licznika wartości NEW_PERIOD.

6.6.1. Działanie szybkiego licznika

Szybkie liczniki HSC (*High Speed Counter*) mogą zliczać zdarzenia, których szybkość występowania jest większa od szybkości wykonywania cyklicznego OB. Jeśli zdarzenia, które mają być zliczane, występują wolniej niż szybkość wykonywania bloku OB, to można użyć standardowych instrukcji liczników CTU, CTD lub CTUD. Jeżeli szybkość występowania zdarzeń jest większa od szybkości wykonywania bloku OB, to należy użyć szybszego licznika HSC. Instrukcja CTRL_HSC pozwala programowi użytkownika zmienić programowo niektóre parametry HSC.

Przykładowo liczniki HSC mogą zostać użyte jako wejście dla przetwornika obrotowego. Przetwornik ten posiada określoną liczbę impulsów na obrót i impuls kasujący, który występuje raz na jeden obrót. Zegar(y) i impuls kasujący z przetwornika obrotowego są wejściami HSC.

HSC jest ładowany pierwszym z kilku ustawień, a wyjścia są aktywowane na okres czasu, w którym bieżące zliczenie jest mniejsze niż wartość bieżącego ustawienia. HSC dostarcza przerwanie, gdy bieżące zliczenie jest równe wartości bieżącego ustawienia, gdy pojawia się reset, a także, gdy następuje zmiana kierunku obrotów.

Po osiągnięciu stanu „bieżące zliczenie jest równe wartości ustawienia” następuje zdarzenie przerwania, ładowana jest nowa wartość ustawienia oraz ustalany jest następny stan na wyjściach. Kiedy wystąpi zdarzenie przerwania przez reset, przywraca się pierwsze ustawienie oraz pierwsze stany na wyjściach, i cykl się powtarza.

Ponieważ przerwania zdarzają się dużo rzadziej niż zliczanie HSC, dokładna kontrola szybkich operacji może zostać zrealizowana z małym wpływem czasu skanu CPU. Metoda zamieszczenia przerwania pozwala przy każdym załadowaniu nowej nastawy na przeprowadzenie przerwania w oddzielnym działaniu, co ułatwia kontrolę stanu licznika. Alternatywnie wszystkie przerwania mogą zostać zrealizowane w jednym zdarzeniu przerwania.

Wybór kanałów wejściowych HSC

Do wyboru kanałów wejściowych HSC należy użyć poniższej tabeli i zapewnić, aby kanały wejściowe CPU i SB, które są podłączane, mogły obsługiwać maksymalne szybkości impulsów w sygnałach procesowych.

Uwaga**Kanały wejściowe CPU i SB (firmware w wersji V4 lub nowszej) mają konfigurowalne czasy reakcji filtra na wejściach**

Wcześniejsze wersje firmware mają stałe kanały wejściowe HSC i stałe czasy reakcji filtra, które nie mogą być zmienione.

Wersje firmware V4 lub nowsze umożliwiają przypisanie kanałów wejściowych i czasów reakcji filtra. Domyślne ustawienie filtra wejściowego na 6,4 ms może być zbyt wolne dla sygnałów procesowych. Czasy reakcji filtra dla wejść HSC muszą być zoptymalizowane dla aplikacji HSC. Więcej informacji na ten temat znajduje się w części „Konfigurowanie czasów reakcji filtra na wejściach cyfrowych”.

Tabela 6.28. Wejście CPU: częstotliwość maksymalna

CPU	Kanał wejściowy CPU	Tryb 1 lub 2-fazowy	Tryb kwadraturowy fazy A/B
1211C	la.0 do la.5	100 kHz	80 kHz
1212C	la.0 do la.5	100 kHz	80 kHz
	la.6, la.7	30 kHz	20 kHz
1214C and 1215C	la.0 do la.5	100 kHz	80 kHz
	la.6 do lb.5	30 kHz	20 kHz
1217C	la.0 do la.5	100 kHz	80 kHz
	la.6 do lb.1	30 kHz	20 kHz
	lb.2 do lb.5 (.2+, .2- do .5+, .5-)	1 MHz	1 MHz

Tabela 6.29. Wejście płytki sygnałowej (SB): częstotliwość maksymalna (płytki opcjonalna)

Płytki sygnałowa (SB)	Kanał wejściowy SB	Tryb 1 lub 2-fazowy	Tryb kwadraturowy fazy A/B
SB 1221, 200 kHz	le.0 do le.3	200 kHz	160 kHz
SB 1223, 200 kHz	le.0, le.1	200 kHz	160 kHz
SB 1223	le.0, le.1	30 kHz	20 kHz

Wybór funkcjonalności HSC

Wszystkie HSC działają w ten sam sposób dla tego samego trybu zliczania. Tryb licznika, sterowanie kierunkiem, oraz początkowy kierunek są przypisywane w konfiguracji CPU dla właściwości funkcji HSC.

Istnieją cztery podstawowe typy HSC:

- Jednofazowe liczniki z wewnętrznym sterowaniem kierunku.
- Jednofazowe liczniki z zewnętrznym sterowaniem kierunku.
- Dwufazowe liczniki z dwoma wejściami zegarowymi.
- Liczniki kwadraturowe A/B.

Użytkownik może użyć każdego typu HSC z lub bez wejścia kasującego. Podczas aktywacji wejściem kasującym (z pewnymi ograniczeniami, patrz tabela

poniżej), bieżąca wartość jest czyszczona aż do momentu deaktywacji wejścia kasującego.

- Funkcje częstotliwości: Niektóre tryby HSC pozwalają na jego konfigurację (typ zliczania) w celu raportowania wartości częstotliwości zamiast bieżącej wartości impulsów. Dostępne są trzy okresy pomiaru częstotliwości okresu: 0,01, 0,1 i 1,0 sekundy.

Okres pomiaru częstotliwości określa jak często HSC zlicza i raportuje nową wartość częstotliwości. Raportowana wartość częstotliwości jest średnią wartością określoną przez całkowitą liczbę impulsów w ostatnim okresie pomiarowym. Jeżeli częstotliwość gwałtownie się zmieniła to raportowana wartość będzie wartością pośrednią pomiędzy najwyższą i najniższą częstotliwością zmierzoną w danym okresie. Częstotliwość jest zawsze raportowana w jednostkach Hertz (impulsy na sekundę) niezależnie od ustawień zakresu pomiaru częstotliwości.

- Tryby zliczania: Poniższa tabela przedstawia wejścia stosowane przez zegar, kierunek zliczania i wejście zerujące związane z HSC.

Jedno wejście nie może zostać użyte do dwóch różnych funkcji, ale każde wejście nieużywane w danym czasie przez HSC może być użyte do innych celów. Na przykład, jeśli HSC1 jest w trybie, który wykorzystuje dwa wejścia wbudowane, ale nie korzysta z trzeciego wejścia zewnętrznego resetu (domyślne przypisanie I0.3), wtedy I0.3 może być wykorzystywane dla przerwań wyzwalanych zboczem sygnału lub dla HSC 2.

Tabela 6.30. Tryby zliczania dla HSC

Typ	Wejście 1	Wejście 2	Wejście 3	Funkcja
Jednofazowe liczniki z wewnętrznym sterowaniem kierunku	Zegar	(Opcjonalnie: kierunek)	–	Zliczanie lub częstotliwość
			Reset	Zliczanie
Jednofazowe liczniki z zewnętrznym sterowaniem kierunku	Zegar	Kierunek	–	Zliczanie lub częstotliwość
			Reset	Zliczanie
Dwufazowe liczniki z dwoma wejściami zegarowymi.	Zegar liczący w górę	Zegar liczący w dół	–	Zliczanie lub częstotliwość
			Reset	Zliczanie
Liczniki kwadraturowe A/B	Faza A	Faza B	–	Zliczanie lub częstotliwość
			Reset ¹	Zliczanie

¹ Dla enkodera: Faza Z, Home

Adresy wejściowe HSC

Podczas konfiguracji CPU istnieje możliwość włączenia i skonfigurowania wejść sprzętowych *Hardware inputs* dla każdego HSC.

Wszystkie wejścia HSC muszą być podłączone do zacisków modułu CPU lub opcjonalnej płytki sygnałowej, którą włącza się od strony frontowej modułu CPU.

Uwaga

Jak pokazano w poniższej tabeli, domyślne przypisania dla ewentualnych sygnałów dla różnych CPU pokrywają się. Przykładowo, opcjonalne wejście resetujące dla HSC1 używa tego samego wejścia jak jedno z wejść HSC2.

Dla CPU w wersji V4 lub nowszej, można ponownie przypisać wejścia HSC podczas konfiguracji CPU. Nie ma konieczności użycia domyślnych przypisań wejść.

Należy zawsze upewnić się, czy HSC zostały tak skonfigurowane, że jedno wejście **nie** jest używane przez kilka HSC.

Poniższe tabele przedstawiają domyślne przypisanie wejść HSC dla wbudowanych I/O jednostek CPU oraz opcjonalnej płytki sygnałowej (SB). (Jeśli wybrany model SB posiada tylko 2 wejścia, dostępne są tylko wejścia 4.0 i 4.1).

Definicje dla tabeli wejść HSC:

- Jednofazowe: C jest wejściem zegara (*Clock*), [d] jest wejściem kierunku (opcjonalne), a [R] jest wejściem zewnętrznego resetu (opcjonalne).
(Reset jest dostępny tylko w trybie zliczania.)
- Dwufazowe: CU jest wejściem zegara odliczającego wprzód (*Clock Up*), CD jest wejściem zegara odliczającego wstecz, a [R] jest opcjonalnym wejściem zewnętrznego resetu (opcjonalne).
(Reset jest dostępny tylko w trybie zliczania.)
- Kwadraturowe AB-fazowe: A jest wejściem A zegara (*Clock A*), B jest wejściem B zegara (*Clock B*), a [R] jest wejściem zewnętrznego resetu (opcjonalne). (Reset jest dostępny tylko w trybie zliczania.)

Tabela 6.31. CPU 1211C: domyślne przypisania adresów HSC

Tryb licznika HSC		Wbudowane wejście CPU (domyślne 0.x)						Opcjonalne wejście SB (domyślne 4.x) ¹			
		0	1	2	3	4	5	0	1	2	3
HSC 1	1-fazowy	C	[d]		[R]			C	[d]		[R]
	2-fazowy	CU	CD		[R]			CU	CD		[R]
	AB-fazowy	A	B		[R]			A	B		[R]
HSC 2	1-fazowy		[R]	C	[d]				[R]	C	[d]
	2-fazowy		[R]	CU	CD				[R]	CU	CD
	AB-fazowy		[R]	A	B				[R]	A	B
HSC 3	1-fazowy					C	[d]	C	[d]		[R]
	2-fazowy										
	AB-fazowy										
HSC4	1-fazowy					C	[d]	C	[d]		[R]
	2-fazowy					CU	CD				
	AB-fazowy					A	B				
HSC5	1-fazowy							C	[d]		[R]
	2-fazowy							CU	CD		[R]
	AB-fazowy							A	B		[R]
HSC6	1-fazowy								[R]	C	[d]
	2-fazowy								[R]	CU	CD
	AB-fazowy								[R]	A	B

¹ SB tylko z 2 wejściami cyfrowymi dostarcza tylko wejść 4.0 i 4.1.

Tabela 6.32. CPU 1212C: domyślne przypisania adresów HSC

Tryb licznika HSC		Wbudowane wejście CPU (domyślne 0.x)							Opcjonalne wejście SB (domyślne 4.x) ¹				
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1	1-fazowy	C	[d]		[R]					C	[d]		[R]
	2-fazowy	CU	CD		[R]					CU	CD		[R]
	AB-fazowy	A	B		[R]					A	B		[R]
HSC 2	1-fazowy		[R]	C	[d]						[R]	C	[d]
	2-fazowy		[R]	CU	CD						[R]	CU	CD
	AB-fazowy		[R]	A	B						[R]	A	B
HSC 3	1-fazowy					C	[d]		[R]	C	[d]		[R]
	2-fazowy					CU	CD		[R]				
	AB-fazowy					A	B		[R]				
HSC4	1-fazowy						[R]	C	[d]	C	[d]		[R]
	2-fazowy						[R]	CU	CD				
HSC5	AB-fazowy						[R]	A	B				
	1-fazowy									C	[d]		[R]
	2-fazowy									CU	CD		[R]
	AB-fazowy									A	B		[R]
HSC6	1-fazowy										[R]	C	[d]
	2-fazowy										[R]	CU	CD
	AB-fazowy										[R]	A	B

¹ SB tylko z 2 wejściami cyfrowymi dostarcza tylko wejść 4.0 i 4.1.

Tabela 6.33. CPU 1214C, CPU 1215C, oraz CPU1217C: domyślne przypisania adresów HSC (tylko wejścia wbudowane, opcjonalne adresy SB są pokazane w następnym tabeli)

Tryb licznika HSC		Bajt 0 wejścia cyfrowego (domyślne: 0.x)							Bajt 1 wejścia cyfrowego (domyślne: 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1	1-fazowy	C	[d]		[R]										
	2-fazowy	CU	CD		[R]										
	AB-fazowy	A	B		[R]										
HSC 2	1-fazowy		[R]	C	[d]										
	2-fazowy		[R]	CU	CD										
	AB-fazowy		[R]	A	B										
HSC 3	1-fazowy					C	[d]		[R]						
	2-fazowy					CU	CD		[R]						
	AB-fazowy					A	B		[R]						
HSC 4	1-fazowy						[R]	C	[d]						
	2-fazowy						[R]	CU	CD						
	AB-fazowy						[R]	A	B						

Tryb licznika HSC		Bajt 0 wejścia cyfrowego (domyślne: 0.x)							Bajt 1 wejścia cyfrowego (domyślne: 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 5	1-fazowy									C	[d]	[R]			
	2-fazowy									CU	CD	[R]			
	AB-fazowy									A	B	[R]			
HSC 6	1-fazowy												C	[d]	[R]
	2-fazowy												CU	CD	[R]
	AB-fazowy												A	B	[R]

Tabela 6.34. Opcjonalne SB w jednostkach CPU w tabeli powyżej: domyślne przypisanie adresów HSC

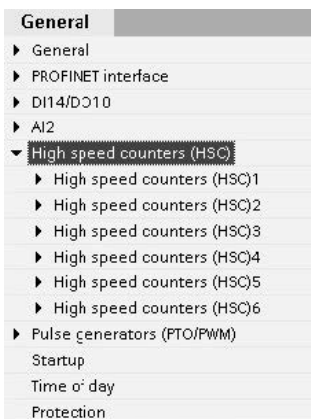
HSC		Opcjonalne wejścia SB (domyślne: 4.x) ¹			
		0	1	2	3
HSC 1	1-fazowy	C	[d]		[R]
	2-fazowy	CU	CD		[R]
	AB-fazowy	A	B		[R]
HSC 2	1-fazowy		[R]	C	[d]
	2-fazowy		[R]	CU	CD
	AB-fazowy		[R]	A	B
HSC 5	1-fazowy	C	[d]		[R]
	2-fazowy	CU	CD		[R]
	AB-fazowy	A	B		[R]
HSC 6	1-fazowy		[R]	C	[d]
	2-fazowy		[R]	CU	CD
	AB-fazowy		[R]	A	B

¹ SB tylko z 2 wejściami cyfrowymi dostarcza tylko wejść 4.0 i 4.1.

Uwaga

Cyfrowe punkty I/O wykorzystywane przez urządzenia szybkich liczników są przypisywane podczas konfiguracji urządzenia CPU. Kiedy adresy cyfrowych punktów I/O zostaną przypisane do urządzeń HSC, to wartości przypisanych adresów punktów I/O nie mogą być modyfikowane przez funkcję wymuszenia w tablicy monitorującej (*watch table*).

6.6.2. Konfiguracja HSC



CPU pozwala na konfigurację do 6 HSC. Należy edytować zakładkę *Properties* dla CPU, aby skonfigurować parametry każdego HSC.

Instrukcja CTRL_HSC w programie użytkownika pozwala na kontrole działania HSC.

Załączanie określonego HSC odbywa się przez wybranie opcji *Enable* dla tego HSC.



OSTRZEŻENIE

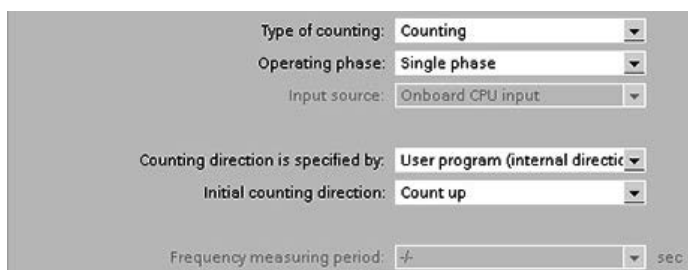
Zagrożenia wynikające ze zmian ustawienia czasu reakcji filtra dla cyfrowych kanałów wejściowych

Jeśli czas reakcji filtra dla kanału wejścia cyfrowego zostanie zmieniony w stosunku do poprzedniego ustawienia, to może być konieczne, aby została wprowadzona nowa wartość wejściowa poziomu „0” o łącznym czasie trwania do 20,0 ms zanim filtr będzie w pełni reagował na nowe dane wejściowe. W tym czasie, krótkie impulsy „0” trwające krócej niż 20,0 ms mogą nie zostać wykryte lub zliczone.

Ta zmiana czasu reakcji filtra może spowodować nieoczekiwane działanie maszyny lub procesu, co może spowodować śmierć lub poważne zranienie personelu i/lub uszkodzenia mienia.

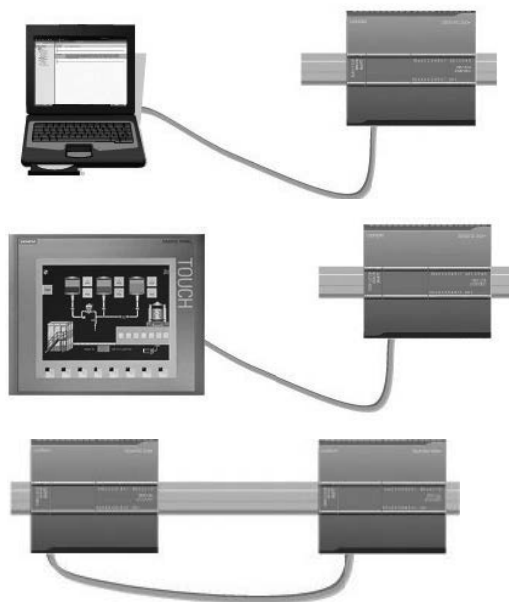
Aby zapewnić natychmiastowe zastosowanie nowego czasu reakcji filtra, należy wyłączyć i ponownie włączyć zasilanie CPU.

Po załączeniu HSC należy skonfigurować jego inne parametry, takie jak funkcja zliczania, wartości początkowe, opcje kasowania i zdarzenia przerwania.



Więcej dodatkowych informacji o konfiguracji HSC znajduje się w sekcji poświęconej konfiguracji CPU (strona 72).

Komunikacja pomiędzy urządzeniami – standard PROFINET 7



Do bezpośredniego połączenia pomiędzy urządzeniem i CPU:

- Projekt musi zawierać CPU.
- Programowane urządzenie nie jest częścią projektu, ale jest obsługiwane przez STEP 7.

Bezpośrednie połączenie pomiędzy panelem HMI i CPU wymaga, aby projekt zawierał obydwa składniki.

Bezpośrednie połączenie dwóch CPU wymaga:

- Projekt musi zawierać obydwa CPU.
- Należy skonfigurować połączenie sieciowe pomiędzy obydwooma CPU.

S7-1200 jest sterownikiem obsługującym PROFINET IO i wykorzystuje STEP 7 do konfiguracji komunikacji pomiędzy urządzeniami programowanymi, interfejsami HMI i innymi CPU. S7-1200 może komunikować się także z urządzeniami, które są tworzone przez innych dostawców. Przełącznik ethernetowy nie jest wymagany przy połączeniu bezpośrednim urządzenia programującego lub połączenia HMI i CPU. Przełącznik ethernetowy jest wymagany przy połączeniu więcej niż dwóch CPU lub HMI.

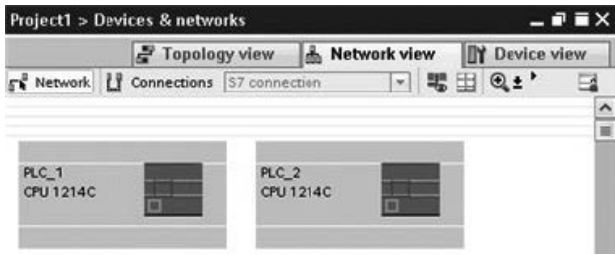
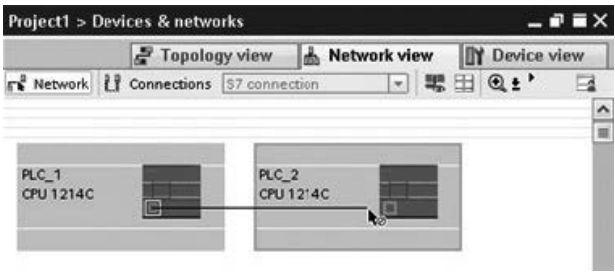
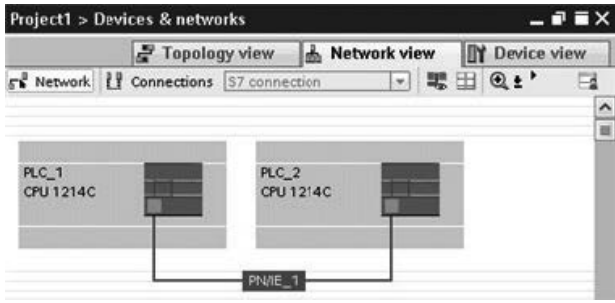
Dodając moduł PROFIBUS CM, CPU może pracować jako urządzenie typu master lub slave w sieci PROFIBUS.

Inne interfejsy komunikacyjne (CM, CP lub CB) zapewniają obsługę wielu protokołów, takich jak Point-to-Point (PTP), Modbus, USS, a także GPRS (modem).

7.1. Tworzenie połączenia sieciowego

W celu utworzenia połączenia sieciowego między urządzeniami projektu należy wybrać *Network view*. Po stworzeniu połączenia sieciowego należy użyć zakładki *Properties* w oknie inspektora w celu konfiguracji parametrów sieci.

Tabela 7.1. Tworzenie połączenia sieciowego

Akcja	Wynik akcji
<p>Należy wybrać <i>Network view</i>, aby wybrać urządzenia do połączenia.</p>	
<p>Wybór portu na jednym urządzeniu i przeciągnięcie połączenia do portu na drugim urządzeniu.</p>	
<p>Zwolnienie klawisza myszy tworzy połączenie sieciowe.</p>	

7.2. Opcje komunikacji

S7-1200 obsługuje kilka typów komunikacji pomiędzy CPU, a programowalnymi urządzeniami, interfejsami HMI i innymi CPU.



OSTRZEŻENIE

Jeśli atakujący może uzyskać fizyczny dostęp do sieci, to jest zagrożenie, że może odczytywać i zapisywać dane.

TIA Portal, CPU i urządzenia HMI (z wyjątkiem urządzeń HMI używających instrukcji GET/PUT) zapewniają bezpieczną komunikację, która zabezpiecza przed atakami typu *replay attack* oraz atakami typu *man-in-the-middle* (podśluch i modyfikacja bez wiedzy stron komunikacji). Po aktywacji komunikacji, odbywa się wymiana podpisanych wiadomości w postaci zwykłego tekstu, co umożliwia atakującemu odczytywać dane, ale zabezpiecza przed nieautoryzowanym zapisem danych. To TIA Portal, a nie proces komunikacji, szyfruje dane chronionych bloków zawierających informacje technologiczne.

Wszystkie inne formy komunikacji (wymiana danych I/O przez PROFIBUS, PROFINET, AS-i, lub inne magistrale I/O, instrukcje GET/PUT, bloki nadawcze (T-Block), oraz moduły komunikacyjne (CM)) nie mają żadnych zabezpieczeń. Te formy komunikacji należy chronić przez ograniczenie dostępu fizycznego. Jeśli atakujący może uzyskać fizyczny dostęp do sieci wykorzystując te formy komunikacji, to jest zagrożenie, że może odczytywać i zapisywać dane.

Informacje i zalecenia dotyczące bezpieczeństwa, można znaleźć w publikacji „Operational Guidelines for Industrial Security” (Wytyczne operacyjne dotyczące bezpieczeństwa przemysłowego) na stronie pomocy technicznej firmy Siemens.

PROFINET

PROFINET jest używany do wymiany danych za pośrednictwem programu użytkownika z innymi partnerami komunikacji poprzez Ethernet:

- PROFINET obsługuje 16 urządzeń PROFINET IO (wejść/wyjść) i maksymalnie 256 podmodułów. PROFIBUS umożliwia użycie 3 niezależnych urządzeń PROFIBUS DP Master, obsługując 32 urządzenia Slave przypadających na jedno urządzenie DP Master, i maksymalnie 512 modułów przypadających na jedno urządzenie DP Master.
- Komunikację S7.
- Protokół UDP.
- Protokół ISO przez TCP (RFC 1006).
- Protokół TCP.

Sterownik PROFINET RT IO

Jako sterownik używający standardu PROFINET RT, CPU komunikuje się z maksymalnie 16 urządzeniami PN poprzez lokalną sieć PN lub przez PN/PN coupler

(link). Więcej informacji o PROFINET i PROFIBUS International (PI), znajduje się pod adresem (www.us.profinet.com).

PROFIBUS

PROFIBUS jest używany do wymiany danych za pośrednictwem programu użytkownika z innymi partnerami komunikacji poprzez sieć PROFIBUS:

- Z modułami CM 1242-5, CPU pracuje jako PROFIBUS DP Slave.
- Z modułami CM 1243-5, CPU pracuje jako PROFIBUS DP Master class1.
- Moduły PROFIBUS DP Slave, PROFIBUS DP Master, oraz AS-i (3 moduły komunikacyjne z lewej strony) oraz PROFINET są oddzielnymi sieciami komunikacyjnymi, które nie ograniczają się wzajemnie.

AS-i

S7-1200 CM 1243-2 AS-i Master umożliwia podłączenie CPU S7-1200 do sieci AS-i.

Komunikacja CPU z CPU S7

Można utworzyć połączenie komunikacyjne do stacji partnerskiej i używać instrukcji GET i PUT do komunikacji z jednostkami CPU S7.

Komunikacja Teleservice

Połączenie TeleSerwis przez GPRS polega na tym, że stacja inżynierska z zainstalowanym programem STEP 7 komunikuje się ze stacją SIMATIC S7-1200 z modułem CP 1242-7 za pośrednictwem sieci GSM i Internetu. Połączenie przebiega przez serwer sterowania zdalnego, który służy jako urządzenie pośredniczące i jest podłączony do Internetu.

IO-Link

S7-1200 SM 1278 4xIO-Link Master umożliwia urządzeniom IO-Link połączenie z CPU S7 1200.

Patrz również

Internetowa strona wsparcia technicznego firmy Siemens (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)

7.3. Liczba asynchronicznych połączeń komunikacyjnych

CPU obsługuje następujące maksymalne liczby jednoczesnych, asynchronicznych, komunikacyjnych połączeń w sieci PROFINET oraz PROFIBUS:

- 8 połączeń dla Otwartej Komunikacji Użytkownika (Open User Communications), aktywnych lub pasywnych: TSEND_C, TRCV_C, TCON, TDISCON, TSEND, i TRCV.

- CPU pozwala na obsługę następującej liczby równoczesnych, asynchronicznych połączeń komunikacyjnych
- 8 połączeń CPU-do-CPU S7 dla danych GET/PUT klienta

Uwaga

CPU S7-1200, S7-300, oraz S7-400 używają instrukcji GET i PUT do komunikacji CPU-do-CPU S7. CPU S7-200 wykorzystuje instrukcje ETHx_XFER do komunikacji CPU-do-CPU S7.

- Połączenia HMI: W środowisku programistycznym TIA Portal można skonfigurować liczbę połączeń HMI wymaganych w aplikacji.
- Połączenia PG: CPU umożliwia połączenie z 1 urządzeniem programującym (PG).
- Połączenia http: CPU umożliwia połączenia dla serwera WWW.

Uwaga

Połączenia dla systemu otwartej komunikacji użytkowników (*Open User Communications*), połączenia S7, urządzenia HMI, urządzenia programujące oraz połączenia komunikacyjne serwera Web (http) mogą używać wiele zasobów połączeniowych w zależności od bieżąco wykorzystywanych funkcji.

7.4. Instrukcje do obsługi PROFINET i PROFIBUS

Instrukcje do PROFINET

Instrukcje TSEND_C i TRCV_C ułatwiają komunikacje po PROFINET łącząc w sobie funkcje TCON, TDISCON z funkcjami TSEND lub TRCV.

- TSEND_C ustala ze stacją partnerską połączenie komunikacyjne TCP lub ISO on TCP, wysyła dane i może zakończyć połączenie. Połączenie po skonfigurowaniu i ustaleniu jest automatycznie utrzymywane i monitorowane przez CPU. TSEND_C łączy w jedną instrukcję funkcje TCON, TDISCON i TSEND.
- TRCV_C ustala z partnerskim CPU połączenie komunikacyjne TCP lub ISO on TCP, odbiera dane i może zakończyć połączenie. Połączenie po skonfigurowaniu i ustaleniu jest automatycznie utrzymywane i monitorowane przez CPU. Instrukcja TSEND_C łączy w jedną instrukcję funkcje TCON, TDISCON i TRCV.

Instrukcje TCON, TDISCON, TSEND oraz TRCV są nadal dostępne.

Do transmisji lub odbioru danych po protokole UDP należy użyć instrukcji TSEND i TURCV. Działają one asynchronicznie (tak jak: TSEND, TRCV, TCON, TDISCON), co oznacza że wykonanie polecenia odbywa się przez kilka wywołań instrukcji.

W celu zmiany konfiguracji parametrów IP z poziomu programu użytkownika należy użyć instrukcji IP_CONF. Ta instrukcja działa także asynchronicznie. Jej wykonanie trwa przez kilka wywołań.

Instrukcje PROFIBUS

Instrukcja DPNRM_DG (czytanie diagnostyki) powoduje odczyt bieżących danych diagnostycznych urządzenia DP slave w formacie opisanym w EN 50 170 Volume 2, PROFIBUS.

Instrukcje rozproszonych I/O dla protokołów PROFINET, PROFIBUS i GPRS

Program użytkownika może używać następujących Instrukcji dla protokołów PROFINET, PROFIBUS i GPRS:

- Instrukcje RDREC (*read record*) oraz WRREC (*write record*) przenoszą określone rekordy danych pomiędzy komponentami, takimi jak moduł na szynie głównej lub rozproszony komponent (PROFIBUS DP lub PROFINET IO).
- Instrukcja RALRM (odczyt alarmu) służy do odczytu przerwania i informacji o nim ze współpracującego urządzenia DP slave lub PROFINET IO. Informacje te zawierają w wyjściowych parametrach informacje rozruchowe wywołanego OB oraz informacje o źródle przerwania.
- Instrukcje DPRD_DAT (*read consistent data*) oraz DPWR_DAT (*write consistent data*) służą do transferu bloków danych większych niż 64 bajty do oraz ze standardowego urządzenia DP Slave.
- Tylko w przypadku PROFIBUS, instrukcja DPNRM_DG służy do odczytu bieżących danych diagnostycznych z urządzenia DP Slave w formacie określonym normą EN 50 170 Vol. 2.

7.5. PROFINET

7.5.1. System otwartej komunikacji użytkowników

CPU S7-1200 ma zintegrowany port PROFINET obsługujący różne standardy komunikacyjne oparte na sieci Ethernet:

- Transport Connection Protocol (TCP).
- ISO on TCP (RFC 1006).
- User Datagram Protocol (UDP).

Tabela 7.2. Protokoły i instrukcje komunikacyjne

Wariant protokołu	Przykład użycia	Wprowadzanie danych do obszaru odbiorczego	Instrukcje komunikacyjne	Typ adresowania
TCP	Komunikacja CPU-CPU Przenoszenie ramek	Tryb „Ad hoc”	Tylko TRCV_C, TRCV	Przypisuje numery portów do urządzeń lokalnych (aktywny) i partnerskich (pasywne).
		Odbiór danych o określonej długości	TSEND_C, TRCV_C, TCON, TDISCON, TSEND, i TRCV	
ISO on TCP	Komunikacja CPU-CPU Podział wiadomości i powtórne jej złożenie	Tryb „Ad hoc”	Tylko TRCV_C, TRCV	Przypisuje punkty dostępu do usługi transportu (TSAP) do urządzeń lokalnych (aktywny) i partnerskich (pasywne).
		Kontrolowane przez protokół	TSEND_C, TRCV_C, TCON, TDISCON, TSEND oraz TRCV	
UDP	Komunikacja CPU-CPU Komunikacje programów użytkownika	Protokół użytkownika	TUSEND i TURCV	Przypisuje numery portów do urządzeń lokalnych (aktywny) i partnerskich (pasywne), ale połączenie nie jest dedykowane
Komunikacja S7	Komunikacja CPU-CPU Odczyt zapis danych z/do CPU	Odbiór i transmisja danych określonej długości	GET oraz PUT	Przypisuje punkty dostępu do usługi transportu (TSAP) do urządzeń lokalnych (aktywny) i partnerskich (pasywne).
PROFINET RT	Komunikacja CPU-urządzenie PROFINET IO	Odbiór i transmisja danych określonej długości	Wbudowane	Wbudowane

7.5.1.1. Tryb ad hoc

Protokoły TCP oraz ISO on TCP otrzymują zwykle pakiety danych o określonej długości, w zakresie od 1 do 8192 bajtów. Instrukcje komunikacyjne TRCV_C i TRCV przewidują jednak także tryb komunikacji ad hoc, w którym długość otrzymanego pakietu danych zmienia się od 1 do 1472 bajtów.

Uwaga

Jeśli dane są przechowywane w „zoptymalizowanym“ DB (tylko symbolicznie), użytkownik może otrzymać te dane tylko jako tablica typów danych, takich jak Byte, Char, USInt oraz SInt.

Konfiguracja instrukcji TRCV_C lub TRCV dla trybu ad hoc odbywa się poprzez ustawienie wartości parametru LEN na 65 535.

Jeśli instrukcje TRCV_C lub TRCV nie są często wywoływane w trybie ad hoc, użytkownik może otrzymywać więcej niż jeden pakiet w jednym wywołaniu. Przykładowo jeśli użytkownik otrzymuje pięć 100 bajtowych pakietów w jednym wywołaniu, TCP dostarczy te pakiety w jednym 500 bajtowym pakiecie, podczas gdy protokół ISO on TCP przebuduje ten pakiet w pięć 100 bajtowych pakietów.

7.5.1.2. Identyfikatory ID połączeń dla systemu otwartej komunikacji użytkowników

Podczas tworzenia w programie użytkownika instrukcji TSEND_C, TRCV_C lub TCON, STEP 7 tworzy instancje DB, która konfiguruje kanał komunikacyjny (lub połączenie) pomiędzy urządzeniami. Należy użyć zakładki *Properties* aby skonfigurować parametry połączenia. Wśród parametrów znajduje się także identyfikator ID połączenia dla danego połączenia.

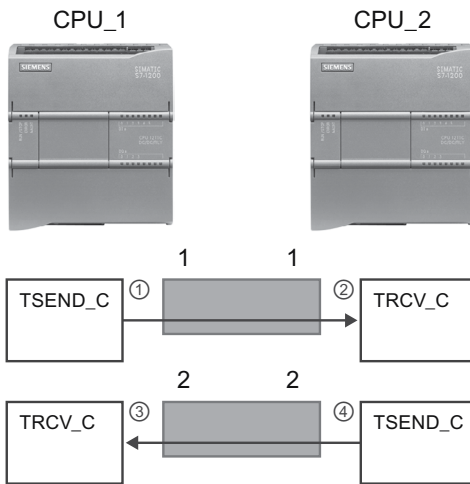
- Identyfikator ID połączenia musi być unikalny dla CPU. Każde połączenie, które jest tworzone, musi mieć inny DB i identyfikator ID połączenia.
- Zarówno lokalny (*Local*) CPU, jak i partnerski (*Partner*) CPU może korzystać z tego samego identyfikatora ID połączenia dla tego samego połączenia, ale identyfikatory ID połączeń nie wymagają dopasowania. Identyfikator ID połączenia jest istotny tylko dla instrukcji PROFINET w programie użytkownika danego CPU.
- Jako identyfikatora ID połączenia CPU można użyć dowolnej liczby. Jednak, skonfigurowanie identyfikatorów ID połączeń jako kolejnych liczb od „1” pozwala w łatwy sposób śledzić liczbę używanych połączeń dla określonego CPU.

Uwaga

Każda instrukcja TSEND_C, TRCV_C lub TCON w programie użytkownika tworzy nowe połączenie. Ważne jest, aby dla każdego połączenia używać poprawnego identyfikatora ID połączenia.

Poniższy przykład pokazuje komunikację pomiędzy dwoma CPU, które wykorzystuje 2 oddzielne połączenia do wysyłania i odbierania danych.

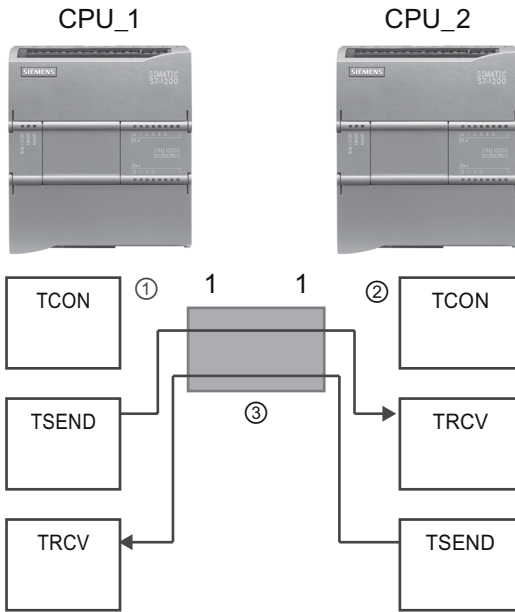
- Instrukcja TSEND_C w CPU_1 łączy się z TRCV_V w CPU_2 w pierwszym połączeniu („ID połączenia 1” w CPU_1 i CPU_2).
- Instrukcja TRCV_C w CPU_1 łączy się z TSEND_C w CPU_2 w drugim połączeniu („ID połączenia 2” w CPU_1 i CPU_2).



- ① TSEND_C w CPU_1 tworzy połączenie i przypisuje identyfikator ID połączenia dla tego połączenia (ID połączenia 1 dla CPU_1).
- ② TRCV_C w CPU_2 tworzy połączenie do CPU_2 i przypisuje identyfikator ID połączenia (ID połączenia 1 dla CPU_2).
- ③ TRCV_C w CPU_1 tworzy drugie połączenie dla CPU_1 i przypisuje inny identyfikator ID połączenia dla tego połączenia (ID połączenia 2 dla CPU_1).
- ④ TSEND_C w CPU_2 tworzy drugie połączenie i przypisuje inny identyfikator ID połączenia dla tego połączenia (ID połączenia 2 dla CPU_2).

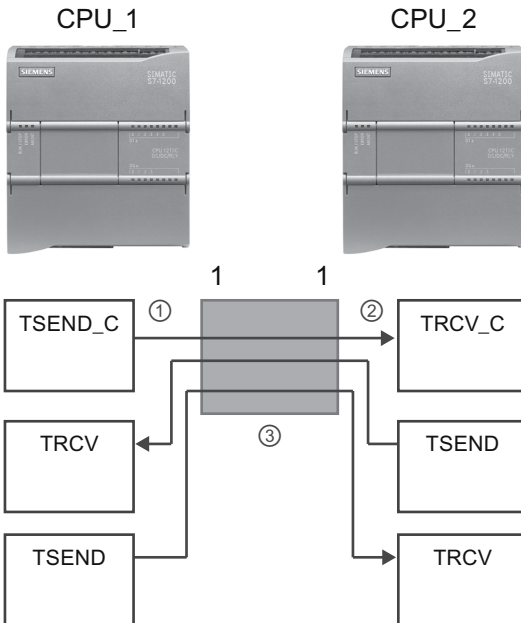
Poniższy przykład pokazuje komunikację pomiędzy dwoma CPU, które wykorzystują 1 połączenie do wysyłania i odbierania danych.

- Każde CPU używa instrukcji TCON do konfiguracji połączenia pomiędzy sobą.
- Instrukcja TSEND w CPU_1 łączy się z instrukcją TRCV w CPU_2 przy użyciu identyfikatora ID połączenia („ID połączenia 1”), które zostało skonfigurowane przez instrukcję TCON w CPU_1. Instrukcja TRCV w CPU_2 łączy się z instrukcją TSEND w CPU_1, przy użyciu identyfikatora ID połączenia („ID połączenia 1”), które zostało skonfigurowane przez instrukcję TCON w CPU_2.
- Instrukcja TSEND w CPU_2 łączy się z instrukcją TRCV w CPU_1 przy użyciu identyfikatora ID połączenia („ID połączenia 1”), które zostało skonfigurowane przez instrukcję TCON w CPU_2. Instrukcja TRCV w CPU_1 łączy się z instrukcją TSEND w CPU_2, przy użyciu identyfikatora ID połączenia („ID połączenia 1”), które zostało skonfigurowane przez instrukcję TCON w CPU_1.



- ① TCON w CPU_1 tworzy połączenie i przypisuje identyfikator ID połączenia dla tego połączenia w CPU_1 (ID = 1).
- ② TCON w CPU_2 tworzy połączenie i przypisuje identyfikator ID połączenia dla tego połączenia na CPU_2 (ID = 1).
- ③ TSEND i TRCV w CPU_1 używają identyfikatora ID połączenia utworzonego przez TCON w CPU_1 (ID = 1). TSEND i TRCV na CPU_2 używają identyfikatora ID połączenia utworzonego przez TCON w CPU_2 (ID = 1).

Jak to zostało pokazane na poniższym przykładzie, istnieje także możliwość użycia pojedynczych instrukcji TSEND i TRCV do komunikacji przez połączenie utworzone instrukcjami TSEND_C lub TRCV_C. Instrukcje TSEND i TRCV nie tworzą same nowego połączenia, więc muszą użyć DB oraz identyfikator ID połączenia, który został utworzony przez instrukcję TSEND_C, TRCV_C lub TCON.



- ① TSEND_C w CPU_1 tworzy połączenie i przypisuje identyfikator ID połączenia dla tego połączenia (ID = 1).
- ② TRCV_C w CPU_2 tworzy połączenie i przypisuje identyfikator ID połączenia dla tego połączenia w CPU_2 (ID = 1).
- ③ TSEND i TRCV w CPU_1 używają identyfikatora ID połączenia utworzonego przez TSEND_C w CPU_1 (ID = 1). TSEND i TRCV w CPU_2 używają identyfikatora ID połączenia utworzonego przez TRCV_C w CPU_2 (ID = 1).

7.5.1.3. Parametry połączenia PROFINET

Instrukcje TSEND_C, TRCV_C oraz TCON wymagają, aby związane z połączeniem parametry były określone. Inaczej urządzenie partnerskie nie zostanie połączone. Parametry te są określone dla protokołów TCP, ISO-on-TCP i UDP w strukturze TCON_Param. Zwykle dla określenia tych parametrów należy użyć zakładki *Configuration* we właściwościach instrukcji. Jeśli zakładka *Configuration* nie jest dostępna, należy określić strukturę TCON_Param programowo.

Struktura opisująca połączenie TCON_Param

Tabela 7.3. Struktura opisująca połączenie (TCON_Param)

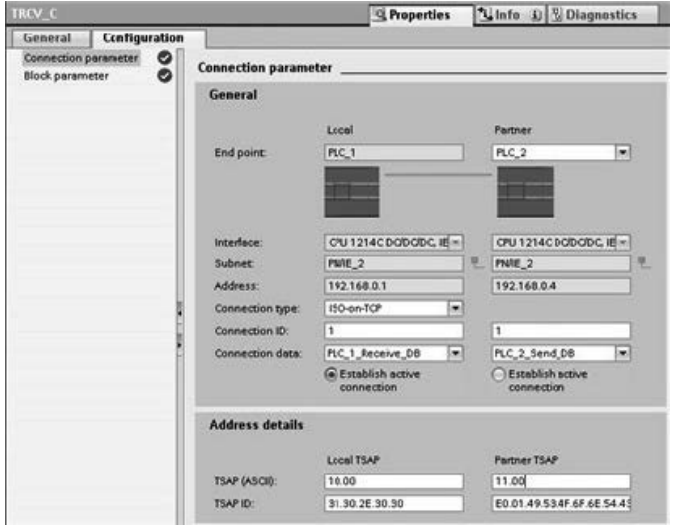
Bajt	Typ danych i parametru		Opis
0...1	block_length	UInt	Długość: 64 bajtów (stała)
2...3	Id	CONN_OUC (Word)	Odniesienie do tego połączenia: Zakres wartości: 1 (domyślnie) do 4095. Określa wartość tego parametru dla instrukcji TSEND_C, TRCV_C lub TCON pod określonym ID.
4	connection_type	USInt	Typ połączenia: <ul style="list-style-type: none"> • 17: TCP (domyślnie). • 18: ISO-on-TCP. • 19: UDP.
5	active_est	Bool	ID dla danego typu połączenia: <ul style="list-style-type: none"> • TCP i ISO-on-TCP: <ul style="list-style-type: none"> – FALSE: Połączenie pasywne, – TRUE: Aktywne połączenie (domyślnie), • UDP: FALSE.
6	local_device_id	USInt	ID lokalnego interfejsu PROFINET lub interfejsu Industrial Ethernet: 1 (domyślnie)
7	local_tsap_id_len	USInt	Długość parametru local_tsap_id, w bajtach; możliwe wartości: <ul style="list-style-type: none"> • TCP: 0 (aktywne, domyślne) lub 2 (pasywne). • ISO-on-TCP: 2 do 16. • UDP: 2.
8	rem_subnet_id_len	USInt	Ten parametr nie jest używany
9	rem_staddr_len	USInt	Długość adresu partnerskiego punktu końcowego, w bajtach: <ul style="list-style-type: none"> • 0: nieokreślony (parametr rem_staddr jest nieistotny). • 4 (domyślny): Obowiązujący adres IP znajduje się w parametrze rem_staddr (tylko dla TCP oraz ISO-on-TCP).
10	rem_tsap_id_len	USInt	Długość używanego parametru rem_tsap_id; w bajtach; możliwe wartości: <ul style="list-style-type: none"> • TCP: 0 (pasywny) lub 2 (aktywny, domyślny). • ISO-on-TCP: 2 do 16. • UDP: 0.
11	next_staddr_len	USInt	Ten parametr nie jest używany

Bajt	Typ danych i parametru		Opis
12...27	local_tsap_id	Tablica [1..16] z Byte	<p>Lokalny składnik adresu połączenia:</p> <ul style="list-style-type: none"> • TCP i ISO-on-TCP: numer lokalnego portu (możliwe wartości: 1 do 49151; zalecane wartości: 2000...5000): <ul style="list-style-type: none"> – local_tsap_id[1] = starszy bajt numeru portu w notacji hexadecymalnej; – local_tsap_id[2] = młodszy bajt numeru portu w notacji hexadecymalnej; – local_tsap_id[3-16] = nieistotny • ISO-on-TCP: lokalne TSAP-ID: <ul style="list-style-type: none"> – local_tsap_id[1] = B#16#E0; – local_tsap_id[2] = listwy i gniazda lokalnych punktów końcowych (bity 0 do 4: numer gniazda, bity 5 to 7: numer listwy); – local_tsap_id[3-16] = rozszerzenie TSAP, opcjonalnie • UDP: Ten parametr nie jest używany. <p>Uwaga: Należy zapewnić, aby żadna wartość local_tsap_id dla jednego CPU nie powtarzała się.</p>
28...33	rem_subnet_id	Tablica [1..6] z USInt	Ten parametr nie jest używany.
34...39	rem_staddr	Tablica [1..6] z USInt	<p>Tylko TCP oraz ISO-on-TCP: Adres IP dla partnerskiego punktu końcowego (nieistotny dla pasywnych połączeń). Przykładowo adres IP 192.168.002.003 jest przechowywany w następujących składnikach tablicy:</p> <p>rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = nieistotne</p>
40...55	rem_tsap_id	Tablica [1..16] z BYTE	<p>Partnerski składnik adresu połączenia:</p> <ul style="list-style-type: none"> • TCP: numer partnerskiego portu. Zakres: 1 do 49151; Zalecane wartości: 2000 do 5000): <ul style="list-style-type: none"> – rem_tsap_id[1] = starszy bajt numeru portu w notacji hexadecymalnej; – rem_tsap_id[2] = młodszy bajt numeru portu w notacji hexadecymalnej; – rem_tsap_id[3-16] = nieistotne • ISO-on-TCP: partnerskie TSAP-ID: <ul style="list-style-type: none"> – rem_tsap_id[1] = B#16#E0 – rem_tsap_id[2] = listwy i gniazda partnerskich punktów końcowych (bity 0 do 4: numer gniazda, bity 5 to 7: numer listwy); – rem_tsap_id[3-16] = rozszerzenie TSAP, opcjonalnie • UDP: Ten parametr nie jest używany.
56...61	next_staddr	Tablica [1..6] z BYTE	Ten parametr nie jest używany.
62...63	Spare	Word	Zastrzeżone: W#16#0000

7.5.2. Konfigurowanie ścieżki połączenia

Po umieszczeniu w programie użytkownika instrukcji TSEND_C, TRCV_C lub TCON, okno inspektora wyświetla właściwości połączenia za każdym razem, gdy zostanie zaznaczona jakakolwiek część instrukcji. Należy określić parametry komunikacyjne w zakładce *Configuration*, która wyświetli się po wskazaniu *Properties* (właściwości) instrukcji komunikacyjnej.

Tabela 7.4. Konfiguracja ścieżki komunikacyjnej (używając właściwości instrukcji)

TCP, ISO-on-TCP, i UDP	Właściwości protokołu
<p>Dla protokołów ethernetowych TCP, ISO-on-TCP, oraz UDP należy użyć pola instrukcji <i>Properties</i> (TSEND_C, TRCV_C, lub TCON), aby skonfigurować połączenia typu „Local/Partner”.</p> <p>Rysunek przedstawia pole <i>Connection properties</i> dla zakładki konfiguracyjnej dla połączenia ISO-on-TCP.</p>	

Uwaga

Podczas konfiguracji właściwości połączenia dla jednego CPU, STEP 7 umożliwia użytkownikowi na wybór określonego DB dla połączenia z partnerskim CPU (jeśli istnieje), lub na stworzenie takiego DB. Partnerskie CPU musi już wcześniej zostać umieszczone w projekcie i nie może być *unspecified CPU*.

Użytkownik nadal musi wstawić instrukcje TSEND_C, TRCV_C lub TCON do programu partnerskiego CPU. Po wykonaniu tego kroku użytkownik powinien wybrać DB, który został stworzony przez tą konfigurację.

Tabela 7.5. Konfiguracja ścieżki połączenia dla komunikacji S7 (konfiguracja urządzenia)

Komunikaty S& (GET i PUT)	Właściwości połączenia
<p>Dla komunikacji S7 należy użyć edytora sieci „Devices & networks” do skonfigurowania połączenia typu Local/Partner. Po naciśnięciu przycisku <i>Highlighted: Connection</i> uzyskuje się dostęp do pola <i>Properties</i>.</p> <p>Zakładka <i>General</i> ma kilka właściwości:</p> <ul style="list-style-type: none"> • <i>General</i> (pokazane). • <i>Local ID</i> – lokalne ID. • <i>Special connection properties</i> – specjalne właściwości połączenia. • <i>Address details</i> (pokazane) – szczegóły adresu. 	<p>The figure displays two screenshots of the SIMATIC Manager configuration interface for an S7 connection. The top screenshot shows the 'Connection path' tab, where the 'Local' and 'Partner' PLCs are identified as 'PLC_1' and 'PLC_2' respectively. The interface shows the 'End point', 'Interface' (set to 'PROFINET interface'), 'Interface type' (set to 'Ethernet/IP'), 'Subnet' (set to 'PINE_1'), and 'Address' (set to '192.168.0.1' for Local and '192.168.0.5' for Partner). The bottom screenshot shows the 'Address details' tab, where the 'End point' is 'PLC_1' and 'PLC_2'. It also shows 'Rack/slot' fields, 'Connection res. (hex)' dropdowns, 'TSAP' fields (set to 'SIMATICACC10101'), and 'Subnet ID' fields (set to 'C7AF-0000-0001' for Local and 'C7AF-0004-0001' for Partner).</p>

Więcej informacji oraz lista instrukcji komunikacyjnych znajduje się w dziale „Protokoły” (strona 128) w sekcji „PROFINET” lub w „Tworzenie połączenia S7” (strona 147) w sekcji „Komunikacja S7”.

Tabela 7.6. Parametry dla wielokrotnego połączenia CPU

Parametr		Definicja
Adres		Przypisany adres IP
Ogólny	End point	Nazwa przypisana do partnera (odbierającego) CPU.
	Interface	Nazwa przypisana do interfejsu.
	Subnet	Nazwa przypisana do podsieci.
	Interface type	<i>Tylko dla komunikacji S7. Typ interfejsu.</i>
	Connection type	Typ protokołu Ethernet.
	Connection ID	Numer ID.
	Connection data	Miejsce przechowywania danych dla lokalnego i partnerskiego CPU.
	Establish active connection	Przycisk opcji ustawienia lokalnego lub partnerskiego CPU jako aktywne połączenie.
Adres szczegółowy	End point	<i>Tylko dla komunikacji S7.</i> Nazwa przypisana do partnera (odbierającego) CPU.
	Rack/slot	<i>Tylko dla komunikacji S7.</i> Położenie gniazda i szyny.
	Connection resource	<i>Tylko dla komunikacji S7.</i> Komponent TSAP używany do konfiguracji komunikacji S7 z CPU urządzeń S7-300 lub S7-400.
	Port (decymalny):	TCP i UDP: Port partnerskiego CPU w formacie dziesiętnym.
	TSAP ¹ i Subnet ID:	ISO on TCP (RFC 1006) oraz komunikacja S7: TSAP dla lokalnego i partnerskiego CPU ustawione w formatach ASCII i hexadecymalnym.

¹ Podczas konfiguracji połączenia dla S7-1200 CPU z protokołem ISO-on-TCP należy używać tylko znaków w formacie ASCII dla rozszerzeń TSAP pasywnie skomunikowanych partnerów.

Punkty dostępu do usługi transportu (TSAP)

Wykorzystując punkty dostępowe serwisu transportowego TSAP (*Transport Service Access Point*), protokół TCP i komunikacja S7 pozwala zrealizować wiele połączeń z jednym adresem IP (do 64 K połączeń). Punkty TSAP jednoznacznie identyfikują te połączenia komunikacyjnych punktów końcowych z adresem IP.

W części *Address Details* okna dialogowego *Connection Parameters*, użytkownik definiuje punkty TSAP, które będą wykorzystywane. TSAP połączenia z CPU jest wprowadzany w polu *Local TSAP*. TSAP przypisany połączeniu w partnerskim CPU jest wprowadzany w polu *Partner TSAP*.

Numery portów

Dla protokołów TCP oraz UDP, parametry konfiguracji lokalnego (aktywnego) połączenia CPU muszą określać zdalny adres IP i numer portu partnerskiego (pasywnego) połączenia CPU.

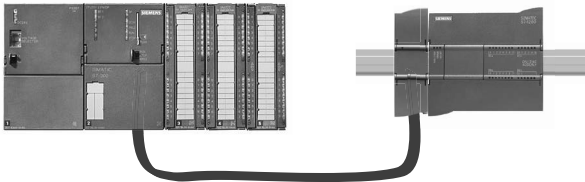
W sekcji *Address Details* okna dialogowego *Connection Parameters*, użytkownik definiuje porty, które będą wykorzystywane. Port połączenia w CPU jest wprowadzany w polu *Local Port*. Port przypisany do połączenia partnerskiego CPU jest wprowadzony w polu *Partner Port*.

7.6. PROFIBUS

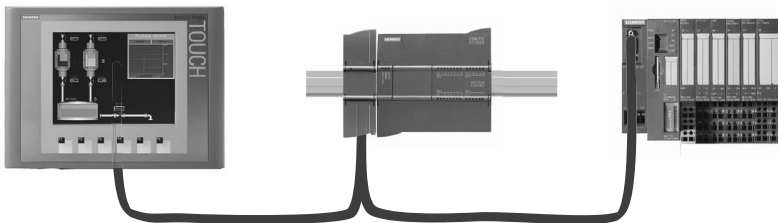
System PROFIBUS używa nadrzędnej magistrali do odpytania urządzeń typu slave rozproszonych w sposób wielopunktowy na magistrali szeregowej RS485. Urządzeniem PROFIBUS slave jest każde urządzenie peryferyjne (przetwornik I/O, zawór, napęd lub inne urządzenie pomiarowe), które przetwarza informację i wysyła stan swoich wyjść do urządzenia master. Urządzenia typu slave pełnią rolę pasywnych stacji w sieci, ponieważ nie mają praw dostępu do magistrali. Mogą one tylko potwierdzić otrzymane wiadomości, lub wysłać odpowiedź na żądanie urządzenia master. Wszystkie urządzenia PROFIBUS slave mają ten sam priorytet, a cała sieć komunikacyjna pochodzi od urządzenia master.

Urządzenie PROFIBUS master tworzy aktywną stację w sieci. PROFIBUS DP definiuje dwa typy urządzeń master. Urządzenia klasy 1 (zwykle programowalny sterownik PLC lub PC ze specjalnym oprogramowaniem) zajmuje się zwykłą komunikacją lub wymianą danych z przeznaczonymi do tego urządzeniami typu slave. Urządzenie master klasy 2 (zwykle urządzenie konfiguracyjne, takie jak laptop albo programująca konsola użyta do celów rozruchowych, konserwacyjnych lub diagnostycznych) jest specjalnym urządzeniem używanym głównie do uruchamiania urządzeń slave i w celach diagnostycznych.

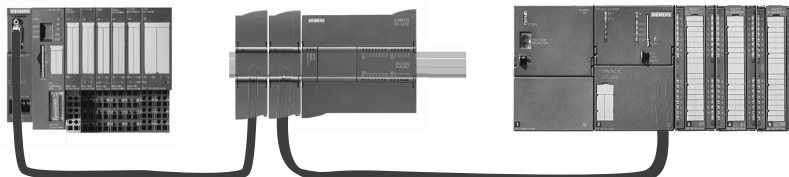
Użytkownik jeśli chce, może konfigurować moduł w systemie innej firmy, wykorzystując plik GSD dostępny dla modułu CM 1242-5 (DP Slave) na płycie CD, dostarczanej wraz z modułem oraz na stronach obsługi klienta firmy Siemens (Siemens Industrial Automation Customer Support) w Internecie: (<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=6GK72425DX300XE0&caller=view>). Na poniższym rysunku przedstawiono S7-1200 jako DP slave dla sterownika S7-300.



S7-1200 może być podłączony do sieci PROFIBUS jako DP master za pomocą modułu komunikacyjnego CM 1243-5. Moduł CM 1243-5 (DP master) może być partnerem komunikacyjnym z urządzeniami DP V0/V1 slave. Na poniższym rysunku przedstawiono sterownik S7-1200 jako master sterujący urządzeniem slave – ET200S.



Jeśli CM 1242-5 i CM 1243-5 są obecne równocześnie w systemie, to S7-1200 może pracować równocześnie jako urządzenie typu slave dla wyżej usytuowanych systemów DP master i jako urządzenie typu master dla niżej usytuowanych systemów DP slave.



Dla wersji V4.0, można skonfigurować maksymalnie trzy moduły CM PROFIBUS dla jednej stacji, w której może być dowolna kombinacja modułów CM DP master lub DP slave. Każde urządzenie DP master (nadrzędne) w implementacji z CPU z wersją firmware V3.0 lub wyższą może sterować maksymalnie 32 urządzeniami Slave (podrzędnymi).

Dane konfiguracyjne modułów komunikacyjnych PROFIBUS CM są przechowywane na lokalnym CPU. Pozwala to, w razie konieczności, na prostą zamianę modułów komunikacyjnych.

7.6.1. Usługi komunikacyjne modułów CM PROFIBUS

Moduły CM PROFIBUS używają protokołu PROFIBUS DP-V1.

Typy komunikacji z DP-V1

Następujące typy komunikacji z DP-V1 są dostępne:

- Komunikacja cykliczna (CM 1242-5 oraz CM 1243-5)

Obydwa moduły PROFIBUS wspierają cykliczną komunikację dla transferu danych procesu pomiędzy urządzeniami DP slave i DP master.

Cykliczna komunikacja jest przeprowadzana przez system operacyjny jednostki centralnej. Żadne bloki programowe nie są w tym celu wymagane. Dane I/O są odczytywane lub zapisywane bezpośrednio z/do obrazu procesu w CPU.
- Komunikacja niecykliczna (tylko CM 1243-5)

Moduł DP master może także obsługiwać acykliczną komunikację, która używa bloków programowych:

 - Instrukcja „RALRM“ służy do obsługi przerwań.
 - Instrukcje „RDREC“ oraz „WRREC“ służą do transferu konfiguracji i danych diagnostycznych.

Funkcje nie obsługiwane przez CM 1243-5: SYNC/FREEZE i Get_Master_Diag.

Inne usługi komunikacyjne modułu CM 1243-5

Moduł komunikacyjny CM 1243-5 master udostępnia także dodatkowe usługi komunikacyjne:

- Komunikacja S7

- Usługi PUT/GET

Moduł DP master funkcjonuje jako klient i serwer dla zapytań pochodzących od innych sterowników S7 i innych komputerów klasy PC. Komunikacja odbywa się w standardzie PROFIBUS.

- Komunikacja PG/OP

Funkcje PG pozwalają na załadowanie danych konfiguracyjnych i programu użytkownika z PG, a także na transfer danych diagnostycznych do PG.

Możliwi partnerzy komunikacyjni dla komunikacji OP to: panele HMI oraz komputerowe panele SIMATIC z systemami WinCC Flexible lub SCADA, które wspierają komunikację S7.

7.6.2. Podręczniki modułów komunikacyjnych CM PROFIBUS

Dodatkowe informacje

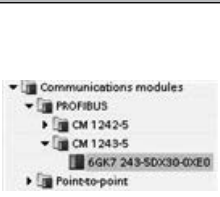
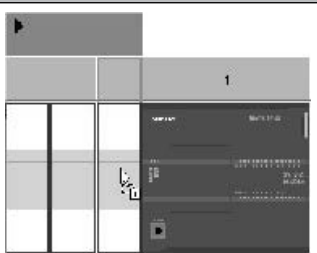
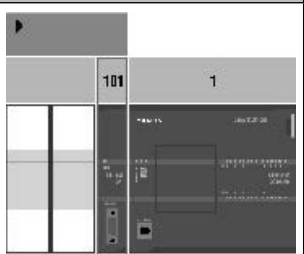
Szczegółowe informacje na temat modułów komunikacyjnych CM PROFIBUS można znaleźć w podręcznikach poszczególnych urządzeń. Można znaleźć je na internetowych stronach obsługi klienta firmy Siemens (Siemens Industrial Automation Customer Support) odpowiednio wprowadzając następujące identyfikatory:

- CM 1242-5 (<http://support.automation.siemens.com/WW/view/en/49852105>)
- CM 1243-5 (<http://support.automation.siemens.com/WW/view/en/49851842>)

7.6.3. Dodawanie modułów CM 1243-5 (DP master) oraz DP slave

W celu dodania urządzenia DP Slave, w portalu „Devices and networks” należy użyć katalogu sprzętu. Moduły te są dołączane po lewej stronie CPU. Aby umieścić moduł w konfiguracji sprzętowej należy wybrać ten moduł w katalogu sprzętowym poprzez podwójne kliknięcie na nim lub przeciągając i upuszczając go na podświetlone gniazdo.

Tabela 7.7. Dodawanie modułu komunikacyjnego CM 1243-5 (DP master) do konfiguracji urządzenia



Moduł	Wybranie modułu	Dodanie modułu	Rezultat operacji
CM 1243-5 (DP master)			

Katalog sprzętu może posłużyć również do dodania urządzeń typu DP slave. Przykładowo dla dodania urządzenia ET200 S DP slave należy w katalogu sprzętu (*Hardware Catalog*) rozwinąć następujące pola:

- Distributed I/O – rozproszone I/O
- ET200 S
- Interface modules – moduły interfejsu
- PROFIBUS

Następnie należy wybrać „6ES7 151-1BA02-0AB0” (IM151-1 HF) z listy numerów części i dodać ET200 S DP slave tak, jak jest to pokazane na poniższym rysunku.

Tabela 7.8. Dodawanie ET200 S DP slave do konfiguracji urządzenia

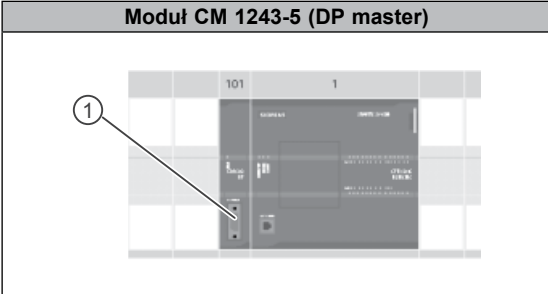
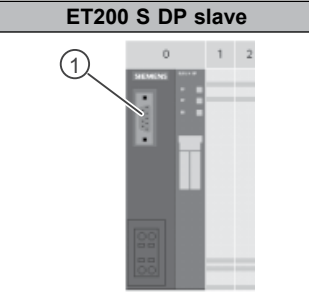
Wstawianie DP slave		Rezultat operacji	
			

7.6.4. Przypisanie adresów PROFIBUS do modułu CM 1243-5 i urządzeń DP slave

Konfiguracja interfejsu PROFIBUS

Po skonfigurowaniu połączeń logicznych sieci dla dwóch urządzeń PROFIBUS użytkownik może konfigurować parametry interfejsów PROFIBUS. W tym celu należy nacisnąć fioletowy blok modułu CM 1243-5, a zakładka *Properties* w oknie inspektora wyświetli dostępne interfejsy PROFIBUS. Interfejs PROFIBUS dla urządzenia DP slave można skonfigurować w ten sam sposób.

Tabela 7.9. Konfiguracja interfejsów PROFIBUS modułu CM 1243-5 (DP master) i urządzenia ET200 S DP slave

Moduł CM 1243-5 (DP master)	ET200 S DP slave
	

① Port PROFIBUS

Przypisanie adresów PROFIBUS

W sieci PROFIBUS każde urządzenie ma przypisany adres PROFIBUS. Zakres adresu mieści się pomiędzy 0 a 127 z następującymi wyjątkami:

- Adres 0: Zastrzeżony dla konfiguracji sieci i/lub narzędzi programowania dołączonych do magistrali

- Adres 1: Zastrzeżony przez SIEMENS AG dla pierwszego urządzenia typu master
- Adres 126: Zastrzeżone dla urządzeń fabrycznych, które nie mają przełącznika ustawień i muszą być zaadresowane przez sieć.
- Adres 127: Zastrzeżony dla urządzeń rozsyłających wiadomości do wszystkich urządzeń w sieci. Nie może zostać przypisany do urządzeń wykonujących inne działanie.

Pomijając zastrzeżone adresy pracujące urządzenia PROFIBUS mogą być adresowane numerami od 2 do 125.

W oknie właściwości należy wybrać pole konfiguracji „PROFIBUS address”. STEP 7 wyświetli okno konfiguracji adresów PROFIBUS, które służy do przypisania adresów PROFIBUS do danego urządzenia.

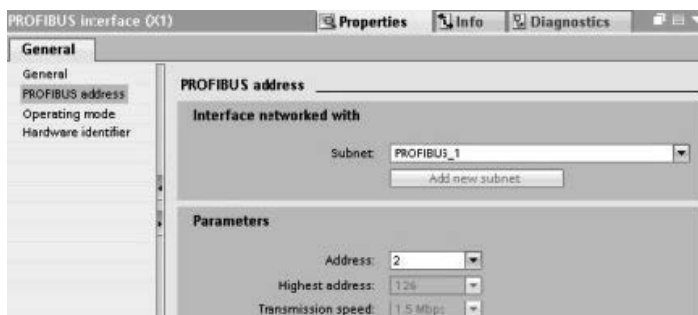


Tabela 7.10. Parametry adresów PROFIBUS

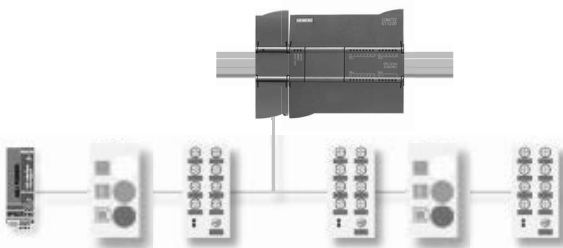
Parametr	Opis	
Podsieć	Należy nazwać podsieć, do której podłączone jest urządzenie. Należy nacisnąć przycisk <i>Add new subnet</i> , aby stworzyć nową podsieć. Jej stan domyślny to „Not connected” (niepołączona). Możliwe są dwa typy połączenia: <ul style="list-style-type: none"> • Domyślny: „Not connected” zapewnia lokalne połączenie. • Podsieć jest wymagana, gdy do sieci przyłączone są dwa lub więcej urządzeń. 	
Parametr	Adres	Przypisany do urządzenia adres PROFIBUS.
	Najwyższy adres	Najwyższy adres PROFIBUS jest określany na podstawie aktywnych stacji PROFIBUS (np. DP master). Pasywne urządzenia DP slaves mają niezależnie adresy PROFIBUS o numerach od 1 do 125, nawet jeśli najwyższy adres PROFIBUS jest ustawiony na np. 15. Najwyższy adres PROFIBUS jest istotny dla przekazywania tokenów (przekazanie wysłanych praw), a same tokeny przekazywane są wyłącznie do aktywnych stacji. Określenie najwyższego adresu PROFIBUS optymalizuje magistralę.
	Szybkość transmisji	Szybkość transmisji skonfigurowanej sieci PROFIBUS wynosi od 9,6 kb/s do 12 Mb/s. Ustawienie szybkości transmisji danych zależy od właściwości użytych węzłów PROFIBUS. Szybkość transmisji nie powinna być większa od szybkości obsługiwanej przez najwolniejszy węzeł. Szybkość transmisji jest zwykle ustawiana dla urządzenia master sieci PROFIBUS. W ten sposób wszystkie urządzenia DP slave automatycznie będą używały tej samej prędkości transmisji (auto-baud).

7.7. AS-i

Moduł S7-1200 CM 1243-2 AS-i Master umożliwia połączenie CPU S7-1200 w sieci AS-i.

AS-i (*Actuator Sensor Interface*) – interfejs aktywatora (urządzenia wykonawczego) i/lub czujnika, to system połączenia sieciowego z pojedynczym urządzeniem nadrzędnym (Master) na najniższym poziomie w systemach automatyki. Moduł CM 1243-2 służy jako urządzenie nadrzędne Master w sieci AS-i. Czujniki i aktywatory (urządzenia podrzędne AS-i Slave) można podłączyć do CPU za pomocą modułu CM 1243-2, używając pojedynczego kabla AS-i. Moduł CM 1243-2 obsługuje wszystkie dane koordynacji sieci AS-i oraz przekaźnikowe, a także przekazywane informacje statusów z aktywatorów i czujników do CPU przez adresy I/O przypisane do CM 1243-2. W zależności od typu urządzenia podrzędnego Slave, możliwy jest dostęp do wartości cyfrowych (binarnych) lub analogowych. Urządzenia podrzędne AS-i Slave stanowią kanały wejściowe i wyjściowe systemu AS-i, i są aktywne tylko wtedy, gdy są wywoływane przez moduł CM 1243-2.

Na rysunku poniżej, sterownik S7-1200 jest urządzeniem nadrzędnym Master w sieci AS-i sterującym panelem operatorskim AS-i oraz modułami cyfrowych i analogowych wejść i wyjść (I/O) będącymi urządzeniami podrzędnymi Slave.



7.7.1. Dodanie modułu CM 1243-2 AS-i Master oraz AS-i Slave

Aby dodać moduły CM1243-2 AS-i Master do CPU należy użyć katalogu sprzętu „Catalog”. Moduły te są dołączane z lewej strony CPU, i można użyć maksymalnie trzech modułów CM1243-2 Master AS-i. Aby wstawić moduł do konfiguracji sprzętowej, należy wybrać moduł w katalogu sprzętu i albo kliknąć dwukrotnie albo przeciągnąć moduł do podświetlonego slotu (po lewej stronie CPU).

Tabela 7.11. Dodawanie modułu CM1243-2 Master AS-i do konfiguracji urządzenia



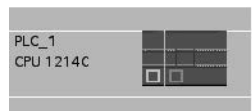
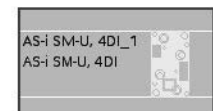
Moduł	Wybranie modułu	Wstawienie modułu	Wynik
CM 1243-2 AS-i Master	<ul style="list-style-type: none"> ▼ Communications modules ▶ PROFIBUS ▶ Point-to-point ▼ AS interface ▼ CM 1243-2 ▶ 3BK7 243-2AA30-0x00 		

W celu dodania urządzenia AS-i Slave należy także użyć katalogu sprzętu „Catalog”. Aby dodać, na przykład, urządzenie Slave z opisem *I/O module, compact, digital, input*, w katalogu sprzętu należy rozwinąć następujące foldery:

- Field devices Field devices (urządzenia instalowane na obiekcie)
- AS-Interface slaves (urządzenia AS-i Slave)

Następnie, z listy numerów katalogowych części należy wybrać „3RG9 001-0AA00” (AS-i SM-U, 4DI) i dodać urządzenie Slave z opisem *I/O module, compact, digital, input*, jak pokazano na rysunku poniżej.

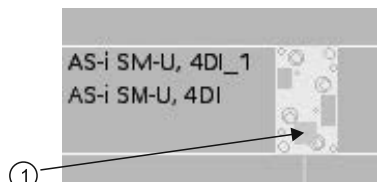
Tabela 7.12. Dodawanie AS-i Slave do konfiguracji urządzenia

Wstawienie AS-i Slave		Wyni	
			

7.7.2. Nadawanie adresu AS-i urządzeniu AS-i Slave

Konfiguracja interfejsu AS-i Slave

Aby skonfigurować parametry interfejsu AS-i, należy kliknąć żółty prostokąt AS-i na urządzeniu AS-i Slave. Zakładka *Properties* (Właściwości) w oknie Inspektor powoduje wyświetlenie interfejsu AS-i.



① Port AS-i

Nadawanie adresu AS-i Slave

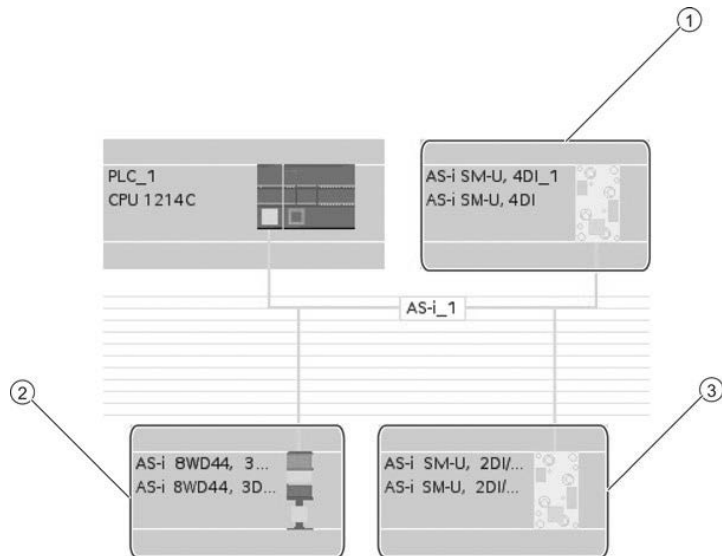
W sieci AS-i, każde urządzenie ma nadany adres AS-i Slave. Adres ten może być w zakresie od 0 do 31, jednak adres 0 jest zarezerwowany tylko dla nowych urządzeń Slave. Adresy urządzeń Slave mogą być w zakresie od 1 (A lub B) do 31 (A lub B), co umożliwi zastosowanie łącznie do 62 urządzeń podrzędnych Slave.

„Standardowe” urządzenia AS-i używają całych adresów i mają adresy numeryczne bez oznaczenia A lub B. Urządzenia AS-i „węzłów A lub B” używają składnik A lub B w każdym adresie, dzięki czemu każdy z 31 adresów, może być użyty dwukrotnie. Zakres adresów obejmuje przedziały od 1A do 31A oraz od 1B do 31B.

Każdy adres z zakresu od 1 do 31 może być nadany urządzeniu AS-i Slave, innymi słowy, nie ma znaczenia, czy urządzenia Slave mają adresy zaczynające się

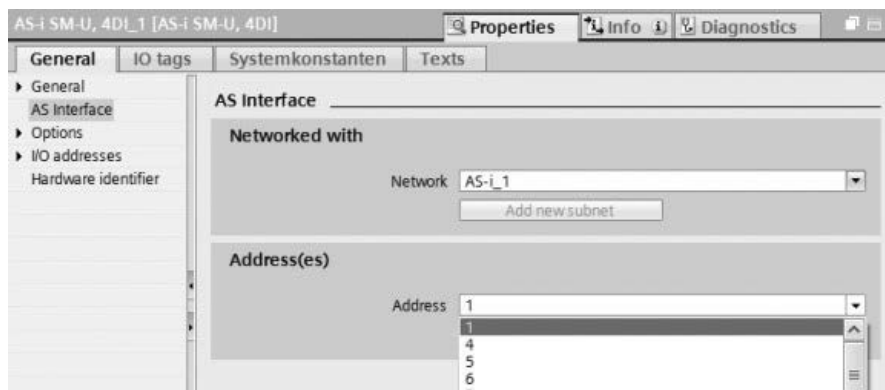
od 21 lub czy pierwszemu urządzeniu Slave nadano adres 1 (kolejność adresów nie ma znaczenia).

W poniższym przykładzie, trzem urządzeniom AS-i zostały przypisane następujące adresy: „1” (urządzenie standardowe), „2A” (urządzenie węzła A), oraz „3” (urządzenie standardowe):



- ① Adres 1 urządzenia AS-i Slave; urządzenie: AS-i SM-U, 4DI; nr zamówieniowy: 3RG9 001-0AA00
- ② Adres 2A urządzenia AS-i Slave; urządzenie: AS-i 8WD44, 3DO, A/B; nr zamówieniowy: 8WD4 428-0BD
- ③ Adres 3 urządzenia AS-i Slave; urządzenie: AS-i SM-U, 2DI/2DO; nr zamówieniowy: 3RG9 001-0AC00

Miejsce wprowadzenia adresu AS-i Slave pokazano poniżej.



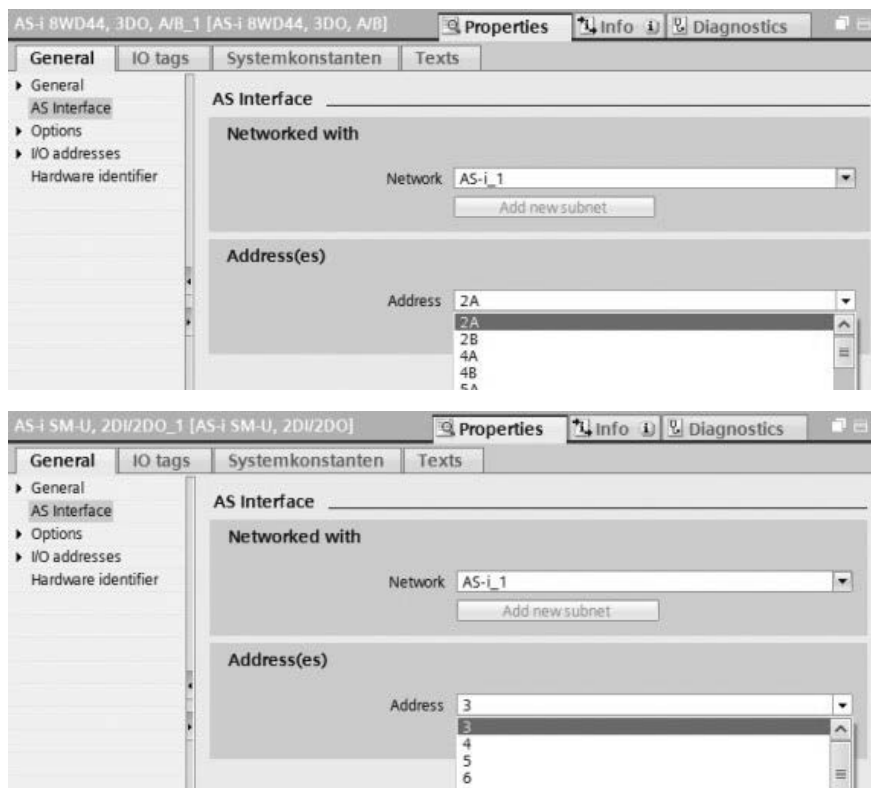


Tabela 7.13. Parametry interfejsu AS-i

Parametr	Opis
Network (Sieć)	Nazwa sieci, w której podłączone jest urządzenie
Address(es) (Adres(y))	Nadawany adres AS-i urządzeniu Slave z zakresu od 1 (A lub B) do 31 (A lub B), dla łącznie do 62 urządzeń podrzędnych

7.8. Komunikacja S7

7.8.1. Instrukcje GET i PUT

Użytkownik może używać instrukcji GET oraz PUT do komunikacji pomiędzy różnymi jednostkami S7 przez połączenie typu PROFINET lub PROFIBUS. Jest to możliwe tylko wtedy, gdy z właściwości *Protection* (Ochrona) lokalnego CPU została wybrana opcja dla partnerskiego CPU *Permit access with PUT/GET communication* (Pozwolenie na dostęp w komunikacji za pomocą instrukcji PUT/GET):

- Uzyskiwanie dostępu do danych w zdalnym CPU: CPU S7-1200 może używać adresów bezwzględnych w polu wejściowym ADDR_x do adresowania zmiennych zdalnych CPU (S7-200/300/400/1200).

- Uzyskiwanie dostępu do danych w standardowym DB: CPU S7-1200 może używać adresów bezwzględnych w polu wejściowym ADDR_x do adresowania zmiennych DB w standardowym DB zdalnego CPU S7.
- Uzyskiwanie dostępu do danych w zoptymalizowanym DB: CPU S7-1200 nie może uzyskać dostępu do zmiennych DB w zoptymalizowanym DB zdalnego CPU S7-1200.
- Uzyskiwanie dostępu do danych w lokalnym CPU: CPU S7-1200 może używać albo adresów bezwzględnych albo adresów symbolicznych jako wejść do pól wejściowych RD_x lub SD_x, odpowiednio, instrukcji GET lub PUT.

STEP 7 podczas wstawiania instrukcji automatycznie tworzy DB.

Uwaga

Dla zapewnienia spójności danych należy zawsze się upewnić, czy dana operacja została ukończona (NDR = 1 dla GET, lub DONE = 1 dla PUT). Dopiero w tym wypadku można uzyskać dostęp do danych lub rozpocząć następną operację odczytu/zapisu.

Uwaga

Opcja działania instrukcji GET/PUT w programie CPU w wersji V4.0 nie jest automatycznie aktywna.

Opcja działania instrukcji GET/PUT w programie CPU w wersji V3.0 jest dla CPU w wersji V4.0 automatycznie aktywna.

Aby to zmienić, należy przejść do konfiguracji CPU w *Device configuration* (Konfiguracja urządzeń), zakładki *Properties* (Właściwości) w oknie Inspektor, i następnie wybrać właściwość *Protection* (Ochrona), tym samym uaktywniając dostęp do instrukcji GET/PUT (strona 78).

7.8.2. Tworzenie połączenia S7

Mechanizmy połączeń

Aby uzyskać dostęp do zdalnych partnerów połączenia za pomocą instrukcji PUT/GET, użytkownik musi mieć uprawnienia.

Domyślnie opcja *Permit access with PUT/GET communication* (Pozwolenie na dostęp do komunikacji za pomocą instrukcji PUT/GET) jest wyłączona. W tym przypadku, dostęp do odczytu i zapisu danych CPU jest możliwy tylko dla połączeń komunikacyjnych, które wymagają konfiguracji lub programowania, zarówno lokalnego (Local) CPU, jak i partnerskiego (Partner) CPU. Dostęp, na przykład, za pomocą instrukcji BSEND/BRCV jest możliwy.

Połączenia, w których lokalny CPU jest tylko serwerem (co oznacza, że dla lokalnego CPU nie ma możliwości konfiguracji/programowania komunikacji z partnerem komunikacyjnym), nie są możliwe podczas pracy CPU. Dotyczy to na przykład:

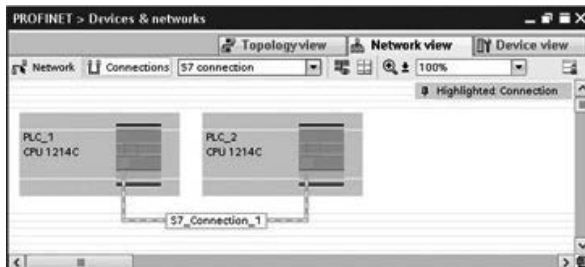
- dostępu za pomocą instrukcji PUT/GET, FETCH/WRITE lub FTP poprzez moduły komunikacyjne,
- dostępu za pomocą instrukcji PUT/GET z innych jednostek CPU S7,
- dostępu do HMI poprzez komunikację za pomocą instrukcji PUT/GET.

Jeśli występuje potrzeba, aby umożliwić dostęp do danych CPU od strony klienta i tym samym nie ograniczać usług komunikacyjnych CPU, należy postępować zgodnie z zaleceniami ochrony dostępu do CPU S7-1200.

Typy połączeń

Typ połączenia wybrany przez użytkownika tworzy połączenie komunikacyjne ze stacją partnerską. Połączenie jest rozpoczęte, ustawione i automatycznie monitorowane. W portalu *Devices and Networks* należy użyć pola *Network view*, aby stworzyć połączenie sieciowe pomiędzy urządzeniami w projekcie. Najpierw należy kliknąć zakładkę *Connections*, a następnie z listy rozwijalnej znajdującej się po prawej stronie wybrać typ połączenia (przykładowo połączenie S7). Użytkownik powinien nacisnąć zielony kwadracik (PROFINET) na pierwszym urządzeniu, a następnie przeciągnąć linię do zielonego kwadracika PROFINET w drugim urządzeniu. Po zwolnieniu przycisku myszy nastąpi połączenie obydwu urządzeń w sieć PROFINET.

Więcej informacji znajduje się w rozdziale „Tworzenie połączenia sieciowego”.



Naciśnij przycisk *Highlighted: Connection*, aby uzyskać dostęp do okna dialogowego *Properties* dla instrukcji komunikacyjnej.

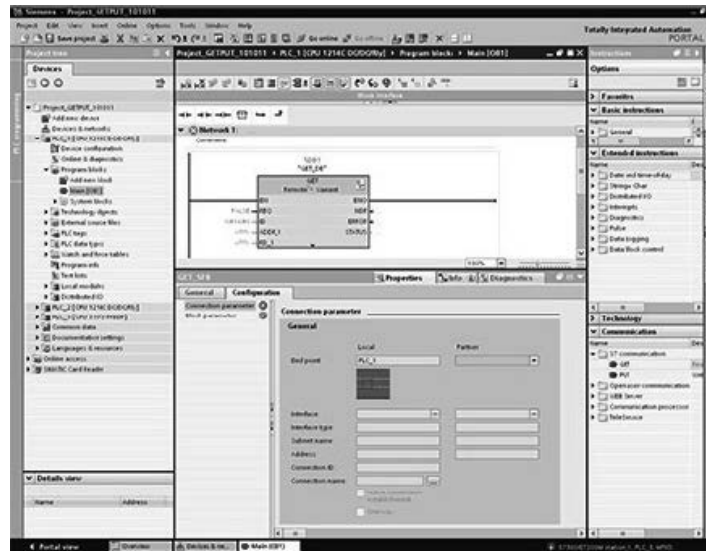
Patrz również

Rozdział „Tworzenie połączenia sieciowego” (strona 123)

7.8.3. Przypisanie parametrów połączenia dla instrukcji GET/PUT

Przypisanie parametrów połączenia dla instrukcji GET/PUT jest pomocne użytkownikowi w konfiguracji połączeń do komunikacji S7 z CPU do CPU.

Po wstawieniu bloku GET lub PUT, rozpoczyna się przypisanie parametrów połączenia dla instrukcji GET/PUT:



Kiedy zostanie wybrana dowolna część instrukcji, to w oknie Inspektor zostaną wyświetlane właściwości połączenia. Parametry komunikacji dla instrukcji komunikacji można określić w zakładce *Configuration* okna *Properties*.

Po wstawieniu do programu bloku GET lub PUT, zostaje automatycznie wyświetlona zakładka *Configuration* (Konfiguracja), oraz strona *Connection parameters* (Parametry połączenia). Strona ta pozwala użytkownikowi skonfigurować wymagane połączenie S7 oraz parametr *Connection ID* (Identyfikator połączenia), do którego odnosi się parametr bloku „ID”. Strona *Block parameters* (Parametry bloku) umożliwia użytkownikowi skonfigurowanie dodatkowych parametrów bloku.

Uwaga

Opcja działania instrukcji GET/PUT w programie CPU w wersji V4.0 nie jest automatycznie aktywna.

Opcja działania instrukcji GET/PUT w programie CPU w wersji V3.0 będzie dla CPU w wersji V4.0 automatycznie aktywna.

Ponieważ opcja działania instrukcji GET/PUT w programie CPU w wersji V4.0 nie jest dla CPU w wersji V4.0 automatycznie aktywna. Aby to zmienić, należy przejść do konfiguracji CPU w *Device configuration* (Konfiguracja urządzeń), zakładki *Properties* (Właściwości) w oknie Inspektor, i następnie wybrać właściwość *Protection* (Ochrona), tym samym uaktywniając dostęp do instrukcji GET/PUT (strona 78).

7.9. GPRS

7.9.1. Połączenie z siecią GSM

Komunikacja WAN bazująca na IP za pomocą GPRS

Używając procesora komunikacyjnego CP 1242-7, sterownik S7-1200 może uzyskać dostęp do sieci GSM. CP 1242-7 pozwala na komunikację w sieci WAN pomiędzy odległymi stacjami i centrami sterującymi oraz stacjami pośrednimi.

Komunikacja pomiędzy pośrednimi stacjami jest możliwa tylko poprzez sieć GSM. Aby możliwa była komunikacja pomiędzy sterownią a odległą stacją, centrum sterujące musi być wyposażone w komputer PC z dostępem do Internetu.

CP 1242-7 obsługuje następujące sposoby przesyłania danych przez sieć GSM:

- GPRS (*General Packet Radio Service*)

Pakietowe przesyłanie danych „GPRS“ jest przeprowadzane przez sieć GSM.

- SMS (*Short Message Service*)

CP 1242-7 może otrzymywać i wysyłać wiadomości SMS. Partner komunikacyjny może być telefonem komórkowym lub sterownikiem S7-1200.

CP 1242-7 może być użyty w przedsiębiorstwach na całym świecie i obsługuje następujące pasma częstotliwości:

- 850 MHz,
- 900 MHz,
- 1800 MHz,
- 1900 MHz.

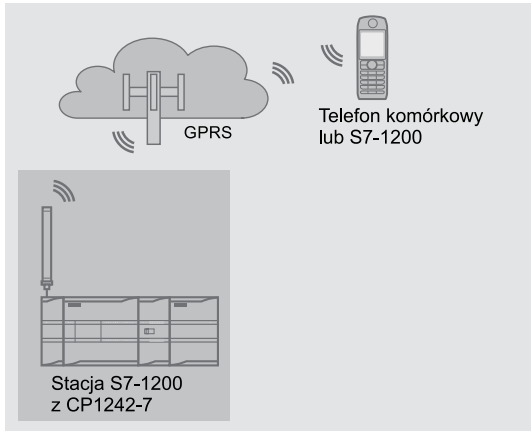
Wymagania

Wyposażenie używane w stacjach lub centrum sterującym zależy od określonej aplikacji.

- Przy komunikacji z lub przez sterownię, centrum sterowania musi być wyposażone w komputer PC z dostępem do Internetu.
- Poza wyposażeniem stacji, odległe stacje S7-1200 z CP 1242-7 muszą spełniać dodatkowe wymagania, aby móc komunikować się przez sieć GSM:
 - Musi zostać podpisana odpowiednia umowa z właściwym dostawcą sieci GSM.
Jeśli używany jest GPRS, umowa musi zezwalać na jego użycie.
Jeśli komunikacja pomiędzy urządzeniami zachodzi tylko po sieci GSM, dostawca sieci GSM musi przyznać stały adres IP dla procesorów komunikacyjnych. W tym wypadku komunikacja pomiędzy urządzeniami nie prowadzi przez centrum sterowania.
 - Karta SIM jest dostarczana w ramach umowy.
Kart SIM jest wkładana do CP 1242-7.
 - Należy zapewnić dostępność sieci GSM w obrębie występowania stacji.

Poniżej znajduje się kilka przykładów konfiguracji stacji z CP 1242-7.

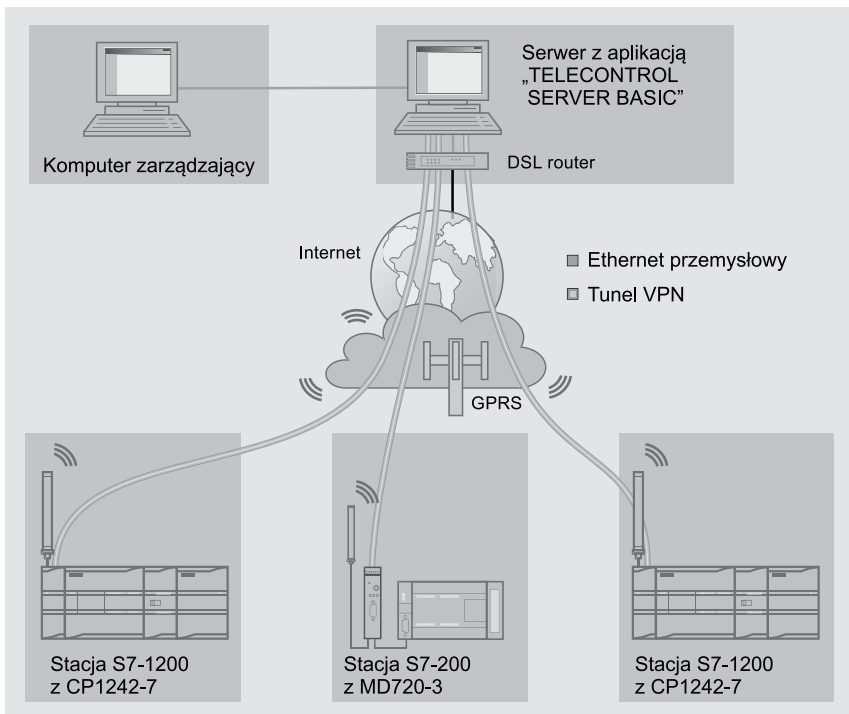
Wysyłanie wiadomości przez SMS



Rys. 7.1. Wysyłanie wiadomości przez SMS ze stacji S7-1200

SIMATIC S7-1200 z procesorem komunikacyjnym CP 1242-7 może wysłać wiadomości przez SMS do skonfigurowanego telefonu komórkowego lub do skonfigurowanej stacji S7-1200.

Zdalna kontrola (teleservice) przez centrum sterujące



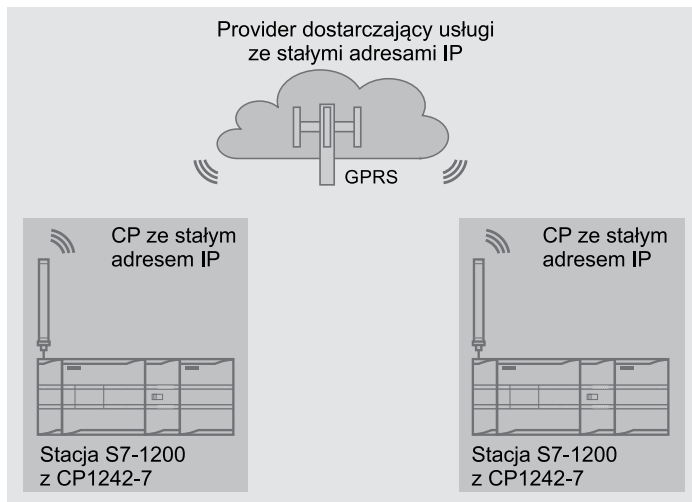
Rys. 7.2. Komunikacja pomiędzy stacją S7-1200 i centrum sterowania

W aplikacjach zdalnego sterowania, stacje S7-1200 z procesorem komunikacyjnym CP 1242-7 komunikują się z centrum sterowania poprzez sieć GSM oraz Internet. Aplikacja TELECONTROL SERVER BASIC jest zainstalowana na serwerze zdalnego sterowania w stacji master. Powoduje to następujące sposoby użytkowania:

- Zdalna komunikacja pomiędzy stacją i centrum sterującym.
W tym przypadku dane z pola są wysyłane przez stacje do serwera sterowania zdalnego w stacji master poprzez sieć GSM oraz Internet. Serwer sterowania zdalnego jest użyty w celu sterowania i monitorowania odległych stacji.
- Komunikacja pomiędzy stacją a komputerem PC w centrum sterowania przez klienta OPC.
W pierwszym przypadku stacje komunikowały się z serwerem sterowania zdalnego. Używając serwera OPC z aplikacji TELECONTROL SERVER BASIC, serwer zdalnego sterowania wymienia dane z komputerem PC podłączonym do stacji master. PC w centrum sterowania może mieć przykładowo zainstalowane oprogramowanie WinCC i zintegrowanego klienta OPC.
- Komunikacja stacji pośrednich przez centrum sterowania.

Aby pozwolić na komunikację stacji pośrednich, serwer sterowania zdalnego przekazuje wiadomości ze stacji nadającej do stacji odbierającej.

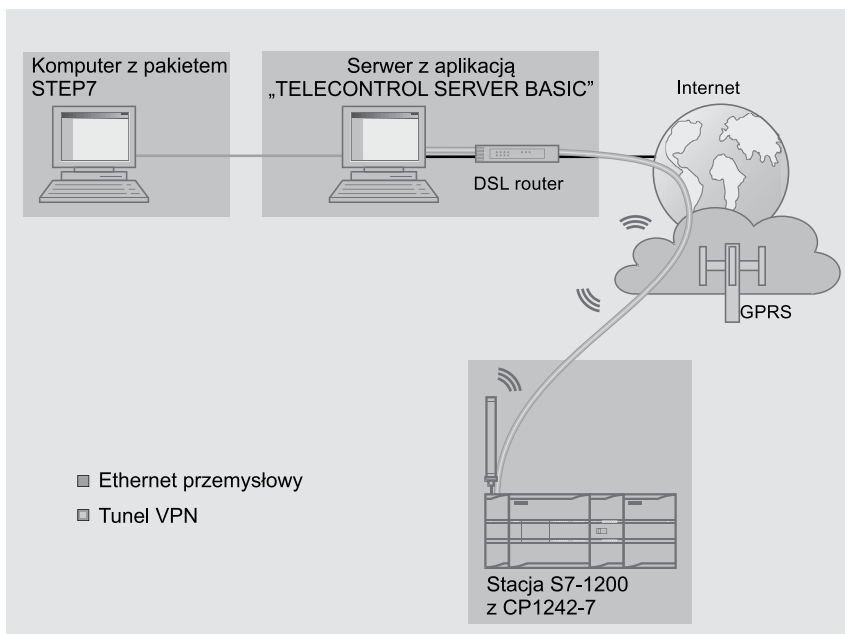
Bezpośrednia komunikacja pomiędzy stacjami



Rys. 7.3. Bezpośrednia komunikacja pomiędzy dwoma stacjami S7-1200

W tej konfiguracji dwie stacje SIMATIC S7-1200 komunikują się ze sobą bezpośrednio po sieci GSM używając procesora komunikacyjnego CP 1242-7. Każdy CP 1242-7 ma stały adres IP. Odpowiednia usługa od dostawcy sieci GSM musi na to zezwalać.

TeleService poprzez GPRS



Rys. 7.4. TeleService poprzez GPRS

TeleService poprzez GPRS: stacja inżynierska, na której zainstalowany jest STEP 7 komunikuje się przez sieć GSM oraz Internet ze stacją SIMATIC S7-1200 za pomocą procesora komunikacyjnego CP 1242-7. Połączenie biegnie poprzez serwer sterowania zdalnego, który funkcjonuje jako pośrednik i jest podłączony do Internetu.

CP 1242-7 może być użyty w następujących aplikacjach:

Aplikacje sterowania zdalnego

- Wysyłanie wiadomości poprzez SMS
SIMATIC S7-1200 z procesorem komunikacyjnym CP 1242-7 może odbierać wiadomości SMS z sieci GSM lub wysyłać wiadomości przez SMS do skonfigurowanego telefonu komórkowego lub do skonfigurowanej stacji S7-1200.
- Komunikacja przez centrum sterowania
Odległe stacje S7-1200 komunikują się ze stacją master poprzez sieć GSM i Internet za pomocą serwera zdalnego sterowania. Aplikacja „TELECONTROL SERVER BASIC” służy do transmisji danych przez GPRS i zainstalowana jest na serwerze zdalnego sterowania na stacji master. Serwer zdalnego sterowania komunikuje się z centralą sterowania wyższego poziomu używając funkcji zintegrowanego serwera OPC.
- Komunikacja pomiędzy stacjami pośrednimi i stacjami S7-1200 za pomocą sieci GSM

Komunikacja stacji pośrednich z odległymi stacjami przez CP 1242-7 może odbywać się na dwa sposoby:

- Pośrednia komunikacja przez stację master.

W tej konfiguracji ustawione jest w stacji master trwałe i bezpieczne połączenie pomiędzy stacjami S7-1200 komunikującymi się pomiędzy sobą za pomocą serwera zdalnego sterowania. CP 1242-7 pracuje w trybie „Telecontrol”.

- Bezpośrednia komunikacja pomiędzy stacjami.

Dla bezpośredniej komunikacji pomiędzy urządzeniami (bez przechodzenia przez stację master) używane są karty SIM ze stałym adresem IP, co pozwala stacjom na bezpośrednie adresowanie pomiędzy sobą. Możliwe usługi komunikacyjne i funkcje bezpieczeństwa (np. VPN) zależą od oferty dostawcy sieci. CP 1242-7 pracuje w trybie „GPRS direct”.

TeleService przez GPRS

Połączenie TeleService może być ustawione pomiędzy stacją inżynierską z STEP 7 i oddalonymi stacjami S7-1200 za pomocą procesora komunikacyjnego CP 1242-7. Komunikacja odbywa się za pomocą sieci GSM i Internetu. Połączenie biegnie od stacji inżynierskiej poprzez serwer zdalnego sterowania lub bramę TeleService, które działają jako pośrednik przekazując ramki i ustawiając autoryzację. Komputery PC używają funkcji z aplikacji „TELECONTROL SERVER BASIC”.

Użytkownik może użyć połączenia TeleService dla następujących celów:

- Załadowania konfiguracji lub danych programu z projektu STEP 7 do stacji.
- Zapytania stacji o dane diagnostyczne

Inne usługi i funkcje procesora komunikacyjnego CP 1242-7

- Synchronizacja czasu dnia (*Time-of-day*) dla CP poprzez Internet.

Użytkownik może także ustawić czas w CP w następujący sposób:

- W trybie „Telecontrol” czas dnia jest transferowany przez serwer zdalnego sterowania. CP używa go do ustawienia swojego czasu.
- W trybie „GPRS direct” CP może zarządzać czasem używając instrukcji SNTP.

Aby zsynchronizować czas CPU należy odczytać bieżący czas z CP używając odpowiedniego bloku.

- Tymczasowe buforowanie wiadomości, które zostaną przesłane, jeśli wystąpią problemy z połączeniem.
- Rejestrowanie objętości danych

Objętość przesyłanych danych jest rejestrowana i może zostać oceniona dla specjalnych celów.

Konfiguracja i zastąpienie modułu

Do konfiguracji modułu jest wymagane następujące narzędzie konfiguracji:

- Program STEP 7 w wersji V11.0 SP1 lub wyższej

Dla STEP 7 V11.0 SP1, jest również wymagany pakiet wsparcia „CP 1242-7” (HSP0003001).

Do transferu danych procesowych poprzez GPRS należy użyć instrukcji komunikacyjnych zdalnego sterowania w programie użytkownika na danej stacji.

Dane konfiguracyjne CP 1242-7 są przechowywane na lokalnym CPU. Pozwala to w razie konieczności na łatwą zamianę procesora komunikacyjnego.

Użytkownik może wstawić do każdego S7-1200 do trzech modułów CP 1242-7. Pozwala to przykładowo na utworzenie dodatkowych redundantnych ścieżek komunikacyjnych.

Połączenia elektryczne

- Zasilanie CP1242-7

Moduł CP ma oddzielne złącze dla zewnętrznego zasilacza 24 V DC.

- Interfejs bezprzewodowy sieci GSM

Wymagana jest dodatkowa antena dla komunikacji w sieci GSM. Jest ona podłączana przez kieszeń SMA znajdującą się na CP.

Antena ANT794-4MR GSM/GPRS

Dostępne są następujące anteny, których używają sieci GPRS/GSM. Mogą one być zainstalowane w pomieszczeniach i na zewnątrz:

- Czterozakresowa antena ANT794-4MR



Rys. 7.5. Antena ANT794-4MR GSM/GPRS

Skrócona nazwa	Numer zamówienia	Objaśnienie
ANT794-4MR	6NH9 860-1AA00	Czterozakresowa antena (900, 1800/1900 MHz, UMTS); odporna na warunki atmosferyczne, do pomieszczeń oraz zastosowań zewnętrznych; 5 m kabel przyłączeniowy połączony na stałe do anteny; wtyczka SMA zawiera zatrzaski instalacyjne, śruby oraz kołki ścienne

- Płaska antena ANT794-3M



Rys. 7.6. Płaska antena ANT794-3M

Skrócona nazwa	Numer zamówienia	Objaśnienie
ANT794-3M	6NH9 870-1AA00	Płaska antena (900, 1800/1900 MHz); odporna na warunki atmosferyczne, do pomieszczeń a także zastosowań zewnętrznych; 1,2 metrowy kabel przyłączeniowy połączony na stałe do anteny; wtyczka SMA zawierająca podkładkę samoprzylepną, możliwe śruby montażowe

Anteny muszą być zamawiane oddzielnie.

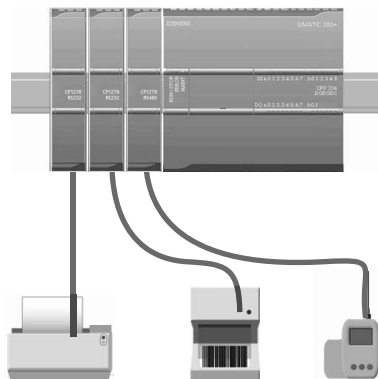
7.10. Protokoły komunikacyjne PtP, USS oraz Modbus

7.10.1 Komunikacja PtP (*Point-to-Point*)

CPU obsługuje komunikację PTP (*Point-to-Point* – Punkt-Punkt) z wykorzystaniem bazującego na znakach protokołu szeregowego:

- PtP (strona 158)
- USS (strona 159)
- Modbus (strona 160)

PtP zapewnia maksymalną swobodę i elastyczność, ale wymaga implementacji w programie użytkownika.



PtP stwarza szeroki zakres możliwości:

- Pozwala wysyłać informacje bezpośrednio do urządzeń zewnętrznych, takich jak drukarka.
- Pozwala odbierać informacje bezpośrednio z innych urządzeń, takich jak czytniki kodu paskowego, czytniki RFID, kamery lub systemy wizyjne innych producentów i wiele innych.
- Pozwala wymieniać informację, wysyłać i odbierać dane, z innymi urządzeniami, takimi jak GPS, kamery i systemy wizyjne innych producentów, modemy radiowe i wiele innych.

Komunikacja PtP jest komunikacją szeregową wykorzystującą standardowe urządzenia UART, które zapewniają różne prędkości transmisji oraz opcje kontroli parzystości. Moduły komunikacyjne (CM 1241) RS232 i RS422/485 oraz płytkę komunikacyjną (CB 1241) RS485 dostarczają interfejsy elektryczne wykorzystywane do realizacji komunikacji PtP.

7.10.2. Korzystanie z szeregowych interfejsów komunikacyjnych

Interfejs dla komunikacji PtP zapewniają dwa moduły komunikacyjne (CM) i jedna płytkę komunikacyjną (CB):

- CM 1241 RS232,
- CM 1241 RS422/485,
- CB 1241 RS485.

Można podłączyć do trzech modułów komunikacyjnych CM (dowolnego typu) i jedną płytkę komunikacyjną CB, co daje w sumie do 4 interfejsów komunikacyjnych. CM należy instalować z lewej strony CPU lub innego CM. Płytkę komunikacyjną CB jest instalowana od przodu CPU. Szczegółowe informacje dotyczące instalacji i deinstalacji modułów są podane w rozdziale *Instalacja* podręcznika systemu S7-1200.

Szeregowe interfejsy komunikacyjne mają następujące cechy:

- Izolowane porty.
- Obsługują protokół Point-to-Point.
- Są konfigurowane i programowane za pomocą instrukcji procesora komunikacji PtP.

- Sygnalizują aktywność nadawania i odbioru diodami LED.
- Są wyposażone w diagnostyczną diodę LED (tylko CM).
- Są zasilane z CPU. Nie jest konieczne zewnętrzne źródło zasilania.

Por. „Dane Techniczne” modułów komunikacyjnych.

Wskaźniki LED

Moduły komunikacyjne wyposażono w trzy wskaźniki LED.

- Diagnostyczna dioda LED (DIAG): Ta dioda LED błyska na czerwono, dopóki nie zostanie zaadresowana przez CPU. Po włączeniu zasilania, CPU sprawdza, czy są zainstalowane moduły CM i adresuje je. Diagnostyczna dioda LED zaczyna błyskać na zielono. Oznacza to, że CPU zaadresował CM, ale jeszcze nie przesłał do niego danych konfiguracyjnych. CPU ładuje konfigurację do skonfigurowanych modułów CM w momencie, kiedy program jest ładowany do CPU. Po załadowaniu do CPU, diagnostyczna dioda LED na module komunikacyjnym powinna świecić w sposób ciągły na zielono.
- Transmisyjna dioda LED (Tx): Dioda świeci, gdy dane są transmitowane na zewnątrz poprzez port komunikacyjny.
- Odbiorcza dioda LED (Rx): Dioda świeci, gdy dane są otrzymywane przez port komunikacyjny.

Płytki komunikacyjne wyposażono w diody sygnalizujące nadawanie (TxD) i odbiór (RxD) danych. Nie wyposażono ich w diodę diagnostyczną.

7.10.3. Instrukcje PtP

Instrukcje PORT_CFG, SEND_CFG oraz RCV_CFG pozwalają użytkownikowi na konfigurację z poziomu programu użytkownika:

- PORT_CFG umożliwia zmianę parametrów portu, takich jak szybkość transmisji,
- SEND_CFG pozwala dynamicznie konfigurować parametry transmisji szeregowej portu komunikacyjnego
- RCV_CFG wykonuje dynamiczną konfigurację parametrów szeregowego odbiornika portu komunikacyjnego Point-to-Point. Instrukcja konfiguruje warunki, które sygnalizują początek i koniec odbieranej wiadomości. Wiadomości, które spełniają te warunki będą odbierane za pomocą instrukcji RCV_PTP.

Zmiany dynamicznej konfiguracji nie są trwale przechowywane w CPU. Po cyklu zasilania zostanie przywrócona początkowa statyczna konfiguracja.

Instrukcje SEND_PTP, RCV_PTP oraz RCV_RST kontrolują komunikację:

- SEND_PTP przesyła określony bufor do CM lub CB. Program CPU kontynuuje działanie podczas gdy moduł wysyła dane z określoną szybkością.
- RCV_PTP sprawdza czy CM lub CB odebrał jakieś wiadomości. Jeżeli wiadomość jest dostępna, to zostanie przesłana z modułu do CPU.
- RCV_RST kasuje bufor odbiorczy.

Każdy moduł CM lub płytka CB może buforować maksymalnie 1 KB. Ten bufor może być zorganizowany tak, by przechowywać wiele odebranych wiadomości.

Instrukcje SGN_SET oraz SGN_GET są obsługiwane tylko przez CM RS232. Należy użyć tych instrukcji do odczytania lub ustawienia stanu sygnałów komunikacyjnych.

7.10.4. Instrukcje USS

S7-1200 obsługuje protokół USS i dostarcza instrukcje, które są zaprojektowane specjalnie do komunikacji z napędami przez port RS485 modułu CM lub płytki CB. Korzystając z biblioteki USS użytkownik może sterować fizycznym napędem i odczytywać/zapisywać parametry napędu. Każdy moduł komunikacyjny CMRS485 lub płytka komunikacyjna CBRS485 obsługuje maksymalnie 16 napędów.

- Instrukcja USS_PORT obsługuje bieżącą komunikację pomiędzy CPU i wszystkimi napędami podłączonymi do CM lub CB. Należy wstawić oddzielną instrukcję USS_PORT dla każdego CM lub CB w aplikacji użytkownika. Program użytkownika musi wywoływać tę funkcję dostatecznie szybko by uniknąć przekroczenia limitu czasu napędu. Ta instrukcja może być wywoływana w głównym lub dowolnym OB obsługi przerw.
- Instrukcja USS_DRV zapewnia programowi użytkownika dostęp do wyspecyfikowanego napędu znajdującego się w sieci USS. Wejścia i wyjścia instrukcji USS_DRV określają stan i pozwalają sterować napędem. Jeżeli w sieci jest 16 napędów, to program użytkownika musi zawierać co najmniej 16 wywołań USS_DRV – jedno na każdy napęd.

Bloki powinny być wywoływane z taką częstotliwością, jak jest wymagana dla realizacji funkcji sterowania napędem. Instrukcja USS_DRV może być wywoływana tylko z głównego OB.

- Instrukcje USS_RPM oraz USS_WPM USS_RPM i USS_WPM odczytują i zapisują parametry operacyjne zdalnego napędu. Te parametry sterują pracą napędu. Są one zdefiniowane w instrukcji napędu. Program użytkownika może zawierać tyle wywołań tych funkcji ile to jest konieczne, ale w dowolnej chwili dla jednego napędu może być aktywna tylko jedna funkcja zapisu lub odczytu. Funkcje USS_RPM i USS_WPM mogą być wywoływane tylko z głównego OB.

Instancja DB zawiera tymczasowe magazynowanie i bufory dla wszystkich napędów podłączonych do CM lub CB w sieci USS. Instrukcje USS dla napędów używają instancji DB do współdzielenia informacji.

Obliczanie czasu potrzebnego do komunikacji z napędem

Komunikacja z napędem odbywa się asynchronicznie w stosunku do cyklu programu CPU. Zwykle CPU wykona kilka cykli programu, zanim zostanie ukończona pełna transakcja komunikacyjna z jednym napędem.

Interwał USS_PORT jest to czas potrzebny do wykonania jednej transakcji komunikacyjnej napędu. W poniższej tabeli przedstawiono minimalne interwały

USS_PORT dla każdej szybkości transmisji komunikacji. Wywołanie funkcji USS_PORT częściej niż wynosi interwał USS_PORT nie powoduje zwiększenia liczby transakcji. Limit czasowy napędu jest to czas jaki może być przeznaczony na transakcję w warunkach, gdy błędy komunikacyjne spowodowały, że podjęte były 3 próby dokończenia transakcji. Domyślnie, protokół biblioteki USS podejmuje automatycznie do 2 prób na każdą transakcję.

Tabela 7.14. Wymagania czasowe

Szybkość transmisji (bps)	Obliczony minimalny interwał USS_PORT (milisekundy)	Limit czasowy pojedynczego napędu (milisekundy)
1200	790	2370
2400	405	1215
4800	212,5	638
9600	116,3	349
19 200	68,2	205
38 400	44,1	133
57 600	36,1	109
115 200	28,1	85

7.10.5. Instrukcje obsługujące protokół Modbus

CPU obsługuje komunikację Modbus w różnych sieciach:

- Modbus RTU (*Remote Terminal Unit*) jest standardowym protokołem komunikacji sieciowej, który używa połączeń RS232 oraz RS485 do szeregowej transmisji danych pomiędzy urządzeniami sieciowymi Modbus. Użytkownik może dodać port PtP (*Point to Point*) do CPU poprzez wykorzystanie płytki komunikacyjnej (CB) RS485 lub modułu komunikacyjnego (CM) RS232 lub RS485.

Modbus RTU używa sieci typu master/slave, gdzie komunikacja jest nawiązywana przez pojedyncze urządzenie Master, a urządzenia typu slave mogą jedynie odpowiadać na żądania urządzenia master. Urządzenie master wysyła żądanie pod jeden adres slave i tylko ten adres slave może odpowiedzieć na tą komendę.

- Modbus TCP (*Transmission Control Protocol*) jest standardowym protokołem komunikacyjnym używającym złącza PROFINET do komunikacji TCP/IP. Żadne dodatkowe moduły komunikacyjne nie są potrzebne.

Modbus TCP używa połączeń serwer-klient jako ścieżki komunikacyjnej dla protokołu Modbus. Można stworzyć, poza połączeniem pomiędzy STEP 7 i CPU, wiele dodatkowych połączeń serwer-klient. Mieszane połączenia pomiędzy klientem i serwerem mogą występować w liczbie, na którą zezwala CPU. Każde połączenie MB_SERVER musi mieć niepowtarzalną instancję DB oraz numer portu IP. Jeden port IP obsługuje tylko jedno połączenie. Każde połączenie MB_SERVER (z unikalną instancją DB i portem IP) musi być wykonywane indywidualnie.

⚠ OSTRZEŻENIE

Jeśli atakujący może uzyskać fizyczny dostęp do sieci, to jest zagrożenie, że może odczytywać i zapisywać dane.

TIA Portal, CPU i urządzenia HMI (z wyjątkiem urządzeń HMI używających instrukcji GET/PUT) zapewniają bezpieczną komunikację, która zabezpiecza przed atakami przez *replay attack* oraz atakami typu *man-in-the-middle* (podśluch i modyfikacja bez wiedzy stron komunikacji). Po aktywacji komunikacji, odbywa się wymiana podpisanych wiadomości w postaci zwykłego tekstu, co umożliwia atakującemu odczytywać dane, ale zabezpiecza przed nieautoryzowanym zapisem danych. To TIA Portal, a nie proces komunikacji, szyfruje dane chronionych bloków zawierających informacje technologiczne.

Wszystkie inne formy komunikacji (wymiana danych I/O przez PROFIBUS, PROFINET, AS-i, lub inne magistrale I/O, instrukcje GET/PUT, bloki nadawcze (T-Block), oraz moduły komunikacyjne (CM)) nie mają żadnych zabezpieczeń. Te formy komunikacji należy chronić przez ograniczenie dostępu fizycznego. Jeśli atakujący może uzyskać fizyczny dostęp do sieci wykorzystując te formy komunikacji, to jest zagrożenie, że może odczytywać i zapisywać dane.

Informacje i zalecenia dotyczące bezpieczeństwa, można znaleźć w publikacji *Operational Guidelines for Industrial Security* (Wytyczne operacyjne dotyczące bezpieczeństwa przemysłowego) na stronie pomocy technicznej firmy Siemens: www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf).

Uwaga

Protokół Modbus TCP pracuje poprawnie z firmware CPU w wersji V1.02 lub nowszej. Próba wykonania instrukcji Modbus za pomocą starszego oprogramowania sprzętowego zakończy się błędem.

Tabela 7.15. Instrukcje obsługujące Modbus

Typ komunikacji	Instrukcja
Modbus RTU (RS232 lub RS485)	MB_COMM_LOAD: Instrukcja MB_COMM_LOAD konfiguruje parametry portu PtP, takie jak: szybkość transmisji, parzystość oraz sterowanie przepływem. Po skonfigurowaniu portu CPU dla protokołu Modbus RTU, może on używać tylko instrukcji MB_MASTER lub MB_SLAVE.
	MB_MASTER: Instrukcja umożliwia CPU pracę jako urządzenie Modbus RTU master i komunikację z jednym lub więcej urządzeniami Modbus slave.
	MB_SLAVE: Instrukcja umożliwia CPU pracę jako urządzenie Modbus RTU slave i komunikację z urządzeniem Modbus master.

Typ komunikacji	Instrukcja
Modbus TCP (PROFINET)	MB_CLIENT: Tworzy połączenie TCP typu klient-serwer, wysyła komunikaty polecenia, odbiera odpowiedzi i steruje odłączeniami z sieci.
	MB_SERVER: Łączy na żądanie z klientem Modbus TCP, odbiera komunikaty Modbus i wysyła odpowiedzi.

Instrukcje Modbus nie używają zdarzeń przerwania komunikacji aby sterować procesem komunikacji. Program musi odpytywać instrukcjami MB_MASTER/MB_SLAVE lub MB_Client/MB_Server aby otrzymać kompletne informacje.

Klient Modbus TCP (master) musi sterować połączeniem klient-serwer za pomocą parametru DISCONNECT. Podstawowe działania klienta Modbus są pokazane poniżej:

1. Rozpoczęcie połączenia z danym adresem IP i numerem portu IP dla serwera (slave).
2. Rozpoczęcie transmisji komunikatów Modbus od klienta i odbiór odpowiedzi serwera.
3. Jeśli jest to wymagane, rozpoczęcie rozłączania połączenia klienta z serwerem, aby rozpocząć połączenie z innym serwerem.

STEP 7 pozwala na użycie następujących instrukcji sterownika S7-1200, które upraszczają konfigurację i użycie regulatora PID:

- Instrukcja PID_Compact używana do sterowania procesem technicznym o ciągłych wartościach na wejściu i wyjściu.
- Instrukcja PID_3Step do sterowania urządzeniami napędzanymi silnikami, takimi jak zawory, które potrzebują sygnałów dyskretnych dla działania otwierającego i zamykającego.

Uwaga

Zmiany wprowadzone w konfiguracji PID oraz załadowanie danych w trybie RUN nie odnoszą skutku, dopóki nie nastąpi przejście CPU z trybu STOP do trybu RUN. Zmiany wprowadzone w oknie dialogowym *PID Parameters* (Parametry PID) przy użyciu opcji *Start value control* (Regulacja wartości początkowej) odnośną skutek natychmiastowy.

Obydwie instrukcje PID mogą obliczyć składniki P-, I- oraz D podczas rozruchu (jeśli jest ustawiona opcja strojenia przed uruchomieniem – *pretuning*) Użytkownik może także skonfigurować instrukcję dla dostrajania (*fine tuning*), które pozwoli na optymalizację parametrów. Użytkownik nie musi ręcznie dostrajać parametrów.

Uwaga!

Należy wykonywać instrukcje PID w stałych przedziałach czasowych czasu próbkowania (najlepiej w cyklicznym OB).

Ponieważ pętla PID potrzebuje określonego czasu na odpowiedź po zmianie sterowanej wartości, nie należy obliczać wartości wyjściowych w każdym cyklu. Nie należy wstawiać instrukcji PID do OB głównego cyklu programu (takich jak OB 1).

Czas próbkowania algorytmu PID odpowiada czasowi pomiędzy obliczeniem dwóch kolejnych wartości wyjściowych (wartości sterowania).

Algorytm PID

Regulator PID (Proportional/Integral/Derivative – Proporcjonalny/Całkujący/Różniczkujący) mierzy odstęp czasu pomiędzy dwoma wywołaniami i szacuje wyniki

monitorowania czasu próbkowania. Wartość średnia czasu próbkowania jest generowana przy każdej zmianie trybu pracy i podczas wstępnego rozruchu. Wartość ta jest używana jako odniesienie dla funkcji nadzorujących w obliczeniach. Monitorowanie obejmuje pomiar bieżącego czasu pomiędzy dwoma wywołaniami i uśrednianie zdefiniowanego przez sterownik czasu próbkowania.

Wartości wyjściowe regulatora PID składają się z trzech składników:

- P (proporcjonalny): wartość wyjściowa jest proporcjonalna do różnicy pomiędzy wartością zadaną, a wartością bieżącą (uchybu).
- I (całkujący): wartość wyjściowa wzrasta proporcjonalnie do czasu trwania różnicy pomiędzy wartością zadaną i wartością bieżącą, aż do zlikwidowania uchybu.
- D (różniczkujący) wartość wyjściowa wzrasta jako funkcja wzrostu szybkości zmian uchybu. Wartość wyjściowa jest zrównana do wartości zadanej najszybciej jak to tylko możliwe.

Regulator PID dla instrukcji PID_Compact używa następującego wzoru do obliczenia wartości wyjściowej.

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Wartość wyjściowa	x	Wartość bieżąca
w	Wartość zadana	s	Operator Laplace'a
K_p	Proporcjonalne wzmocnienie (Składnik P)	a	Współczynnik opóźnienia całkowania (Składnik D)
T_i	Czas całkowania - czas zdwojenia (Składnik I)	b	Wagi działania proporcjonalnego (Składnik P)
T_d	Czas różniczkowania czas wyprzedzenia (Składnik D)	c	Wagi działania różniczkującego (Składnik D)

Regulator PID dla instrukcji PID_3Step używa następującego wzoru do obliczenia wartości wyjściowej.

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Wartość wyjściowa	x	Wartość bieżąca
w	Wartość zadana	s	Operator Laplace'a
K_p	Proporcjonalne wzmocnienie (Składnik P)	a	Współczynnik opóźnienia całkowania (Składnik D)
T_i	Czas całkowania - czas zdwojenia (Składnik I)	b	Wagi działania proporcjonalnego (Składnik P)
T_d	Czas różniczkowania czas wyprzedzenia (Składnik D)	c	Wagi działania różniczkującego (Składnik D)

8.1. Wstawianie instrukcji PID i obiektów technologicznych

STEP 7 zapewnia dwie instrukcje do obsługi regulatora PID:

- Instrukcja PID_Compact i związane z nią obiekty technologiczne tworzą uniwersalny regulator PID wraz z dostrajaniem nastaw. Obiekty technologiczne zawierają wszystkie ustawienia dla pętli regulacji.
- Instrukcja PID_3Step i związane z nią obiekty technologiczne tworzą regulator PID ze specjalnymi ustawieniami dla zaworów sterowanych silnikami. Obiekty technologiczne zawierają wszystkie ustawienia dla pętli regulacji. Regulator PID_3Step ma także dwa dodatkowe wyjścia logiczne.

Po stworzeniu obiektu technologicznego użytkownik musi skonfigurować jego parametry (strona 165). Należy określić tryb autodostrajania (*pre-tuning* podczas rozruchu lub ręczne dostrajanie – *fine tuning*), aby narzucić tryb pracy regulatora PID (strona 184).

Tabela 8.1. Wstawianie instrukcji PID i obiektów technologicznych

Podczas wstawiania instrukcji PID do programu użytkownika STEP 7 automatycznie tworzy dla instrukcji obiekt technologiczny oraz instancję DB. Instancja DB zawiera wszystkie parametry używane przez instrukcje PID. Każda instrukcja PID musi posiadać własną, niepowtarzalną instancję DB. W przeciwnym wypadku nie będzie działać poprawnie.

Po wstawieniu instrukcji PID i stworzeniu obiektu technologicznego oraz instancji DB, użytkownik może skonfigurować parametry obiektu technologicznego (strona 181).

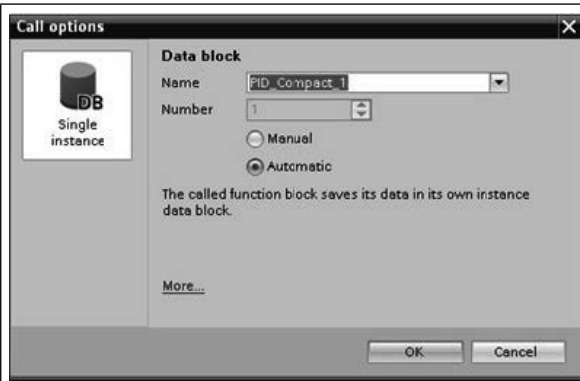
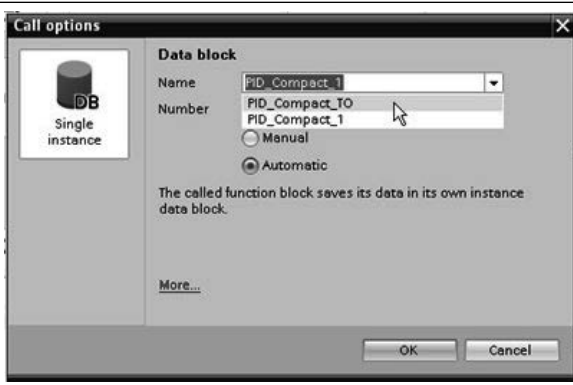

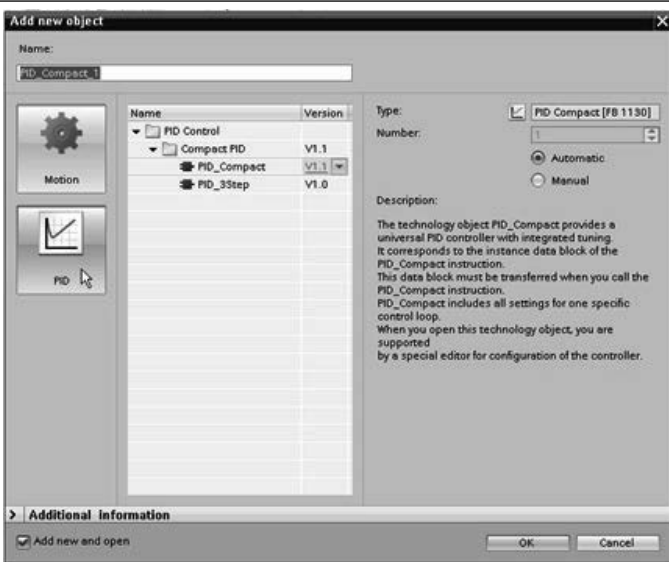


Tabela 8.2. Tworzenie projektu technologicznego z poziomu nawigatora projektu (opcjonalnie)

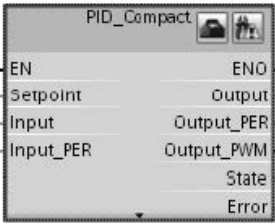
<p>Istnieje także możliwość stworzenie obiektu technologicznego przed wstawieniem instrukcji PID. Przez stworzenie obiektu technologicznego przed wstawieniem instrukcji PID do programu użytkownika można wybrać później odpowiedni obiekt technologiczny podczas późniejszej operacji wstawiania instrukcji PID.</p>	
<p>Aby stworzyć obiekt technologiczny, należy dwukrotnie kliknąć ikonę <i>Add new object</i> w nawigatorze projektu.</p>	

<p>Należy kliknąć ikonę <i>Control</i>, aby wybrać obiekt technologiczny dla danego typu regulatora PID (PID_Compact lub PID_3Step). Użytkownik może stworzyć nazwę dla tego obiektu. Naciśnij <i>OK</i>, aby stworzyć obiekt technologiczny.</p>	
---	---

8.2. Instrukcja PID_Compact

Instrukcja PID_Compact dostarcza uniwersalny regulator PID ze zintegrowanym samostrojaniem w trybie automatycznym i ręcznym.

Tabela 8.3. Instrukcja PID_Compact

LAD/FBD	SCL	Opis
<p>"PID_Compact_T0"</p> 	<pre>"PID_Compact_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, Output=>_real_out_, Output_PER=>_word_out_, Output_PWM=>_bool_out_, SetpointLimit_H=>_bool_out_, SetpointLimit_L=>_bool_out_, InputWarning_H=>_bool_out_, InputWarning_L=>_bool_out_, State=>_int_out_, Error=>_bool_out_, ErrorBits=>_dword_out_);</pre>	<p>PID_Compact umożliwia użycie regulatora PID z autostrajaniem w trybie automatycznym i manualnym. PID_Compact jest regulatorem typu PIDT1 z mechanizmem zabezpieczenia przed <i>anti-windup</i> oraz ważeniem składników P oraz D.</p>

¹ Podczas wstawiania instrukcji PID do programu użytkownika STEP 7 automatycznie tworzy dla instrukcji obiekt technologiczny i instancję DB. Instancja DB zawiera wszystkie parametry obiektu technologicznego.

² W przykładzie SCL, "PID_Compact_1" jest nazwą DB instancji.

Tabela 8.4. Typy danych dla określonych parametrów

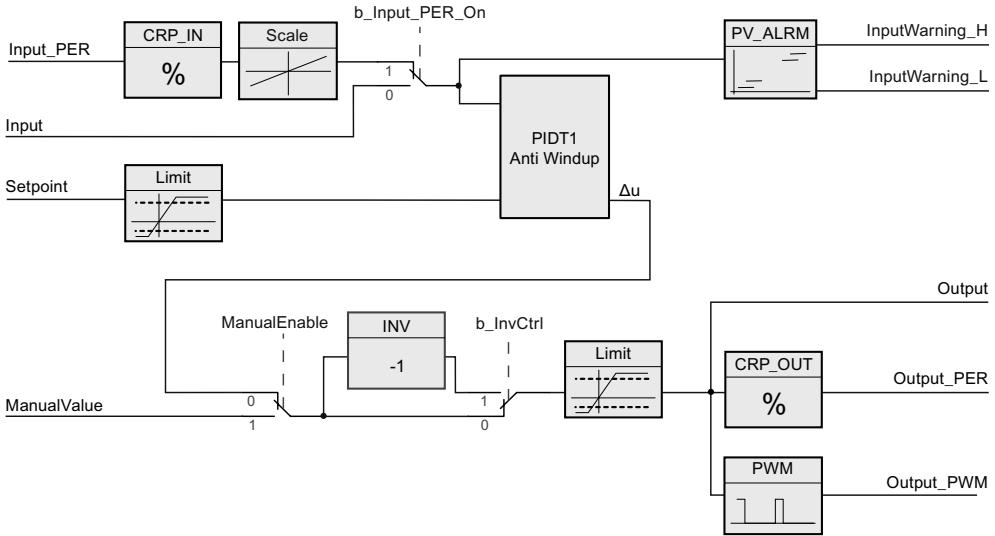
Parametr i jego typ		Typ danych	Opis
Setpoint	IN	Real	Wartość zadana dla regulatora PID w trybie automatycznym. (Wartość domyślna: 0.0)
Input	IN	Real	Tag programu użytkownika jest wykorzystywany jako źródło wartości procesowej. (Wartość domyślna: 0.0). Jeśli jest używany parametr Input, to należy ustawić Config.InputPerOn = FALSE.
Input_PER	IN	Word	Wejście analogowe jest używane jako źródło wartości procesowej. (Wartość domyślna: W#16#0). Jeśli jest używany parametr Input_PER, to należy ustawić Config.InputPerOn = TRUE.
Disturbance	IN	Real	Zmienna zakłócenia lub wartość przesterowania
ManualEnable	IN	Bool	Włącza lub wyłącza ręczny tryb pracy. (Wartość domyślna: FALSE): <ul style="list-style-type: none"> Zmiana zbocza z FALSE na TRUE aktywuje "tryb ręczny", State = 4, Mode pozostaje niezmieniony. Dopóki ManualEnable = PRAWDA, dopóty nie można zmienić trybu pracy za pomocą zbocza narastającego na wejściu ModeActivate lub użyć okna dialogowego uruchomienia. Zmiana zbocza z TRUE na FALSE aktywuje tryb pracy przypisany przez Mode. Uwaga: Zaleca się, aby tryb pracy zmieniać tylko za pomocą ModeActivate.
ManualValue	IN	Real	Wartość wyjściowa dla ręcznego trybu pracy. (Wartość domyślna: 0.0). Można użyć wartości z przedziału od Config.OutputLowerLimit do Config.OutputUpperLimit.
ErrorAck	IN	Bool	Resetuje ErrorBits i wyjścia ostrzeżeń. Zmiana zbocza z FALSE na TRUE
Reset	IN	Bool	Ponownie uruchamia regulator. (Wartość domyślna: FALSE): <ul style="list-style-type: none"> Zmiana zbocza z FALSE na TRUE: <ul style="list-style-type: none"> Przełącza na nieaktywny tryb pracy Resetuje ErrorBits oraz wyjścia ostrzeżeń Usuwa część całkowitą Zachowuje parametry PID Dopóki Reset = TRUE, dopóty PID_Compact pozostaje w nieaktywnym trybie pracy (State = 0). Zmiana zbocza z TRUE na FALSE: <ul style="list-style-type: none"> PID_Compact przełącza na tryb pracy, który jest zapisany w parametrze Mode.
ModeActivate	IN	Bool	PID_Compact przełącza na tryb pracy, który jest zapisany w parametrze Mode. Zmiana zbocza z FALSE na TRUE:
Mode	IN	Int	Pożądaný tryb PID; Aktywowany przednim zboczem na wejściu ModeActivate.
ScaledInput	OUT	Real	Skalowana wartość procesowa. (Wartość domyślna: 0.0)
Output ¹	OUT	Real	Wartość wyjściowa w formacie REAL. (Wartość domyślna: 0.0)
Output_PER ¹	OUT	Word	Analogowa wartość wyjściowa. (Wartość domyślna: W#16#0)
Output_PWM ¹	OUT	Bool	Wartość wyjściowa dla modulacji szerokości impulsów. (Wartość domyślna: FALSE) Wartość wyjściowa utworzona w oparciu o czasy włączania i wyłączania.

Tabela 8.4. Typy danych dla określonych parametrów

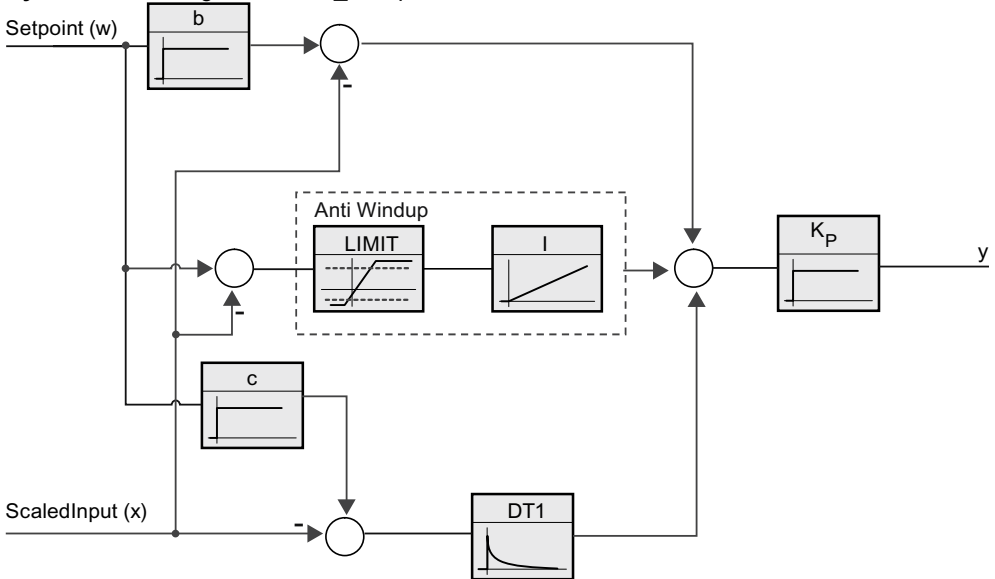
Parametr i jego typ		Typ danych	Opis
SetpointLimit_H	OUT	Bool	Górny limit wartości zadanej. (Wartość domyślna: FALSE.) Jeśli SetpointLimit_H = TRUE, to osiągnięto wartość bezwzględną górnego limitu wartości zadanej (Setpoint \geq Config.SetpointUpperLimit). Wartość zadana jest ograniczona do Config.SetpointUpperLimit.
SetpointLimit_L	OUT	Bool	Dolny limit wartości zadanej. (Wartość domyślna: FALSE) Jeśli SetpointLimit_L = TRUE, to osiągnięto wartość bezwzględną dolnego limitu wartości zadanej (Setpoint \leq Config.SetpointLowerLimit). Wartość zadana jest ograniczona do Config.SetpointLowerLimit.
InputWarning_H	OUT	Bool	Jeśli InputWarning_H = TRUE, to wartość procesowa osiągnęła lub przekroczyła górny limit wartości ostrzeżenia. (Wartość domyślna: FALSE)
InputWarning_L	OUT	Bool	Jeśli InputWarning_L = TRUE, to wartość procesowa osiągnęła lub przekroczyła dolny limit wartości ostrzeżenia. (Wartość domyślna: FALSE)
State	OUT	Int	Aktualny tryb pracy regulatora PID. (Wartość domyślna: 0). Można zmienić tryb pracy za pomocą parametru wejściowego Mode i narastającego zbocza na wejściu ModeActivate: <ul style="list-style-type: none"> • State = 0: Nieaktywny • State = 1: Wstępne strojenie • State = 2: Ręczne dokładne strojenie • State = 3: Tryb automatyczny • State = 4: Tryb ręczny • State = 5: Zamienna wartość wyjściowa z monitorowaniem błędów
Error	OUT	Bool	Jeśli Error = TRUE, to występuje przynajmniej jeden komunikat o błędzie w tym cyklu programu. (Wartość domyślna: FALSE). Uwaga: Parametr Error w PID w wersji V1.x był polem ErrorBits, które zawierało kody błędów. Teraz znacznik boolowski wskazuje, że wystąpił błąd.
ErrorBits	OUT	DWord	Tabela parametrów ErrorBits instrukcji PID_Compact (strona 170) definiuje komunikaty o błędach, które występują. (Wartość domyślna: DW#16#0000 (brak błędu)). Parametr ErrorBits jest trwale zapamiętany i jest resetowany zboczem narastającym na wejściach Reset lub ErrorAck. Uwaga: W wersji V1.x, parametr ErrorBits został zdefiniowany jako parametr Error i nie istnieje.

¹ Wyjścia parametrów Output, Output_PER oraz Output_PWM mogą być używane równolegle.

Regulator PID_Compact



Rys. 8.1. Praca regulatora PID_Compact



Rys. 8.2. Praca regulatora PID_Compact jako regulator PIDT1 z algorytmem anti-windup

8.3. Parametry ErrorBit instrukcji PID_Compact

Jeśli w toku przetwarzania występuje kilka błędów, to wartości kodów błędów są wyświetlane za pomocą binarnego dodawania. Wyświetlenie kodu błędu 0003, na przykład, oznacza, że błędy 0001 i 0002 są nadal w toku przetwarzania.

Tabela 8.5. Parametry ErrorBit instrukcji PID_Compact

ErrorBit (DW#16#...)	Opis
0000	Brak błędu.
0001 ^{1, 2}	Parametr wejściowy Input ma wartość poza granicami wartości procesowej. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Niepoprawna wartość parametru „Input_PER”. Należy sprawdzić, czy na wejściu analogowym występuje błąd
0004 ⁴	Błąd podczas dokładnego strojenia. Oscylacje wartości procesowej nie mogą być utrzymane.
0008 ⁴	Błąd podczas rozpoczęcia wstępnego strojenia. Wartość procesowa jest zbyt blisko wartości zadanej. Należy rozpocząć dokładne strojenie.
0010 ⁴	Wartość zadana została zmieniona podczas strojenia. Uwaga: Można ustawić dopuszczalne odchylenie wartości zadanej tagu CancelTuningLevel.
0020	Wstępne strojenie nie jest dozwolone podczas dokładnego strojenia. Uwaga: Jeśli ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_Compact pozostaje w trybie dokładnego strojenia.
0080 ⁴	Błąd podczas wstępnego strojenia. Niewłaściwa konfiguracja limitów wartości wyjściowej. Należy sprawdzić, czy limity wartości wyjściowej są poprawnie skonfigurowane i odpowiadają logice sterowania.
0100 ⁴	Błąd podczas dokładnego strojenia spowodował niepoprawne parametry.
0200 ^{2, 3}	Niepoprawna wartość parametru Input: wartość ma niepoprawny format liczbowy.
0400 ^{2, 3}	Obliczenie wartości wyjściowej nie powiodło się. Należy sprawdzić parametry PID.
0800 ^{1, 2}	Błąd czasu próbkowania: Instrukcja PID_Compact nie jest wywoływana w czasie próbkowania OB cyklicznego przerwania.
1000 ^{2, 3}	Niepoprawna wartość parametru Setpoint: wartość ma niepoprawny format liczbowy.
10000	Niepoprawna wartość parametru ManualValue: wartość ma niepoprawny format liczbowy. Uwaga: Jeśli ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_Compact używa SubstituteOutput jako wartości wyjściowej. Gdy tylko zostanie przypisana poprawna wartość parametru ManualValue, to PID_Compact będzie używać jej jako wartości wyjściowej.

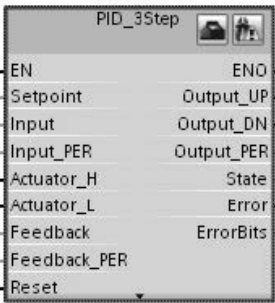
ErrorBit (DW#16#...)	Opis
20000	<p>Niepoprawna wartość tagu SubstituteValue: wartość ma niepoprawny format liczbowy.</p> <p>PID_Compact używa dolnego limitu wartości wyjściowej jako wartości wyjściowej. Uwaga: Jeśli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE i błąd już nie występuje, to PID_Compact powraca do trybu automatycznego.</p>
40000	<p>Niepoprawna wartość parametru Disturbance: wartość ma niepoprawny format liczbowy.</p> <p>Uwaga: Jeśli tryb automatyczny był aktywny i ActivateRecoverMode = FALSE przed wystąpieniem błędu, to parametr Disturbance jest ustawiony na zero. PID_Compact pozostaje w trybie automatycznym.</p> <p>Uwaga: Jeśli tryb wstępnego strojenia, lub dokładnego strojenia był aktywny i ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_Compact przełącza na tryb pracy, który jest zapisany w parametrze Mode. Jeśli Disturbance w obecnej fazie nie ma żadnego wpływu na wartość wyjściową, to strojenie nie jest anulowane.</p>

- ¹ Uwaga: Jeśli tryb automatyczny był aktywny przed wystąpieniem błędu i ActivateRecoverMode = TRUE, to PID_Compact pozostaje w trybie automatycznym.
- ² Uwaga: Jeśli tryb wstępnego strojenia lub dokładnego strojenia był aktywny przed wystąpieniem błędu i ActivateRecoverMode = TRUE, to PID_Compact przełącza na tryb pracy, który jest zapisany w parametrze Mode.
- ³ Uwaga: Jeśli tryb automatyczny był aktywny przed wystąpieniem błędu i ActivateRecoverMode = TRUE, to PID_Compact wyprowadza skonfigurowane zamienne wartości wyjściowe. Gdy błąd już nie występuje, to PID_Compact przełącza na tryb automatyczny.
- ⁴ Uwaga: Jeśli ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_Compact anuluje strojenie i przełącza na tryb pracy, który jest zapisany w parametrze Mode.

8.4. Instrukcja PID_3Step

PID_3Step umożliwi użycie regulatora PID z autodostrajaniem, który został dostosowany do regulacji zaworów i elementów wykonawczych sterowanych silnikiem.

Tabela 8.6. Instrukcja PID_3Step

LAD/FBD	SCL	Opis
<p>"PID_3Step_TO"</p> 	<pre> „PID_3Step_1”(SetpoInt:=_real_in_, Input:=_real_in_, ManualValue:=_real_in_, Feedback:=_real_in_, InputPer:=_word_in_, FeedbackPer:=_word_in_, ManualEnable:=_bool_in_, ManualUP:=_bool_in_, ManualDN:=_bool_in_, ActuatorH:=_bool_in_, ActuatorL:=_bool_in_, Reset:=_bool_in_, ScaledInput=>_real_out_, ScaledFeedback=>_real_out_, ErrorBits=>_dword_out_, OutputPer=>_word_out_, State=>_int_out_, OutputUP=>_bool_out_, OutputDN=>_bool_out_, SetpoIntLimitH=>_bool_out_, SetpoIntLimitL=>_bool_out_, InputWarningH=>_bool_out_, InputWarningL=>_bool_out_, Error=>_bool_out_); </pre>	<p>PID_3Step umożliwia użycie regulatora PID z autodostrajaniem, który został dostosowany do regulacji zaworów i elementów wykonawczych sterowanych silnikiem. PID_3Step jest regulatorem typu PID1 z mechanizmem zabezpieczenia <i>anti-windup</i> oraz ważeniem składników P oraz D.</p>

¹ Podczas wstawiania instrukcji PID do programu użytkownika STEP 7 automatycznie tworzy dla instrukcji obiekt technologiczny i instancję DB. Instancja DB zawiera wszystkie parametry obiektu technologicznego.

² W przykładzie SCL, "PID_3Step_1" jest nazwą DB instancji.

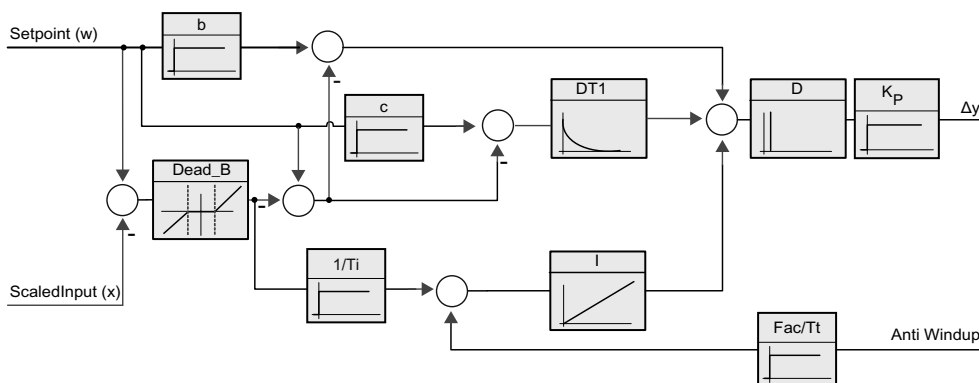
Tabela 8.7. Typy danych dla określonych parametrów

Parametr i jego typ	Typ danych	Opis	
Setpoint	IN	Real	Wartość zadana dla regulatora PID w trybie automatycznym. (Wartość domyślna: 0.0)
Input	IN	Real	Tag programu użytkownika jest wykorzystywany jako źródło wartości procesowej. (Wartość domyślna: 0.0) Jeśli jest używany parametr Input, to należy ustawić Config. InputPerOn = FALSE.
Input_PER	IN	Word	Wejście analogowe jest używane jako źródło wartości procesowej. (Wartość domyślna: W#16#0) Jeśli jest używany parametr Input_PER, to należy ustawić Config. InputPerOn = TRUE.
Actuator_H	IN	Bool	Cyfrowy sygnał zwrotny położenia zaworu dla górnej pozycji końcowej. Jeśli Actuator_H = TRUE, zawór jest w górnej pozycji końcowej i nie może kontynuować ruchu w tym kierunku. (Wartość domyślna: FALSE).
Actuator_L	IN	Bool	Cyfrowy sygnał zwrotny położenia zaworu dla dolnej pozycji końcowej. Jeśli Actuator_L = TRUE, zawór jest w dolnej pozycji końcowej i nie może kontynuować ruchu w tym kierunku. (Wartość domyślna: FALSE).

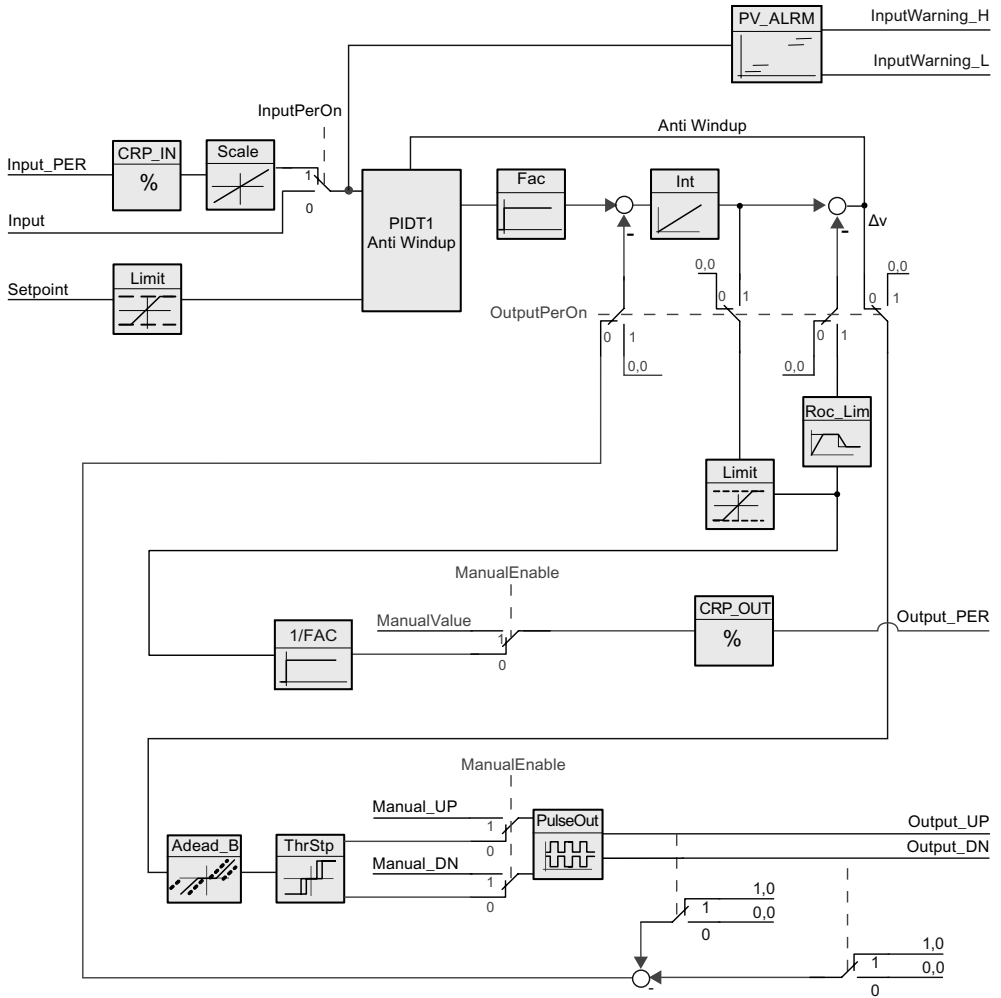
Parametr i jego typ		Typ danych	Opis
Feedback	IN	Real	Sprężenie zwrotne od położenia zaworu. (Wartość domyślna: 0.0). Jeśli jest używany parametr Feedback, to należy ustawić Config.FeedbackPerOn = FALSE.
Feedback_PER	IN	Word	Analogowe sprężenie zwrotne od pozycji zaworu. (Wartość domyślna: W#16#0). Jeśli jest używany parametr Feedback_PER, to należy ustawić Config.FeedbackPerOn = TRUE. Feedback_PER jest skalowane za pomocą następujących tagów: <ul style="list-style-type: none"> • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut
Disturbance	IN	Real	Zmienna zakłócenia lub wartość przesterowania
ManualEnable	IN	Bool	Włącza lub wyłącza ręczny tryb pracy. (Wartość domyślna: FALSE): <ul style="list-style-type: none"> • Zmiana zbocza z FALSE na TRUE aktywuje „tryb ręczny”, State = 4, Mode pozostaje niezmieniony. Dopóki ManualEnable = PRAWDA, dopóty nie można zmienić trybu pracy za pomocą zbocza narastającego na wejściu ModeActivate lub użyć okna dialogowego uruchomienia. • Zmiana zbocza z TRUE na FALSE aktywuje tryb pracy przypisany przez Mode. <p>Uwaga: Zaleca się, aby tryb pracy zmieniać tylko za pomocą ModeActivate.</p>
ManualValue	IN	Real	Wartość bieżąca procesu dla ręcznego trybu pracy. (Wartość domyślna: 0.0). W trybie ręcznym użytkownik określa bezwzględna pozycję zaworu. ManualValue jest obliczany tylko jeśli użytkownik używa OutputPer, lub jeśli dostępne jest sprzężenie zwrotne . Wartość domyślna: 0.0
ManualUP	IN	Bool	<ul style="list-style-type: none"> • Manual_UP = TRUE: <ul style="list-style-type: none"> – Zawór jest otwarty, nawet jeśli używa się Output_PER lub sprzężenia zwrotnego położenia. Jeśli została osiągnięta górna pozycja końcowa, to zawór nie jest już dłużej przesuwany. – Patrz również Config.VirtualActuatorLimit • Manual_UP = FALSE: <ul style="list-style-type: none"> – Jeśli używa się Output_PER lub sprzężenia zwrotnego położenia, zawór przesuwa się do ManualValue. W przeciwnym razie, zawór nie przesuwa się. <p>Uwaga: Jeśli Manual_UP i Manual_DN są jednocześnie ustawione na TRUE, to zawór nie przesuwa się.</p>
ManualDN	IN	Bool	<ul style="list-style-type: none"> • Manual_DN = TRUE: <ul style="list-style-type: none"> – Zawór jest otwarty, nawet jeśli używa się Output_PER lub sprzężenia zwrotnego położenia. Jeśli została osiągnięta górna pozycja końcowa, to zawór nie jest już dłużej przesuwany. – Patrz również Config.VirtualActuatorLimit • Manual_DN = FALSE: <ul style="list-style-type: none"> – Jeśli używa się Output_PER lub sprzężenia zwrotnego położenia, zawór przesuwa się do ManualValue. W przeciwnym razie, zawór nie przesuwa się.

Parametr i jego typ		Typ danych	Opis
ErrorAck	IN	Bool	Resetuje ErrorBits i wyjścia ostrzeżeń. Zmiana zbocza z FALSE na TRUE
Reset	IN	Bool	Ponownie uruchomia regulator. (Wartość domyślna: FALSE): <ul style="list-style-type: none"> Zmiana zbocza z FALSE na TRUE: <ul style="list-style-type: none"> Przełącza na nieaktywny tryb pracy Resetuje ErrorBits oraz wyjścia ostrzeżeń Usuwa część całkowitą Zachowuje parametry PID Dopóki Reset = TRUE, dopóty PID_3Step pozostaje w nieaktywnym trybie pracy (State = 0). Zmiana zbocza z TRUE na FALSE: <ul style="list-style-type: none"> PID_3Step przełącza na tryb pracy, który jest zapisany w parametrze Mode.
ModeActivate	IN	Bool	PID_3Step przełącza na tryb pracy, który jest zapisany w parametrze Mode. Zmiana zbocza z FALSE na TRUE:
Mode	IN	Int	Pożądaný tryb PID; Aktywowany przednim zboczem na wejściu ModeActivate.
ScaledInput	OUT	Real	Skalowana wartość procesowa.
ScaledFeedback	OUT	Real	Zgłaszanie skalowanej wartości położenia Uwaga: Dla urządzenia wykonawczego bez sprzężenia zwrotnego położenia, pozycja urządzenia wykonawczego wskazywana przez ScaledFeedback jest bardzo nieprecyzyjna. W tym przypadku, ScaledFeedback może być użyte wyłącznie do zgrubnego oszacowania aktualnej pozycji.
Output_UP	OUT	Bool	Cyfrowa wartość wyjściowa otwarcia zaworu. (Wartość domyślna: FALSE). Jeśli Config.OutputPerOn = FALSE, to jest używany parameter Output_UP.
Output_DN	OUT	Bool	Cyfrowa wartość wyjściowa zamknięcia zaworu. (Wartość domyślna: FALSE). Jeśli Config.OutputPerOn = FALSE, to jest używany parameter Output_DN.
Output_PER	OUT	Word	Analogowa wartość wyjściowa. Jeśli Config.OutputPerOn = TRUE, to jest używany parameter Output_PER.
SetpointLimitH	OUT	Bool	Górny limit wartości zadanej. (Wartość domyślna: FALSE.) Jeśli SetpointLimit_H = TRUE, to osiągnięto wartość bezwzględną górnego limitu wartości zadanej (Setpoint ≥ Config.SetpointUpperLimit) Uwaga: Wartość zadana jest ograniczona (Setpoint ≥ Config.SetpointUpperLimit).
SetpointLimitL	OUT	Bool	Dolny limit wartości zadanej. (Wartość domyślna: FALSE) Jeśli SetpointLimitL = TRUE, to osiągnięto wartość bezwzględną dolnego limitu wartości zadanej (Setpoint ≤ Config.SetpointLowerLimit). Uwaga: Wartość zadana jest ograniczona (Setpoint ≤ Config.SetpointLowerLimit).
InputWarningH	OUT	Bool	Jeśli InputWarningH = TRUE, to wartość wejściowa osiągnęła lub przekroczyła górny limit wartości ostrzeżenia. (Wartość domyślna: FALSE)

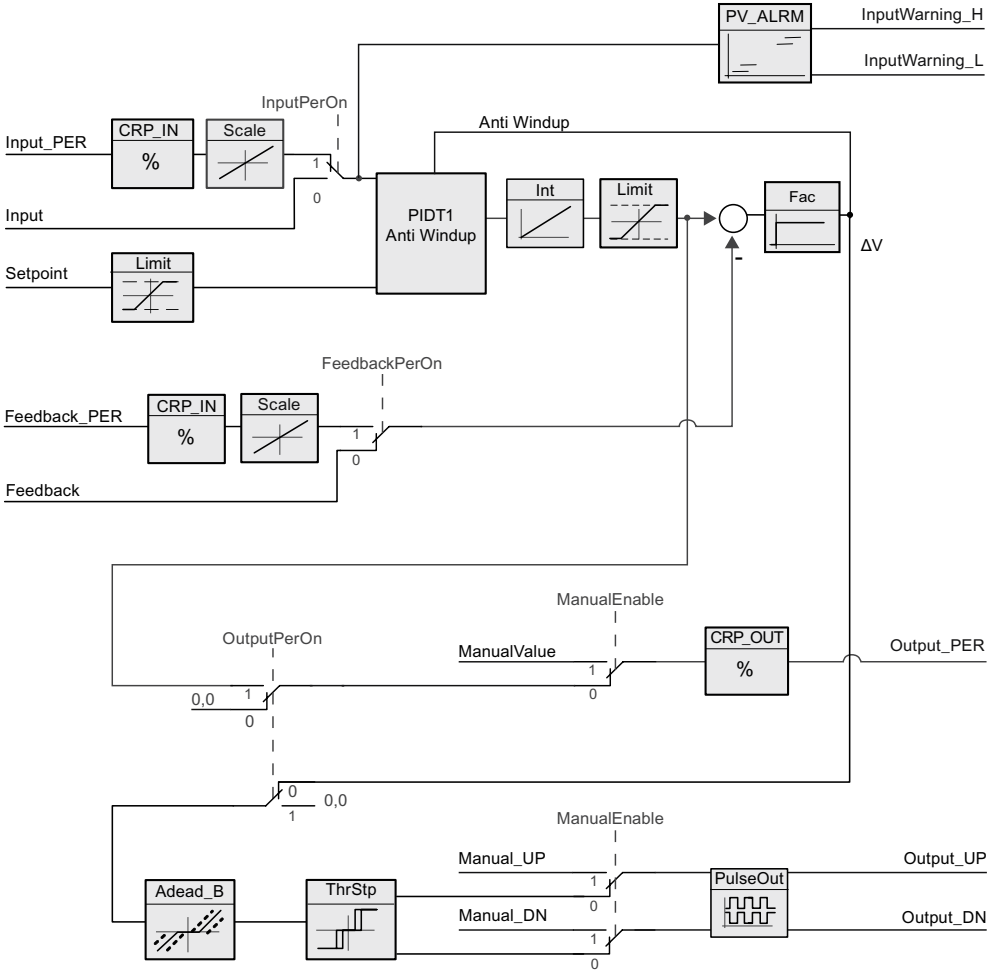
Parametr i jego typ	Typ danych	Opis	
InputWarningL	OUT	Bool	Jeśli InputWarningL = TRUE, to wartość wejściowa osiągnęła lub przekroczyła dolny limit wartości ostrzeżenia. (Wartość domyślna: FALSE)
State	OUT	Int	Aktualny tryb pracy regulatora PID. (Wartość domyślna: 0) Można zmienić tryb pracy za pomocą parametru wejściowego Mode i narastającego zbocza na wejściu ModeActivate: <ul style="list-style-type: none"> • State = 0: Nieaktywny • State = 1: Wstępne strojenie • State = 2: Ręczne dokładne strojenie • State = 3: Tryb automatyczny • State = 4: Tryb ręczny • State = 5: Zamienna wartość wyjściowa • State = 6: Pomiar czasu zmiany stanów • State = 7: Monitorowanie błędów • State = 8: Zamienna wartość wyjściowa z monitorowaniem błędów \ • State = 10: Tryb ręczny bez sygnałów położenia końcowego
Error	OUT	Bool	Jeśli Error = TRUE, to występuje przynajmniej jeden komunikat o błędzie. (Wartość domyślna: FALSE) Uwaga: Parametr Error w PID w wersji V1.x był polem ErrorBits, które zawierało kody błędów. Teraz znacznik boolowski wskazuje, że wystąpił błąd.
ErrorBits	OUT	DWord	Tabela parametrów ErrorBits instrukcji PID_3Step (strona 179) definiuje komunikaty o błędach, które występują. (Wartość domyślna: DW#16#0000 (brak błędu)). Parametr ErrorBits jest trwale zapamiętany i jest resetowany zboczem narastającym na wejściach Reset lub ErrorAck. Uwaga: W wersji V1.x, parametr ErrorBits został zdefiniowany jako parametr Error i nie istnieje.



Rys. 8.3. Praca regulatora PID_3Step jako regulator PIDT1 z algorytmem anti-windup



Rys. 8.4. Praca regulatora PID_3Step bez sprzężenia zwrotnego od pozycji



Rys. 8.5. Praca regulatora PID_3Step z sprzężeniem zwrotnym od pozycji

8.5. Parametry ErrorBits instrukcji PID_3STEP

Jeśli kilka błędów jest aktywnych, wartości kodu błędu są wyświetlane jako średnie z dodawania binarnego. Przykładowo: wyświetlenie kodu błędu 0003 znaczy, że w toku przygotowywania są błędy 0001 i 0002.

Tabela 8.8. Parametry ErrorBit instrukcji PID_3STEP

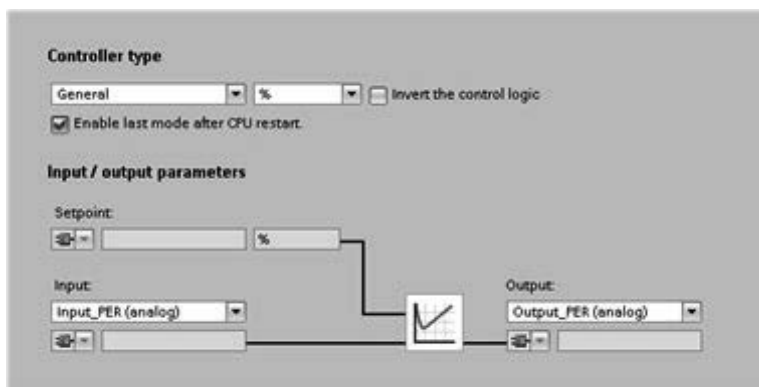
Wartość ErrorBit (DW#16#...)	Opis
0000	Brak błędu.
0001 1, 2	Wartość parametru „Input” jest poza granicami wartości procesowych: Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 2, 3	Nieprawidłowa wartość parametru „Input_PER”. Należy sprawdzić, czy na wejściu analogowym występuje błąd.
0004 4	Błąd podczas dokładnego strojenia. Oscylacje wartości procesowej nie mogą być utrzymane.
0010 4	Wartość zadana została zmieniona podczas strojenia. Uwaga: Można ustawić dopuszczalne odchylenie wartości zadanej tagu CancelTuningLevel.
0020	Wstępne strojenie nie jest dozwolone podczas dokładnego strojenia. Uwaga: Jeśli ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_3Step pozostaje w trybie dokładnego strojenia.
0080 4	Błąd podczas wstępnego strojenia. Niewłaściwa konfiguracja limitów wartości wyjściowej. Należy sprawdzić, czy limity wartości wyjściowej są poprawnie skonfigurowane i odpowiadają logice sterowania.
0100 4	Błąd podczas dokładnego strojenia spowodował niepoprawne parametry.
0200 2, 3	Niepoprawna wartość parametru Input: wartość ma niepoprawny format liczbowy.
0400 2, 3	Obliczenie wartości wyjściowej nie powiodło się. Należy sprawdzić parametry PID.
0800 1, 2	Błąd czasu próbkowania: Instrukcja PID_3Step nie jest wywoływana w czasie próbkowania OB cyklicznego przzerwania.
1000 2, 3	Niepoprawna wartość parametru Setpoint: wartość ma niepoprawny format liczbowy.
2000 1, 2, 5	Niepoprawna wartość parametru Feedback_PER. Należy sprawdzić, czy na wejściu analogowym występuje błąd.
4000 1, 2, 5	Niepoprawna wartość parametru Feedback: wartość ma niepoprawny format liczbowy.
8000 1, 2	Błąd w cyfrowym układzie sprzężenia zwrotnego położenia (zgłaszania położenia). Actuator_H = TRUE i Actuator_L = TRUE. Urządzenie wykonawcze nie może być zmieniane dla zamiennej wartości wyjściowej i pozostaje w aktualnej pozycji. W tym stanie tryb ręczny nie jest możliwy. Aby zmienić ten stan urządzenia wykonawczego trzeba wyłączyć opcję „Actuator end stop” (Config.ActuatorEndStopOn = FALSE) lub przełączyć na tryb ręczny bez sygnałów pozycji końcowej (Mode = 10).

Wartość ErrorBit (DW#16#...)	Opis
10000	<p>Niepoprawna wartość parametru ManualValue: wartość ma niepoprawny format liczbowy.</p> <p>Urządzenie wykonawcze nie może być zmieniane dla wartości wprowadzanej ręcznie i pozostaje w bieżącej pozycji.</p> <p>Należy przypisać poprawną wartość parametrowi ManualValue lub wprowadzić urządzenie wykonawcze w tryb ręczny za pomocą parametrów Manual_UP i Manual_DN.</p>
20000	<p>Niepoprawna wartość tagu SavePosition: wartość ma niepoprawny format liczbowy.</p> <p>Urządzenie wykonawcze nie może być zmieniane dla zamiennej wartości wyjściowej i pozostaje w aktualnej pozycji.</p>
40000	<p>Niepoprawna wartość parametru Disturbance: wartość ma niepoprawny format liczbowy.</p> <p>Uwaga: Jeśli tryb automatyczny był aktywny i ActivateRecoverMode = FALSE przed wystąpieniem błędu, to parametr Disturbance jest ustawiony na zero. PID_3Step pozostaje w trybie automatycznym.</p> <p>Uwaga: Jeśli tryb wstępnego strojenia lub dokładnego strojenia był aktywny i ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_3Step przełącza na tryb pracy, który jest zapisany w parametrze Mode. Jeśli Disturbance w obecnej fazie nie ma żadnego wpływu na wartość wyjściową, to strojenie nie jest anulowane.</p> <p>Błąd nie ma wpływu na pomiar czasu zmiany stanów.</p>

- ¹ Uwaga: Jeśli tryb automatyczny był aktywny przed wystąpieniem błędu i ActivateRecoverMode = TRUE, to PID_3Step pozostaje w trybie automatycznym.
- ² Uwaga: Jeśli tryb wstępnego strojenia, lub dokładnego strojenia, lub pomiaru czasu zmiany stanów był aktywny i ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_3Step przełącza na tryb pracy, który jest zapisany w parametrze Mode.
- ³ Uwaga: Jeśli tryb automatyczny był aktywny przed wystąpieniem błędu i ActivateRecoverMode = TRUE, to PID_3Step przełącza na tryb „Zastępcza wartość wyjściowa z monitorowaniem błędów” lub „Monitorowanie błędów”. Gdy błąd już nie występuje, to PID_3Step przełącza na tryb automatyczny.
- ⁴ Uwaga: Jeśli ActivateRecoverMode = TRUE przed wystąpieniem błędu, to PID_3Step anuluje strojenie i przełącza na tryb pracy, który jest zapisany w parametrze Mode.
- ⁵ Urządzenie wykonawcze nie może być zmieniane dla zamiennej wartości urządzenia wykonawczego tylko za pomocą Manual_UP i Manual_DN, a nie ManualValue.

8.6. Konfigurowanie regulatora PID

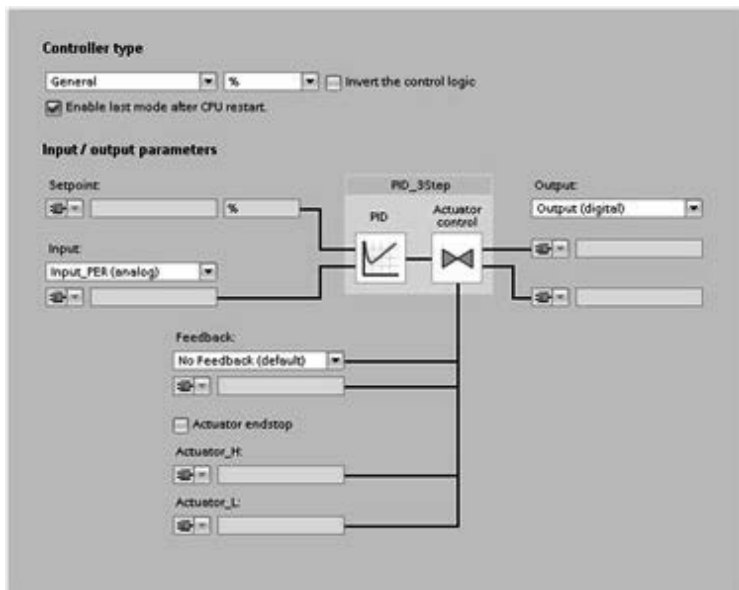
Parametry obiektu technologicznego określają pracę regulatora PID. Użyj ikony do otwarcia edytora konfiguracji.



Rys. 8.6. Edytor konfiguracji PID_Compact (Ustawienia podstawowe)

Tabela 8.9. Przykładowa konfiguracja ustawień instrukcji PID_Compact

Ustawienia		Opis
Podstawowe	Typ regulatora	Wybiera jednostka inżynierska
	Odwroćenie logiki sterującej	Pozwala na wybranie działania odwrotnego pętli PID. <ul style="list-style-type: none"> • Jeśli nie jest zaznaczone, PID działa w trybie działania bezpośredniego, a wyjścia pętli PID są zwiększane jeśli wartość wejściowa jest mniejsza od zadanej. • Jeśli jest zaznaczone, wyjście pętli PID jest zwiększane jeśli wartość wejściowa jest większa od zadanej.
	Załączenie ostatniego trybu po ponownym uruchomieniu CPU	Uruchamia ponownie pętlę PID jeśli nastąpił reset CPU lub zakres wejściowy został przekroczony i wartość powróciła do ustalonych granic.
	Input (wejście)	Wybór parametru Input lub Input_PER (analogowe) dla wartości procesu. Input_PER może pochodzić bezpośrednio z modułu wejść analogowych.
	Output (wyjście)	Wybór parametru Output lub Output_PER (analogowe) dla wartości procesu. Output_PER może kierować wartości bezpośrednio do modułu wyjść analogowych.
Wartości procesu	Skaluje zarówno zakres jak i limity wartości procesu. Jeśli wartość procesu wyniesie powyżej górnego limitu lub poniżej dolnego pętla PID przejdzie do trybu nieaktywnego, a wartość wyjściowa z regulatora zostanie ustawiona na 0. Aby użyć Input_PER, użytkownik musi przeskalować analogowe wartości procesu (<i>input value</i>).	



Rys. 8.7. Edytor konfiguracji PID_3Step (ustawienia podstawowe)

Tabela 8.10. Przykładowa konfiguracja ustawień instrukcji PID_3Step

Ustawienia		Opis
Podstawowe	Typ regulatora	Wybiera jednostka inżynierska
	Odwroćenie logiki sterującej	Pozwala na wybranie działania odwrotnego pętli PID. <ul style="list-style-type: none"> • Jeśli nie jest zaznaczone, PID działa w trybie działania bezpośredniego, a wyjścia pętli PID są zwiększane jeśli wartość wejściowa jest mniejsza od zadanej. • Jeśli jest zaznaczone, wyjście pętli PID jest zwiększane jeśli wartość wejściowa jest większa od zadanej.
	Załączenie ostatniego trybu po ponownym uruchomieniu CPU	Uruchamia ponownie pętlę PID jeśli nastąpił reset CPU lub zakres wejściowy został przekroczony i wartość powróciła do ustalonych granic. Ustawia tryb: Określa tryb, na który ma przejść regulator PID po restarcie.
	Input (wejście)	Wybór parametru Input lub Input_PER (analogowe) dla wartości procesu. Input_PER może pochodzić bezpośrednio z modułu wejść analogowych.
	Output (wyjście)	Wybór cyfrowych wyjść (Output_UP i Output_DN) lub analogowych wyjść (Output_PER) dla wartości wyjściowych.
	Feedback (sprężenie zwrotne)	Wybór typu stanu urządzenia zwracanego do pętli PID <ul style="list-style-type: none"> • Brak sprężenia (domyślny) • Sprężenie zwrotne • Feedback_PER
Process value	Skaluje zarówno zakres jak i limity wartości procesu. Jeśli wartość procesu wyniesie powyżej górnego limitu lub poniżej dolnego pętla PID przejdzie do nieaktywnego trybu, a wartość wyjściowa z regulatora zostanie ustawiona na 0. Aby użyć Input_PER, użytkownik musi przeskalować analogowe wartości procesu (<i>input value</i>).	

Ustawienia		Opis
Actuator	Motor transition time	Ustawia czas od otwarcia do zamknięcia zaworu (należy odszukać tę wartość w danych technicznych zaworu lub w opisie bezpośrednio na zaworze).
	Minimum ON time	Ustawia najmniejsze przemieszczenie zaworu (należy odszukać tę wartość w danych technicznych zaworu lub w opisie bezpośrednio na zaworze).
	Minimum OFF time	Ustawia najmniejszą wartość czasu przestoju zaworu (należy odszukać tę wartość w danych technicznych zaworu lub w opisie bezpośrednio na zaworze).
	Reaction to error (Reakcja na błąd)	Definiuje zachowanie zaworu w momencie wystąpienia błędu lub zerowania pętli PID. Jeśli użytkownik zdecydował się na użycie zastępczej pozycji, należy ją wprowadzić w polu <i>Safety position</i> . Dla analogowego sprzężenia zwrotnego lub analogowego wyjścia należy wybrać wartość pomiędzy dolnym i górnym limitem wyjścia. Dla cyfrowych wyjść użytkownik może wybierać tylko z wartości 0% (off) lub 100% (on).
	Scale Position Feedback ¹	<ul style="list-style-type: none"> • <i>High stop</i> oraz <i>Lower limit stop</i> definiują maksymalne otwarcie zaworu (całkowite otwarcie) i największe przekrycie zaworu (całkowite zamknięcie). <i>High stop</i> musi być większy niż <i>Lower limit stop</i>. • <i>High limit process value</i> oraz <i>Low limit process value</i> definiują górną i dolną pozycję zaworu podczas dostrajania i w trybie automatycznym. • <i>FeedbackPER</i> (<i>Low</i> oraz <i>High</i>) definiują analogowe sprzężenie zwrotne od pozycji zaworu. <i>FeedbackPER High</i> musi być większe niż <i>FeedbackPER Low</i>.
Advanced (Zaawansowane)	Monitoring process value (Wartość procesowa monitorowania)	W tym oknie użytkownik może wprowadzić własne parametry strojenia PID. Aby to umożliwić, pole wyboru <i>Enable Manual Entry</i> (Aktywuj ręczne wprowadzanie) musi być zaznaczone.
	PID parameters (Parametry PID)	

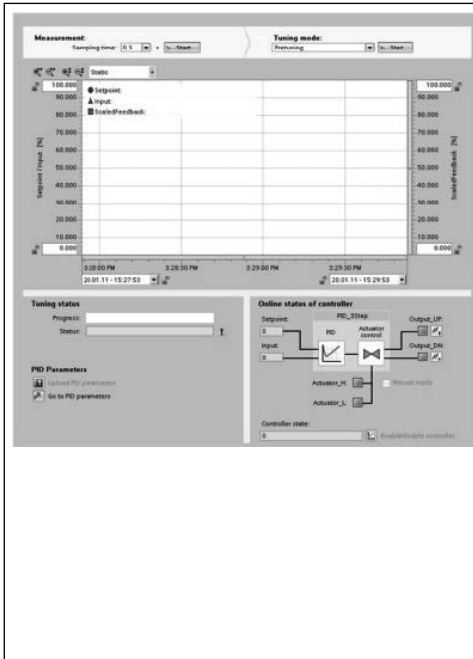
¹ *Scale Position Feedback* jest edytowalne tylko jeśli użytkownik uaktywnił w ustawieniach podstawowych opcję *Feedback*.

8.7. Uruchomienie regulatora PID

Należy użyć edytora uruchomienia w celu skonfigurowania regulatora PID do autostrojenia podczas rozruchu i autostrojenia podczas pracy. Aby otworzyć edytora uruchomienia należy kliknąć odpowiednią ikonę w instrukcji lub nawigatorze projektu.



Tabela 8.11. Przykładowy ekran uruchomienia (PID_3Step)



- Measurement: wyświetla wartość zadaną, wartość procesu (wartość wejściową) oraz trend wartości wyjściowej w czasie rzeczywistym. Należy wprowadzić czas próbkowania i nacisnąć przycisk *Start*.
- Tuning mode: Aby nastroić pętlę PID należy wybrać *Pre-tuning* lub *Fine tuning* (ręcznie) i nacisnąć przycisk *Start*. Regulator PID przechodzi przez różne fazy, aby obliczyć odpowiedź systemu i czasy odświeżania. Odpowiednie nastawy regulatora są obliczane w oparciu o te wartości.

Po zakończeniu procesu strojenia, użytkownik może zachować te parametry klikając przycisk *Upload PID parameters* w sekcji *PID Parameters* edytora uruchomienia.

Jeżeli wystąpi błąd podczas strojenia, to wartość wyjściowa z regulatora wyniesie 0. Regulator przejdzie w tryb „nieaktywny”. Ten stan wywoła błąd.

Kontrola wartości początkowych parametrów PID

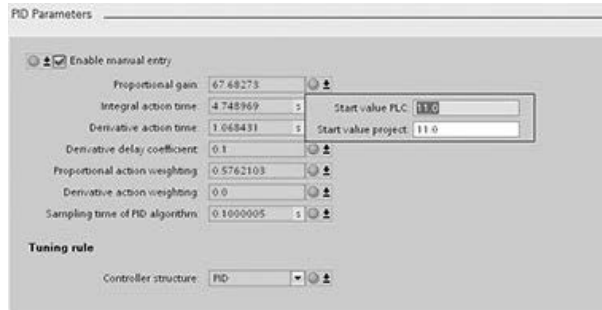
Można edytować rzeczywiste wartości parametrów konfiguracyjnych PID tak, aby działanie regulatora PID mogło być optymalizowane w trybie online.

Należy otworzyć „Technology objects” (Obiekty technologiczne) regulatora PID i jego obiekt „Configuration” (Konfiguracja). Aby uzyskać dostęp do kontroli wartości początkowych, należy kliknąć ikonę okularów w lewym górnym rogu okna dialogowego:



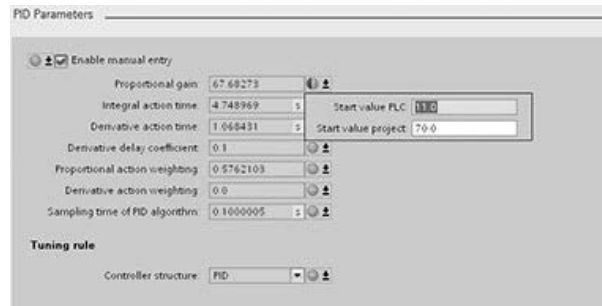
Teraz można zmieniać wartości dowolnych parametrów konfiguracyjnych regulatora PID, jak pokazano na rysunku poniżej.

Można porównać wartość rzeczywistą z wartością początkową projektu (offline) i wartością początkową PLC (online) każdego parametru. Jest to konieczne do porównania różnic online/offline bloku danych obiektu technologicznego (TO-DB) oraz uzyskania informacji o wartościach, które będą używane jako bieżące wartości przy następczej zmianie stanów Stop-Start PLC. Ponadto, ikona porównania daje wizualne wskazanie ułatwiające wykrycie różnic online i offline:



Na rysunku powyżej przedstawiono okno parametrów PID z ikonami porównania pokazującymi, które wartości się różnią pomiędzy projektami online i offline. Zielona ikona wskazuje, że wartości są takie same; ikona niebieska lub pomarańczowa oznacza, że wartości są różne.

Dodatkowo, kliknięcie przycisku parametru ze strzałką w dół powoduje otwarcie małego okna, które pokazuje wartość początkową projektu „Start value project” (offline) oraz wartość początkową PLC „Start value PLC” (online) każdego parametru:



Wbudowany serwer WWW ułatwiający komunikację z Internetem 9

Serwer sieci WEB pozwala na dostęp do danych o CPU oraz danych procesowych przechowywanych przez CPU. Zbiór standardowych stron Web jest zintegrowany z oprogramowaniem sprzętowym CPU (*firmware*). Za pomocą tych stron użytkownik uzyskuje dostęp do CPU poprzez przeglądarkę sieciową na swoim komputerze lub urządzeniu przenośnym. Zestaw stron Web pozwala upoważnionym użytkownikom na użycie wielu funkcji:

- Zamiana trybu pracy sterownika (STOP oraz RUN).
- Nadzorowanie i modyfikowanie stanu zmiennych PLC.
- Podgląd i ściągnięcie wszelkich dzienników zdarzeń gromadzonych przez CPU.
- Podgląd bufora diagnostycznego CPU.
- Aktualizowanie oprogramowania sprzętowego CPU.

Serwer sieci Web pozwala także na stworzenie definiowanych przez użytkownika stron Web, które mają dostęp do danych CPU. Użytkownik może projektować takie strony za pomocą wybranego przez siebie programu do HTML. Należy wstawić do kodu HTML wcześniej zdefiniowane komendy „AWP” (*Automation Web Programming* – programowanie automatyki w sieci Web), aby mieć dostęp do danych w CPU.

Użytkownik konfiguruje użytkowników i poziomy uprawnień dla web serwera w oknie *Device Configuration* (Konfiguracja urządzeń) dla CPU w programie STEP 7.

Wymagane przeglądarki internetowe

Web serwer obsługuje następujące przeglądarki internetowe na komputery PC:

- Internet Explorer w wersji 8.0 i wyższej
- Mozilla Firefox wersji 3.0 i wyższej
- Opera w wersji 11.0 i wyższej
- Google Chrome w wersji 21.0 i wyższej
- Apple Safari w wersji 6.0 i wyższej

Web serwer obsługuje następujące przeglądarki internetowe na urządzenia przenośne:

- Internet Explorer w wersji 8.0 i wyższej, dla paneli HMI
- Mobile Safari, iOS w wersji 5.0 i wyższej
- Mobile Android Browser, w wersji 2.3.4 i wyższej
- Mobile Google Chrome, Android w wersji 4.0 i wyższej

W części dotyczącej ograniczeń mających wpływ na korzystanie z web serwera (strona 189) podane są ograniczenia przeglądarek, które mogą zakłócać wyświetlanie standardowych, lub zdefiniowanych przez użytkownika, stron internetowych.

9.1. Łatwość użycia standardowych stron Web

Używanie standardowych stron Web jest łatwe! Użytkownik musi tylko podczas konfiguracji CPU włączyć serwer Web.



Strona startowa wyświetla reprezentację połączanego CPU oraz listę ogólnych informacji o CPU.

Po zalogowaniu się jako użytkownik z wymaganymi uprawnieniami, można również zmienić tryb pracy CPU (STOP i RUN) oraz uaktywnić miganie diod LED.



Strona *Variable Status* (Zmienne stanu) pozwala na obserwowanie lub zmianę każdego I/O lub danych pamięci w CPU. Konieczne jest zalogowanie się z uprawnieniami do odczytu danych w celu monitorowania wartości oraz z uprawnieniami do zapisu danych w celu modyfikowania wartości. Użytkownik może wprowadzić adres bezpośredni (np. I0.0), nazwę zmiennej lub zmienną z określonego bloku programu. Wartości danych są odświeżane automatycznie, chyba że użytkownik odznaczy tę opcję.

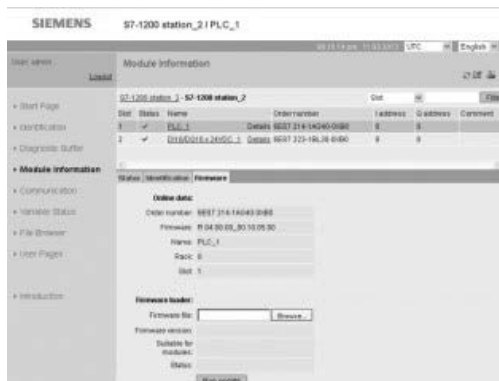


Strona *Diagnostic Buffer* (Bufor diagnostyczny) wyświetla bufor diagnostyczny, i jest dostępna dla użytkowników z uprawnieniami dostępu do danych diagnostycznych. Użytkownik może określić zakres wyświetlanych wpisów.

Lista diagnostycznych wpisów jest listą zdarzeń, które wystąpiły oraz czasu i daty ich wystąpienia. Należy wybrać pojedyncze zdarzenie, aby wyświetlić o nim szczegółowe informacje.



Strona *File Browser* (Przeglądarka plików) umożliwia przeglądanie, pobieranie i edycję plików w pamięci ładowania CPU, takich jak logi danych (strona 109) i receptury. Jeśli dla CPU nie jest używany 4 poziom zabezpieczenia, to wszyscy użytkownicy mogą przeglądać pliki ze strony przeglądarki plików. Użytkownicy z uprawnieniami do modyfikowania plików mogą pliki usuwać, edytować i zmieniać ich nazwy.



Strona *Module Information* (Informacje o modułach) oprócz wyświetlania informacji na temat modułów stacji umożliwia aktualizację oprogramowania sprzętowego (firmware) dla CPU lub innych modułów, które obsługują aktualizację oprogramowania sprzętowego. Użytkownicy z uprawnieniami dostępu do danych diagnostycznych mogą przeglądać dane modułu. Użytkownicy z uprawnieniami do wykonywania aktualizacji oprogramowania sprzętowego mogą aktualizować oprogramowanie sprzętowe.

Inne strony Web wyświetlają informację o CPU (takie jak numer seryjny, wersja i numer zamówienia), o parametrach komunikacyjnych (takie jak adresy sieciowe, własności fizyczne, interfejsy komunikacyjne i statystykę połączeń) oraz o modułach podłączonych na lokalnej liście.

 **Ostrzeżenie**

Nieautoryzowany dostęp do CPU przez web serwer

Nieautoryzowany dostęp do CPU lub zmiana wartości zmiennych PLC na nieprawidłowe może zakłócić działanie procesu, i może spowodować śmierć, poważne obrażenia ciała i/lub uszkodzenie mienia.

Ponieważ aktywacja web serwera umożliwia upoważnionym użytkownikom wykonywanie zmian w trybie pracy, wprowadzanie zapisów do danych PLC, oraz aktualizację firmware, firma Siemens zaleca przestrzegać następujących zasad bezpieczeństwa:

- Umożliwić dostęp do web serwera tylko za pośrednictwem protokołu HTTPS.
- Zabezpieczyć identyfikatory ID użytkowników web serwera za pomocą silnego hasła. Silne hasła, czyli trudne do odgadnięcia przez inne osoby, składają się z co najmniej dziesięciu znaków, w tym pomieszczone litery, cyfry i znaki specjalne nie tworzące słów, które można znaleźć w słowniku, ani nazw i identyfikatorów, które mogą być uzyskane z informacji osobistych. Hasła należy utrzymywać w tajemnicy i często je zmieniać.
- Nie rozszerzać domyślnych, minimalnych uprawnień użytkownika *Everybody* (Każdy).
- Wykonywać testy na występowanie błędów oraz sprawdzać poprawność zakresów wartości dla zmiennych w programie, ponieważ użytkownicy stron internetowych mogą zmienić wartości zmiennych PLC na niepoprawne.
- Używać bezpiecznej sieci Virtual Private Network (VPN), aby połączyć się z web serwerem obsługującym PLC S7-1200 z lokalizacji poza zabezpieczoną siecią.

9.2. Ograniczenia, które mogą mieć wpływ na korzystanie z serwera sieci Web

Poniższe aspekty IT mogą mieć wpływ na korzystanie z serwera sieci Web:

- Zwykle użytkownik musi użyć adresu IP danego CPU w celu uzyskania dostępu do standardowych stron sieci Web, stron Web definiowanych przez użytkownika, lub adresu IP routera bezprzewodowego z numerem portu. Jeśli przeglądarka sieciowa nie pozwala na bezpośredni dostęp do adresu IP, należy skontaktować się z administratorem sieci. Jeśli lokalne struktury obsługują DNS, użytkownik może połączyć się z danym adresem IP poprzez wpis DNS.
- Zapory, ustawienia proxy i inne określone dla strony ograniczenia mogą zablokować dostęp do CPU. Należy skontaktować się ze swoim administratorem sieciowym w celu rozwiązania tych problemów.
- Standardowe strony Web używają skryptów JavaScript oraz "ciasteczek" (cookies). Jeśli skrypty JavaScript lub pliki cookie są w ustawieniach przeglą-

darki sieci Web wyłączone, to należy je uaktywnić. Jeśli nie można ich uaktywnić, to niektóre funkcje będą ograniczone. Używanie skryptów JavaScript lub „ciasteczek” w zdefiniowanych przez użytkownika stronach Web nie jest konieczne. Jeśli jednak są one używane, to ich obsługa musi zostać włączona dla danej przeglądarki.

- Protokół Secure Sockets Layer (SSL) jest obsługiwany przez strony sieci Web. Użytkownik może uzyskać dostęp do standardowej lub zdefiniowanej przez użytkownika strony Web poprzez format adresowania URL typu `http://ww.xx.yy.zz` lub `https://ww.xx.yy.zz`, gdzie „ww.xx.yy.zz” reprezentuje adres IP dla CPU.
- Siemens zapewnia certyfikat bezpieczeństwa dla zabezpieczonego dostępu do serwera sieci Web. Użytkownik może ściągnąć i importować certyfikaty do opcji internetowych swojej przeglądarki. Jeśli użytkownik zdecyduje się nie importować certyfikatu, to przy każdym połączeniu się z serwerem Web za pomocą `https://` ukaże się powiadomienie o weryfikacji zabezpieczeń.

Lista połączeń

Serwer sieci Web obsługuje maksymalnie 30 aktywnych połączeń HTTP. Te 30 połączeń mogą być wykorzystywane na różne sposoby, w zależności od używanej przeglądarki internetowej, oraz liczby różnych obiektów na stronie (pliki .css, obrazy, dodatkowe pliki .html). Niektóre połączenia utrzymują się podczas wyświetlania strony, a inne, po rozpoczęciu łączności, stają się nieaktywne.

Jeśli, na przykład, użytkownik korzysta z przeglądarki Mozilla Firefox 8, która obsługuje maksymalnie sześć trwałych połączeń, to można użyć pięciu zakładek lub okien przeglądarki, zanim web serwer zacznie przerywać połączenia. W przypadku, kiedy strona nie wykorzystuje wszystkich sześć połączeń, można użyć dodatkowej przeglądarki lub zakładek przeglądarki.

Należy również pamiętać, że liczba aktywnych połączeń może mieć wpływ na wydajność strony.

Uwaga

Przed zamknięciem serwera sieci Web należy się wylogować

Jeśli użytkownik zalogował się do serwera sieci Web, to przed zamknięciem przeglądarki sieci Web powinien wylogować się. Web serwer obsługuje maksymalnie siedem jednoczesnych logowań.

9.2.1. Ograniczenia funkcji związane z wyłączeniem obsługi JavaScript w opcjach internetowych

Standardowe strony internetowe są realizowane za pomocą kodu HTML, skryptów JavaScripts, oraz plików cookie. Jeśli witryna użytkownika ogranicza użycie skryptów JavaScripts oraz plików cookie, wtedy należy uaktywnić je dla stron, aby poprawnie funkcjonowały. Jeśli nie można uaktywnić JavaScripts dla przeglądarki internetowej, wtedy funkcje kontrolowane przez skrypty JavaScripts nie będą wykonywane.

Ogólnie

Strony nie aktualizują się dynamicznie. Aby zobaczyć aktualne dane, stronę należy ręcznie odświeżać za pomocą ikony odświeżania.

Strona Diagnostic Buffer

Strona *Diagnostic Buffer* (Bufor diagnostyczny) wykorzystuje JavaScript następująco:

- Wyświetlanie szczegółów zdarzenia: Za pomocą JavaScript, wybierany jest wiersz w buforze diagnostycznym, aby wyświetlić szczegóły w dolnej części strony. Bez JavaScript, aby zobaczyć dane zdarzenia w dolnej części, należy kliknąć hiperłącze pola zdarzenia wpisu bufora diagnostycznego.
- Zmiana zakresu wyświetlanych wpisów bufora diagnostycznego: Za pomocą JavaScript użytkownik używa rozwijanej listy w górnej części strony do wybrania zakresu wyświetlanych wpisów bufora diagnostycznego, oraz automatycznie aktualizuje się strona. Bez JavaScript, aby wybrać zakres wyświetlanych wpisów bufora diagnostycznego, należy użyć listy rozwijanej w górnej części strony, i następnie kliknąć link *GO*, aby zaktualizować stronę bufora diagnostycznego z wybranym zakresem z rozwijanej listy.

Należy pamiętać, że hiperłącze *GO* i hiperłącza pól zdarzeń są widoczne tylko wtedy, gdy JavaScript jest uaktywniony. Nie są one konieczne, a zatem nie ma ich, kiedy JavaScript jest uaktywniony.

Strona Module Information

Bez włączonej obsługi JavaScript, występują następujące ograniczenia:

- Nie można filtrować danych.
- Nie można sortować pól.

Strona Variable Status

Bez włączonej obsługi JavaScript, występują następujące ograniczenia:

- Po wprowadzeniu każdej zmiennej, należy ręcznie uaktywnić wiersz *New variable*, aby wprowadzić nową zmienną.
- Wybranie formatu wyświetlania nie powoduje automatycznego wyświetlania wartości w wybranym formacie. Należy kliknąć przycisk *Monitor value*, aby zaktualizować wyświetlanie wartości w nowym formacie.

9.2.2. Funkcje ograniczone poprzez wyłączenie obsługi „ciasteczek”

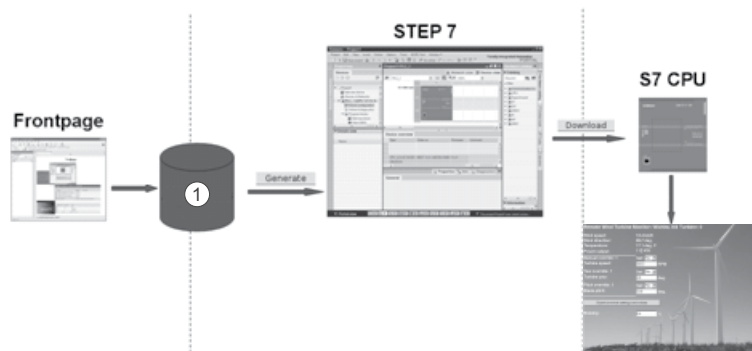
Jeśli obsługa plików cookie w przeglądarce sieci Web jest wyłączona, to występują następujące ograniczenia:

- Nie można zalogować się.
- Nie można zmienić ustawienia języka.
- Nie można zmienić czasu UTC na czas PLC. Bez plików cookie, wszystkie czasy są czasami UTC.

9.3. Uproszczone tworzenie stron definiowanych przez użytkownika

9.3.1. Łatwe tworzenie dowolnych, zdefiniowanych przez użytkownika stron sieci Web

Serwer WWW sterownika S7-1200 dostarcza środków do tworzenia własnej aplikacji określonych stron HTML, które rejestrują dane z PLC. Należy użyć wybranego edytora HTML do stworzenia tych stron. Następnie należy je załadować do CPU, gdzie będą dostępne z poziomu standardowych stron sieci Web.



① Pliki HTML z osadzonymi komendami AWP

Proces ten wymaga wykonania kilku zadań:

- Stworzenia stron HTML w edytorze HTML (np. MS Frontpage).
- Osadzenia komend AWP w komentarzach HTML w kodzie HTML: Komendy AWP są stałym zestawem komend wymaganych dla dostępu do informacji z CPU.
- Skonfigurowania STEP 7 do odczytania i przetworzenia stron HTML.
- Wygenerowania bloków programowych ze stron HTML.
- Kompilacji i ściągnięcia bloków programowych do CPU.
- Dostępu do stron Web zdefiniowanych przez użytkownika poprzez komputer PC lub urządzenie przenośne.

Użytkownik może użyć wybranego przez siebie pakietu oprogramowania do tworzenia własnych stron HTML, które będą używane z serwerem sieci Web. Należy się upewnić, że kod HTML jest kompilowany do standardów HTML określonych przez W3C (World Wide Web Consortium). STEP 7 nie przeprowadza żadnej weryfikacji składni HTML dla strony użytkownika. Użytkownik może użyć pakietu oprogramowania, które służy do projektowania w trybie WYSIWYG lub projektowania układu projektu, ale musi mieć możliwość edycji kodu w czystej postaci HTML. Większość narzędzi do tworzenia WEB obsługuje ten typ edycji, w przeciwnym razie, zawsze można skorzystać z prostego edytora tekstu do edycji kodu HTML. Kod HTML musi zawierać następującą linię, która ustawia kodowanie strony w UTF-8:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Należy zawsze zapisywać plik z edytora w formacie kodowania UTF-8.

STEP 7 kompiluje wszystko ze stron HTML do obsługiwanych przez siebie bloków danych. Te bloki danych składają się z jednego sterującego bloku danych, który kieruje wyświetlaniem stron Web i jednego lub większej ilości bloków danych, które zawierają skompilowane strony Web. Należy uważać na obszerne zbiory stron HTML, najczęściej to te zawierające duże obrazy, które wymagają dużej ilości miejsca w obszarze pamięci obsługującego je DB. Jeśli całkowita wielkość pamięci CPU nie jest wystarczająca dla zdefiniowanej przez użytkownika strony Web, należy użyć karty pamięci, która zapewnia dodatkową zewnętrzną pamięć.

Aby kod HTML mógł być używany przez S7-1200 należy zamieścić w nim komendy AWP jako komentarze HTML. Po stworzeniu strony należy zapisać ją na komputerze i zapamiętać ścieżkę do folderu, w którym została ona zapisana.

Odświeżanie strony Web zdefiniowanej przez użytkownika

Strony stworzone przez użytkownika nie są automatycznie odświeżane. To użytkownik decyduje o tym, kiedy strona ma być odświeżona. Strony, które wyświetlają dane PLC, odświeżane są okresowo, co pozwala na odczyt bieżących danych. Dla stron HTML, które służą do wprowadzania danych, odświeżanie mogłoby zaburzyć to wprowadzanie. Jeśli użytkownik zdecyduje się na automatyczne odświeżanie całej strony, może dodać do nagłówka HTML poniższą linię kodu, gdzie „10” oznacza liczbę sekund pomiędzy odświeżeniami:

```
<meta http-equiv="Refresh" content="10">
```

Użytkownik może także użyć języka JavaScript lub innych technologii HTML do sterowania stroną lub odświeżaniem danych. W tym wypadku należy zapoznać się z dokumentacją HTML i JavaScript.

9.3.2. Ograniczenia związane ze stronami definiowanymi przez użytkownika

Ograniczenia dla standardowych stron Web są nałożone także na te zdefiniowane przez użytkownika. Dodatkowo strony Web zdefiniowane przez użytkownika zmuszają do rozważenia innych kwestii.

Wielkość obszaru pamięci

Strony zdefiniowane przez użytkownika stają się blokami danych po kliknięciu przycisku „Generate blocks”. Bloki te zajmują pewną ilość obszaru pamięci. Jeśli sterownik wyposażony jest w kartę pamięci, użytkownik ma do dyspozycji całą pojemność karty pamięci, która jest używana jako zewnętrzna pamięć do załadowania zdefiniowanych przez użytkownika stron Web.

Jeśli karta pamięci nie jest dostępna, bloki te zajmują wewnętrzny obszar pamięci, którego rozmiar jest ograniczony zgodnie z rodzajem CPU.

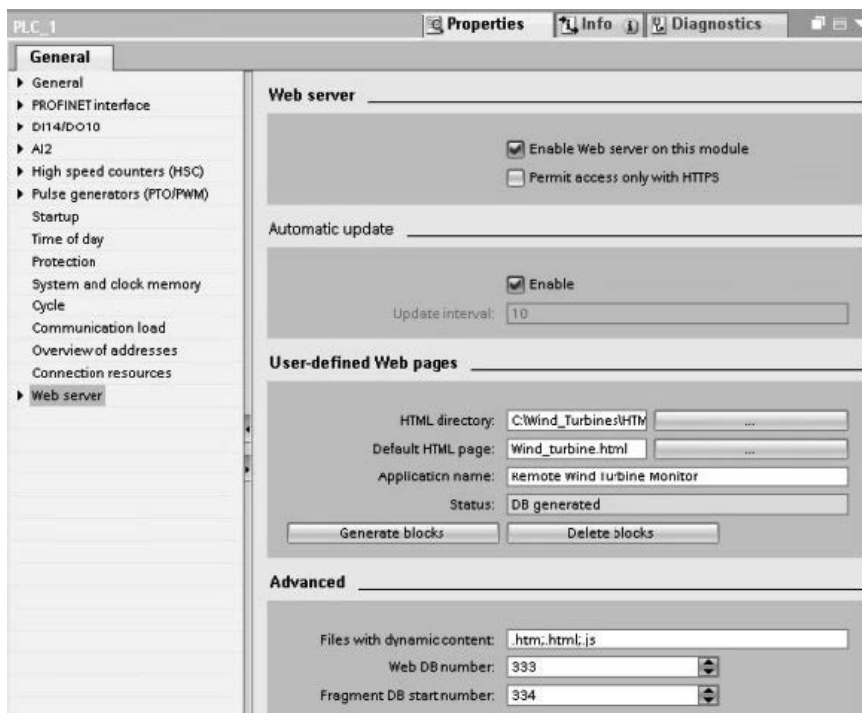
Użytkownik może sprawdzić rozmiar obszaru pamięci i dostępną, wolną pamięć za pomocą narzędzia Online and Diagnostic. Można także zobaczyć właściwości każdego pojedynczego bloku, który został przez STEP 7 wygenerowany ze zdefiniowanej przez użytkownika strony Web i określić jego rozmiar.

Uwaga

W celu zmniejszenia zapotrzebowania na pamięć dla zdefiniowanych przez użytkownika stron Web, należy zmniejszyć rozmiar obrazów użytych na tych stronach.

9.3.3. Konfiguracja zdefiniowanych przez użytkownika stron Web

Aby skonfigurować zdefiniowane przez użytkownika strony Web należy edytować w CPU właściwości serwera Web.



Po włączeniu funkcjonalności serwera Web należy wprowadzić następujące informacje:

- Lokacja domyślnej strony startowej w celu utworzenia bloku DB dla stron Web zdefiniowanych przez użytkownika.
- Nazwę aplikacji (opcjonalnie). Nazwa aplikacji jest używana dla późniejszej organizacji lub grupowania stron Web. Jeśli stworzono nazwę aplikacji, adres URL powinien pojawić się w poniższej wersji:
http://ww.xx.yy.zz/awp/<nazwa aplikacji>/<nazwa strony>.html.
- Rozszerzenia nazwy pliku w celu sprawdzenia obecności komend AWP. Domyślnie STEP 7 analizuje pliki z rozszerzeniami .htm, .html lub .js. Jeśli używane są dodatkowe rozszerzenia, należy je wymienić.


- Numery identyfikacyjne dla sterującego DB i inicjująca część DB.

Po skonfigurowaniu serwera sieci Web należy nacisnąć przycisk *Create blocks* w celu wygenerowania bloków DB ze stron HTML. Po tej operacji strony Web stają się częścią programu użytkownika. Sterujące bloki danych dla działań stron Web i „częściowe” DB zawierają wszystkie strony HTML.

9.3.4. Używanie instrukcji WWW

Instrukcja WWW umożliwia, aby strony definiowane przez użytkownika były dostępne ze standardowych stron Web. Aby umożliwić dostęp do stron Web zdefiniowanych przez użytkownika, program użytkownika musi wykonać instrukcję WWW tylko raz. Można jednak sprawić, aby strony Web zdefiniowane przez użytkownika były dostępne tylko w określonych okolicznościach. Wtedy program użytkownika może sterować wywołaniem instrukcji WWW w zależności od wymagań aplikacji.

Tabela 9.1. Instrukcja WWW

LAD/FBD	SCL	Opis
	<pre>ret_val := #WWW(ctrl_db:=_uint_in_);</pre>	<p>Rozpoznaje sterujący DB, który jest używany dla stron Web zdefiniowanych przez użytkownika. Sterujący blok danych jest parametrem wejściowym instrukcji WWW i określa zawartość stron jako reprezentacja we fragmencie bloku danych. Podobna procedura dotyczy statusu i informacji sterujących.</p>

Program użytkownika zwykle używa sterującego DB tak jakby został stworzony przez operację *Create blocks* – nie potrzeba dodatkowych ustawień. Program użytkownika może jednak ustawić globalne komendy w sterującym DB, które będą deaktywować serwer sieci Web, lub też później ponownie go załączać. Dla stron zdefiniowanych przez użytkownika, które zostały stworzone jako część DB, program użytkownika kontroluje ich zachowanie poprzez tabele żądań w sterującym DB.

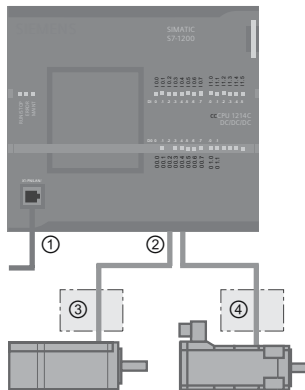
CPU posiada funkcje sterowania napędami, zarówno silników krokowych jak i serwonapędów z interfejsem impulsowym. Funkcje sterowania napędami sterują i nadzorują pracę napędów.

- Obiekt mechaniczny systemu pozycjonowania osi „Axis” (The „Axis” technology object) konfiguruje dane mechaniczne napędu, interfejs napędu, dane dynamiczne oraz inne właściwości napędu.
- Użytkownik konfiguruje wyjścia impulsowe i wyjścia sterujące kierunkiem sterowanego napędu.
- Program użytkownika używa instrukcji sterowania ruchem do sterowania osiami i rozpoczęcia wykonania zadań dla napędu.

Uwaga

Zmiany wprowadzone w konfiguracji sterowania napędami oraz załadowanie danych w trybie RUN nie odnoszą skutku, dopóki nie nastąpi przejście CPU z trybu STOP do trybu RUN.

Należy użyć interfejsu PROFINET do ustawienia połączenia online pomiędzy CPU i urządzeniem programującym. Dla sterowania napędem dostępne są dodatkowe funkcje online CPU oraz dodatkowe funkcje uruchomienia i diagnostyczne.



- ① PROFINET
- ② Wyjścia impulsowe i sterujące kierunkiem
- ③ Sekcja zasilania silnika krokowego
- ④ Sekcja zasilania serwonapędu

Opcje DC/DC/DC dla CPU S7-1200 mają zintegrowane wyjścia do sterowania kierunkiem ruchu napędów. Wariant przekaźnikowy CPU wymaga płytki sygnałowej z wyjściami DC dla sterowania napędem.

Płytkę sygnałową rozszerza liczbę zintegrowanych I/O o kilka dodatkowych punktów I/O. SB z dwoma wejściami cyfrowymi może zostać użyta do sterowania impulsowego i sterowania kierunkiem silnika. SB z czterema wyjściami cyfrowymi może obsłużyć sterowanie impulsowe i sterowanie kierunkiem dla dwóch silników. Wbudowane wyjścia przekaźnikowe nie mogą zostać użyte w roli wyjść impulsowych w sterowaniu silnikami. Niezależnie od tego, czy używa się I/O wbudowanych, I/O płytki sygnałowej (SB), lub kombinacji obu rodzajów I/O, można zdefiniować maksymalnie do czterech generatorów impulsów.

Cztery generatory impulsowe mają domyślne przypisania I/O; jednakże, mogą być skonfigurowane dla każdego wyjścia cyfrowego CPU lub SB. Generatory impulsów na CPU nie mogą być przypisane do modułów rozszerzeń (SM) lub rozproszonych I/O.

Uwaga

Wyjścia impulsowe nie mogą być używane przez inne instrukcje w programie użytkownika

Po skonfigurowaniu wyjść CPU lub płytki sygnałowej jako generatorów impulsów (do użycia z PWM lub podstawowymi instrukcjami sterowania ruchem), odpowiadające adresy wyjść nie sterują już dłużej wyjściami. Jeśli program użytkownika zapisuje wartość do wyjścia używanego jako generatora impulsów, CPU nie zapisuje tej wartości do wyjścia fizycznego.

Tabela 10.1. Największa możliwa liczba sterowanych napędów

Typ CPU		I/O wbudowane; Bez zainstalowanej płytki SB		Z płytką SB (2 x wyjścia DC)		Z płytką SB (4 x wyjścia DC)	
		Z kierunkiem	Bez kierunku	Z kierunkiem	Bez kierunku	Z kierunkiem	Bez kierunku
CPU 1211C	DC/DC/DC	2	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1212C	DC/DC/DC	3	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1214C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1217C	DC/DC/DC	4	4	4	4	4	4

Uwaga

Maksymalnie cztery generatory impulsów

Niezależnie od tego, czy używa I/O wbudowanych, I/O płytki sygnałowej (SB), lub kombinacji obu rodzajów, można zdefiniować maksymalnie do czterech generatorów impulsów.

Tabela 10.2. Częstotliwości maksymalne dla wyjść CPU

CPU	Wyjścia CPU	Częstotliwość maksymalna dla wyjść impulsowych i kierunkowych	Częstotliwość maksymalna dla wyjść kwadraturowych A/B, w górę/w dół, i impulsowych/kierunkowych
1211C	Qa.0 do Qa.3	100 kHz	100 kHz
1212C	Qa.0 do Qa.3	100 kHz	100 kHz
	Qa.4, Qa.5	20 kHz	20 kHz
1214C i 1215C	Qa.0 do Qa.3	100kHz	100 kHz
	Qa.4 do Qb.1	20 kHz	20 kHz
1217C	DQa.0 do DQa.3 (.0+, .0– do .3+, .3–)	1 MHz	1 MHz
	DQa.4 do DQb.1	100 kHz	100 kHz

Tabela 10.3. Częstotliwości maksymalne dla wyjść płytki sygnałowej (SB)

Płytki sygnałowa (SB)	Wyjścia płytki sygnałowej (SB)	Częstotliwość maksymalna dla wyjść impulsowych i kierunkowych	Częstotliwość maksymalna dla wyjść kwadraturowych A/B w górę / w dół, i impulsowych / kierunkowych
SB 1222, 200 kHz	DQe.0 do DQe.3	20 0kHz	200 kHz
SB 1223, 200 kHz	DQe.0, DQe.1	200 kHz	200 kHz
SB 1223	DQe.0, DQe.1	20 kHz	20 kHz

Tabela 10.4. Zakres częstotliwości dla wyjść impulsowych

Wyjście impulsowe	Częstotliwość
Zintegrowane z CPU	4 PTO: $2 \text{ Hz} \leq f \leq 1 \text{ MHz}$, 4 PTO: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$, lub dowolna kombinacja tych wartości dla 4 PTO.1 2
Standardowa SB	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
Szybka SB	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$

¹ W tabeli poniżej podano cztery możliwe kombinacje częstotliwości wyjściowych CPU 1217C.

² W tabeli poniżej podano cztery możliwe kombinacje częstotliwości wyjściowych 1211C CPU, CPU 1212C, 1214C CPU lub 1215C.

Przykładowe konfiguracje częstotliwości wyjściowych CPU 1217C

Uwaga

Z zastosowaniem wbudowanych wyjść różnicowych, CPU 1217C może generować sygnały impulsowe o częstotliwości do 1 MHz.

Poniższe przykłady pokazują cztery możliwe kombinacje częstotliwości wyjściowych:

- przykład 1: 4 PTO 1 MHz, bez wyjścia kierunkowego,
- przykład 2: 1 PTO 1 MHz, 2 PTO 100 kHz, oraz 1 PTO 20 kHz, wszystkie z wyjściem kierunkowym,
- przykład 3: 4 PTO 200 kHz, bez wyjścia kierunkowego,
- przykład 4: 2 PTO 100 kHz, oraz 2 PTO 200 kHz, wszystkie z wyjściem kierunkowym.

P = Impulsowe D = Kierunkowe		Wbudowane wyjścia CPU										Wyjścia szybkiej płytki SB				Wyjścia standardowej płytki SB outputs	
		Wyjścia (Q) 1 MHz				Wyjścia (Q) 100 kHz						Wyjścia (Q) 200 kHz				Wyjścia (Q) 20 kHz	
		0.0+	0.1+	0.2+	0.3+	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		0.0	0.1-	0.2-	0.3-												
Przykład 1: 4 PTO 1 MHz (bez wyjść kierunk.)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
Przykład 2: 1 PTO 1 MHz; 2 PTO 100 kHz i 1 PTO 20 kHz (wszystkie z wyjściami kierunk.)	PTO1	P	D														
	PTO2					P	D										
	PTO3							P	D								
	PTO4															P	D
Przykład 3: 4 PTO 200 kHz (bez wyjść kierunk.)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
Przykład 4: 2 PTO 100 kHz; 2 PTO 200 kHz (wszystkie z wyjściami kierunk.)	PTO1					P	D										
	PTO2							P	D								
	PTO3											P	D				
	PTO4													P	D		

Przykładowe konfiguracje częstotliwości wyjściowych CPU 1211C, CPU 1212C, CPU 1214C, oraz CPU 1215C

Poniższe przykłady pokazują cztery możliwe kombinacje częstotliwości wyjściowych:

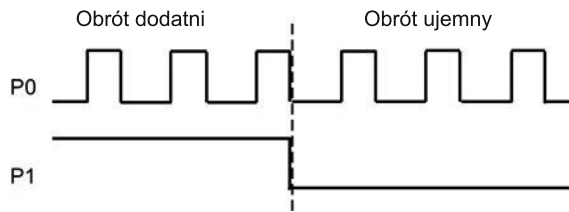
- przykład 1: 4 PTO 100 kHz, bez wyjścia kierunkowego,
- przykład 2: 2 PTO 100 kHz oraz 2 PTO 20 kHz, wszystkie z wyjściem kierunkowym,
- przykład 3: 4 PTO 200 kHz, bez wyjścia kierunkowego,
- przykład 4: 2 PTO 100 kHz oraz 2 PTO 200 kHz, wszystkie z wyjściem kierunkowym.

P = Impulsowe D = Kierunkowe		Wbudowane wyjścia CPU										Wyjścia szybkiej płytki SB				Wyjścia wolnej płytki SB	
		Wyjścia (Q) 100 kHz				Wyjścia (Q) 20 kHz						Wyjścia (Q) 200 kHz				Wyjścia (Q) 20 kHz	
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		CPU 1211C															
		CPU 1212C				CPU 1212C											
		CPU 1214C				CPU 1214C		CPU 1214C									
		CPU 1215C				CPU 1215C		CPU 1215C									
Przykład 1: 4 PTO 100 kHz (bez wyjść kierunk)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
Przykład 2: 2 PTO 100 kHz; 2 PTO 20 kHz (wszystkie z wyjściami kierunk)	PTO1	P	D														
	PTO2			P	D												
	PTO3					P	D										
	PTO4							P	D								
Przykład 3: 4 PTO 200 kHz (bez wyjść kierunk)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
Przykład 4: 2 PTO 100 kHz; 2 PTO 200 kHz (wszystkie z wyjściami kierunk)	PTO1	P	D														
	PTO2			P	D												
	PTO3											P	D				
	PTO4														P	D	

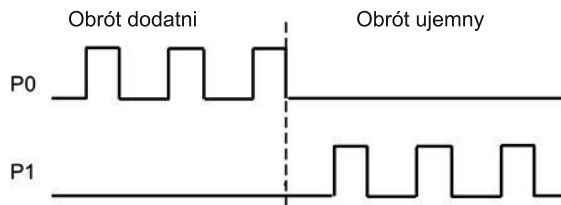
10.1. Sterowanie fazowe

Interfejs sterowania fazowego „Phasing” silnikami krokowymi lub serwonapędami zawiera cztery następujące opcje:

- PTO (*pulse A and direction B*): Jeśli zostanie wybrana opcja PTO (*pulse A and direction B*) (impulsowe A i kierunkowe B), to wtedy jedno wyjście (P0) steruje impulsami a drugie wyjście (P1) steruje kierunkiem. P1 jest w stanie wysokim (aktywnym), gdy impulsy są generowane w kierunku dodatnim. P1 jest w stanie niskim (nieaktywnym), gdy impulsy są generowane w kierunku ujemnym:

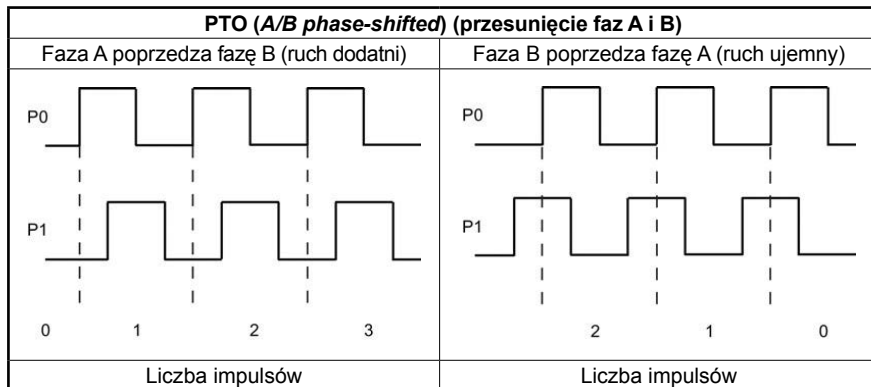


- PTO (*count up A and count down B*): Jeśli zostanie wybrana opcja PTO (*count up A and count down B*) (zliczanie w górę A i zliczanie w dół B), to wtedy jedno wyjście (P0) generuje impulsy dla kierunku dodatniego, a na drugie wyjście (P1) generuje impulsy dla kierunku dodatniego:

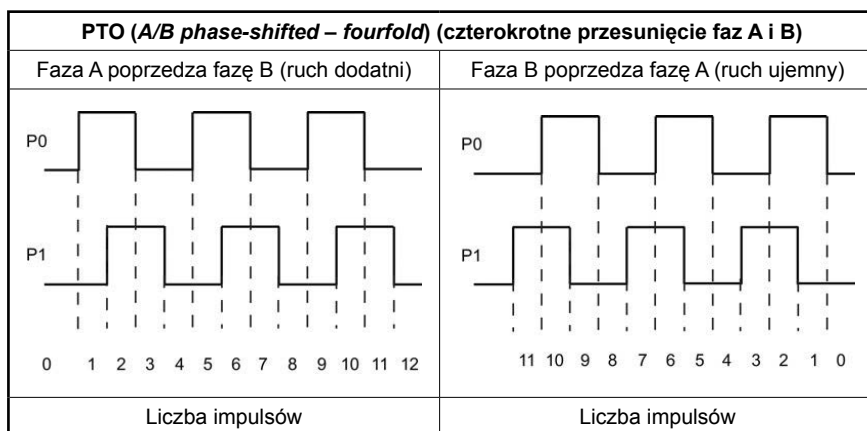


- PTO (*A/B phase-shifted*): Jeśli zostanie wybrana opcja PTO (*A/B phase-shifted*) (przesunięcie faz A i B), to wtedy oba wyjścia generują impulsy z określoną częstotliwością, ale z przesunięciem faz o 90 stopni. Jest to konfiguracja 1X, co oznacza, że jeden impuls występuje w czasie pomiędzy dodatnimi przejściami na wyjściu P0. W tym przypadku, kierunek określa się przez to, które wyjście jako pierwsze przechodzi w stan wysoki. Dla kierunku dodatniego stan wysoki na P0 poprzedza stan wysoki na P1. Dla kierunku ujemnego stan wysoki na P1 poprzedza stan wysoki na P0.

Liczba generowanych impulsów zależy od liczby przejść (zmian stanów) z 0 na 1 dla fazy A. Zależność fazowa wyznacza kierunek ruchu:



- PTO (A/B phase-shifted – fourfold):** Jeśli zostanie wybrana opcja PTO (A/B phase-shifted – fourfold) (czterokrotne przesunięcie faz A i B), to wtedy oba wyjścia generują impulsy z określoną częstotliwością, ale z przesunięciem faz o 90 stopni. Jest to konfiguracja 4X, co oznacza, że jeden impuls występuje w wyniku przejść (dodatnich i ujemnych) na każdym wyjściu. W tym przypadku, kierunek określa się przez to, które wyjście jako pierwsze przechodzi w stan wysoki. Dla kierunku dodatniego stan wysoki na P0 poprzedza stan wysoki na P1. Dla kierunku ujemnego stan wysoki na P1 poprzedza stan wysoki na P0. Konfiguracja czterokrotna bazuje na przejściach (zmianach stanów) dodatnich i ujemnych, zarówno dla fazy A, jak i fazy B. Zależność fazowa (faza A poprzedza fazę B lub faza B poprzedza fazę A) wyznacza kierunek ruchu.



- PTO (pulse and direction (direction de-selected)):** Jeśli wyjście kierunkowe w PTO zostanie dezaktywowane za pomocą opcji (pulse and direction (direction de-selected)) (impulsowe i kierunkowe (kierunkowe dezaktywowane)), to wtedy wyjście P0 steruje generowaniem impulsów. Wyjście P1 nie jest uży-

wane i jest dostępne dla innych celów programowych. W tym trybie CPU akceptuje tylko polecenia ruchu w kierunku dodatnim. Po wyborze tego trybu, system sterowania ruchem uniemożliwia dokonywanie niedozwolonych konfiguracji z kierunkiem ujemnym. Można zapisywać dane wyjściowe, jeśli aplikacja sterowania ruchem dotyczy tylko jednego kierunku. Na rysunku poniżej pokazano przypadek jednej fazy (jedno wyjście) (przy założeniu polaryzacji dodatniej):



10.2. Konfiguracja generatora impulsów

1. Dodaj obiekt mechaniczny:

- W drzewie projektu należy rozwinąć węzeł *Technology Objects* i wybrać *Add new object*.
- Należy wybrać ikonę *Axis* (można zmienić nazwę) i kliknąć *OK* aby otworzyć edytor konfiguracji dla obiektu osiowego.
- Należy wyświetlić właściwości *Select PTO for Axis Control* znajdujące się pod zakładką *Basic parameters* i wybrać pożądany impuls.

Uwaga

Jeśli PTO nie zostało wcześniej skonfigurowane w właściwościach CPU, to PTO jest skonfigurowane do użycia jednego z wbudowanych wyjść.

W przypadku korzystania z płytki sygnałowej wyjść, należy wybrać przycisk *Device Configuration* w celu przejścia do właściwości CPU. W *Parameter assignment*, w sekcji *Pulse options*, należy skonfigurować źródło wyjścia dla wyjścia płytki sygnałowej.

- Skonfiguruj pozostałe parametry podstawowe i rozszerzone.
2. Należy zaprogramować aplikację poprzez wstawienie instrukcji *MC_Power* do bloku kodu.
- Dla wejścia *Axis* należy wybrać oś, która została wcześniej skonfigurowana.
 - Ustawienia wejścia *ENABLE* na *TRUE* umożliwia działanie innych funkcji sterujących napędem.
 - Ustawienia wejścia *ENABLE* na *FALSE* powoduje odwołanie wszystkich instrukcji sterujących napędem.

Uwaga

Jedna oś może zawierać tylko jedną instrukcję MC_Power.

3. Wstaw inne instrukcje sterujące napędem, aby uzyskać określony ruch.

Uwaga

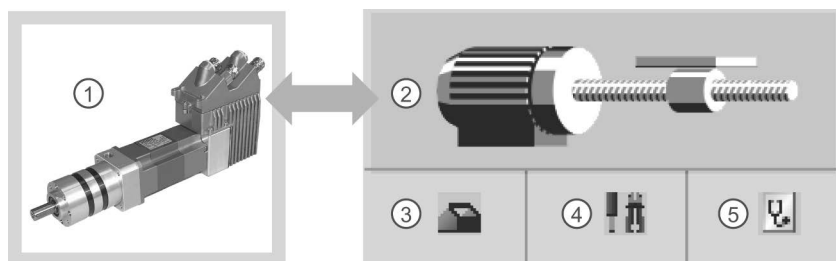
Konfiguracja generatora impulsów dla wyjść płytki sygnałowej: W konfiguracji urządzenia należy wybrać z właściwości CPU opcję *Pulse generators (PTO/PWM)* i włączyć generator impulsów. Dla każdej z następujących wersji CPU sterownika S7-1200: V1.0, V2.0, V2.1, oraz V2.2, są dostępne dwa generatory impulsowe. Dla CPU V3.0 sterownika S7-1200 są dostępne cztery generatory impulsów. W tym samym obszarze konfiguracji w sekcji *Pulse options*, wybrać ustawienie generatora impulsów *Pulse generator used as: „PTO”*.

Uwaga

CPU przelicza zadania sterowania ruchem w „plastrach” (*slices*) lub segmentach 10 ms. Podczas gdy jeden „plaster” jest wykonywany, następny czeka w kolejce do wykonania. Jeśli użytkownik przerwie wykonywane przez poszczególne osie zadania napędu (przez wykonanie nowego zadania dla tej osi), nowe zadanie sterowania napędem może nie zostać wykonane w czasie do 20 ms.

10.3. Konfiguracja osi napędu

Step 7 zapewnia dla obiektów technologii osiowych narzędzia konfiguracyjne, narzędzia uruchamiające i narzędzia diagnostyczne.



① Napęd

② Obiekt mechaniczny

③ Konfiguracja

④ Uruchomienie

⑤ Diagnostyka

Uwaga

Dla oprogramowania sprzętowego CPU w wersji V2.2 lub niższej, PTO wymaga wewnętrznego działania szybkich liczników. Oznacza to, że odpowiadający mu szybki licznik nie może być używany do jakiegokolwiek innego działania.

Przypisanie HSC do PTO jest stałe. W momencie, w którym PTO1 jest aktywowane, zostaje ono przypisane do szybkiego licznika HSC1. Jeśli PTO2 zostanie aktywowane, będzie ono przypisane do HSC2. Użytkownik nie może monitorować bieżącej wartości (np. w ID 1000) kiedy generowane są impulsy.

CPU S7-1200 w wersji V3.0 i wyższej nie mają tego ograniczenia. Jeśli wyjścia impulsowe są skonfigurowane w tych CPU, to wszystkie HSC pozostają dostępne do wykorzystania w programie.

Tabela 10.5. Narzędzia STEP 7 do sterowania ruchem

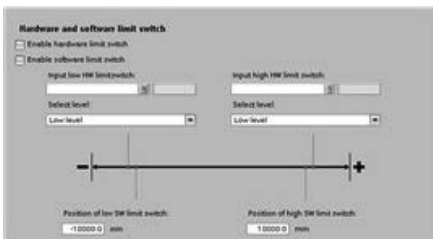
Narzędzie	Opis
<i>Configuration</i> – konfiguracja	Konfiguruje następujące właściwości obiektu mechaniczny: <ul style="list-style-type: none"> • Wybór używanego PTO i konfiguracja interfejsu napędu. • Właściwości mechaniki i przełożenia napędu (lub maszyny lub systemu). • Właściwości granicznego położenia, dynamiki i bazowania. Należy zapisać konfigurację w bloku danych obiektu technologicznego.
<i>Commissioning</i> – uruchamianie	Sprawdza działanie osi bez konieczności tworzenia programu użytkownika. Podczas rozpoczęcia pracy narzędzia wyświetlany jest panel sterujący. Oferuje on następujące instrukcje: <ul style="list-style-type: none"> • Załączanie i wyłączenie osi. • Ruch osi w trybie jog – ręczne przesuwanie. • Pozycja osi w układzie względnym i bezwzględnym. • Oś bazowa. • Potwierdzanie błędów. Prędkość i przyspieszenie/opóźnienie mogą być określone przez komendy napędu. Panel sterujący wyświetla także bieżący stan osi.
<i>Diagnostics</i> – diagnostyka	Monitoruje bieżący stan i informacje o błędzie dla osi i napędu.



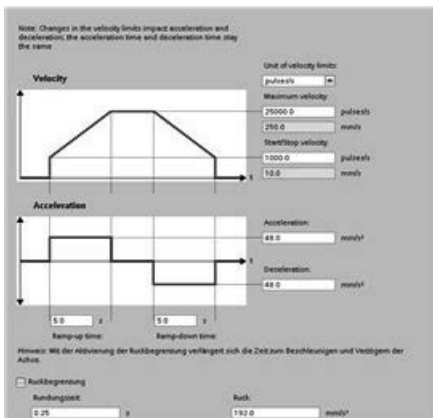
Po stworzeniu obiektu technologicznego, użytkownik konfiguruje osie poprzez definiowanie parametrów, takich jak PTO i definiowanie interfejsu napędu. Użytkownik musi skonfigurować także pozostałe parametry osi, takie jak graniczne położenie, dynamikę i bazowanie.

Uwaga

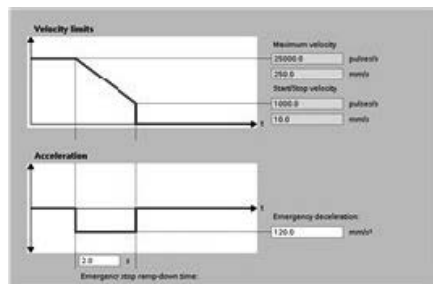
Użytkownik może przeprowadzić adaptację wartości parametrów wejściowych do nowych wymiarów systemu z poziomu programu.



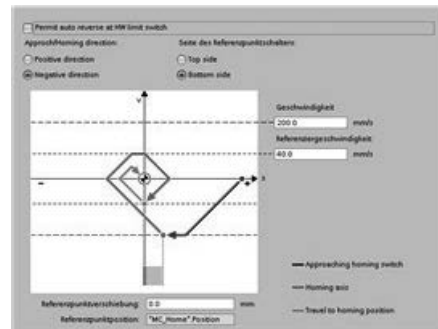
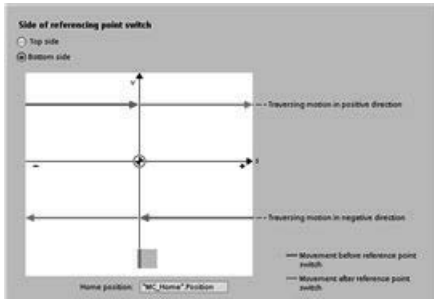
Konfiguracja właściwości sygnałów z napędu i nadzorowania pozycji (sprzętowe i programowe krańcówki).



Użytkownik może skonfigurować dynamikę napędu i zachowanie przy komendzie awaryjnego stopu.



Użytkownik może także zdefiniować sposób bazowania (pasywny lub aktywny).



Należy użyć panelu sterującego *Commissioning* (uruchamianie) aby sprawdzić funkcjonalność niezależnie od programu użytkownika.



Naciśnięcie ikony *Startup* powoduje uruchomienie osi.

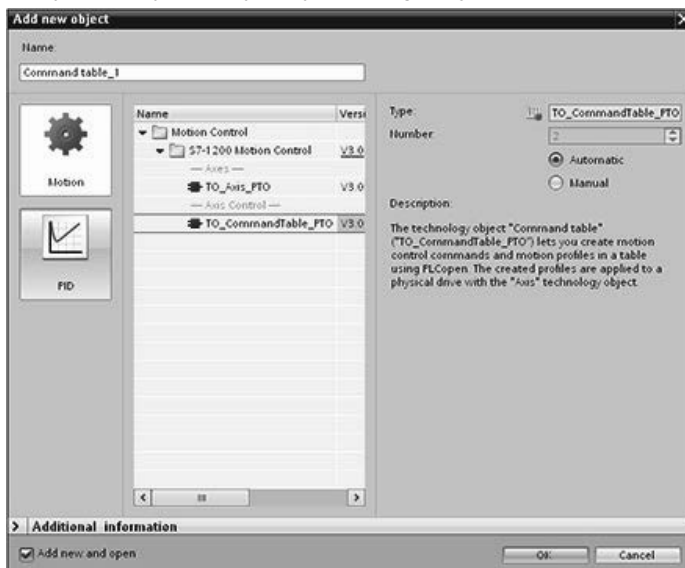
Panel sterowania wyświetla bieżący stan osi. Użytkownik poza załączeniem i wyłączeniem osi może także sprawdzać pozycjonowanie osi (zarówno wartość bezwzględna jak i względną) oraz określać jej prędkość, przyspieszenie lub opóźnienie. Panel sterowania pozwala także na potwierdzenie błędów.

10.4. Konfiguracja obiektu technologicznego TO_CommandTable_PTO

Instrukcję CommandTable można skonfigurować za pomocą obiektów technologicznych.

Dodanie obiektu technologicznego

1. W drzewie projektu, rozwinąć węzeł *Technological Objects* i wybrać *Add new object*.
2. Wybrać ikonę *CommandTable* (w razie potrzeby zmienić nazwę), a następnie kliknąć *OK*, aby otworzyć edytor konfiguracji dla obiektu CommandTable.



Kroki planowania aplikacji

Można utworzyć pożądaną sekwencję ruchów w oknie konfiguracji *Command Table*, i następnie sprawdzić rezultat w widoku graficznym na wykresie trendu.

Można wybrać typy poleceń, które mają być wykorzystywane do przetwarzania tabeli poleceń. Można wprowadzić do 32 kroków. Polecenia są wykonywane w kolejności, łatwo tworząc złożony profil ruchu.

Tabela 10.6. Typy poleceń instrukcji MC_CommandTable

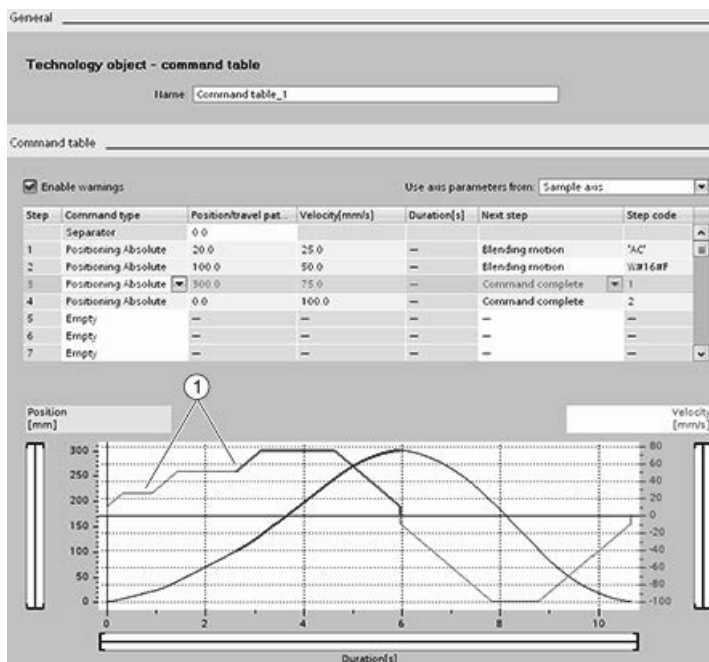
Typ polecenia	Opis
<i>Empty</i> (Puste)	<i>Empty</i> (Puste) służy jako symbol zastępujący wszelkie polecenia, które mają być dodane. Wpis empty jest ignorowany, gdy jest przetwarzana tabela poleceń
<i>Halt</i> (Zatrzymaj)	Powoduje zatrzymanie osi. Uwaga: Polecenie to jest wykonywane tylko po poleceniu <i>Velocity setpoint</i>
<i>Positioning Relative</i> (Pozycjonowanie względne)	Ustawia oś na podstawie odległości. Polecenie powoduje przesunięcie osi na określoną odległość z określoną prędkością
<i>Positioning Absolute</i> (Pozycjonowanie bezwzględne)	Ustawia oś na podstawie miejsca. Polecenie powoduje przesunięcie osi do określonego miejsca z określoną prędkością
<i>Velocity setpoint</i> (Wartość zadana prędkości)	Powoduje przesuwanie osi z określoną prędkością
<i>Wait</i> (Czekaj)	Powoduje wyczekiwanie do zakończenia określonego okresu. Polecenie <i>Wait</i> nie zatrzymuje aktywnego ruchu poprzecznego
<i>Separator</i> (Separator)	Dodaje wiersz oddzielający powyżej zaznaczonego wiersza. Wiersz oddzielający umożliwia zdefiniowanie więcej niż jednego profilu w jednej tabeli poleceń

Na rysunku poniżej, *Command complete* (Polecenie wykonane) jest używane jako przejście do następnego kroku. Ten typ przejścia pozwala urządzeniu zwalniać do prędkości uruchomienia/zatrzymania (start/stop), a następnie ponownie przyspieszać na początku kolejnego kroku (etapu).



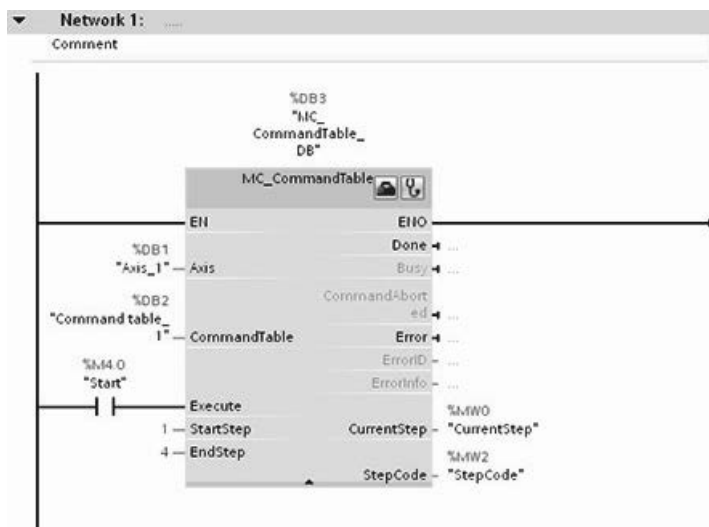
① Oś (axis) zwalnia do prędkości uruchomienia/zatrzymania pomiędzy krokami.

Na rysunku poniżej, *Blending motion* (Łączenie ruchu) jest stosowane jako przejście do następnego kroku. Ten typ przejścia umożliwia urządzeniu utrzymać jego prędkość do rozpoczęcia następnego kroku, w wyniku czego następuje płynne przejście urządzenia z jednego kroku do następnego. Stosowanie łączenia ruchu (*blending motion*) może skrócić całkowity czas potrzebny do całkowitego wykonania profilu. Bez łączenia ruchu, przykładowy przebieg procesu zajmuje siedem sekund. Z łączeniem ruchu, czas wykonania procesu jest skrócony o jedną sekundę, co w sumie daje sześć sekund.



① Oś (axis) kontynuuje ruch i przyspiesza lub zwalnia do prędkości następnego kroku, oszczędzając czas i mechaniczne zużycie.

Wykonywanie tabeli poleceń (CommandTable) jest sterowane przez instrukcję MC_CommandTable, jak pokazano poniżej:



10.5. Instrukcje sterowania ruchem *Motion Control*

10.5.1. Przegląd instrukcji sterowania ruchem

Instrukcje sterowania ruchem MC (*Motion Control*) używają przypisanego bloku danych technologicznych oraz dedykowanych PTO (wyjść ciągów impulsów) jednostki CPU do kontroli ruchu osi.

- MC_Power (strona 212) aktywuje lub dezaktywuje oś do sterowania ruchem.
- MC_Reset (strona 215) resetuje wszystkie błędy sterowania ruchem. Wszystkie błędy sterowania ruchem, które są wykrywane są potwierdzane.
- MC_Home (strona 216) ustanawia zależność pomiędzy programem sterowania osią a mechanicznym systemem pozycjonowania osi.
- MC_Halt (strona 218) anuluje wszystkie procesy ruchu i powoduje zatrzymanie ruchu osi. Pozycja zatrzymania nie jest określona.
- MC_MoveAbsolute (strona 220) rozpoczyna ruch do pozycji bezwzględnej. Zadanie kończy się po osiągnięciu pozycji docelowej.
- MC_MoveRelative (strona 222) rozpoczyna ruch pozycjonujący względem pozycji początkowej.
- MC_MoveVelocity (strona 224) powoduje ruch osi z określoną prędkością.
- MC_MoveJog (strona 226) wprowadza tryb ręczny (jog mode) do celów testowania i uruchamiania.
- MC_CommandTable (strona 228) wykonuje polecenia ruchu osi jako sekwencję ruchów.
- MC_ChangeDynamic (strona 231) zmienia ustawienia dynamiczne osi.
- MC_WriteParam (strona 233) zapisuje określoną liczbę parametrów do zmiany funkcjonalności osi z programu użytkownika.
- MC_ReadParam (strona 234) odczytuje określoną liczbę parametrów, które wskazują aktualną pozycję, prędkość i inne dane osi zdefiniowane dla wejścia Axis.

Poziomy oprogramowania sprzętowego CPU

W przypadku korzystania z CPU S7-1200 z oprogramowaniem sprzętowym (firmware) w wersji V4.0, należy wybrać instrukcję ruchu w wersji v4.0.

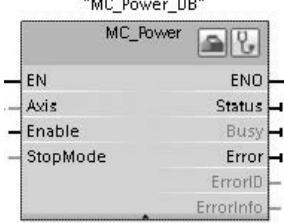
W przypadku korzystania z CPU S7-1200 z oprogramowaniem sprzętowym (firmware) w wersji V3.0 lub poprzedniej, należy wybrać instrukcję ruchu odpowiednio w wersji V3.0, V2.0 lub V1.0.

10.5.2. Instrukcja MC_Power (zwolnienie/zablokowanie osi)

Uwaga

Jeśli w wyniku błędu oś zostaje wyłączona, zostanie włączona automatycznie ponownie po wyeliminowaniu błędu i zatwierdzeniu. Wymaga to, żeby w trakcie tego procesu parametr wejścia Enable miał zachowaną wartość TRUE.

Tabela 10.7. Instrukcja MC_Power

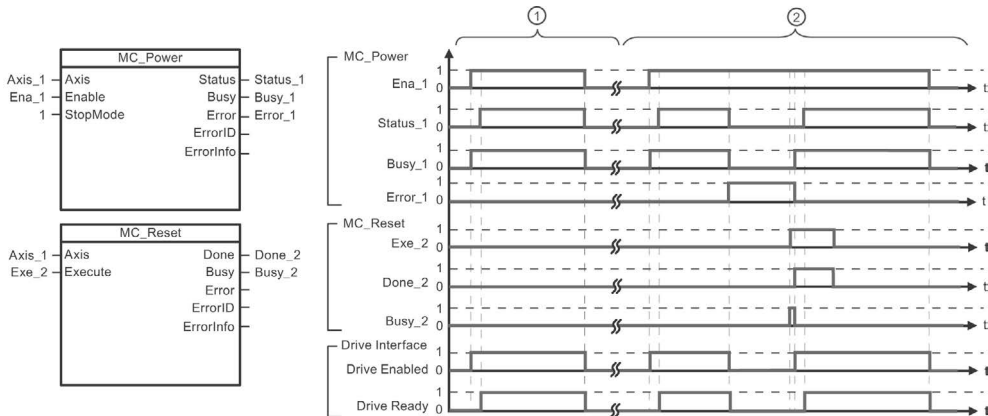
LAD/FBD	SCL	Opis
<p>"MC_Power_DB"</p> 	<pre>"MC_Power_DB" (Axis:=_multi_fb_in_, Enable:=_bool_in_, StopMode:=_int_in_, Status=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Instrukcja MC_Power steruje napędem oraz załącza lub wyłącza osie. Przed załączeniem lub wyłączeniem osi należy sprawdzić następujące warunki:</p> <ul style="list-style-type: none"> • Obiekt mechaniczny został poprawnie skonfigurowany. • Nie ma żadnego oczekującego błędu, który wywołałby zatrzymanie urządzenia. <p>Zadanie sterowania napędem nie może zatrzymać wykonania instrukcji MC_Power. Wyłączenie osi (parametr wejściowy Enable = FALSE) powoduje przerwanie wykonywania wszelkich zadań sterowania napędem dla powiązanych obiektów mechanicznych</p>

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_Power_DB" jest nazwą DB instancji.

Tabela 10.8. Parametry instrukcji MC_Power

Parametr i jego typ	Typ danych	Opis	
Axis	IN	TO_Axis_1	Obiekt mechaniczny
Enable	IN	Bool	<ul style="list-style-type: none"> FALSE (domyślna): Wszystkie aktywne zadania są porzucane zgodnie z sparametryzowanym trybem „StopMode” i oś ulega zatrzymaniu. TRUE: Sterowanie napędem próbuje załączyć oś.
StopMode	IN	Int	<ul style="list-style-type: none"> 0: Awaryjny stop –jeśli żądanie wyłączenia jest w toku, oś zatrzymuje się ze skonfigurowanym awaryjnym zwalnianiem. Oś jest wyłączona i nieruchoma. 1: Natychmiastowy stop – Jeśli żądanie zatrzymania jest w toku, oś jest wyłączana bez zwalniania. Wyjścia impulsowe są zatrzymywane natychmiastowo. 2: Zatrzymanie awaryjne ze sterowaniem zrywu: Jeśli żądanie dezaktywacji osi jest w toku, to oś hamuje przy skonfigurowanym opóźnieniu zatrzymania awaryjnego. Jeżeli sterowanie zrywu jest aktywne, to pod uwagę jest brany zryw skonfigurowany. Oś jest dezaktywowana po osiągnięciu stanu spoczynku.
Status	OUT	Bool	<p>Stan załączenia osi:</p> <ul style="list-style-type: none"> FALSE: Oś jest wyłączona. <ul style="list-style-type: none"> Oś nie wykonuje zadań sterowania napędem i nie przyjmuje żadnych nowych zadań (wyjątek stanowi zadanie MC_Reset). Oś nie jest bazowana. Po wyłączeniu osi, stan nie zmienia się na FALSE aż do zatrzymania osi. TRUE: Oś jest załączona. <ul style="list-style-type: none"> Oś jest gotowa do wykonywania zadań sterowania napędem. Po załączeniu osi, stan nie zmienia się na TRUE, aż do momentu, w którym rozpocznie się przetwarzanie sygnału <i>Drive ready</i>. Jeśli interfejs napędu <i>Drive ready</i> nie został wcześniej skonfigurowany, to jego stan zmieni się na TRUE natychmiast.
Busy	OUT	Bool	<p>FALSE: MC_Power nie jest aktywne. TRUE: MC_Power jest aktywna.</p>
Error	OUT	Bool	<p>FALSE: Brak błędu. TRUE: Wystąpił błąd w instrukcji sterowania napędem („MC_Power”) lub w związonym z nią obiekcie mechanicznym> Przyczyna błędu może zostać odczytana za pomocą parametrów <i>ErrorID</i> oraz <i>ErrorInfo</i>.</p>
ErrorID	OUT	Word	ID błędu dla parametru <i>Error</i> .
ErrorInfo	OUT	Word	Informacje o ID błędu dla parametru <i>ErrorID</i> .



- ① Oś zostaje wyłączona i załączona ponownie. Po zasignalizowaniu gotowości napędu do CPU, powodzenie załączania może być odczytane z „Status_1”.
- ② Po załączeniu osi wystąpił błąd, który spowodował jej wyłączenie. Błąd został wyłączony i zatwierdzony przez „MC_Reset”. Oś została ponownie załączona.

Aby załączyć oś ze skonfigurowanym interfejsem napędu, należy wykonać następujące kroki:

1. Sprawdzenie powyższych wymagań.
2. Przypisanie wejściowemu parametrowi „StopMode” żądanej wartości. Ustawienie parametru wejściowego Enable na TRUE.

Zmiana wyjścia *Drive enabled* na TRUE załącza zasilanie napędu.

CPU czeka na sygnał *Drive ready* z napędu.

Po uzyskaniu tego sygnału przez skonfigurowane wejście CPU, oś staje się aktywna. Wyjściowy parametr „Status” i zmienna obiektu technologicznego <Nazwa osi>.StatusBits.Enable uzyskują wartość TRUE.

Aby załączyć osie bez skonfigurowanego interfejsu napędu, należy wykonać następujące kroki:

1. Sprawdzenie powyższych wymagań
2. Przypisanie wejściowemu parametrowi „StopMode” żądanej wartości. Ustawienie parametru wejściowego Enable na TRUE. Oś jest załączona. Wyjściowy parametr „Status” i zmienna obiektu technologicznego <Nazwa osi>.StatusBits.Enable uzyskują wartość TRUE.


Aby wyłączyć osie należy wykonać następujące kroki:

1. Doprowadzić oś do zatrzymania.
Użytkownik może określić, czy oś jest nieruchoma przez zmienną obiektu technologicznego <Nazwa osi>.StatusBits.StandStill.
2. Wprowadzenie parametru wejściowego „Enable” = FALSE po zatrzymaniu osi.

3. Jeśli wejściowe parametry „Busy” oraz „Status” oraz zmienna obiektu technologicznego <Nazwa osi>.StatusBits.Enable mają wartość FALSE, wyłączenie osi jest zakończone.

10.5.3. Instrukcja MC_Reset (potwierdzanie błędów)

Tabela 10.9. Instrukcja MC_Reset

LAD/FBD	SCL	Opis
 <p>"MC_Reset_DB"</p> <p>MC_Reset</p> <p>EN END</p> <p>Axis Done</p> <p>Execute Busy</p> <p> Error</p> <p> ErrorID</p> <p> ErrorInfo</p>	<pre>"MC_Reset_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Restart:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Instrukcja MC_Reset potwierdza błąd pracy z zatrzymaniem osi (<i>Operating error with axis stop</i>) oraz błąd konfiguracji (<i>Configuration error</i>). Błędy, które wymagają potwierdzenia znajdują się na liście <i>ErrorIDs and ErrorInfos</i> w <i>Remedy</i> (środki naprawcze).</p> <p>Przed użyciem instrukcji MC_Reset, użytkownik musi wyeliminować źródło nadsyłania błędu konfiguracyjnego wymagającego potwierdzenia (przykładowo zmieniając niewłaściwą wartość przyspieszenia w obiekcie mechanicznym na wartość właściwą).</p> <p>Od wersji V3.0 lub wyższej, polecenie Restart umożliwia załadowanie konfiguracji osi do pamięci roboczej w trybie RUN.</p>

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_Reset_DB" jest nazwą DB instancji.

Zadanie MC_Reset nie może zostać przerwane przez żadne inne zadanie sterowania ruchem. Nowe zadanie MC_Reset nie przerywa żadnych innych aktywnych zadań sterowania napędem.

Tabela 10.10. Parametry instrukcji MC_Reset

Parametr i jego typ	Typ danych	Opis	
Axis – oś	IN	TO_Axis_1	Obiekt mechaniczny
Execute – wykonanie	IN	Bool	Rozpoczęcie zadania po wystąpieniu zbrocza narastającego.
Restart	IN	Bool	TRUE = Ładowanie konfiguracji osi z pamięci ładowania do pamięci roboczej. Polecenie może być wykonane tylko wtedy, gdy oś jest dezaktywowana.
			FALSE = Potwierdzenie występowania błędów
Done – wykonane	OUT	Bool	TRUE = Zadanie zostało wykonane.
Busy – zajęte	OUT	Bool	TRUE = Zadanie jest wykonywane.
Error – błąd	OUT	Bool	TRUE = Wystąpił błąd podczas wykonywania zadania. Przyczyna wystąpienia błędu może zostać znaleziona w parametrach „ErrorID” i „ErrorInfo”.
ErrorID	OUTP	Word	ID błędu dla parametru „Error”.
ErrorInfo	OUT	Word	Informacje o ID błędu dla parametru „ErrorID”.

Aby potwierdzić błąd za pomocą MC_Reset należy wykonać następujące kroki:

1. Sprawdzić powyższe wymagania.
2. Rozpocząć potwierdzanie błędu w przypadku zbocza narastającego na wejściu parametru Execute.
3. Błąd został potwierdzony, gdy w Done przyjmie wartość TRUE i zmienna obiektu technologicznego <Nazwa osi>.StatusBits.Error będzie równa FALSE.

10.5.4. Instrukcja MC_Home (bazowanie osi)

Tabela 10.11. Instrukcja MC_Home

LAD/FBD	SCL	Opis
	<pre>"MC_Home_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Position:= _real_in_, Mode:= _int_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>Instrukcja MC_Home służy do dopasowania współrzędnych osi do rzeczywistej pozycji napędu. Bazowanie nie jest potrzebne w przypadku stosowania bezwzględnego pozycjonowania osi.</p> <p>Instrukcja MC_Home może być użyta tylko w wypadku, gdy oś została wcześniej włączona.</p>

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_Home_DB" jest nazwą DB instancji.

Dostępne są następujące typy bazowania:

- Bezwzględne bazowanie bezpośrednie (*Mode=0*): Bieżąca pozycja osi jest ustawiona zgodnie z wartością parametru „Position”.
- Względne bazowanie bezpośrednie (*Mode=1*): Bieżąca pozycja osi jest odsunięta o wartość parametru „Position”.
- Bierne bazowanie (*Mode=2*): W tym trybie instrukcja MC_Home nie wykonuje żadnego ruchu bazującego. Ruch postępowy wymagany w tym kroku musi zostać zaimplementowany przez inne instrukcje sterujące napędem. Jeśli krańcówka punktu odniesienia zostanie wykryta, to oś ukończyła bazowanie.
- Aktywne bazowanie (*Mode=3*): Procedura bazowania jest wykonywana automatycznie

Tabela 10.12. Parametry instrukcji MC_Home

Parametr i jego typ	Typ danych	Opis	
Axis – oś	IN	TO_Axis_PTO	Obiekt mechaniczny
Execute – wykonanie	IN	Bool	Rozpoczęcie zadania po wystąpieniu zbocza narastającego
Position – pozycja	IN	Real	<ul style="list-style-type: none"> Mode = 0, 2, lub 3 (Bezwzględna pozycja osi po wykonaniu operacji bazowania). Mode = 1 (Wartość korekty dla bieżącej pozycji osi). Wartości graniczne: $-1.0e^{12} \leq \text{Pozycje} \leq 1.0e^{12}$
Mode – tryb	IN	Int	Tryb bazowania: <ul style="list-style-type: none"> 0: Bezwzględne bazowanie bezpośrednie. Nowa pozycja osi ma wartość z parametru „Position”. 1: Względne bazowanie bezpośrednie. Nowa pozycja osi wynosi: bieżąca wartość pozycji osi + wartość pozycji z parametru „Position”. 2: Pasywne bazowanie. Bazowanie zgodne z konfiguracją osi. W następstwie bazowania wartość z parametru „Position” jest ustawiana jako nowa pozycja osi. 3: Aktywne bazowanie. Podchodzenie do punktu odniesienia następuje zgodnie z konfiguracją osi. W następstwie bazowania wartość z parametru „Position” jest ustawiana jako nowa pozycja osi.
Done – wykonane	OUT	Bool	TRUE = Zadanie zostało wykonane.
Busy – zajęte	OUT	Bool	TRUE = Zadanie jest wykonywane.
CommandAborted – przerwanie komendy	OUT	Bool	TRUE = Wykonanie zadania zostało przerwane przez inne zadanie.
Error – błąd	OUT	Bool	TRUE = Wystąpił błąd podczas wykonywania zadania. Przyczyna wystąpienia błędu może zostać znaleziona w parametrach „ErrorID” i „ErrorInfo”.
ErrorID	OUT	Word	ID błędu dla parametru „Error”.
ErrorInfo	OUT	Word	Informacje o ID błędu dla parametru „ErrorID”.

Uwaga

Bazowanie osi będzie przerwane, jeśli nastąpią następujące zdarzenia:

- Wyłączenie osi poprzez instrukcję MC_Power.
- Przełączenie pomiędzy trybem automatycznego i ręcznego sterowania.
- Po uruchomieniu aktywnego bazowania (po udanym wykonaniu operacji bazowania, bazowanie osi jest ponownie dostępne).
- Po cyklu zasilania CPU (wyłączenie i włączenie CPU).
- Po zmianie trybu pracy sterownika (RUN-do-STOP lub STOP-do-RUN).

Do bazowania osi należy wykonać następujące kroki:

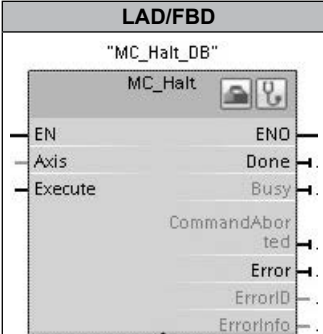
1. Sprawdzenie wymagań przedstawionych powyżej.
2. Zainicjowanie niezbędnych wartości wejściowych i rozpoczęcie bazowania w przypadku wystąpienia zbocza narastającego na wejściu parametru Execute.
3. Jeśli w parametrze wyjściowym Done ustawi się na TRUE i zmienna obiektu technologicznego <Nazwa osi>. StatusBits.HomingDone także będzie miała wartość TRUE, to proces bazowania został zakończony.

Tabela 10.13. Priorytety wykonywania instrukcji MC_Home

Tryb	Opis	
0 lub 1	Zadanie MC_Home nie może zostać przerwane przez żadne inne zadanie sterowania ruchem. Nowe zadanie MC_Home nie przerywa żadnych innych aktywnych zadań sterowania ruchem. Zadania ruchu związane z pozycją są przywracane po bazowaniu zgodnie z nową pozycją bazującą (wartość wejściowego parametru „Position”).	
2	Zadanie MC_Home może być przerwane przez następujące zadania sterowania napędem: Zadanie MC_Home, Tryb (Mode) = 2, 3: Nowe zadanie MC_Home przerywa kolejne aktywne zadanie sterowania napędem. Zadanie MC_Home, Tryb (Mode) = 2: Zadania napędu związane z pozycją są przywracane po bazowaniu zgodnie z nową pozycją bazującą (wartość wejściowego parametru „Position”).	
3	Zadanie MC_Home może być przerwane przez następujące zadania sterowania ruchem: <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 	Nowe zadanie MC_Home może być przerwane przez następujące aktywne zadania sterowania ruchem: <ul style="list-style-type: none"> • MC_Home Mode = 2, 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog

10.5.5. Instrukcja MC_Halt (zatrzymanie osi)

Tabela 10.14. Instrukcja MC_Halt

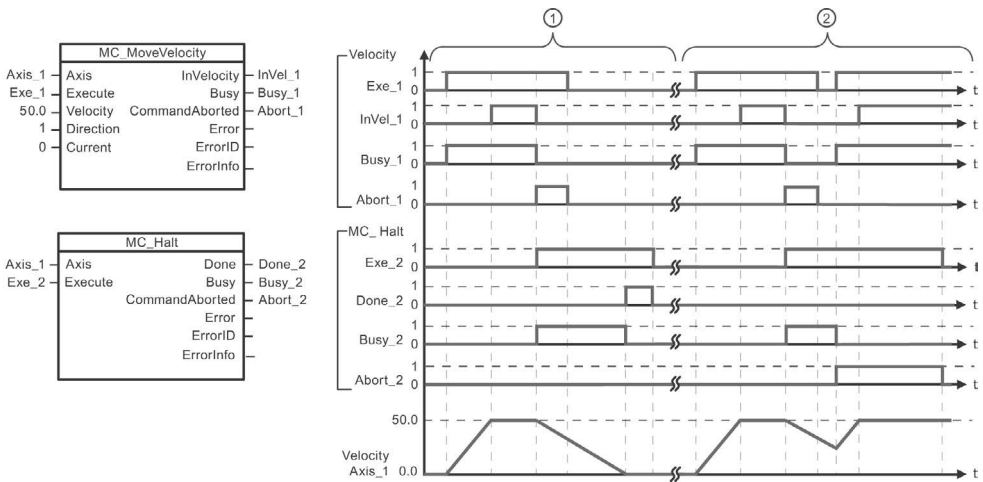
LAD/FBD	SCL	Opis
	<pre>"MC_Halt_DB" (Axis:= multi_fb_in_, Execute:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Instrukcja MC_Halt kasuje wszystkie procesy sterowania ruchem i powoduje zatrzymanie ruchów osi. Pozycja zatrzymania nie jest określona. Instrukcja MC_Halt może być użyta tylko w wypadku, gdy os została wcześniej załączona.</p>

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_Halt_DB" jest nazwą DB instancji.

Tabela 10.15. Parametry instrukcji MC_Halt

Parametr i jego typ	Typ danych	Opis
Axis – oś	IN	TO_Axis_1
Execute – wykonanie	IN	Bool
Done – wykonane	OUT	Bool
Busy – zajęte	OUT	Bool
CommandAborted	OUT	Bool
Error – błąd	OUT	Bool
ErrorID	OUT	Word
ErrorInfo	OUT	Word



Następujące wartości zostały ustawione w oknie konfiguracyjnym „Dynamics > General”:
Przyspieszenie = 10,0 oraz Opóźnienie = 5,0

- ① Oś jest hamowana przez zadanie MC_Halt dopóki nie zostanie zatrzymana. Stan zatrzymania jest sygnalizowany poprzez „Done_2”.
- ② Zadanie MC_Halt które hamuje oś zostaje przerwane przez inne zadanie sterowania ruchem. Przerwanie jest sygnalizowane przez „Abort_2”.

Priorytety wykonywania instrukcji MC_Halt

Zadanie MC_Halt może być przerwane przez następujące zadania sterowania napędem:


- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Nowe zadanie MC_Halt może być przerwane przez następujące aktywne zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.6. Instrukcja MC_MoveAbsolute (ustawienie osi w pozycji bezwzględnej)

Tabela 10.16. Instrukcja MC_MoveAbsolute

LAD/FBD	SCL	Opis
<p>"MC_MoveAbsolute_DB"</p> 	<pre>"MC_MoveAbsolute_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Velocity:=_real_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Instrukcja MC_MoveAbsolute rozpoczyna ruch w celu osiągnięcia pozycji bezwzględnej.</p> <p>Instrukcja MC_MoveAbsolute może być użyta tylko w wypadku, gdy oś została wcześniej załączona i bazowana.</p>

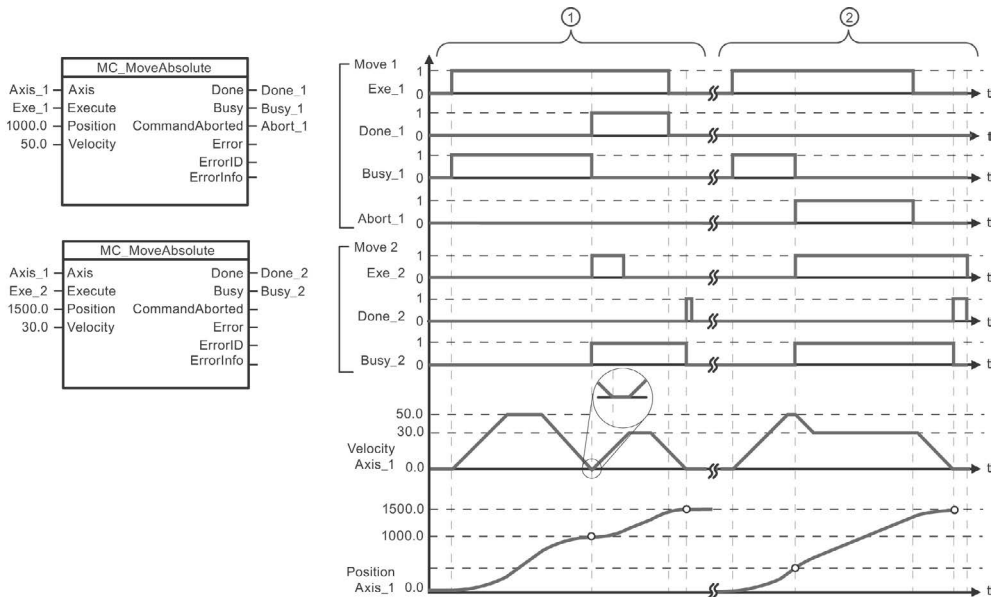
¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_MoveAbsolute_DB" jest nazwą DB instancji.

Tabela 10.17. Parametry instrukcji MC_MoveAbsolute

Parametr i jego typ	Typ danych	Opis
Axis – oś	IN TO_Axis_1	Objekt mechaniczny.
Execute – wykonanie	IN Bool	Rozpoczęcie zadania po wystąpieniu zbrocza narastającego (Wartość domyślna: FALSE).
Position – pozycja	IN Real	Bezwzględna pozycja docelowa (Wartość domyślna: 0.0). Graniczne wartości: $-1,0e^{12} \leq \text{Pozycja} \leq 1,0e^{12}$.
Velocity – prędkość	IN Real	Prędkość osi (Wartość domyślna: 10.0). Prędkość nie zawsze jest osiągana ze względu na dobrane wartości przyspieszeń i opóźnień oraz drogę dojścia do zadanej pozycji. Graniczne wartości: $\text{Prędkość Start/Stop} \leq \text{Prędkość} \leq \text{maksymalna Prędkość}$.

Parametr i jego typ	Typ danych	Opis	
Done – wykonane	OUT	Bool	TRUE = Osiągnięto zadaną pozycję.
Busy – zajęte	OUT	Bool	TRUE = Zadanie jest wykonywane.
CommandAborted	OUT	Bool	TRUE = Wykonanie zadania zostało przerwane przez inne zadanie.
Error - błąd	OUT	Bool	TRUE = Wystąpił błąd podczas wykonywania zadania. Przyczyna wystąpienia błędu może zostać znaleziona w parametrach „ErrorID” i „ErrorInfo”.
ErrorID	OUT	Word	ID błędu dla parametru „Error” (Wartość domyślna: 0000).
ErrorInfo	OUT	Word	Informacje o ID błędu dla parametru „ErrorID” (Wartość domyślna: 0000).



Następujące wartości zostały skonfigurowane w oknie konfiguracyjnym *Dynamics > General*: Przyspieszenie = 10.0 oraz Opóźnienie = 10.0.

- ① Zadanie MC_MoveAbsolute powoduje przesunięcie osi do bezwzględnej pozycji 1000.0. Osiągnięcie przez oś zadanej pozycji jest sygnalizowane przez „Done_1”. Po spełnieniu warunku „Done_1” = TRUE, uruchamiane jest następne zadanie MC_MoveAbsolute z pozycją docelową 1500.0. Z powodu krótkich czasów odpowiedzi (np. czas cyklu programu użytkownika, etc.) oś dociera szybko do zadanej pozycji (zob. powiększony szczegół). Osiągnięcie zadanej pozycji jest sygnalizowane przez „Done_2”.
- ② Aktywne zadanie MC_MoveAbsolute jest przerywane przez kolejne zadanie MC_MoveAbsolute. Jest to sygnalizowane przez „Abort_1”. Oś zostaje przesunięta z nową prędkością do nowej pozycji zadanej 1500.0. Dotarcie do tej pozycji jest sygnalizowane przez „Done_2”.

Priorytety wykonywania instrukcji MC_MoveAbsolute

Zadanie MC_MoveAbsolute może być przerwane przez następujące zadania sterowania napędem:


- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Nowe zadanie MC_MoveAbsolute może być przerwane przez następujące aktywne zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.7. Instrukcja MC_MoveRelative (ustawienie osi względem pozycji początkowej)

Tabela 10.18. Instrukcja MC_MoveRelative

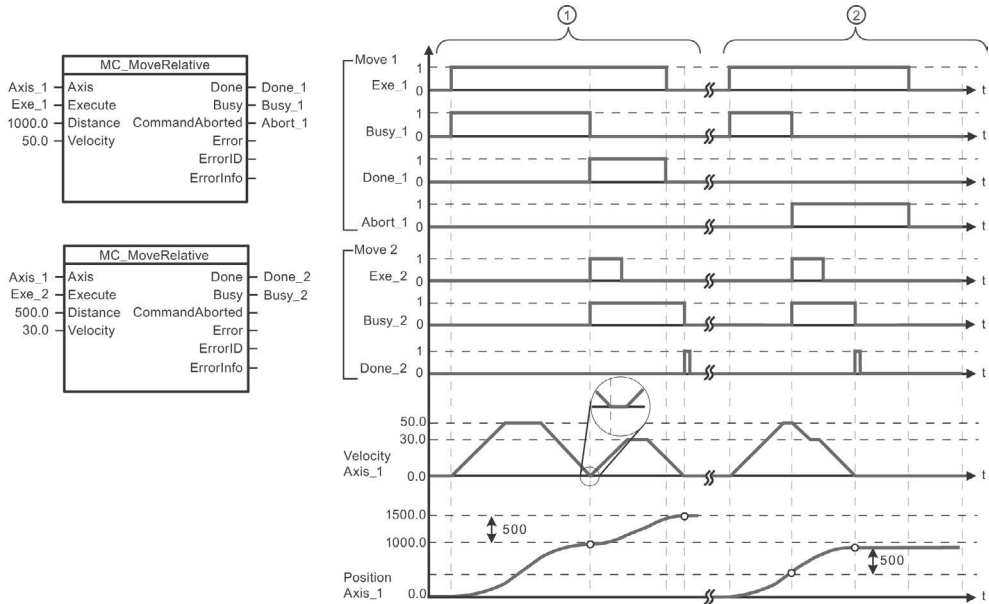
LAD/FBD	SCL	Opis
	<pre>"MC_MoveRelative_DB" (Axis:= multi_fb_in_, Execute:= bool_in_, Distance:= real_in_, Velocity:= real_in_, Done=> bool_out_, Busy=> bool_out_, CommandAborted=> bool_out_, Error=> bool_out_, ErrorID=> word_out_, ErrorInfo=> word_out_);</pre>	<p>Instrukcja MC_MoveRelative rozpoczyna ruch pozycjonujący względem pozycji startowej. Instrukcja MC_MoveRelative może być użyta tylko w wypadku, gdy oś została wcześniej załączona oraz naprowadzona.</p>

- STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.
- W przykładzie SCL, "MC_MoveRelative_DB" jest nazwą DB instancji.

Tabela 10.19. Parametry instrukcji MC_MoveRelative

Parametr i jego typ	Typ danych		Opis
Axis – oś	IN	TO_Axis_1	Obiekt mechaniczny.
Execute – wykonanie	IN	Bool	Rozpoczęcie zadania po wystąpieniu zbocza narastającego (Wartość domyślna: FALSE).
Position - pozycja	IN	Real	Długość przesunięcia dla operacji pozycjonowania (Wartość domyślna: 0.0) Graniczne wartości: $-1,0e^{12} \leq \text{Pozycja} \leq 1,0e^{12}$.
Velocity - prędkość	IN	Real	Prędkość osi (Wartość domyślna: 10.0) Ta prędkość nie, zawsze jest osiągnięta ze względu na dobrane wartości przyspieszeń i opóźnień oraz drogę dojścia do zadanej pozycji. Graniczne wartości: Prędkość Start/stop \leq Prędkość \leq maksymalna prędkość

Parametr i jego typ	Typ danych		Opis
Done – wykonane	OUT	Bool	TRUE = Osiągnięto zadaną pozycję
Busy – zajęte	OUT	Bool	TRUE = Zadanie jest wykonywane.
CommandAborted	OUT	Bool	TRUE = Wykonanie zadania zostało przerwane przez inne zadanie.
ErrorID	OUT	Word	ID błędu dla parametru „Error” (Wartość domyślna: 0000)
ErrorInfo	OUT	Word	Informacje o ID błędu dla parametru „ErrorID”(Wartość domyślna: 0000)



Następujące wartości zostały skonfigurowane w oknie konfiguracyjnym „Dynamics > General”: Przyspieszenie = 10,0 oraz Opóźnienie = 10,0.

- ① Zadanie MC_MoveRelative powoduje przesunięcie osi o odległość (*Distance*) 1000.0. Osiągnięcie przez oś zadanej pozycji jest sygnalizowane przez „Done_1”. Po spełnieniu warunku „Done_1” = TRUE, uruchamiane jest następne zadanie MC_MoveRelative z odległością 500.0. Z powodu krótkich czasów odpowiedzi (np. czas cyklu programu użytkownika, etc.) oś dociera szybko do zadanej pozycji (zob. powiększony szczegół). Osiągnięcie zadanej pozycji jest sygnalizowane poprzez „Done_2”.
- ② Aktywne zadanie MC_MoveRelative jest przerywane przez kolejne zadanie MC_MoveRelative. Jest to sygnalizowane przez „Abort_1”. Oś zostaje przesunięta z nową prędkością odległość (*Distance*) 500.0. Dotarcie do tej pozycji jest sygnalizowane przez „Done_2”.

Priorytety wykonywania instrukcji MC_MoveRelative

Zadanie MC_MoveRelative może być przerwane przez następujące zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Nowe zadanie MC_MoveRelative może być przerwane przez następujące aktywne zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.8. Instrukcja MC_MoveVelocity

Tabela 10.20. Instrukcja MC_MoveVelocity

LAD/FBD	SCL	Opis
	<pre>"MC_MoveVelocity_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Velocity:= _real_in_, Direction:= _int_in_, Current:= _bool_in_, InVelocity=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>Instrukcja MC_MoveVelocity utrzymuje ruch osi z określoną stałą prędkością.</p> <p>Instrukcja MC_MoveVelocity może być użyta tylko w wypadku, gdy oś została wcześniej załączona.</p>

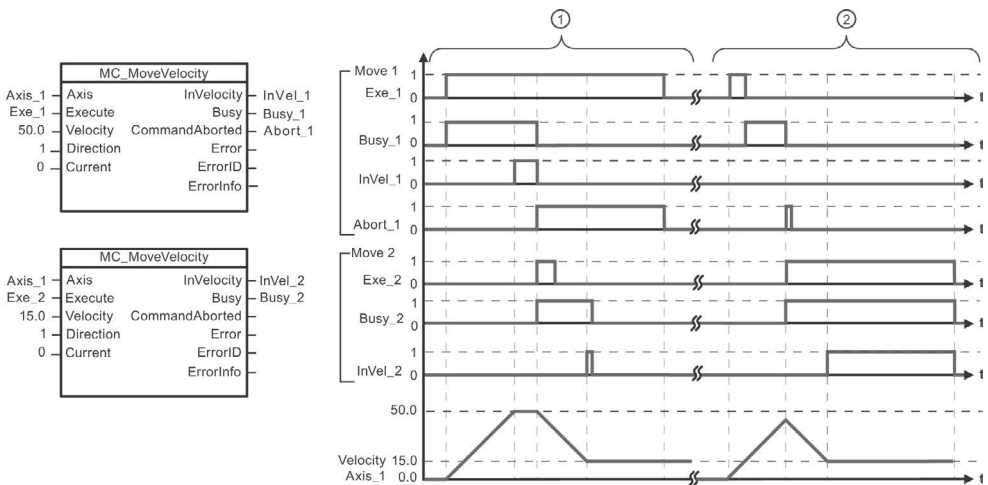
¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_MoveVelocity_DB" jest nazwą DB instancji.

Tabela 10.21. Parametry instrukcji MC_MoveVelocity

Parametr i jego typ	Typ danych	Opis	
Axis	IN	TO_Axis_1	Obiekt mechaniczny.
Execute	IN	Bool	Rozpoczęcie zadania po wystąpieniu zbocza narastającego (Wartość domyślna: FALSE).
Velocity	IN	Real	Określenie prędkości ruchu osi (Wartość domyślna: 10.0). Graniczne wartości: $-1,0e^{12} \leq \text{Prędkość} \leq 1,0e^{12}$. (Prędkość = 0 jest dozwolona).
Direction	IN	Int	Określenie kierunku: <ul style="list-style-type: none"> • 0: Kierunek obrotu odpowiada znakowi wartości parametru „Velocity” (Wartość domyślna). • 1: Dodatni kierunek obrotu (Znak wartości parametru „Velocity” jest ignorowany). • 2: Ujemny kierunek obrotu (Znak wartości parametru „Velocity” jest ignorowany).

Parametr i jego typ		Typ danych	Opis
Current	IN	Bool	Utrzymanie bieżącej prędkości <ul style="list-style-type: none"> FALSE: Utrzymanie bieżącej prędkości jest nieaktywne. Używane są wartości parametrów „Velocity” oraz „Direction” (Wartość domyślna) TRUE: Utrzymanie bieżącej prędkości jest aktywowane. Wartości parametrów „Velocity” oraz „Direction” nie są brane pod uwagę. Jeśli oś przywraca ruch do bieżącej prędkości, parametr „InVelocity” zwraca wartość TRUE.
InVelocity	OUT	Bool	TRUE: <ul style="list-style-type: none"> Jeśli „Current” = FALSE: prędkość określona w parametrze „Velocity” została osiągnięta. Jeśli „Current” = TRUE: Oś dochodzi do bieżącej prędkości podczas startu.
Busy	OUT	Bool	TRUE = Zadanie jest wykonywane.
CommandAborted	OUT	Bool	TRUE = Wykonanie zadania zostało przerwane przez inne zadanie.
Error	OUT	Bool	TRUE = Wystąpił błąd podczas wykonywania zadania. Przyczyna wystąpienia błędu może zostać znaleziona w parametrach „ErrorID” i „ErrorInfo”.
ErrorID	OUT	Word	ID błędu dla parametru „Error” (Wartość domyślna: 0000)
ErrorInfo	OUT	Word	Informacje o ID błędu dla parametru „ErrorID”(Wartość domyślna: 0000)



Następujące wartości zostały skonfigurowane w oknie konfiguracyjnym *Dynamics > General*: Przyspieszenie = 10.0 oraz Opóźnienie = 10.0

- ① Aktywne zadanie MC_MoveVelocity jest sygnalizowane poprzez „InVel_1” jeśli zadana prędkość została osiągnięta. Zadanie zostaje przerwane przez następne zadanie MC_MoveVelocity. Kiedy zostanie osiągnięta nowa zadana prędkość 15.0, sygnalizowane jest to poprzez „InVel_2”. Następnie oś kontynuuje ruch z nową stałą prędkością.
- ② Aktywne zadanie MC_MoveVelocity jest przerywane przez kolejne zadanie MC_MoveVelocity, aby osiągnąć nową prędkość. Jest to sygnalizowane przez „Abort_1”. Oś zostaje przesunięta z nową, stałą prędkością.

Priorytety wykonywania instrukcji MC_MoveVelocity

Zadanie MC_MoveVelocity może być przerwane przez następujące zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Nowe zadanie MC_MoveVelocity może być przerwane przez następujące aktywne zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Uwaga

Działanie przy prędkości zerowej (Velocity = 0,0)

Zadanie MC_MoveVelocity z ustawionym parametrem „Velocity” = 0,0 (tak jak zadanie MC_Halt) przerywa aktywne zadanie sterowania napędem i hamuje oś z nastawionym opóźnieniem. Po zatrzymaniu osi, parametr wyjściowy „InVelocity” zostaje ustawiony na TRUE na przynajmniej jeden cykl programu.

Parametr „Busy” jest ustawiony na TRUE podczas hamowania i zmienia się na FALSE wraz z parametrem „InVelocity”. Jeśli parametr „Execute” = TRUE, „InVelocity” oraz „Busy” zostają podtrzymane.

Jeśli zadanie MC_MoveVelocity zostało uruchomione, bit stanu „SpeedCommand” zostaje ustawiony w obiekcie mechanicznym. Bit stanu „ConstantVelocity” jest ustawiany po zatrzymaniu osi. Obydwa bity są przystosowywane do nowych sytuacji wraz z rozpoczęciem nowego zadania.

10.5.9. Instrukcja MC_MoveJog

Tabela 10.22. Instrukcja MC_MoveJog

LAD/FBD	SCL	Opis
	<pre>"MC_MoveJog_DB" (Axis:=_multi_fb_in_, JogForward:=_bool_in_, JogBackward:=_bool_in_, Velocity:=_real_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Instrukcja MC_MoveJog porusza oś ze stałą określoną prędkością w trybie Jog. Instrukcja ta jest szczególnie przydatna podczas testowania i rozruchu.</p> <p>Instrukcja MC_MoveJog może być użyta tylko w wypadku, gdy oś została wcześniej załączona.</p>

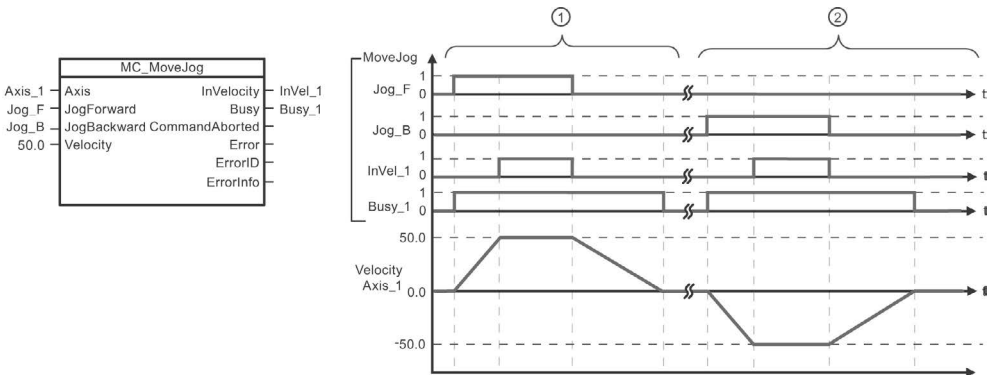
¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_MoveJog_DB " jest nazwą DB instancji.

Tabela 10.23. Parametry instrukcji MC_MoveJog

Parametr i jego typ		Typ danych	Opis
Axis	IN	TO_Axis_1	Obiekt mechaniczny
JogForward ¹	IN	Bool	Dopóki ten parametr ma wartość TRUE, oś porusza się w kierunku dodatnim z prędkością określoną w parametrze „Velocity”. Znak wartości parametru „Velocity” jest ignorowany (wartość domyślna: False).
JogBackward ¹	IN	Bool	Dopóki ten parametr ma wartość TRUE, oś porusza się w kierunku ujemnym z prędkością określoną w parametrze „Velocity”. Znak wartości parametru „Velocity” jest ignorowany (wartość domyślna: False).
Velocity	IN	Real	Ustawiona wartość prędkości dla trybu jog (Domyślna wartość: 10.0). Graniczne wartości: Prędkość Start/stop ≤ Prędkość ≤ maksymalna Prędkość.
InVelocity	OUT	Bool	TRUE = Prędkość określona w parametrze „Velocity” została osiągnięta.
Busy	OUT	Bool	TRUE = Zadanie jest Wykonywane.
CommandAborted	OUT	Bool	TRUE = Wykonanie zadania zostało przerwane przez inne zadanie.
Error	OUT	Bool	TRUE = Wystąpił błąd podczas wykonywania zadania. Przyczyna wystąpienia błędu może zostać znaleziona w parametrach „ErrorID” i „ErrorInfo”.
ErrorID	OUT	Word	ID błędu dla parametru „Error” (Wartość domyślna: 0000)
ErrorInfo	OUT	Word	Informacje o ID błędu dla parametru „ErrorID”(Wartość domyślna: 0000).

¹ Jeżeli obydwa parametry JogForward oraz JogBackward są ustawione na TRUE, to oś zwalnia z ustawionym opóźnieniem. Błędy zostają wskazane w parametrach „Error”, „ErrorID”, oraz „ErrorInfo”.



Następujące wartości zostały skonfigurowane w oknie konfiguracyjnym *Dynamics > General*: Przyspieszenie = 10,0 oraz Opóźnienie = 5,0.

- ① Oś porusza się w kierunku dodatnim w trybie jog poprzez parametr „Jog_F”. Kiedy zostanie osiągnięta zadana prędkość = 50,0, zostaje to zasygnalizowane przez „InVelo_1”. Po skasowaniu Jog_F oś hamuje do zatrzymania.
- ② Oś porusza się w kierunku ujemnym w trybie jog poprzez parametr „Jog_B”. Kiedy zostanie osiągnięta zadana prędkość = 50,0, zostaje to zasygnalizowane przez „InVelo_1”. Po skasowaniu Jog_B oś hamuje do zatrzymania.

Priorytety wykonywania instrukcji MC_MoveJog

Zadanie MC_MoveJog może być przerwane przez następujące zadania sterowania ruchem:

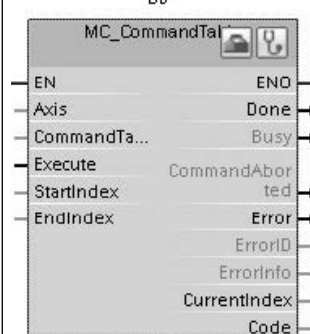
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Nowe zadanie MC_MoveJog może być przerwane przez następujące aktywne zadania sterowania ruchem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.10. MC_CommandTable (wykonywanie poleceń ruchu osi jako sekwencji ruchów)

Tabela 10.24. Instrukcja MC_CommandTable

LAD/FBD	SCL	Opis
	<pre>"MC_CommandTable_DB" (Axis:=_multi_fb_in_, CommandTable:=_multi_fb_in_, Execute:=_bool_in_, StartIndex:=_uint_in_, EndIndex:=_uint_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_, CurrentIndex=>_uint_out_, Code=>_word_out_);</pre>	<p>Instrukcja wykonuje serię pojedynczych ruchów napędu sterowanego osiowo, które mogą stworzyć sekwencję ruchu.</p> <p>Poszczególne ruchy są konfigurowane w oknie polecenia obiektu technologicznego jako wyjścia impulsowe (TO_CommandTable_PTO).</p>

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

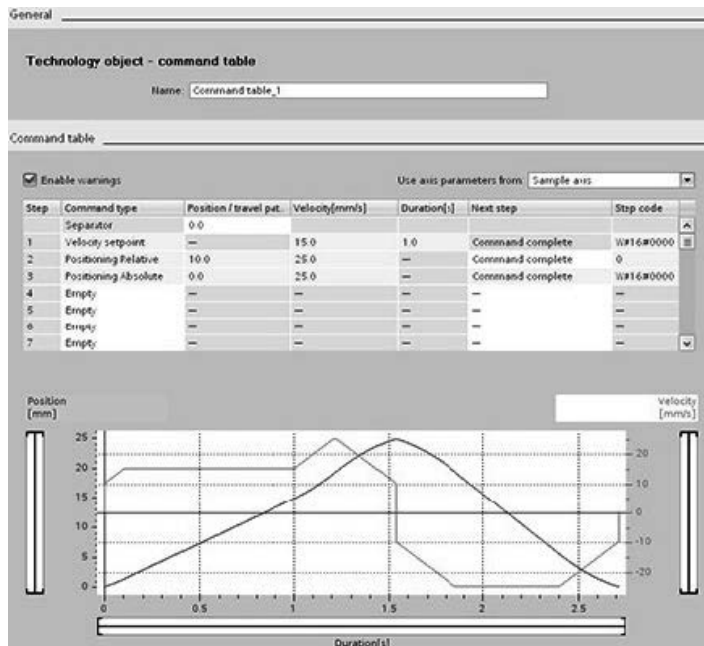
² W przykładzie SCL, "MC_CommandTable_DB " jest nazwą DB instancji.

Tabela 10.25. Parametry instrukcji MC_CommandTable

Parametr i jego typ	Typ danych	Wartość początkowa	Opis	
Axis	IN	TO_Axis_1	Objekt mechaniczny.	
Table	IN	TO_CommandTable_1	Tablice komend dla Obiektu technologii (<i>Command table</i>).	
Execute	IN	Bool	FALSE	Rozpoczęcie pracy wraz ze zboczem narastającym.
StartIndex	IN	Int	1	Rozpoczęcie przetwarzania tabeli komend z krokiem określonym w tym parametrze. Ograniczenia: $1 \leq \text{StartIndex} \leq \text{EndIndex}$

Parametr i jego typ		Typ danych	Wartość początkowa	Opis
EndIndex	IN	Int	32	Zakończenie przetwarzania tabeli komend z krokiem określonym w tym parametrze. Ograniczenia: StartIndex ≤ EndIndex ≤ 32.
Done	OUT	Bool	FALSE	Przetworzenie MC_CommandTable zakończyło się powodzeniem.
Busy	OUT	Bool	FALSE	TRUE = Zadanie jest wykonywane.
CommandAborted	OUT	Bool	FALSE	TRUE = Wykonanie zadania zostało przerwane przez inne zadanie.
Error	OUT	Bool	FALSE	TRUE = Wystąpił błąd podczas wykonywania zadania. Przyczyna wystąpienia błędu może zostać znaleziona w parametrach „ErrorID” i „ErrorInfo”.
ErrorID	OUT	Word	16#0000	Identyfikator błędu.
ErrorInfo	OUT	Word	16#0000	Informacje o błędzie.
Step	OUT	Int	0	Wielkość bieżącego kroku w procesie.
Code	OUT	Word	16#0000	Zdefiniowany przez użytkownika identyfikator bieżącego kroku w procesie.

Można utworzyć pożądaną sekwencję ruchów w oknie konfiguracji *Command Table*, i następnie sprawdzić rezultat w widoku graficznym na wykresie trendu.



Można wybrać typy poleceń, które mają być wykorzystywane do przetwarzania tabeli poleceń. Można wprowadzić do 32 zadań. Polecenia są wykonywane w kolejności.

Tab. 10.26. Typy poleceń instrukcji MC_CommandTable

Typ polecenia	Opis
<i>Empty</i> (Puste)	<i>Empty</i> (Puste) służy jako symbol zastępujący wszelkie polecenia, które mają być dodane. Wpis empty jest ignorowany, gdy jest przetwarzana tabela poleceń.
<i>Halt</i> (Zatrzymaj)	Powoduje zatrzymanie osi. Uwaga: Polecenie to jest wykonywane tylko po poleceniu <i>Velocity setpoint</i> .
<i>Positioning Relative</i> (Pozycjonowanie względne)	Ustawia oś na podstawie odległości. Polecenie powoduje przesunięcie osi na określoną odległość z określoną prędkością.
<i>Positioning Absolute</i> (Pozycjonowanie bezwzględne)	Ustawia oś na podstawie miejsca. Polecenie powoduje przesunięcie osi do określonego miejsca z określoną prędkością.
<i>Velocity setpoint</i> (Wartość zadana prędkości)	Powoduje przesuwanie osi z określoną prędkością.
<i>Wait</i> (Czekaj)	Powoduje wyczekiwanie do zakończenia określonego okresu. Polecenie <i>Wait</i> nie zatrzymuje aktywnego ruchu poprzecznego.
<i>Separator</i> (Separator)	Dodaje wiersz oddzielający powyżej zaznaczonego wiersza. Wiersz oddzielający umożliwia zdefiniowanie więcej niż jednego profilu w jednej tabeli poleceń.

Wymagania, które należy spełnić przed wykonaniem MC_CommandTable:

- Obiekt mechaniczny TO_Axis_PTO V2.0 musi być poprawnie skonfigurowany.
- Obiekt mechaniczny TO_CommandTable_PTO musi być poprawnie skonfigurowany.
- Oś musi zostać zwolniona.

Priorytety wykonywania instrukcji MC_CommandTable

Zadanie MC_CommandTable może być przerwane przez następujące zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable

Nowe zadanie MC_CommandTable może być przerwane przez następujące aktywne zadania sterowania napędem:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable
- Aktualne zadanie sterowania ruchem wraz z rozpoczęciem pierwszego polecenia pozycjonowania względnego *Positioning Relative*, pozycjonowania bezwzględnego *Positioning Absolute*, ustawienia prędkości *Velocity setpoint*, lub zatrzymania *Halt*

10.5.11. Instrukcja MC_ChangeDynamic (zmiana ustawień dynamicznych osi)

Tabela 10.27. MC_ChangeDynamic

LAD/FBD	SCL	Opis
	<pre>"MC_ChangeDynamic_DB" (Execute:=_bool_in_, ChangeRampUp:=_bool_in_, RampUpTime:=_real_in_, ChangeRampDown:=_bool_in_, RampDownTime:=_real_in_, ChangeEmergency:=_bool_in_, EmergencyRampTime:=_real_in_, ChangeJerkTime:=_bool_in_, JerkTime:=_real_in_, Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Zmiana dynamiczne ustawienia sterowania ruchem osi:</p> <ul style="list-style-type: none"> • Zmiana wartości czasu narastania (przyspieszenia) • Zmiana wartości czasu zwolnienia (opóźnienia) • Zmiana wartości czasu zwolnienia awaryjnego zatrzymania (opóźnienia awaryjnego zatrzymania) • Zmiana wartości czasu wygładzania (zrywu)

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL, "MC_ChangeDynamic_DB" jest nazwą DB instancji.

Tabela 10.28. Parametry instrukcji MC_ChangeDynamic

Parametr i jego typ	Typ danej	Opis
Axis	IN	TO_Axis_1 Oś obiektu technologicznego
Execute	IN	Bool Rozpoczęcie wykonywania polecenia dodatnim zboczem sygnału. Wartość domyślna: FALSE
ChangeRampUp	IN	Bool TRUE = Zmiana czasu narastania (ramp up time) zgodnie z parametrem wejściowym <i>RampUpTime</i> . Wartość domyślna: FALSE
RampUpTime	IN	Real Czas (w sekundach), aby przyspieszyć, od stanu spoczynku do uzyskania ustawionej prędkości maksymalnej, bez ograniczenia zrywu. Wartość domyślna: 5,00 Zmiana wpłynie na tag <Axis name>. Config. DynamicDefaults.Acceleration. Skuteczność zmian przedstawiono w opisie tego tagu.
ChangeRampDown	IN	Bool TRUE = Zmiana czasu zwolnienia (ramp down time) zgodnie z parametrem wejściowym <i>RampDownTime</i> . Wartość domyślna: FALSE
RampDownTime	IN	Real Czas (w sekundach), aby zwolnić ruch osi, od ustawionej prędkości maksymalnej do stanu spoczynku (zatrzymania), bez ograniczenia zrywu. Wartość domyślna: 5,00 Zmiana wpłynie na tag <Axis name>. Config. DynamicDefaults.Deceleration. Skuteczność zmian przedstawiono w opisie tego tagu.

Parametr i jego typ		Typ danej	Opis
ChangeEmergency	IN	Bool	TRUE = Zmiana czasu zwolnienia do awaryjnego zatrzymania zgodnie z parametrem wejściowym <i>EmergencyRampTime</i> . Wartość domyślna: FALSE
EmergencyRampTime	IN	Real	Czas (w sekundach), aby zwolnić ruch osi od ustawionej prędkości maksymalnej do stanu spoczynku (zatrzymania), bez ograniczenia zrywu w trybie awaryjnego zatrzymania. Wartość domyślna: 2,00 Zmiana wpłynie na tag <Axis name>. Config. DynamicDefaults.EmergencyDeceleration. Skuteczność zmian przedstawiono w opisie tego tagu.
ChangeJerkTime	IN	Bool	TRUE = Zmiana czasu wygładzania zgodnie z parametrem wejściowym <i>JerkTime</i> . Wartość domyślna: FALSE
JerkTime	IN	Real	Czas wygładzania (w sekundach), dla nachylenia (rampy) charakterystyki przyspieszenia i opóźnienia ruchu osi. Wartość domyślna: 0,25 Zmiana wpłynie na tag <Axis name>. Config. DynamicDefaults.Jerk. Skuteczność zmian przedstawiono w opisie tego tagu.
Done	OUT	Bool	TRUE = Zmienione wartości zostają zapisane w bloku danych DB obiektu technologicznego. Opis tagów pojawi się, kiedy zmiany odniosą skutek. Wartość domyślna: FALSE
Error	OUT	Bool	TRUE = Wystąpił błąd podczas wykonywania polecenia. Przyczynę błędu można znaleźć za pomocą <i>ErrorID</i> oraz <i>ErrorInfo</i> . Wartość domyślna: FALSE
ErrorID	OUT	Bool	Identyfikator błędu. Wartość domyślna: 16#0000
ErrorInfo	IN	Word	Informacje o błędzie. Wartość domyślna: 16#0000

Wymagania, które należy spełnić przed wykonaniem MC_ChangeDynamic:

- Obiekt mechaniczny TO_Axis_PTO V2.0 musi być poprawnie skonfigurowany.
- Oś musi zostać zwolniona.

Priorytety wykonywania instrukcji MC_ChangeDynamic

Polecenie MC_ChangeDynamic nie może być anulowane przez inne polecenie sterowania ruchem.

Nowe polecenie MC_ChangeDynamic nie anuluje innego aktywnego zadania sterowania ruchem.

Uwaga

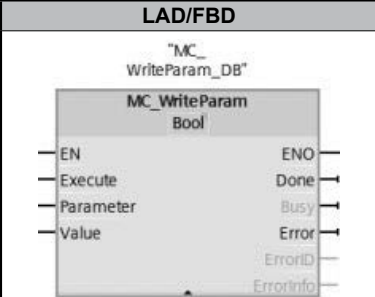
Parametry wejściowe „RampUpTime”, „RampDownTime”, „EmergencyRampTime” oraz „RoundingOffTime” mogą mieć określone wartości, które powodują, że parametry osi „acceleration”, „delay”, „emergency stop-delay” oraz „jerk” będą poza dozwolonymi ograniczeniami.

Użytkownik powinien się upewnić, że wartości parametru MC_ChangeDynamic mieszczą się w zakresie ograniczeń konfiguracji obiektu mechanicznego.

10.5.12. Instrukcja MC_WriteParam (zapisywanie parametrów obiektu technologicznego)

Za pomocą instrukcji MC_WriteParam można zapisać określoną liczbę parametrów do zmiany funkcjonalności osi z programu użytkownika.

Tabela 10.29. Instrukcja MC_WriteParam

LAD/FBD	SCL	Opis
	<pre>"MC_WriteParam_DB" (Parameter:=_variant_in_, Value:=_variant_in_, Execute:=_bool_in_, Done:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>Instrukcja MC_WriteParam zapisuje parametry publiczne (na przykład, wartości przyspieszenia i DB użytkownika).</p>

¹ Po wstawieniu instrukcji, program STEP 7 automatycznie tworzy blok danych DB.

² W przykładzie SCL, "MC_WriteParam_DB" jest nazwą DB instancji.

Można zapisywać wartości parametrów, które są publiczne (*public*). Nie można zapisywać wartości parametrów MotionStatus i StatusBits. W poniższej tabeli wykazane są parametry do zapisu:

Nazwa parametru do zapisu	Nazwa parametru do zapisu
Actor.InverseDirection	DynamicDefaults.Acceleration
Actor.DirectionMode	DynamicDefaults.Deceleration
Actor.DriveParameter. PulsesPerDriveRevolution	DynamicDefaults.Jerk
Sensor[1].ActiveHoming.Mode	DynamicDefaults.EmergencyDeceleration
Sensor[1].ActiveHoming.SideInput	PositionLimitsHW.Active
Sensor[1].ActiveHoming.Offset	PositionLimitsHW.MaxSwitchedLevel
Sensor[1].ActiveHoming.SwitchedLevel	PositionLimitsHW.MinSwitchedLevel
Sensor[1].PassiveHoming.Mode	PositionLimitsSW.Active
Sensor[1].PassiveHoming.SideInput	PositionLimitsSW.MinPosition
Sensor[1].PassiveHoming.SwitchedLevel	PositionLimitsSW.MaxPosition
Units.LengthUnit	Homing.AutoReversal
Mechanics.LeadScrew	Homing.ApproachDirection
DynamicLimits.MinVelocity	Homing.ApproachVelocity
DynamicLimits.MaxVelocity	Homing.ReferencingVelocity

Tabela 10.30. Parametry instrukcji MC_WriteParam

Parametr i jego typ	Typ danych	Opis	
PARAMNAME	IN	Variant	Nazwa parametru, którego wartość jest zapisywana
VALUE	IN	Variant	Wartość zapisywana przypisywanemu parametrowi
EXECUTE	IN	Bool	Rozpoczęcie wykonywania instrukcji. Wartość domyślna: FALSE
DONE	OUT	Bool	Wartość została zapisana. Wartość domyślna: FALSE
BUSY	OUT	Bool	Jeśli TRUE, to instrukcja jest wykonywana. Wartość domyślna: FALSE

Parametr i jego typ		Typ danych	Opis
ERROR	OUT	Real	Jeśli TRUE, to wystąpił błąd. Wartość domyślna: FALSE
ERRORID	OUT	Word	ID błędu
ERRORINFO	OUT	Word	Informacje związane z ERRORID

Tabela 10.31. Kody warunkowe dla parametrów ERRORID oraz ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Opis
0	0	Parametr osi TO-DB został pomyślnie zmieniony
8410[1]	0028[1]	Niepoprawny parametr (parametr osi TO-DB o niepoprawnej długości)
8410[1]	0029[1]	Niepoprawny parametr (to nie jest parametr osi TO-DB)
8410[1]	002B[1]	Niepoprawny parametr (parametr osi TO-DB tylko do odczytu)
8410[1]	002C[1]	Poprawny parametr, ale oś nie jest dezaktywowana
Config Error[2]	Config Error[2]	Poprawny parametr (publiczny parametr osi TO-DB tylko do odczytu) poza zakresem
Config Error[3]	Config Error[3]	Poprawny parametr (publiczny parametr osi TO-DB) poza zakresem

¹ Błąd instrukcji MC_WriteParam


² Błąd instrukcji MC_Power

³ Błąd instrukcji MC_Power oraz MC_MoveXXX lub MC_CommandTable

10.5.13. Instrukcja MC_ReadParam (odczytywanie parametrów obiektu technologicznego)

Za pomocą instrukcji MC_ReadParam można odczytać określoną liczbę parametrów, które wskazują aktualną pozycję, prędkość i inne dane osi zdefiniowane dla wejścia Axis.

Tabela 10.32. Instrukcja MC_ReadParam

LAD/FBD	SCL	Opis
	<pre>"MC_ReadParam_DB" (Enable:=_bool_in_, Parameter:=_variant_in_, Value:=_variant_in_out_, Valid:=_bool_out_, Busy:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>Instrukcja MC_ReadParam umożliwia odczyt pojedynczych wartości stanu, niezależnie od punktu kontroli cyklu.</p>

¹ Po wstawieniu instrukcji, program STEP 7 automatycznie tworzy blok danych DB.

² W przykładzie SCL, "MC_ReadParam_DB " jest nazwą DB instancji.

Instrukcja MC_ReadParam działa w oparciu o wartość logiczną na wejściu *Enable*. Gdy na wejściu *Enable* jest TRUE, to instrukcja odczytuje wartość określonego parametru *Parameter* i przekazuje do miejsca przechowywania wartości *Value*.

Wartość parametru *Position* funkcji diagnostycznej osi *MotionStatus* jest aktualizowana w każdym punkcie kontroli cyklu CCP (*Cycle Control Point*) na podstawie bieżącej wartości szybkiego licznika (HSC).

Wartość parametru *Velocity* funkcji diagnostycznej osi *MotionStatus* jest zadana prędkością na koniec bieżącego segmentu procesu (jest aktualizowana co ok. 10 ms). Instrukcja *MC_ReadParam* również umożliwia odczyt tej wartości.

Jeśli wystąpi błąd, to instrukcja powoduje przejście do stanu błędu, który można zresetować tylko przez nowe narastające zbocze na wejściu *Enable*.

Tabela 10.33. Parametry instrukcji *MC_ReadParam*

Parametr i jego typ	Typ danych	Opis	
ENABLE	IN	Bool	Rozpoczęcie wykonywania instrukcji. Wartość domyślna: FALSE
PARAMETER	IN	Variant	Wskaźnik do parametru TO, który ma być odczytany
VALID	OUT	Bool	Jeśli TRUE, to wartość została odczytana. wartość domyślna: FALSE
BUSY	OUT	Bool	Jeśli TRUE, to instrukcja jest wykonywana. Wartość domyślna: FALSE
ERROR	OUT	Real	Jeśli TRUE, to wystąpił błąd. Wartość domyślna: FALSE
ERRORID	OUT	Word	ID błędu. Wartość domyślna: 0
ERRORINFO	OUT	Word	Informacje związane z errorID. Wartość domyślna: 0
VALUE	INOUT	Variant	Wskaźnik do miejsca, w którym przechowywana jest odczytana wartość

Tabela 10.34. Kody warunkowe dla parametrów *ERRORID* oraz *ERRORINFO*

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Opis
0	0	Parametr został pomyślnie odczytany
8410	0028	Niepoprawny parametr (niepoprawna długość)
8410	0029	Niepoprawny parametr (to nie jest parametr osi TO-DB)
8410	0030	Niepoprawny parametr (nie do odczytu)
8411	0032	Niepoprawny parametr (niepoprawna wartość)

Parametry TO

Funkcje osi *MotionStatus* obejmują cztery zmienne. Można kontrolować zmiany wartości tych zmiennych, które mogą być odczytywane w czasie wykonywania programu:

Nazwa zmiennej	Typ danych	Do odczytu za pomocą instrukcji <i>MC_ReadParam</i>
<i>MotionStatus</i> :	Struktura	Nie
<i>Position</i>	REAL	Tak
<i>Velocity</i>	REAL	Tak
<i>Distance</i>	REAL	Tak
<i>TargetPosition</i>	REAL	Tak

10.6. Sterowanie ruchem przez sterownik S7-1200

10.6.1. Wyjścia CPU wykorzystywane do sterowania ruchem

CPU dostarcza cztery generatory wyjść impulsowych. Każdy generator wyjść impulsowych dostarcza jedno wyjście impulsowe i jedno wyjście kierunkowe do sterowania silnika krokowego lub serwonapędu za pomocą interfejsu impulsowego. Wyjście impulsowe dostarcza napędowi impulsy wymagane do ruchu silnika. Wyjście kierunkowe steruje kierunkiem poruszania się napędu.

Na wyjściu PTO jest generowany sygnał fali prostokątnej o zmiennej częstotliwości. Generowanie impulsów jest sterowane przez informacje konfiguracji i wykonania dostarczone przez konfigurację H/W i/lub SFC/SFB.

Na podstawie wyboru użytkownika, podczas gdy CPU jest w trybie RUN, albo wartości przechowywane w rejestrze obrazu, albo sygnały wyjściowe generatora impulsów sterują wyjściami cyfrowymi. W trybie STOP, generator PTO nie steruje wyjściami.

Wbudowane wyjścia CPU oraz wyjścia na płycie sygnałowej mogą być użyte jako wyjścia impulsowe i kierunkowe. Można wybrać wbudowane wyjścia CPU oraz wyjścia na płycie sygnałowej podczas konfiguracji urządzenia w sekcji Pulse generators (PTO/PWM) na zakładce *Properties*.

W tabeli poniżej przedstawiono domyślne przypisania wejść i wyjść; jednakże cztery generatory impulsowe mogą być skonfigurowane dla każdego wyjścia cyfrowego.

Uwaga

Wyjścia PTO nie mogą być wykorzystywane przez inne instrukcje w programie użytkownika

Przy konfigurowaniu wyjść CPU lub płytki sygnałowej jako generatorów impulsów (do zastosowania z PWM lub podstawowymi instrukcjami sterowania ruchem), odpowiadające adresy wyjściowe nie sterują już wyjściami. Jeśli program użytkownika zapisuje wartość na wyjściu używanym jako generator impulsów, to CPU nie zapisuje tej wartości na wyjściu fizycznym.

Uwaga

Wyjścia kierunkowe PTO mogą zostać zwolnione do innego zastosowania w programie użytkownika.

Każde PTO wymaga przypisania dwóch wyjść: wyjścia impulsowego i wyjścia kierunkowego. Osobno można używać wyjścia impulsowego, ale nie wyjścia kierunkowego. Wtedy można zwolnić wyjście kierunkowe do innego zastosowania w programie użytkownika. Wyjście nie może być jednocześnie używane jako wyjście kierunkowe PTO oraz w programie użytkownika.

Tabela 10.35. Domyślne przypisanie adresów wyjść impulsowych i kierunkowych

Wykorzystanie wyjść do sterowania ruchem		
	Impulsowe	Kierunkowe
PTO1		
Wejścia/wyjścia wbudowane	Q0.0	Q0.1
Wejścia/wyjścia SB	Q4.0	Q4.1
PTO2		
Wejścia/wyjścia wbudowane	Q0.2	Q0.3
Wejścia/wyjścia SB	Q4.2	Q4.3
PTO3		
Wejścia/wyjścia wbudowane	Q0.4 ¹	Q0.5 ¹
Wejścia/wyjścia SB	Q4.0	Q4.1
PTO4		
Wejścia/wyjścia wbudowane	Q0.6 ²	Q0.7 ²
Wejścia/wyjścia SB	Q4.2	Q4.3

¹ CPU 1211C nie ma wyjść Q0.4, Q0.5, Q0.6, oraz Q0.7. Dlatego wyjścia te nie mogą być stosowane w CPU 1211C.

² CPU 1211C nie ma wyjść Q0.4, Q0.5, Q0.6, oraz Q0.7. Dlatego wyjścia te nie mogą być stosowane w CPU 1211C.

³ Ta tabela dotyczy funkcji PTO CPU 1211C, CPU 1212C, CPU 1214C oraz CPU 1215C.

Interfejs napędu

Do sterowania ruchem, można opcjonalnie skonfigurować interfejs napędu z użyciem opcji *Drive enabled* (napęd aktywowany) i *Drive ready* (napęd gotowy). W przypadku korzystania z interfejsu napędu, można dowolnie wybrać wyjście cyfrowe dla opcji *Drive enabled* oraz wejście cyfrowe dla opcji *Drive ready*.

Uwaga

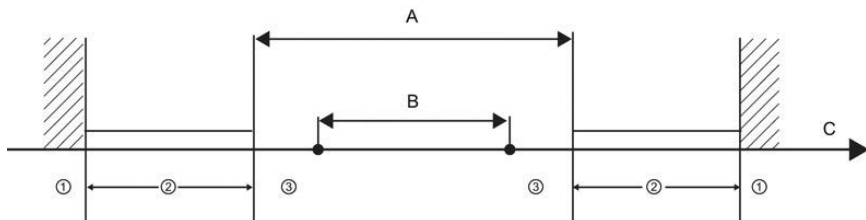
Jeśli wyjście generujące serię impulsów PTO (*Pulse Train Output*) zostało wybrane i przypisane do osi, to oprogramowanie wbudowane (*firmware*) przejmie kontrolę przez odpowiadające wyjścia impulsowe i kierunkowe.

Przy takim przejściu funkcji sterowania, połączenie pomiędzy obrazem procesu a wyjściem I/O jest rozłączone. Podczas gdy użytkownik ma możliwość zapisu obrazu procesu z wyjść impulsowych i kierunkowych za pomocą programu użytkownika lub tablicy monitorującej (*watch table*), to nigdy nie jest on przekazywany do wyjścia I/O. W związku z tym nie jest również możliwe monitorowanie wyniku I/O przez program użytkownika lub tablicę monitorującą. Odczytywane informacje zaledwie odzwierciedlają wartość obrazu procesu i pod żadnym względem nie odpowiadają aktualnemu stanowi wyjścia I/O.

Dla wszystkich innych wyjść CPU, które nie są używane na stałe przez firmwarę CPU, status wyjścia I/O może być jak zwykle kontrolowany lub monitorowany przez obraz procesu.

10.6.2. Sprzętowe i programowe wyłączniki krańcowe do sterowania ruchem

W celu ograniczenia dozwolonego zakresu przesuwania *allowed travel range* oraz zakresu roboczego *working range* dla osi, używa się sprzętowych i programowych wyłączników krańcowych.



- | | |
|---|------------------------------------|
| ① Ogranicznik mechaniczny | A Dozwolony zakres przesuwania osi |
| ② Dolne i górne ograniczenia sprzętowe | B Zakres roboczy osi |
| ③ Dolne i górne ograniczenia programowe | C Odległość |

Przed użyciem sprzętowe i programowe wyłączniki krańcowe muszą być aktywowane przez konfigurację lub w programie użytkownika. Wyłączniki krańcowe są aktywne tylko po przejściu osi do pozycji spoczynkowej (homingu).

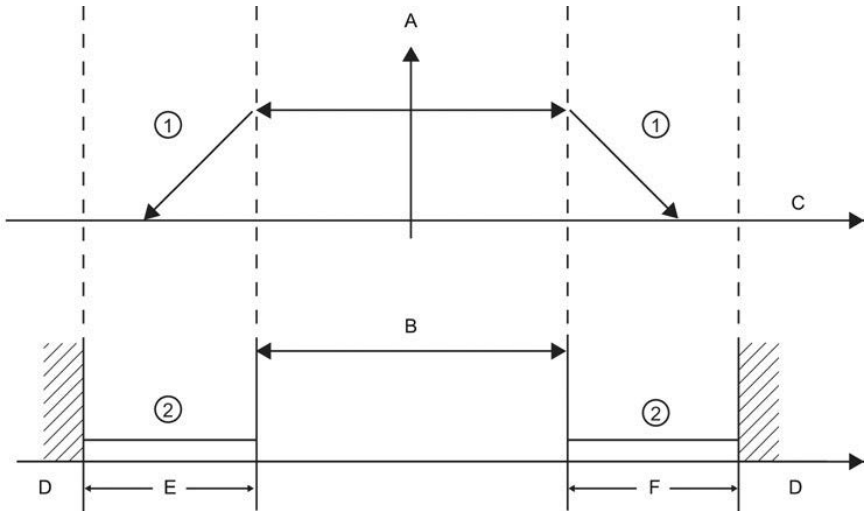
Sprzętowe wyłączniki krańcowe

Sprzętowe wyłączniki krańcowe wyznaczają maksymalny zakres przesuwania osi. Sprzętowe wyłączniki krańcowe są fizycznymi elementami wyłączającymi, które muszą być podłączone do wejść CPU, które mogą obsługiwać przerwania. Należy używać tylko takich wyłączników sprzętowych, które po zbliżeniu pozostają na stałe wyłączone. Ten stan wyłączenia może zostać cofnięty jedynie po powrocie do dozwolonego zakresu przesuwania.

Table 10.36. Dostępne wejścia dla ograniczeń sprzętowych

Opis	RPS	LIM-	LIM+
Wejścia/wyjścia wbudowane		10.0 – 11.5	
Wejścia/wyjścia SB		14.0 – 14.3	

Kiedy następuje zbliżenie do krańcowych wyłączników sprzętowych, os hamuje do stanu spoczynku przy ustalonym opóźnieniu awaryjnym. Określone opóźnienie awaryjne musi być wystarczające, aby pewnie zatrzymać os przed ogranicznikiem mechanicznym. Poniższy wykres przedstawia zachowanie osi po jej zbliżeniu do sprzętowych wyłączników krańcowych.



- ① Oś hamuje do stanu spoczynku przy ustalonym opóźnieniu awaryjnym.
 ② Zakres, w którym sprzętowe wyłączniki krańcowe sygnalizują stan zbliżenia „approached”.
- A Prędkość [Velocity]
 B Dozwolony zakres przesuwania
 C Odległość
 D Ogranicznik mechaniczny
 E Dolny sprzętowy wyłącznik krańcowy
 F Górny sprzętowy wyłącznik krańcowy
 G Operation of motion control for S7-1200



OSTRZEŻENIE

Zagrożenie zmian czasu filtra cyfrowego kanału wejściowego

Jeśli czas filtra kanału wejścia cyfrowego zmieni się od poprzedniego ustawienia, to udostępnienie nowej wartości wejściowej na poziomie „0” może zajmować łącznie do 20,0 ms, zanim filtr będzie w pełni reagował na nowe dane wejściowe. W tym czasie, zdarzenia krótkich impulsów „0” trwające mniej niż 20,0 ms mogą zostać nie wykryte lub nie zliczone.

Ta zmiana czasów filtra może spowodować nieoczekiwane działanie maszyny lub procesu, co może spowodować śmierć lub poważne zranienie personelu i/lub uszkodzenia mienia.

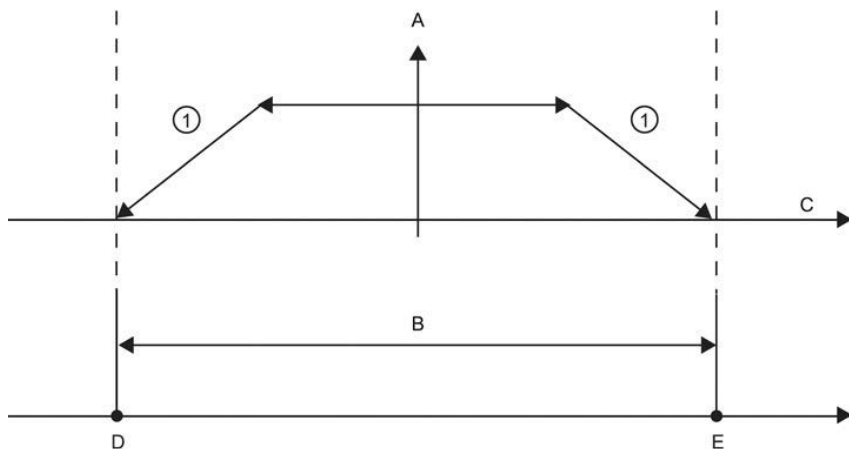
Aby nowy czas filtra został natychmiast wdrożony, należy wyłączyć i ponownie włączyć zasilanie CPU.

Programowe wyłączniki krańcowe

Programowe wyłączniki krańcowe ograniczają zakres pracy „working range” osi. Powinny one znajdować się wewnątrz sprzętowych wyłączników krańcowych w

odniesieniu do zakresu przesuwania. Ponieważ pozycje programowych wyłączników krańcowych można ustawić elastycznie, zakres roboczy osi może być ograniczony indywidualnie, w zależności od bieżącego profilu ruchu. W przeciwieństwie do sprzętowych wyłączników krańcowych, programowe wyłączniki krańcowe są realizowane wyłącznie za pomocą oprogramowania i nie wymagają własnych elementów przełączających.

Jeśli programowe wyłączniki krańcowe są aktywowane, to aktywny ruch jest zatrzymywany na pozycji programowego wyłącznika krańcowego. Oś jest hamowana przy ustalonym opóźnieniu. Poniższy wykres przedstawia zachowanie osi aż do osiągnięcia programowego wyłącznika krańcowego.



① Oś hamuje do stanu spoczynku przy ustalonym opóźnieniu.

A Prędkość [Velocity]

B Zakres roboczy

C Odległość

D Dolny programowy wyłącznik krańcowy

E Górny programowy wyłącznik krańcowy

Jeśli ogranicznik mechaniczny znajduje się po programowych wyłącznikach krańcowych i istnieje ryzyko uszkodzenia mechanicznego, to należy użyć dodatkowych sprzętowych wyłączników krańcowych.

Informacje dodatkowe

Program użytkownika może zastąpić sprzętowe lub programowe ograniczenia pozycji przez włączenie lub wyłączenie funkcjonalności ograniczeń zarówno sprzętowych, jak i programowych. Wyboru dokonuje się w bloku danych DB osi.

Aby włączyć lub wyłączyć funkcjonalność ograniczeń sprzętowych, należy uzyskać dostęp do tagu *Active* (Bool) używając ścieżki DB „<axis name>/Config/PositionLimits_HW”. Stan tagu *Active* włącza lub wyłącza sprzętowe ograniczenia pozycji.

Aby włączyć lub wyłączyć funkcjonalności ograniczeń programowych pozycji, należy uzyskać dostęp do tagu *Active* (Bool) używając ścieżki DB „<axis name>/Config/PositionLimits_SW”. Stan tagu *Active* włącza lub wyłącza programowe ograniczenia pozycji.

Można również zmodyfikować programowe ograniczenia pozycji za pomocą programu użytkownika (na przykład, dodać opcję konfiguracji urządzenia lub skrócić czas przełączania urządzenia). Program użytkownika może również zapisać nowe wartości do tagów minimalnej wartości pozycji *MinPosition* i maksymalnej wartości pozycji „*MaxPosition*” (jednostki techniczne w formacie Real) używając ścieżki DB „<axis name>/Config/PositionLimits_SW”.

10.6.3. Homing

10.6.3.1. Przejście osi do pozycji spoczynkowej

Homing odnosi się do dopasowania współrzędnych osi do rzeczywistego, fizycznego położenia napędu. (Jeśli napęd jest aktualnie na pozycji x, to oś będzie tak ustawiana, by znaleźć się na pozycji x). Dla osi sterowanych pozycją, wprowadzane wartości i wyświetlane wyniki pozycji odnoszą się dokładnie do tych współrzędnych osi.

Uwaga

Zachowanie zgodności pomiędzy współrzędnymi osi a rzeczywistą sytuacją jest niezwykle ważne. Ten krok jest niezbędny, aby zapewnić, że bezwzględna docelowa pozycja osi jest również osiągnięta dokładnie z napędem.

Instrukcja MC_Home inicjuje homing osi.

Istnieją 4 różne funkcje homingu. Pierwsze dwie funkcje pozwalają użytkownikowi ustawić aktualną pozycję osi, a pozostałe dwie pozycjonują oś w stosunku do czujnika odniesienia *Home*.

- Tryb 0 – Odniesienie bezpośrednie, bezwzględne: Po uruchomieniu tego trybu, oś uzyskuje informację, gdzie dokładnie jest. To ustawia wewnętrzną zmienną pozycji na wartość parametru wejściowego *Position* instrukcji MC_Home. Jest to używane do kalibracji i ustawienia maszyny.

Pozycja osi jest ustawiana niezależnie od wyłącznika punktu odniesienia (*Reference Point Switch*). Aktywne ruchy poprzeczne nie są przerywane. Wartość parametru wejściowego *Position* instrukcji MC_Home jest natychmiast ustawiana jako punkt odniesienia osi. Aby przypisać ten punkt odniesienia do dokładnej fizycznej pozycji, w czasie operacji homingu oś na tej pozycji musi być w stanie spoczynku.

- Tryb 1 – Odniesienie bezpośrednie, względne: Po uruchomieniu, w tym trybie używana jest wewnętrzna zmienna pozycji i do niej dodawana jest wartość parametru wejściowego *Position* instrukcji MC_Home. Jest to zazwyczaj używane w celu uwzględnienia przesunięcia maszyny (offsetu).

Pozycja osi jest ustawiana niezależnie od wyłącznika punktu odniesienia. Aktywne ruchy poprzeczne nie są przerywane. Poniższe równanie odnosi się do pozycji osi po homingu: nowa pozycja osi = aktualna pozycja osi + wartość parametru *Position* instrukcji MC_Home.

- Tryb 2 – Odniesienie pasywne: Kiedy oś porusza się i przechodzi przez wyłącznik punktu odniesienia, aktualna pozycja jest ustawiana jako pozycja spoczynkowa (*home position*). Ta funkcja pomaga uwzględnić normalne zużycie maszyny i luzy przekładni oraz zapobiec konieczności ręcznej kompensacji zużycia. Parametr wejściowy *Position* instrukcji MC_Home, jak przedtem, dodaje pozycję wskazaną przez wyłącznik punktu odniesienia, umożliwiając łatwe przesunięcie z pozycji spoczynkowej.

Podczas homingu pasywnego, instrukcja MC_Home nie wykonuje żadnego ruchu homingu. Ruch poprzeczny wymagany na tym etapie musi być realizowany przez użytkownika za pośrednictwem innych instrukcji sterowania ruchem. Kiedy zostanie wykryty wyłącznik punktu odniesienia, to oś przejdzie do pozycji spoczynkowej zgodnie z konfiguracją. Po rozpoczęciu homingu pasywnego, aktywne ruchy poprzeczne nie są przerywane.

- Tryb 3 – Odniesienie aktywne: Ten tryb jest najbardziej precyzyjną metodą homingu osi. Wartości początkowe kierunku i prędkość ruchu są konfigurowane w konfiguracji obiektu technologicznego Technology Object Configuration Extended Parameters-Homing. To zależy od konfiguracji maszyny. Istnieje również możliwość określenia, czy w pozycji spoczynkowej występuje zbocze narastające lub opadające sygnału wyłącznika punktu odniesienia. Praktycznie wszystkie czujniki mają zakres aktywny i jeśli pozycja ustalona Steady State On byłaby użyta do sygnalizowania pozycji spoczynkowej, wtedy byłaby możliwość wystąpienia błędu w pozycji spoczynkowej ponieważ aktywny zakres sygnału pozycji On pokrywałby zakres odległości. Używając albo narastającego albo opadającego zbocza tego sygnału uzyskiwana jest bardziej precyzyjna pozycja spoczynkowa. Jak w przypadku wszystkich innych trybów, wartość parametru wejściowego *Position* instrukcji MC_Home jest dodawana do sprzętowo odniesionej pozycji.

W aktywnym trybie homingu, instrukcja MC_Home realizuje wymagane podejście do punktu odniesienia. Kiedy wyłącznik punktu odniesienia zostanie wykryty, oś przechodzi do pozycji spoczynkowej zgodnie z konfiguracją. Aktywne ruchy poprzeczne są przerywane.

W trybach 0 i 1 nie jest wymagane, aby oś w ogóle się poruszała. Są one zazwyczaj używane do ustawień i kalibracji. W trybach 2 i 3 jest wymagane, aby oś poruszała się i przechodziła przez czujnik, który jest skonfigurowany w obiekcie technologicznym „Axis” jako wyłącznik punktu odniesienia. Punkt odniesienia może być umieszczony w obszarze roboczym osi lub poza normalnym obszarem roboczym, ale w zasięgu ruchu.

10.6.3.2. Konfiguracja parametrów homingu

Parametry dla aktywnego i pasywnego homingu konfiguruje się w oknie konfiguracji *Homing*. Metodę homingu ustawia się za pomocą parametru wejściowego *Mode* (tryb) instrukcji sterowania ruchem. Tutaj, *Mode* = 2 oznacza homing pasywny, a *Mode* = 3 oznacza, homing aktywny.

Uwaga

Aby maszyna nie przekroczyła położenia ogranicznika mechanicznego w przypadku odwrócenia kierunku, należy spełnić jeden z następujących warunków:

- utrzymywać niską prędkość,
- zwiększyć konfigurowane przyspieszanie/opóźnienie,
- zwiększyć odległości pomiędzy sprzętowym wyłącznikiem krańcowym a ogranicznikiem mechanicznym.

Tabela 10.37. Parametry konfiguracyjne dla homingu osi

Parametr	Opis
<i>Input reference point switch</i> (Wejściowy wyłącznik punktu odniesienia) (homing aktywny i pasywny)	<p>Należy wybrać wejście cyfrowe dla wyłącznika punktu odniesienia z rozwijanej listy wyboru. Wejście musi być zdolne do obsługi przerwań. Jako wejścia dla wyłącznika punktu odniesienia można wybrać wbudowane wejścia CPU oraz wejścia wstawionej płytki sygnałowej.</p> <p>Domyślny czas filtrowania (<i>filter time</i>) dla wejść cyfrowych wynosi 6,4 ms. Kiedy wejście cyfrowe jest wykorzystywane jako wyłącznik punktu odniesienia, może to prowadzić do niepożądanych opóźnień, i tym samym nieściśłości. W zależności od zredukowanej prędkości i zakresu działania wyłącznika punktu odniesienia, punkt odniesienia może nie zostać wykryty. Czas filtrowania może być ustawiony pod <i>Input filter</i> w konfiguracji urządzenia dla wejść cyfrowych.</p> <p>Określony czas filtrowania może być krótszy niż czas trwania sygnału wejściowego na wyłączniku punktu odniesienia.</p>
<i>Auto reverse after reaching the hardware limit switches</i> (Automatyczna zmiana kierunku po osiągnięciu sprzętowych wyłączników krańcowych) (Tylko homing aktywny)	<p>Należy zaznaczyć pole wyboru, aby korzystać ze sprzętowego wyłącznika krańcowego jako czujnika zmiany kierunku w celu podejścia do punktu odniesienia. Sprzętowe wyłączniki krańcowe muszą być skonfigurowane i uaktywnione dla zmiany kierunku.</p> <p>Jeśli sprzętowy wyłącznik krańcowy zostanie osiągnięty w trakcie aktywnego homingu, to oś hamuje ze skonfigurowanym opóźnieniem (nie z opóźnieniem awaryjnym) i zmienia kierunek. Następnie w odwrotnym kierunku wykrywany jest wyłącznik punktu odniesienia.</p> <p>Jeśli zmiana kierunku nie jest aktywna, a oś dochodzi do sprzętowego wyłącznika krańcowego podczas aktywnego homingu, podejście do punktu odniesienia jest przerywane z błędem i oś jest hamowana z opóźnieniem awaryjnym.</p>

Parametr	Opis
<i>Approach direction</i> (Kierunek podejścia) (Homing aktywny i pasywny)	Wraz z wyborem kierunku, należy określić kierunek podejścia <i>approach direction</i> używany podczas homingu aktywnego do wyszukiwania wyłącznika punktu odniesienia, jak również kierunku homingu. Kierunek homingu określa kierunek przesuwania osi do podejścia od skonfigurowanej strony wyłącznika punktu odniesienia, w celu przeprowadzenia operacji homingu.
<i>Reference point switch</i> (Wyłącznik punktu odniesienia) (Homing aktywny i pasywny)	<ul style="list-style-type: none"> • Homing aktywny: Należy wybrać, czy oś ma być odnoszona z lewej, czy z prawej strony wyłącznika punktu odniesienia. W zależności od pozycji początkowej osi i konfiguracji parametrów homingu, sekwencja podejścia do punktu odniesienia może się różnić od diagramu w oknie konfiguracji. • Homing pasywny: Przy homingu pasywnym, poprzeczne ruchy dla celów homingu muszą być wdrożone przez użytkownika za pomocą instrukcji sterowania ruchem. Strona wyłącznika punktu odniesienia, po której występuje homing zależy od następujących czynników: <ul style="list-style-type: none"> – konfiguracji kierunku podejścia <i>Approach direction</i>, – konfiguracji wyłącznika punktu odniesienia <i>Reference point switch</i>, – aktualnego kierunku przesuwania podczas homingu pasywnego.
<i>Approach velocity</i> (Prędkość podejścia) (Tylko homing aktywny)	Należy określić prędkość, przy której ma być wyszukiwany wyłącznik punktu odniesienia podczas zbliżenia do punktu odniesienia. Wartości graniczne (niezależnie od wybranej jednostki przez użytkownika): prędkość początkowa/końcowa (start/stop) \leq <i>Approach velocity</i> (prędkość podejścia) \leq prędkość maksymalna
<i>Reduced velocity</i> (Prędkość zredukowana) (Tylko homing aktywny)	Należy określić prędkość, z którą oś zbliża się do wyłącznika punktu odniesienia w celu homingu. Wartości graniczne (niezależnie od wybranej jednostki przez użytkownika): prędkość początkowa/końcowa (start/stop) \leq <i>reduced velocity</i> (prędkość zredukowana) \leq prędkość maksymalna
<i>Home position offset</i> (Przesunięcie pozycji spoczynkowej) (Tylko homing aktywny)	Jeśli pożądana pozycja odniesienia odbiega od pozycji wyłącznika punktu odniesienia, to w tym polu może być określone przesunięcie pozycji spoczynkowej. Jeśli ta wartość nie jest równa 0, to oś wykonuje następujące akcje po homingu na wyłączniku punktu odniesienia: <ol style="list-style-type: none"> 1. Przesuwanie osi ze zredukowaną prędkością o wartość przesunięcia pozycji spoczynkowej. 2. Kiedy pozycja przesunięcia pozycji spoczynkowej zostanie osiągnięta, pozycja osi jest ustawiana na bezwzględną pozycję odniesienia. Bezwzględna pozycja odniesienia jest określona przez parametr <i>Position</i> (pozycja) instrukcji sterowania ruchem „MC_Home”. Wartości graniczne (niezależnie od wybranej jednostki przez użytkownika): $-1,0e^{12} \leq$ <i>Home position offset</i> (przesunięcie pozycji spoczynkowej) $\leq 1,0e^{12}$

Tabela 10.38. Czynniki wpływające na homing

Czynniki wpływające:			Wynik:
Konfiguracja kierunku podejścia „Approach direction”	Konfiguracja wyłącznika punktu odniesienia „Reference point switch”	Aktualny kierunek przesuwania <i>Current travel direction</i>	Homing na wyłączniku punktu odniesienia <i>Homing on Reference point switch</i>
<i>Positive</i> Dodatni	<i>Left (negative) side</i> Lewa (ujemna) strona	<i>Positive direction</i> Kierunek dodatni	<i>Left</i> Lewy
		<i>Negative direction</i> Kierunek ujemny	<i>Right</i> Prawy
<i>Positive</i> Dodatni	<i>Right (positive) side</i> Prawa (dodatnia) strona	<i>Positive direction</i> Kierunek dodatni	<i>Right</i> Prawy
		<i>Negative direction</i> Kierunek ujemny	<i>Left</i> Lewy
<i>Negative</i> Ujemny	<i>Left (negative) side</i> Lewa (ujemna) strona	<i>Positive direction</i> Kierunek dodatni	<i>Right</i> Prawy
		<i>Negative direction</i> Kierunek ujemny	<i>Left</i> Lewy
<i>Negative</i> Ujemny	<i>Right (positive) side</i> Prawa (dodatnia) strona	<i>Positive direction</i> Kierunek dodatni	<i>Left</i> Lewy
		<i>Negative direction</i> Kierunek ujemny	<i>Right</i> Prawy

10.6.3.3. Sekwencja homingu aktywnego

Homing aktywny rozpoczyna się od instrukcji sterowania ruchem „MC_Home” (parametr wejściowy Mode = 3). W tym przypadku parametr wejściowy *Position* określa bezwzględne współrzędne punktu odniesienia. Alternatywnie, do celów testowych homing aktywny można rozpocząć na panelu sterowania.

Poniższy wykres pokazuje przykład charakterystyki statycznej dla aktywnego podejścia do punktu odniesienia przy następujących parametrach konfiguracji:

- kierunek podejścia *Approach direction* = dodatni kierunek podejścia *Positive approach direction*
- wyłącznik punktu odniesienia *Reference point switch* = prawa (dodatnia) strona *Right (positive) side*
- wartość przesunięcia pozycji spoczynkowej *home position offset* > 0

Tabela 10.39. Charakterystyka prędkości przejścia osi do pozycji spoczynkowej

Operacja		Uwagi	
		A	Prędkość podejścia
		B	Prędkość zredukowana
		C	Współrzędne pozycji spoczynkowej
		D	Przesunięcie pozycji spoczynkowej
①	Faza wyszukiwania (niebieski odcinek krzywej): Kiedy rozpoczyna się homing aktywny, oś przyspiesza do skonfigurowanej prędkości podejścia <i>approach velocity</i> i przy tej prędkości wyszukuje wyłącznika punktu odniesienia.		
②	Podjęcie do punktu odniesienia (czerwona część krzywej): Gdy wyłącznik punktu odniesienia zostanie wykryty, to oś w tym przykładzie hamuje i zmienia kierunek poruszania, by przejść do pozycji spoczynkowej po skonfigurowanej stronie wyłącznika punktu odniesienia ze skonfigurowaną prędkością zredukowaną <i>reduced velocity</i> .		
③	Przesuwanie do pozycji punktu odniesienia (zielona część krzywej): Po homingu na wyłączniku punktu odniesienia, oś przesuwana się do współrzędnych punktu odniesienia <i>Reference point coordinates</i> z prędkością zredukowaną <i>reduced velocity</i> . Po osiągnięciu współrzędnych punktu odniesienia, oś zostaje zatrzymana na pozycji określonej przez parametr wejściowy <i>Position</i> instrukcji MC_Home.		

Uwaga

Jeśli wyszukiwanie homingu nie działa zgodnie z oczekiwaniami, należy sprawdzić wejścia przypisane do ograniczników sprzętowych lub do punktu odniesienia. Na wejściach tych przerwania od zbroczy mogą być dezaktywowane w konfiguracji urządzenia.

Należy zbadać dane konfiguracyjne dla obiektu technologicznego osi, aby sprawdzić, które wejścia (o ile takie są) są przypisane dla opcji *HW Low Limit Switch Input*, *HW High Limit Switch Input*, oraz *Input reference point switch*. Następnie należy otworzyć okno konfiguracji urządzenia dla CPU i zbadać każde z przypisanych wejść. Opcje uaktywnienia wykrywania narastającego zbrocza sygnału *Enable rising edge detection* oraz uaktywnienia wykrywania opadającego zbrocza sygnału *Enable falling edge detection* powinny być zaznaczone. Jeśli opcje te nie są zaznaczone, to należy usunąć określone wejścia w konfiguracji osi i wybrać je ponownie.

10.7. Uruchomienie

Funkcja diagnostyczna statusów i bitów błędów

Do monitorowania najważniejszych statusów i komunikatów błędów dla osi używana jest funkcja diagnostyczna statusów i bitów błędów *Status and error bits*. Wyświetlenie funkcji diagnostycznej jest dostępne online w trybie sterowania ręcznego *Manual control* i automatycznego *Automatic control*, kiedy oś jest aktywna.

Tabela 10.40. Status osi

Status	Opis
<i>Enabled</i> (Uaktywniona)	Oś jest uaktywniona i gotowa, aby była sterowana przez zadania sterowania ruchem. (Tag obiektu technologicznego: <Axis name>.StatusBits.Enable)
<i>Homed</i> (Przeszła do pozycji spoczynkowej)	Oś jest w pozycji spoczynkowej i może wykonywać zadania pozycjonowania bezwzględnego za pomocą instrukcji sterowania ruchem „MC_MoveAbsolute”. W przypadku homingu względnego oś nie musi być w pozycji spoczynkowej. Sytuacje wyjątkowe: Podczas homingu aktywnego, status jest ustawiony na FALSE. Jeśli oś znajdująca się w pozycji spoczynkowej będzie przechodzić homing pasywny, to podczas homingu pasywnego status jest ustawiony na TRUE. (Tag obiektu technologicznego: <Axis name>.StatusBits.HomingDone)
<i>Error</i> (Błąd)	Wystąpił błąd w obiekcie technologicznym „Axis”. Więcej informacji o błędach jest dostępnych w przy automatycznym sterowaniu w parametrach ErrorID oraz ErrorInfo instrukcji sterowania ruchem. W trybie ręcznym pole <i>Last Error</i> panelu sterowania wyświetla szczegółowe informacje o przyczynie błędu. (Tag obiektu technologicznego: <Axis name>.StatusBits.Error)
<i>Control panel active</i> (Aktywny panel sterowania)	Tryb sterowania ręcznego <i>Manual control</i> został uaktywniony w panelu sterowania. Panel sterowania ma pierwszeństwo sterowania obiektem technologicznym „Axis”. Osi nie może być sterowane z programu użytkownika. (Tag obiektu technologicznego: <Axis name>.StatusBits.ControlPanelActive)

Tabela 10.41. Status napędu

Status	Opis
<i>Drive ready</i> (Napęd gotowy)	Napęd jest gotowy do pracy. (Tag obiektu technologicznego: <Axis name>.StatusBits.DriveReady)
<i>Error</i> (Błąd)	Napęd zgłosił błąd sygnału gotowości. (Tag obiektu technologicznego: <Axis name>.ErrorBits.DriveFault)

Tabela 10.42. Status ruchu osi

Status	Opis
<i>Standstill</i> (Stan spoczynku lub zatrzymanie)	Oś jest w stanie spoczynku. (Tag obiektu technologicznego: <Axis name>.StatusBits.StandStill)
<i>Accelerating</i> (Przyspieszanie)	Oś przyspiesza. (Tag obiektu technologicznego: <Axis name>.StatusBits.Acceleration)
<i>Constant velocity</i> (Prędkość stała)	Oś porusza się ze stałą prędkością. (Tag obiektu technologicznego: <Axis name>.StatusBits.ConstantVelocity)
<i>Decelerating</i> (Zwalnianie)	Oś zwalnia (spowalnia). (Tag obiektu technologicznego: <Axis name>.StatusBits.Deceleration)

Tabela 10.43. Status trybu ruchu

Status	Opis
<i>Positioning</i> (Pozycjonowanie)	Oś wykonuje zadanie pozycjonowania za pomocą instrukcji sterowania ruchem „MC_MoveAbsolute” lub „MC_MoveRelative” lub z panelu sterowania. (Tag obiektu technologicznego: <Axis name>.StatusBits.PositioningCommand)
<i>Speed Command</i> (Polecenie prędkości)	Oś wykonuje zadanie przy ustawionej prędkości za pomocą instrukcji sterowania ruchem „MC_MoveVelocity” lub „MC_MoveJog” lub z panelu sterowania. (Tag obiektu technologicznego: <Axis name>.StatusBits.SpeedCommand)
<i>Homing</i> (Homing lub przejście osi do pozycji spoczynkowej)	Oś wykonuje zadanie homingu za pomocą instrukcji sterowania ruchem „MC_Home” lub panelu sterowania. (Tag obiektu technologicznego: <Axis name>.StatusBits.Homing)

Tabela 10.44. Bity błędów

Błąd	Opis
<i>Min software limit reached</i> (Osiągnięto minimalne ograniczenie programowe)	Osiągnięto dolny programowy wyłącznik krańcowy. (Tag obiektu technologicznego: <Axis name>.ErrorBits.SwLimitMinReached)
<i>Min software limit exceeded</i> (Przekroczono minimalne ograniczenie programowe)	Przekroczono dolny programowy wyłącznik krańcowy. (Tag obiektu technologicznego: <Axis name>.ErrorBits.SwLimitMinExceeded)
<i>Max software limit reached</i> (Osiągnięto maksymalne ograniczenie programowe)	Osiągnięto górny programowy wyłącznik krańcowy. (Tag obiektu technologicznego: <Axis name>.ErrorBits.SwLimitMaxReached)
<i>Max software limit exceeded</i> (Przekroczono minimalne ograniczenie programowe)	Przekroczono górny programowy wyłącznik krańcowy. (Tag obiektu technologicznego: <Axis name>.ErrorBits.SwLimitMaxExceeded)

Błąd	Opis
<i>Negative hardware limit</i> (Ujemne ograniczenie sprzętowe)	Zbliżono do dolnego sprzętowego wyłącznika krańcowego. (Tag obiektu technologicznego: <Axis name>.ErrorBits.HwLimitMin)
<i>Positive hardware limit</i> (Dodatnie ograniczenie sprzętowe)	Zbliżono do górnego sprzętowego wyłącznika krańcowego. (Tag obiektu technologicznego: <Axis name>.ErrorBits.HwLimitMax)
<i>PTO and HSC already used</i> (Wyjście generujące serię impulsów PTO oraz szybki licznik HSC są już używane)	Druga oś używa tego samego wyjścia generującego serię impulsów PTO oraz szybkiego licznika HSC i jest uaktywniona za pomocą instrukcji „MC_Power”. (Tag obiektu technologicznego: <Axis name>.ErrorBits.HwUsed)
<i>Configuration error</i> (Błąd konfiguracji)	Obiekt technologiczny „Axis” został skonfigurowany niepoprawnie lub edytowalne dane konfiguracyjne zostały zmodyfikowane niepoprawnie podczas wykonywania programu użytkownika. (Tag obiektu technologicznego: <Axis name>.ErrorBits.ConfigFault)
<i>General Error</i> (Błąd ogólny)	Wystąpił wewnętrzny błąd. (Tag obiektu technologicznego: <Axis name>.ErrorBits.SystemFault)

Funkcja diagnostyczna „Motion Status”

Do monitorowania statusu ruchu osi używa się funkcji diagnostycznej statusu ruchu „Motion status”. Wyświetlenie funkcji diagnostycznej jest dostępne online w trybie sterowania ręcznego *Manual control* i automatycznego *Automatic control*, kiedy oś jest aktywna.

Tabela 10.45. Status ruchu

Status	Opis
<i>Target position</i> (Pozycja docelowa)	Pole pozycji docelowej <i>Target position</i> wskazuje aktualną pozycję docelową aktywnego zadania pozycjonowania wykonywanego za pomocą instrukcji sterowania ruchem „MC_MoveAbsolute” lub „MC_MoveRelative” lub z panelu sterowania. Wartość pola <i>Target position</i> jest ważna tylko w czasie wykonywania zadania pozycjonowania. (Tag obiektu technologicznego: <Axis name>.MotionStatus.TargetPosition)
<i>Current position</i> (Aktualna pozycja)	Pole aktualnej pozycji <i>Current position</i> wskazuje aktualną pozycję osi. Jeżeli oś nie jest w pozycji spoczynkowej, to ta wartość wskazuje wartość pozycji względem pozycji uaktywnienia osi. (Tag obiektu technologicznego: <Axis name>.MotionStatus.Position)
<i>Current velocity</i> (Aktualna prędkość)	Pole aktualnej prędkości <i>Current velocity</i> wskazuje rzeczywistą prędkość osi. (Tag obiektu technologicznego: <Axis name>.MotionStatus.Velocity)

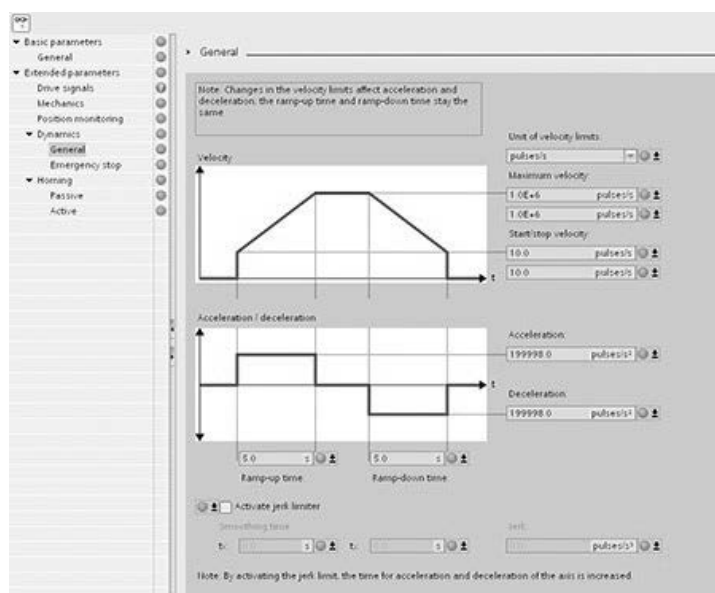
Tabela 10.46. Wartości graniczne wielkości dynamicznych

Wielkość dynamiczna	Opis
<i>Velocity</i> (Prędkość)	Pole prędkości <i>Velocity</i> określa konfigurowaną, maksymalną prędkość osi. (Tag obiektu technologicznego: <Axis name>.Config.DynamicLimits.MaxVelocity)
<i>Acceleration</i> (Przyspieszenie)	Pole przyspieszenia <i>Acceleration</i> wskazuje konfigurowane, aktualne przyspieszenie osi. (Tag obiektu technologicznego: <Axis name>.Config.DynamicDefaults.Acceleration)
<i>Deceleration</i> (Opóźnienie)	Pole opóźnienia <i>Deceleration</i> wskazuje konfigurowane, aktualne opóźnienie osi. (Tag obiektu technologicznego: <Axis name>.Config.DynamicDefaults.Deceleration)

Kontrola wartości początkowych parametrów sterowania ruchem

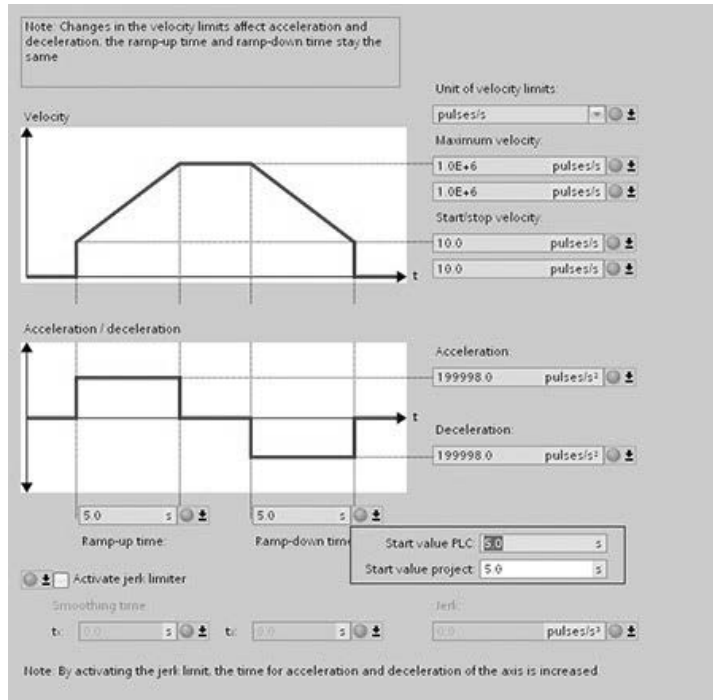
Można edytować rzeczywiste wartości parametrów konfiguracyjnych sterowania ruchem tak, aby przebieg procesu mógł być optymalizowany w trybie online.

Należy otworzyć *Technology objects* (Obiekty technologiczne) dla sterowania ruchem i odpowiadający obiekt *Configuration* (Konfiguracja). Aby uzyskać dostęp do kontroli wartości początkowych, należy kliknąć ikonę okularów w lewym górnym rogu okna dialogowego:



Teraz można zmieniać wartości dowolnych parametrów konfiguracyjnych sterowania ruchem, jak pokazano na rysunku poniżej.

Można porównać wartość rzeczywistą z wartością początkową projektu (offline) i wartością początkową PLC (online) każdego parametru. Jest to konieczne do porównania różnic online/offline bloku danych obiektu technologicznego (TO-DB) oraz uzyskania informacji o wartościach, które będą używane jako bieżące wartości przy następnej zmianie stanów Stop-Start PLC. Ponadto, ikona porównania daje wizualne wskazanie ułatwiające wykrycie różnic online i offline.



Na rysunku powyżej przedstawiono okno parametrów sterowania ruchem z ikonami porównania pokazującymi, które wartości się różnią pomiędzy projektami online i offline. Zielona ikona wskazuje, że wartości są takie same; ikona niebieska lub pomarańczowa oznacza, że wartości są różne.

Dodatkowo, kliknięcie przycisku parametru ze strzałką w dół powoduje otwarcie małego okna, które pokazuje wartość początkową projektu *Start value project* (offline) oraz wartość początkową PLC *Start value PLC* (online) każdego parametru.

Note: Changes in the velocity limits affect acceleration and deceleration, the ramp-up time and ramp-down time stay the same

Unit of velocity limits: pulses/s

Maximum velocity: 1.0E+6 Impulses

Start/stop velocity: 10.0 Impulses

Acceleration: 199998.0 pulses/s²

Deceleration: 199998.0 pulses/s²

Ramp-up time: 5.0 s

Ramp-down time: 5.0 s

Start value FLC: 30 s

Start value project: 5.1 s

Activate jerk limiter:

Smoothing time: t₁: 0.0 s, t₂: 0.0 s

Jerk: 0.0 pulses/s³

Note: By activating the jerk limit, the time for acceleration and deceleration of the axis is increased.

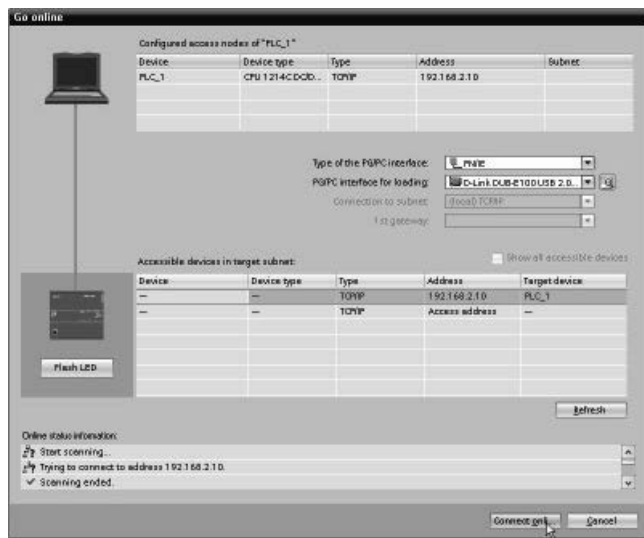
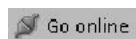
11.1. Tryb online i połączenie z CPU

Połączenie online pomiędzy urządzeniem programującym i CPU jest wymagane do załadowania programów i danych projektów inżynierskich.

Połączenie online pozwala na następujące operacje:

- Testowanie programów użytkownika.
- Wyświetlanie i zmianę trybu pracy CPU (strona 254).
- Wyświetlanie i ustawienie daty i czasu dnia dla CPU (strona 263).
- Wyświetlanie informacji o modułach.
- Porównanie i synchronizacja bloków programów online i offline (strona 262).
- Zgrywanie i wgrywanie bloków programu.
- Wyświetlenie danych diagnostycznych i bufora diagnostycznego (strona 263).
- Użycie tablicy monitorującej (*Watch table*, strona 256) do sprawdzenia programu użytkownika przez sprawdzenie i modyfikację wartości.
- Użycie tabeli wymuszeń do ustawiania wartości w CPU (strona 257).

Podłączenie do określonego CPU, znajdującego się w rozbudowanej sieci, następuje po wybraniu z drzewa nawigacji projektu (*Project Navigation*) odpowiedniego CPU i kliknięciu *Go online* w widoku projektu.



Jeśli CPU jest podłączone po raz pierwszy online, użytkownik musi wybrać typ interfejsu PG/PC i określony interfejs PG/PC z okna dialogowego przed ustawieniem połączenia z CPU bazującego na tym interfejsie.

Urządzenie programujące jest już połączone z CPU. Pomarańczowa ramka wskazuje połączenie online. Można wówczas w widoku *online* lub diagnostycznym uzyskać dostęp do danych systemu docelowego za pomocą karty *Online tools*.

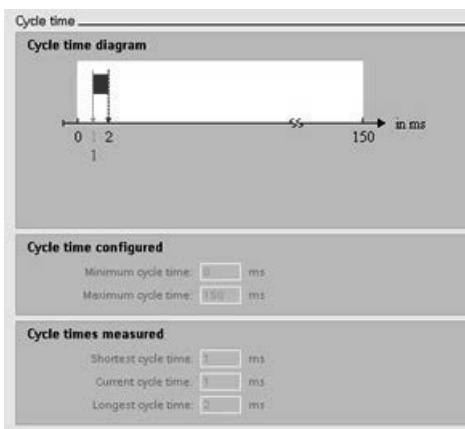
11.2. Interakcja z połączonym CPU

Karta zadań *Online tools* (Narzędzia online) w widoku projektu wyświetla panel operatorski, który pokazuje tryb pracy CPU online. Należy użyć przycisku na panelu operatorskim do zmiany trybu pracy CPU (RUN lub STOP). Ma on także przycisk MRES, który kasuje pamięć.

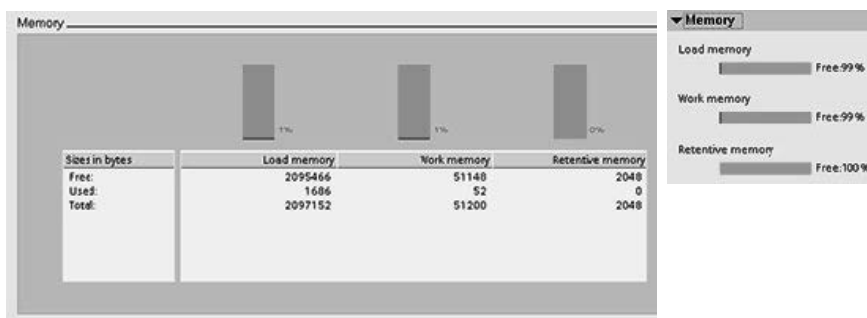


Kolor wskaźnika RUN/STOP pokazuje tryb pracy CPU: żółty wskazuje tryb STOP, zielony wskazuje tryb RUN.

Aby używać panelu operatorskiego, należy ustawić połączenie online pomiędzy programem STEP 7 a CPU. Po wybraniu CPU z konfiguracji urządzenia lub wyświetleniu bloku kodu z podłączonego CPU, użytkownik może wyświetlić panel operatorski z karty zadań *Online tools*.



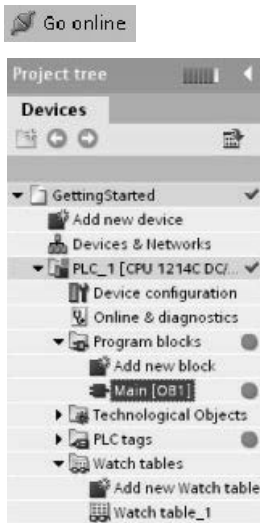
Można monitorować czas cyklu dla połączanego CPU.



Użytkownik może także mieć wgląd w zużycie pamięci przez CPU.

11.3. Połączenie online w celu monitorowania wartości w CPU

Aby monitorować zmienne (tagi) CPU musi mieć połączenie online. W tym celu wystarczy kliknąć przycisk *Go online* w pasku narzędzi.



Po podłączeniu CPU STEP 7 zmienia kolor nagłówków obszarów roboczych na pomarańczowy.

Drzewo projektu wyświetla porównanie pomiędzy projektem offline i połączonym CPU. Zielone kółko oznacza, że projekty z CPU i offline zostały zsynchronizowane. Oznacza to, że obydwa mają tę samą konfigurację i program użytkownika.

Tablica zmiennych pokazuje listę tagów/zmiennych. Mogą być one pokazywane także przez tablice monitorującą, także jako bezpośrednie adresy.

	Name	Address	Display format	Monitor value	Modify value
1	"On"	%I 0	Bool		
2	"Off"	%I 1	Bool		
3	"Run"	%Q0.0	Bool		

Należy nacisnąć w pasku narzędzi przycisk *Monitor all*, aby nadzorować wykonanie programu użytkownika i wyświetlić wartości zmiennych.

	Name	Address	Display format	Monitor value	Modify value
1	"On"	%I 0	Bool	FALSE	
2	"Off"	%I 1	Bool	FALSE	
3	"Run"	%Q0.0	Bool	FALSE	

Pole *Monitor value* pokazuje wartość każdej ze zmiennych.

11.4. Wyświetlanie stanu programu użytkownika

Użytkownik może nadzorować stan zmiennych w edytorach programu LAD i FBD. Należy użyć paska edytora aby wyświetlić edytor programu LAD. Pasek edytora pozwala na zmianę widoku pomiędzy otwartymi edytorami bez konieczności zamykania lub otwierania okien.

W pasku narzędzi edytora programu należy nacisnąć przycisk *Monitoring on/off*, aby wyświetlić stan programu użytkownika.





Sieć w programie użytkownika wyświetla przepływ zasilania na zielono.

Użytkownik może także, naciskając PPM na instrukcji lub parametrze, modyfikować wartości dla instrukcji.

11.5. Tablice monitorujące do monitorowania CPU

Tablica monitorująca (*watch table*) pozwala monitorować i modyfikować wybrane zmienne podczas wykonywania przez CPU programu użytkownika. Zmienne mogą być wejściami (I), wyjściami (Q), pamięcią M, blokiem DB lub peryferyjnymi wejściami (np. „On:P” lub „I 3.4:P”). Nie można dokładnie monitorować fizycznych wyjść (np. Q0.0:P), ponieważ funkcja monitorująca może wyświetlać tylko ostatnią wartość wpisaną z pamięci Q i nie odczytuje aktualnej wartości z fizycznych wyjść.

Obserwacja zmiennych (tagów) nie zmienia sekwencji programu. Prezentuje użytkownikowi informacje o sekwencji programu i danych programu w CPU. Użytkownik może użyć funkcji modyfikowania wartości do testowania wykonania swojego programu.

	Name	Address	Display format	Monitor value	Monitor with trigger	Modify with trigger	Modify value
1	*Start	%I0.0	Bool		Permanent	Permanent	<input type="checkbox"/>
2	*Stop	%I0.1	Bool		Permanent	Permanent	<input type="checkbox"/>
3	*Running	%M0.0	Bool		Permanent	Permanent	<input type="checkbox"/>

Uwaga

Cyfrowe punkty I/O używane przez szybkie liczniki (HSC) oraz urządzenia PWM i PTO są przypisywane podczas konfiguracji. Jeśli adresy cyfrowych I/O są przypisane do tych urządzeń, ich wartości nie mogą być modyfikowane przez funkcję *Force* z tabeli monitorującej.

Użytkownik określa kiedy monitorować lub modyfikować zmienną, wybierając jedną z następujących opcji:

- Na początku lub końcu cyklu programu.
- Po przejściu CPU w tryb STOP.
- „Na stałe” (wartość nie jest kasowana po przejściu w tryb RUN).



Aby stworzyć tablicę monitorującą:

1. Podwójnie kliknąć *Add new watch table* w celu otwarcia nowej tablicy monitorującej.
2. Wprowadzić nazwę zmiennej w celu jej dodania do tablicy monitorującej.

W celu monitorowania zmiennych CPU musi być podłączona w trybie *online*. Dostępne są następujące opcje modyfikowania zmiennych:

- *Modify now* natychmiast zmienia wartości dla wybranych adresów na jeden cykl programu.
- *Modify with trigger* zmienia wartości dla wybranych adresów.
W tej funkcji nie występuje sprzężenie zwrotne wskazujące, że wybrane adresy faktycznie zostały zmodyfikowane. Jeżeli konieczne jest potwierdzenie zmian, to należy stosować funkcję *Modify now*.
- *Enable peripheral outputs* blokuje rozkaz dezaktywacji wyjść i jest dostępna tylko wtedy, kiedy CPU jest w trybie STOP.

Różne funkcje można wybierać za pomocą przycisków znajdujących się na górze tablicy monitorującej.

W celu monitorowania zmiennej, należy wprowadzić jej nazwę oraz wybrać format wyświetlania z rozwijanej listy. W trybie połączenia *online* z CPU, kliknięcie *Monitor* spowoduje wyświetlenie faktycznych wartości punktów danych w polu *Monitor value*.

11.6. Użycie tablic wymuszeń

CPU umożliwia wymuszanie wartości na wejściach i wyjściach poprzez wyspecyfikowanie adresu wejścia lub wyjścia w tabeli monitorującej i zastosowanie wymuszenia. Wymuszenie jest stosowane do obszaru wejściowego obrazu procesu przed wykonaniem programu użytkownika i do obszaru wyjściowego obrazu procesu przed ustawieniem stanu wyjść modułów.

Uwaga

Wymuszone wartości przechowywane są w CPU, a nie w tabeli wymuszeń.

Nie można wymusić wejścia (lub adresu „I”) lub wyjścia (lub adresu „Q”). Można jednak wymusić wartość zewnętrznego adresu wejść lub wyjść. Tablica wymuszeń automatycznie dołącza do adresu „:P” (np.: „On”:P lub „Run”:P).

	F	F				
I	Name	Address	Display format	Monitor value	Force value	F
1	"On".P	%I0.0.P	Bool		TRUE	<input checked="" type="checkbox"/> 1
2	"Off".P	%I0.1.P	Bool			<input type="checkbox"/>
3	"Run".P	%Q0.1.P	Bool	0		<input type="checkbox"/>

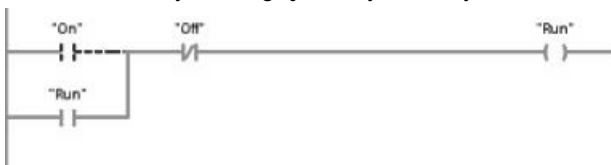
Wartość wejścia lub wyjścia, które ma zostać wymuszona, musi być wpisana w polu *Force value*. Użytkownik może użyć wtedy komórki sprawdzającej w kolumnie *Force*, aby włączyć wymuszenie wejścia lub wyjścia.



Aby wymusić wartość zmiennych w tablicy wymuszeń należy kliknąć przycisk *Start or replace forcing*. Do skasowania wartości wymuszenia, należy wcisnąć przycisk *Stop forcing*.

W tabeli wymuszeń użytkownik może nadzorować stan wymuszonych wartości wejść. Nie można jednak monitorować wymuszonych wartości wyjść.

Można także uzyskać wgląd do wymuszonych wartości w edytorze programu.



Uwaga

Kiedy wejście lub wyjście jest wymuszone w tablicy monitorującej, wtedy działanie wymuszające staje się częścią programu użytkownika. Nawet jeśli program zostanie zamknięty, to wybór wymuszeń pozostaje aktywny w działającym oprogramowaniu CPU dopóty, dopóki nie nastąpi ich wykasowanie poprzez przejście z oprogramowaniem sterującym w tryb *online* i zakończenie funkcji wymuszania.

CPU umożliwia wymuszenie wartości na wejściach i wyjściach poprzez określenie fizycznego adresu wejścia lub wyjścia (I_: P lub Q_: P) w tablicy wymuszającej, a następnie zastosowanie funkcji wymuszenia.

W programie, wartości odczytane z wejść fizycznych są zastępowane wartościami wymuszonymi. Program posługuje się w czasie pracy wartościami wymuszonymi. Kiedy program zapisuje stan wyjścia fizycznego, wtedy wartość wyjściowa jest zastępowana wartością wymuszoną. Na wyjściu fizycznym pojawia się wartość wymuszona i proces używa tej wartości.

Kiedy wejście lub wyjście jest wymuszone w tablicy monitorującej, wtedy działanie wymuszające staje się częścią programu użytkownika. Nawet jeśli program zostanie zamknięty, to wymuszenia pozostają aktywne w działającym oprogramowaniu CP aż nastąpi ich wykasowanie poprzez przejście w tryb *online* i zakończenie funkcji wymuszania. Programy z aktywnymi wymuszeniami załadowane do innego CPU z karty pamięci nadal będą wymuszać stany zmiennych w programie.

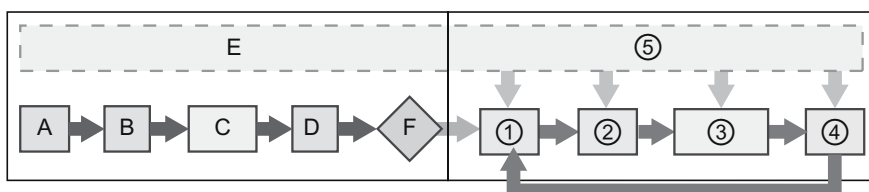
Jeśli program użytkownika jest wykonywany z karty pamięci zabezpieczonej przed zapisem, użytkownik nie może inicjować lub zmieniać wymuszenia I/O

z poziomu tablicy monitorującej, ponieważ nie może nadpisywać wartości w chronionym przed zapisem programie użytkownika. Każda próba wymuszenia wartości chronionej przed zapisem będzie skutkowałą błędem. Jeśli użytkownik używa karty pamięci do transferu programu użytkownika, każdy wymuszony element na tej karcie zostanie przeniesiony także do CPU.

Uwaga

Cyfrowe punkty I/O przypisane do HSC, PWM, oraz PTO nie mogą być wymuszone.

Cyfrowe punkty I/O używane przez szybkie liczniki (HSC) oraz urządzenia PWM i PTO są przypisywane podczas konfiguracji. Kiedy adresy cyfrowych I/O zostaną przypisane do tych urządzeń, wtedy wartości przypisanych adresów I/O nie mogą być zmienione przez funkcję wymuszenia tablicy wymuszającej.



Rozruch (Startup)

- A Funkcja wymuszania (*Force*) nie ma wpływu na kasowanie obszaru pamięci I.
- B Funkcja wymuszania nie ma wpływu na inicjalizację wartości wyjściowych.
- C Podczas wykonywania rozruchowych OB, CPU stosuje wartości wymuszone jeśli program użytkownika uzyskuje dostęp do fizycznych wejść.
- D Przechowywanie przerw w kolejce nie jest zakłócone.
- E Odblokowywanie zapisu do wyjść nie jest zakłócone.

RUN

- ① Podczas zapisywania obszaru pamięci Q do fizycznych wyjść, CPU uaktualniania stan wyjść wartościami wymuszonymi.
- ② Po skopiowaniu stanu wejść fizycznych do pamięci I, CPU stosuje wartości wymuszone.
- ③ Podczas wykonywania programu użytkownika (cyklicznych OB), CPU stosuje wartości wymuszone jeśli program użytkownika uzyskuje dostęp do fizycznych wejść.
- ④ Obsługa żądań komunikacyjnych i autodiagnostyka nie są zakłócone.
- ⑤ Obsługa przerw w dowolnej części cyklu programu nie jest zakłócona.

11.7. Przechwytywanie wartości online bloków DB do skasowania wartości początkowych

Użytkownik może przechwycić bieżące wartości, które są monitorowane w połączonym CPU i zdefiniować je jako wartości początkowe dla globalnego DB.

- CPU musi być podłączone online.
- CPU musi być w trybie RUN
- Użytkownik musi posiadać już otwarte DB w programie.



Użyj przycisku *Show a snapshot of the monitored values* (pokaż podgląd monitorowanych wartości), aby zachować bieżące wartości określonych zmiennych w DB. Dzięki temu będzie można kopiować te wartości z kolumny DB *Start value*.

1. W edytorze DB należy nacisnąć przycisk *Monitor all tags*. Kolumna *Monitor value* wyświetla bieżące wartości danych.
2. Kliknięcie przycisku *Show a snapshot of the monitored values* powoduje wyświetlenie bieżących wartości w kolumnie *Snapshot*.
3. Po kliknięciu przycisku *Monitor all* następuje zatrzymanie monitorowania danych w CPU.
4. Należy skopiować wartości z kolumny *Snapshot*.
 - Należy wybrać wartość do skopiowania.
 - Kliknięcie PPM na wybrana wartość spowoduje wyświetlenie menu kontekstowego.
 - Należy wybrać komendę *Copy*.
5. Należy wkleić skopiowane dane do odpowiedniej komórki kolumny *Start value* (kliknięcie PPM i wybranie z menu kontekstowego komendy *Paste*).
6. Należy zapisać projekt, aby skopiowane wartości zostały skonfigurowane jako wartości startowe dla DB.
7. Należy skompilować i ściągnąć blok DB do CPU. DB będzie używało nowych wartości startowych po ponownym przejściu CPU w tryb RUN.

Uwaga

Wartości pokazane w kolumnie *Monitor value* są zawsze skopiowane z CPU. STEP 7 nie sprawdza, czy wszystkie wartości pochodzą z tego samego cyklu programu.

11.8. Kopiowanie elementów projektu

Użytkownik może skopiować bloki programu z połączonego CPU lub karty pamięci dołączonej do urządzenia programującego.

Należy przygotować projekt *offline* dla skopiowanych bloków poprzez:

1. Dodanie urządzenia CPU zgodnego z połączonym online CPU.
2. Rozszerzenie węzła CPU tak, aby folder *Program blocks* był widoczny.



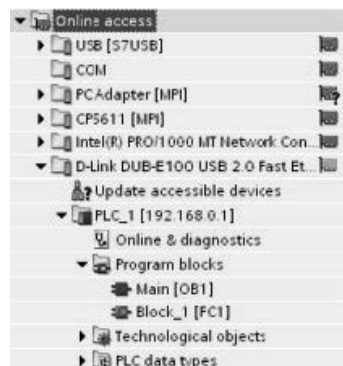
Aby przesłać bloki programu z CPU online do projektu offline, należy wykonać następujące kroki:

1. W projekcie offline kliknąć przycisk folder *Program blocks*.
2. Kliknąć przycisk *Go online*.
3. Kliknąć przycisk *Upload*.
4. Potwierdzić decyzję w oknie dialogowym *Upload*.



Zamiast poprzedniej procedury, można zastosować inną, według następujących kroków:

1. Należy z okna nawigatora projektu rozszerzyć węzeł *Online access*, aby wybrać blok programu w połączonym CPU.
2. Należy rozwinąć węzeł sieci i kliknąć dwukrotnie w *Update accessible devices*.
3. Następnie należy rozwinąć węzeł CPU.
4. Należy przeciągnąć folder *Program blocks* z połączonego online CPU i upuścić go w folderze *Program blocks* projektu offline.
5. W oknie dialogowym *Upload preview* należy wybrać boks *Continue*, a następnie kliknąć przycisk *Upload from device*.



Kiedy nastąpi zakończenie zgrzywania, wszystkie bloki programu, bloki technologiczne oraz zmienne będą wyświetlone w obszarze *offline*.

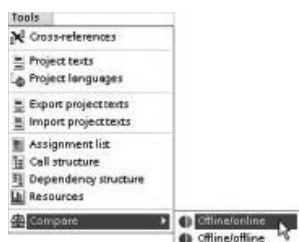


Uwaga

Użytkownik może skopiować bloki programu z połączonego online CPU do istniejącego programu. Folder *Program-blocks* projektu *offline* nie musi być pusty. Istniejący program zostanie jednak usunięty i zastąpiony przez program użytkownika z CPU połączonego online.

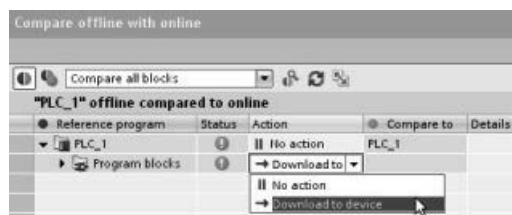
11.9. Porównywanie CPU w trybach offline i online

Istnieje możliwość porównania bloków kodu znajdujących się w połączonym CPU z blokami kodu w programie użytkownika. Jeśli bloki kodu z projektu nie pasują do bloków kodu CPU online, wtedy edytor *Compare* umożliwia zsynchronizować projekt z CPU online przez załadowanie bloków kodu z projektu do CPU, lub przez usunięcie bloków z projektu, które nie istnieją w CPU online.

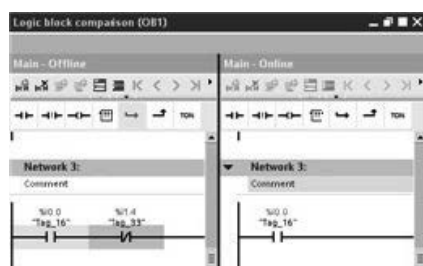


W projekcie należy wybrać CPU.

Użycie komend *Compare Offline/online* otwiera edytor *Compare* (dostęp do tej komendy może być uzyskany zarówno z menu *Tools* menu jak również poprzez kliknięcie PPM na CPU w projekcie użytkownika).



Kliknąć w kolumnie *Action* dla obiektu aby wybrać, czy obiekt usunąć, nie podjąć akcji, lub obiekt załadować do urządzenia.



Kliknięcie przycisku *Synchronize* powoduje zgranie programu z CPU.

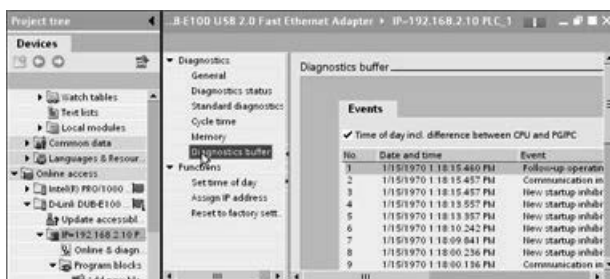
Kliknąć prawym przyciskiem myszy obiekt w kolumnie *Compare to* oraz wybrać przycisk *Start detailed comparison*, aby umieścić bloki kodu obok siebie.

Szczegółowe porównanie podświetla różnice pomiędzy blokami kodu w połączonym CPU i blokami kodu w projekcie użytkownika.

11.10. Wyświetlanie zdarzeń diagnostycznych

CPU posiada bufor diagnostyczny, który zawiera wpisy każdego zdarzenia diagnostycznego, takiego jak przełączenie trybu pracy CPU lub wykrycie błędów zarówno w CPU jak i w dołączonych modułach. Należy być w trybie online aby mieć dostęp do tego bufora.

Każdy wpis zawiera datę i czas wystąpienia zdarzenia, kategorię zdarzenia i jego opis. Wpisy są wyświetlane w porządku chronologicznym, ze zdarzeniami najświeższymi na górze listy.



Jeśli CPU jest zasilane, dostępne jest do 50 ostatnich zdarzeń w tym dzienniku. Jeśli dziennik jest przepełniony następuje nadpisanie najstarszych zdarzeń.

Po utracie zasilania zachowywane są ostatnie zdarzenia.

11.11. Ustawianie adresu IP i czasu dnia

Użytkownik może ustawić adres IP oraz czas dnia w połączonym CPU. Po otwarciu zakładki *Online & diagnostics* z drzewa projektu połączanego CPU, użytkownik może wyświetlić lub zmienić adres IP. Istnieje także możliwość wyświetlenia lub ustawienia parametrów czasu i daty dla połączanego CPU.



Uwaga

Ta operacja dostępna jest tylko dla CPU, które posiadają tylko adres MAC (nie został im przyznany adres IP), albo zostały przywrócone do ustawień fabrycznych.

11.12. Przywracanie ustawień fabrycznych

Sterownik oS7-1200 można zresetować do ustawień fabrycznych w następujących warunkach:

- W CPU nie ma wprowadzonej żadnej karty pamięci.
- CPU ma połączenie online.
- CPU jest w trybie STOP.

Uwaga

Jeżeli CPU jest w trybie RUN i rozpoczyna się operację resetowania (przywracania ustawień), to można umieścić go w trybie STOP po akceptacji monitu o potwierdzenie.

Procedura

Aby zresetować CPU do ustawień fabrycznych, należy wykonać następujące kroki:

1. Otworzyć okno widoku CPU *Online and Diagnostics* (Online i diagnostyka).
2. Z folderu *Functions* (Funkcje) wybrać opcję *Reset to factory settings* (Resetowanie do ustawień fabrycznych)
3. Zaznaczyć pole wyboru *Keep IP address* (Zachowaj adres IP), aby zachować adres IP, lub zaznaczyć pole wyboru *Reset IP address* (Resetuj adres IP), aby usunąć adres IP.
4. Kliknąć przycisk *Reset* (Resetuj).
5. Zaakceptować monit o potwierdzenie, klikając przycisk *OK*.

Wynik

Moduł jest przełączony do trybu STOP, jeśli to konieczne, i zresetowany do ustawień fabrycznych:

- Pamięć robocza, wewnętrzna pamięć ładowania, oraz wszystkie obszary argumentów są wyczyszczone.
- Wszystkie parametry są zresetowane do wartości domyślnych.
- Bufor diagnostyczny jest wyczyszczony.
- Zegar czasu rzeczywistego jest zresetowany.
- Adres IP jest zachowany lub usunięty, w zależności od wybranych ustawień. (Adres MAC (*Media Access Control*) jest stały i nigdy się nie zmienia).

11.13. Aktualizacja oprogramowania sprzętowego

Oprogramowanie sprzętowe (*firmware*) podłączonego CPU można zaktualizować za pomocą programu STEP 7 w trybie online oraz narzędzi diagnostycznych.

Aby przeprowadzić aktualizację oprogramowania sprzętowego, należy wykonać następujące kroki:

1. Otworzyć okno *Online and Diagnostics* (Online i diagnostyka) dla podłączonego CPU.
2. Z folderu *Functions* (Funkcje) wybrać opcję *Update firmware* (Aktualizacja firmware).
3. Kliknąć przycisk *Browse* (Przeglądaj) i przejść do lokalizacji, w której znajduje się plik aktualizacji firmware. Może to być miejsce na dysku twardym, gdzie został pobrany i umieszczony plik aktualizacji firmware dla S7-1200 (<http://support.automation.siemens.com/WW/view/en/34612486/133100>) z witryny internetowej serwisu i wsparcia technicznego (<http://www.siemens.com/automation/>).
4. Wybrać plik, który jest kompatybilny z modulem. W tabeli wykazane są kompatybilne moduły dla wybranego pliku.
5. Kliknąć przycisk *Run update* (Uruchom aktualizację). Należy postępować zgodnie z instrukcjami wyświetlanymi w oknach dialogowych, i w razie potrzeby, zmienić tryb pracy CPU.

W programie STEP 7 podczas aktualizacji firmware jest wyświetlany wskaźnik postępu. Po zakończeniu aktualizacji, pojawi się monit o ponowne uruchomienie modułu z nowym firmware.

Uwaga

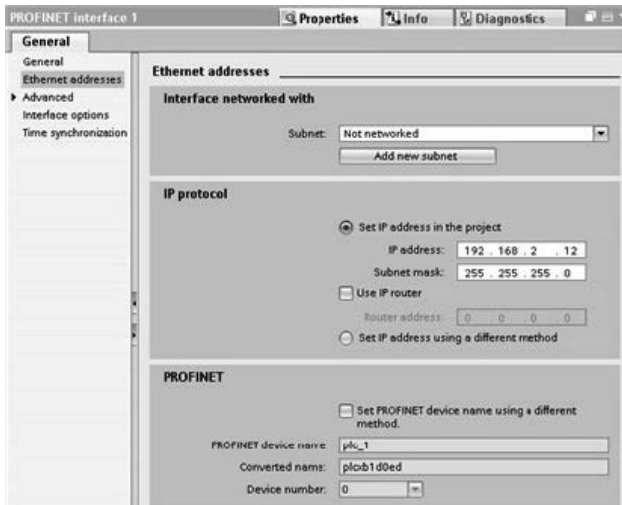
Jeśli użytkownik dokona wyboru, aby nie uruchamiać modułu z nowym firmware, to poprzedni firmware pozostanie aktywny do czasu zresetowania modułu, na przykład przez wyłączenie i ponowne włączenie zasilania. Nowe oprogramowanie firmware stanie się aktywne dopiero po zresetowaniu modułu.

Aktualizację firmware można również wykonać stosując jedną z następujących metod:

- użyć karty pamięci,
- użyć standardowej strony internetowej *Module Information* (Informacje o module) web serwera (strona 187).

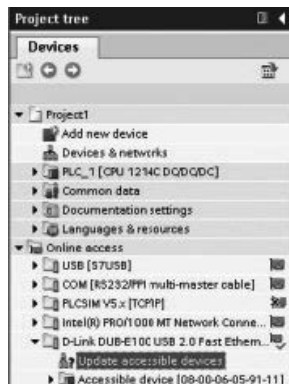
11.14. Załadowanie adresu IP do połączanego online CPU

Aby przypisać adres IP należy przeprowadzić następujące działania:



- Skonfigurowanie dla CPU adresu IP (strona 77).
- Zapisanie i załadowanie konfiguracji do CPU.

Adres IP i maska podsieci muszą być zgodne z adresem IP i maską podsieci urządzenia programującego. Należy skontaktować się z administratorem sieci w celu uzyskania adresu IP i maski podsieci dla CPU użytkownika.



Jeśli CPU jest konfigurowane po raz pierwszy, użytkownik może użyć funkcji *Online access*, aby ustawić adres IP.

Adres IP, który został ściągnięty jako część konfiguracji urządzenia nie zostanie utracony po utracie zasilania przez PLC.

Po ściągnięciu konfiguracji urządzenia wraz z adresem IP, użytkownik może zobaczyć adres IP w folderze *Online access*.

11.15. Użycie „nieznanego CPU” do automatycznej detekcji

Jeśli użytkownik ma do dyspozycji fizyczne CPU, które można podłączyć do urządzenia programującego, to istnieje możliwość łatwego załadowania konfiguracji sprzętowej.

Użytkownik musi najpierw połączyć CPU z urządzeniem programującym i stworzyć nowy projekt.

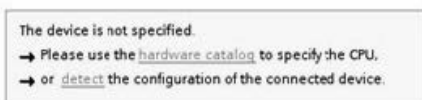
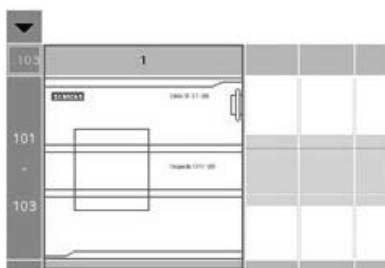


W konfiguracji urządzenia (widok projektu lub widok portalu) należy dodać nowe urządzenie, ale zaznaczając *unspecified CPU* zamiast wyboru określonego urządzenia. STEP 7 utworzy obiekt nieznanego CPU.



Po stworzeniu nieznanego urządzenia użytkownik może załadować konfigurację sprzętową dla połączonego CPU.

- W edytorze programu należy wybrać komendę *Hardware detection* z menu *online*.
- W edytorze konfiguracji urządzenia należy wybrać opcję wykrywania konfiguracji dla połączonego urządzenia.



Po wybraniu w oknie dialogowym odpowiedniego CPU, STEP 7 załaduje konfigurację sprzętową z CPU, włączając w to wszelkie moduły (SM, SB lub CM).

Adres IP nie zostanie zgrany. Użytkownik musi go ręcznie skonfigurować w oknie *Device configuration*.

11.16. Ładowanie w trybie RUN

CPU obsługuje funkcję ładowania w trybie RUN – *Download in RUN mode*. Funkcja ta ma umożliwić dokonywanie drobnych zmian w programie użytkownika przy minimalnym zakłóceniu procesu sterowanego przez program. Jednak implementacja tej funkcji pozwala również na ogromne zmiany w programie, które mogą być destrukcyjne, a nawet niebezpieczne.

OSTRZEŻENIE

Zagrożenia związane ładowaniem w trybie RUN

Podczas ładowania zmian do CPU w trybie RUN, zmiany natychmiast wpływają na działanie procesu. Zmiana programu w trybie RUN może spowodować nieoczekiwane działanie systemu, co może spowodować śmierć lub poważne zranienie personelu i/lub uszkodzenia mienia.

Ładowanie w trybie RUN powinno być wykonywane wyłącznie przez upoważnione osoby, które rozumieją wpływ wprowadzanych w trybie RUN zmian na działanie systemu.

Funkcja *Download in RUN mode* umożliwia dokonywanie zmian w programie oraz ładowanie ich do CPU bez konieczności przełączania do trybu STOP:

- Można dokonywać drobnych zmian w bieżącym procesie, bez konieczności jego zatrzymywania (na przykład, zmienić wartość parametru).
- Można szybciej debugować program (na przykład, odwrócić logikę zmieniając styki przekaźnika normalnie otwarte na normalnie zamknięte).

Użytkownik może wykonać następujące zmiany w blokach i tagach (zmiennych) programu, i następnie załadować je w trybie RUN:

- Tworzyć, nadpisywać i usuwać funkcje (FC), bloki funkcyjne (FB) oraz tabele tagów.
- Tworzyć, nadpisywać i usuwać bloki danych (DB) oraz bloki danych instancji dla bloków funkcyjnych (FB). Struktury DB można dodawać i ładować w trybie RUN. W zależności od ustawień konfiguracyjnych, CPU może przechowywać wartości istniejących tagów bloków oraz inicjalizować nowe tagi bloków wartościami początkowymi, lub też zresetować wartości wszystkich tagów bloków danych do wartości początkowych. W trybie RUN nie można ładować DB web serwera (sterowania lub fragmentu).
- Nadpisywać bloki organizacyjne (OB), przy czym nie można ich tworzyć lub usuwać.

W trybie RUN można ładować maksymalnie dziesięć bloków za jednym razem. W przypadku konieczności ładowania więcej niż dziesięciu bloków, wymagane jest przejście CPU do trybu STOP.

Jeśli mają być ładowane zmiany wprowadzane w rzeczywistym procesie (w odróżnieniu od symulowanego procesu, co można zrobić w trakcie debugowania programu), jest istotne, aby przed ładowaniem przemyśleć możliwe konsekwencje jakie to wywoła, związane z bezpieczeństwem maszyn i operatorów maszyn.

Uwaga

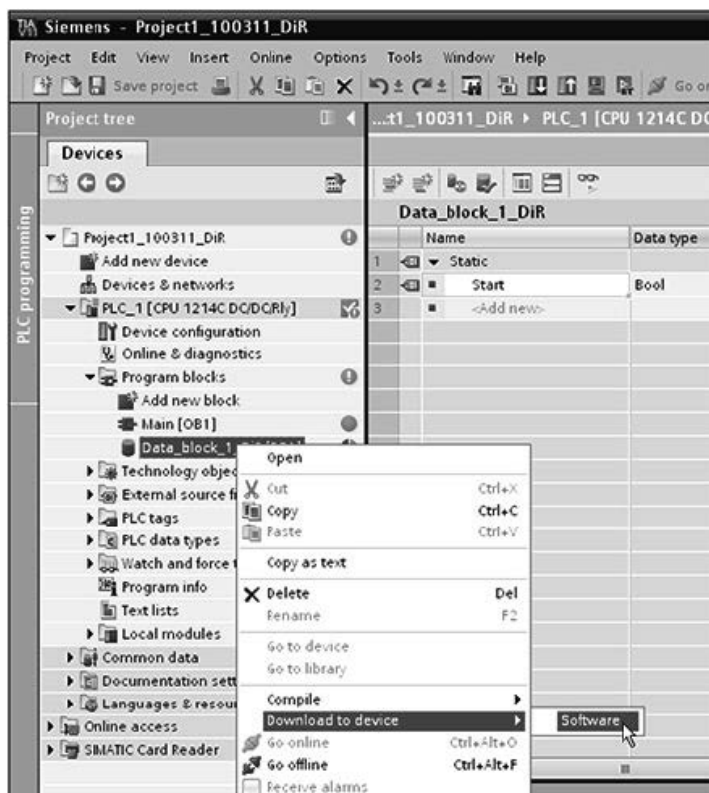
Jeśli CPU jest w trybie RUN i wprowadzono zmiany programu, to STEP 7 zawsze najpierw próbuje je załadować w trybie RUN. Jeśli nie jest to pożądane, należy przełączyć CPU do trybu STOP.

Jeśli wprowadzone zmiany nie są obsługiwane przez funkcję ładowania w trybie RUN, to STEP 7 monitoruje (informuje użytkownika), że CPU musi przejść do trybu STOP.

11.16.1. Zmiana programu w trybie RUN

Aby zmienić program w trybie RUN, najpierw trzeba się upewnić, czy CPU i program spełniają wymagania, a następnie wykonać następujące czynności:

1. Aby załadować program w trybie RUN, należy wybrać jedną z następujących metod:
 - użyć polecenia *Download to device* z menu *Online*,
 - użyć przycisku *Download to device* na pasku narzędzi,
 - kliknąć prawym przyciskiem myszy pozycję *Program blocks* w drzewie projektu i wybrać polecenie *Download to device > Software*.



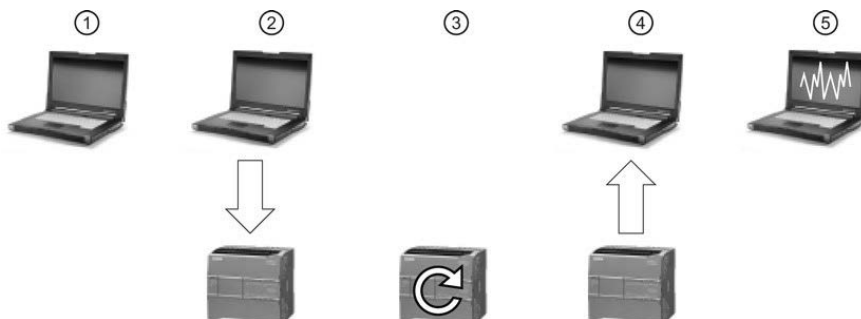
Jeśli program jest poprawnie skompilowany, to STEP 7 zacznie ładować program do CPU.

2. Gdy w STEP 7 pojawi się monit o ładowanie programu lub anulowanie operacji, to aby załadować program do CPU należy kliknąć przycisk *Load* (Ładuj).

11.17. Śledzenie i rejestrowanie zmiennych CPU wyzwalane zdarzeniowo

STEP 7 zapewnia funkcje śledzenia i analizator stanów logicznych, dzięki którym można skonfigurować zmienne PLC do ich śledzenia i rejestrowania. Następnie można załadować śledzone i zarejestrowane dane do urządzenia programującego i użyć narzędzi STEP 7 do analizy, przetwarzania i graficznego zobrazowania danych. Do gromadzenia śledzonych danych i zarządzania nimi wykorzystywany jest folder *Traces* znajdujący się w drzewie projektu programu STEP 7.

Na rysunku poniżej przedstawiono różne etapy funkcji *trace*:



- ① Konfiguracja funkcji *trace* w edytorze śledzenia STEP 7. Można skonfigurować dane do rejestrowania, czas rejestrowania, częstotliwość rejestracji oraz warunek wywołania.
- ② Przesłanie danych konfiguracji funkcji *trace* z programu STEP 7 do PLC.
- ③ PLC wykonuje program, i gdy następuje spełnienie warunku wywołania rejestracji, rozpoczyna się rejestrowanie śledzonych danych.
- ④ Przesłanie zarejestrowanych danych z PLC do STEP 7.
- ⑤ Użycie narzędzi w STEP 7 do analizy danych, przedstawienia danych w formie graficznej i zapisania w pamięci.

Należy pamiętać, że funkcja *trace* jest dostępna od wersji V4.0 CPU S7-1200.

W systemie pomocy programu STEP 7 można znaleźć szczegółowe informacje na temat konfiguracji funkcji *trace*, ładowania danych konfiguracji, gromadzenia danych śledzenia oraz wyświetlenia danych za pomocą analizatora stanów logicznych.

12.1. Przegląd technologii IO-Link

IO-Link jest zdefiniowaną przez Organizację Użytkowników Sieci PROFIBUS (PNO) innowacyjną technologią komunikacyjną dla czujników i urządzeń wykonawczych (aktuatorów). IO-Link jest międzynarodowym standardem, zgodnie z normą IEC 61131-9. Opiera się ona na połączeniu PtP (punkt-punkt) pomiędzy czujnikami i aktuatorami – urządzeniami podrzędnymi (*Slave*), a sterownikiem – urządzeniem nadrzędnym (*Master*). Zatem, takie rozwiązanie komunikacji nie reprezentuje systemu magistrali, a raczej pewne rozwinięcie tradycyjnego połączenia punkt-punkt.

Przez podłączone czujniki i urządzenia wykonawcze, oprócz cyklicznie wymienianych danych operacyjnych, są przesyłane duże ilości parametrów i danych diagnostycznych. Do transmisji danych wykorzystuje się ten sam 3-przewodowy kabel połączeniowy, który jest używany w standardowej technice łączenia czujników.

12.2. Elementy systemu IO-Link

System IO-Link składa się z urządzeń IO-Link (zwykle czujniki, urządzenia wykonawcze, lub ich kombinacje), standardowego 3-przewodowego kabla oraz urządzenia IO-Link Master. Urządzenie Master (nadrzędne) może być urządzeniem o dowolnej konstrukcji i o dowolnym stopniu ochrony.

Urządzenie IO-Link Master może mieć jeden port lub więcej portów. Urządzenie SM 1278 4xIO-Link Master ma cztery porty. Do każdego portu można podłączyć jedno urządzenie IO-LINK lub jeden standardowy czujnik lub jedno urządzenie wykonawcze. IO-Link jest systemem komunikacji punkt-punkt.

12.3. Po włączeniu zasilania

Przy włączeniu zasilania, urządzenie IO-Link jest zawsze w trybie SIO (standardowy tryb I/O). Porty urządzenia Master mogą mieć różne konfiguracje. Więcej informacji znajduje się w opisie IO-Link w podręczniku systemu S7-1200.

Jeśli port jest ustawiony w trybie SIO, to urządzenie Master traktuje ten port jak normalne wejście cyfrowe. Jeśli port jest ustawiony w trybie IO-Link (tryb komunikacji), to urządzenie Master próbuje znaleźć podłączone urządzenie IO-Link. Proces ten nazywany jest pobudzaniem (*wake-up*).

Podczas procesu pobudzania, urządzenie master wysyła określony sygnał i czeka na odpowiedź urządzenia Slave. Początkowo urządzenie Master próbuje to robić przy możliwie największej szybkości transmisji danych. Jeśli to się nie powiedzie, urządzenie Master próbuje poszukiwanie przy kolejno mniejszej szybkości transmisji danych. Master próbuje przesłać sygnał do urządzenia trzy razy przy każdej

szybkości transmisji. Urządzenie Slave zawsze działa tylko przy jednej, określonej szybkości transmisji danych. Jeśli urządzenie Master otrzyma odpowiedź (to jest, jeśli urządzenie Slave zostanie pobudzone), to wtedy obydwie urządzenia rozpoczynają komunikację. Na początku wymieniane są parametry komunikacyjne, a następnie rozpoczyna się cykliczna wymiana danych procesowych.

Jeśli urządzenie Slave zostanie odłączone podczas pracy, to urządzenie Master wykrywa przerwanie komunikacji, raportuje to do sterownika zgodnie ze specyfikacją Fieldbus i próbuje ponownie cyklicznie pobudzić urządzenie. Po kolejnym udanym pobudzeniu, parametry komunikacyjne są ponownie odczytywane, w odpowiednich przypadkach potwierdzane, i następnie cykliczna wymiana danych rozpoczyna się ponownie.

12.4. Protokół IO-Link

W systemie IO-Link mogą być wymieniane trzy typy danych:

- cykliczne dane procesowe (dane procesowe na wejściach i wyjściach) → dane cykliczne,
- parametry urządzeń (obiekty danych na żądanie) → dane acykliczne,
- zdarzenia → dane acykliczne.

Urządzenie IO-Link wysyła dane tylko na żądanie urządzenia IO-Link Master. Dane procesowe są przesyłane po ramce IDLE wysłanej przez urządzenie Master, natomiast dane parametrów i zdarzeń są przysyłane na jawne żądanie urządzenia Master.

12.5. Konfiguracja Fieldbus

Urządzenie IO-Link Master w sieci Fieldbus funkcjonuje jak zwykły węzeł Fieldbus i jest zintegrowane przez odpowiedni opis urządzenia w odpowiednim konfiguratorze sieci. Tworzone pliki opisują właściwości komunikacyjne i inne parametry IO-Link Master, np. liczbę portów. One nie wykazują, które urządzenia IO-Link są podłączone.

Opis urządzeń IO-Link (IODD – *IO-Link Device Description*) umożliwia jednak uzyskanie w pełni przejrzystej reprezentacji architektury systemu, uwzględniając pojedyncze urządzenia IO-Link. Z pomocą IODD i narzędzia konfiguracji IO-Link S7-PCT, można skonfigurować które urządzenie IO-Link ma być podłączone do którego portu urządzenia IO-Link Master.

Szczegółowe informacje na temat konfiguracji można znaleźć w systemie pomocy S7-PCT oraz w podręczniku systemu S7-1200.

12.6. IO-Link w programie STEP 7

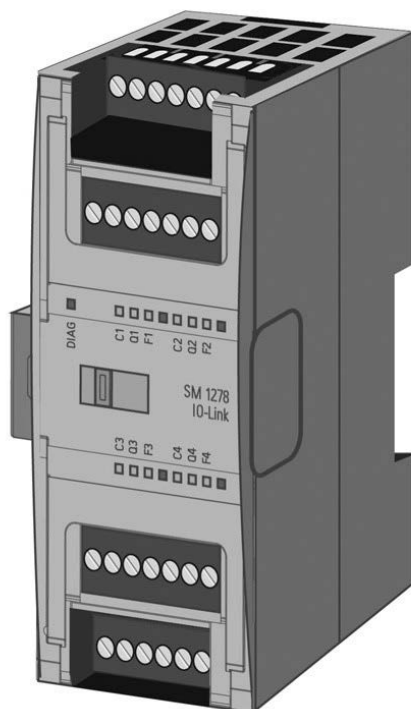
Urządzenie IO-Link Master programuje komunikację acykliczną z urządzeniem IO-Link za pomocą bloku funkcyjnego (FB) wywołania IOL_CALL w programie sterownika S7-1200 w środowisku STEP 7. Blok funkcyjny IOL_CALL wskazuje

używane w programie urządzenie IO-Link Master, oraz które porty urządzenia Master są używane do wymiany danych.

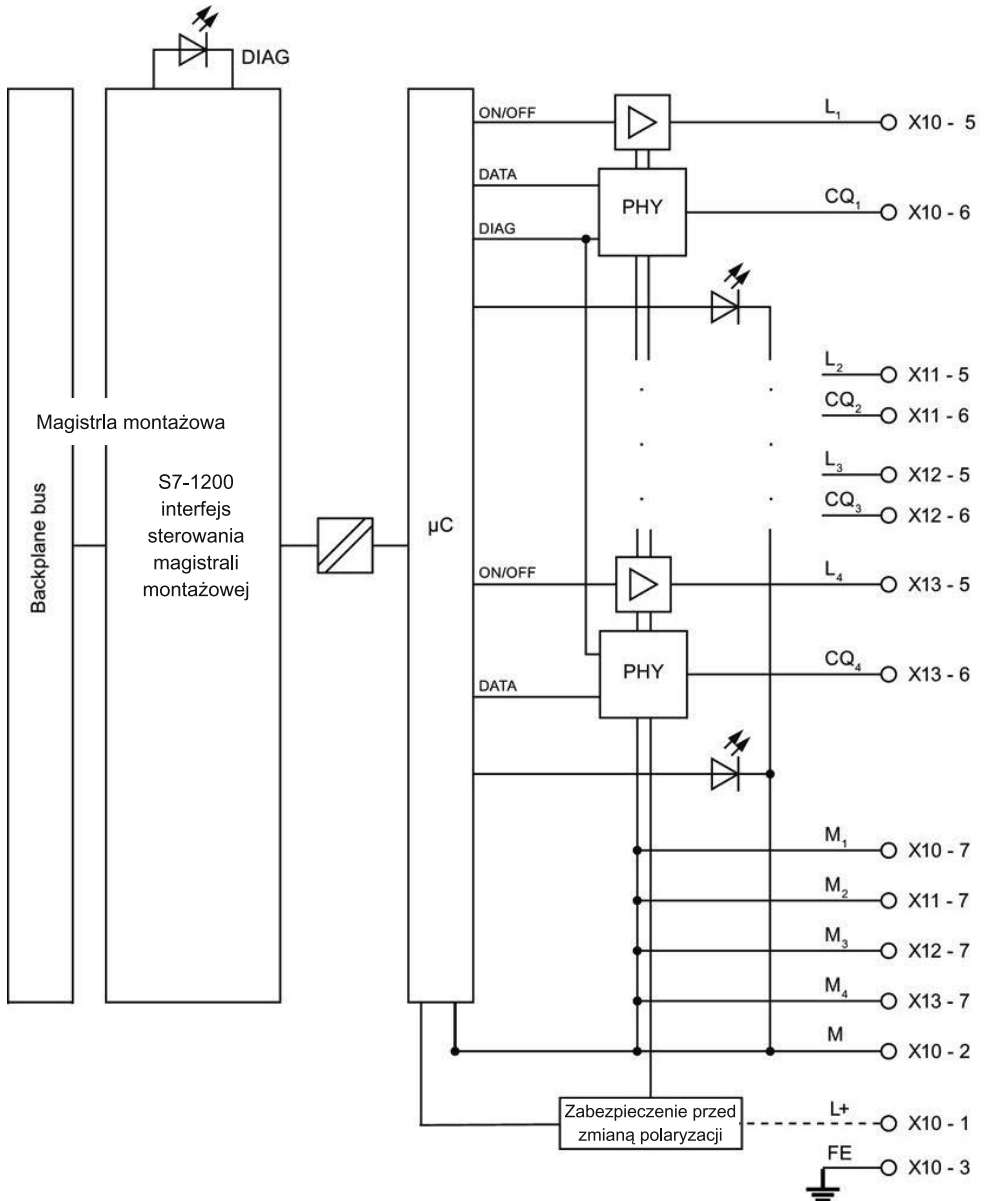
Na stronie pomocy technicznej online firmy Siemens (<http://support.automation.siemens.com>) znajdują się szczegółowe informacje na temat używania bloku funkcyjnego (FB) IOL_CALL. Aby uzyskać dostęp do informacji o produktach IO-Link i ich wykorzystaniu, w polu wyszukiwania na stronie internetowej serwisu należy wprowadzić „IO-Link”.

12.7. SM 1278 4xIO-Link Master

Urządzenie SM 1278 4xIO-Link Master to 4-portowy moduł, który działa zarówno jako moduł sygnałowy (rozszerzeń), jak i moduł komunikacyjny. Każdy port może działać w trybie IO-Link, jako pojedyncze wejście cyfrowe 24 VDC lub wyjście cyfrowe 24 VDC. Można podłączyć do czterech urządzeń IO-Link (połączenie 3-przewodowe) lub do czterech standardowych urządzeń wykonawczych (aktuatorów) lub standardowych enkoderów.

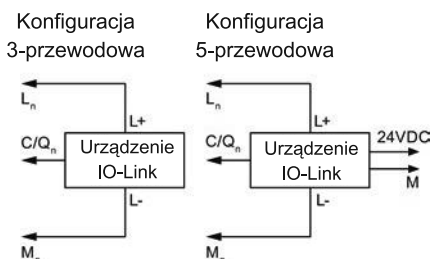


Schemat blokowy SM 1278 4xIO-Link Master

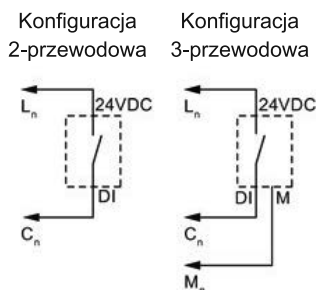


Przykłady połączeń

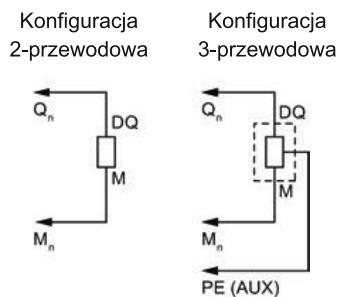
Na rysunku poniżej przedstawiono konfiguracje dla trybu pracy IO-Link (3-przewodowa i 5-przewodowa), gdzie n = numer portu:



Na rysunku poniżej przedstawiono konfiguracje trybu pracy DI (2-przewodowej i 3-przewodowej), gdzie n = numer portu:



Na rysunku poniżej przedstawiono konfiguracje trybu pracy DQ (2-przewodowej i 3-przewodowej), gdzie n = liczba portów:



Dodatkowe informacje na temat używania i konfigurowania SM 1278 4xIO-Link Master

Szczegółowe informacje na temat urządzenia SM 1278 4xIO-Link Master, w tym obejmujące schematy, połączenia, parametryzację i alarmy diagnostyczne, znajdują się w podręczniku systemu S7-1200.

Dane techniczne

A

A.1. Dane techniczne ogólne

Zgodność z normami

Struktura systemu automatyki S7-1200 jest zgodna z podanymi niżej normami i procedurami testowymi. Kryteria testowania systemu automatyki S7-1200 są oparte o te normy i procedury testowe.

Należy pamiętać, że nie wszystkie modele S7-1200 mogą być certyfikowane według tych norm, i że status certyfikacji może ulec zmianie bez powiadomienia. To użytkownik jest odpowiedzialny za ustalenie mającej zastosowanie certyfikacji, na podstawie etykiet (tabliczek znamionowych) umieszczonych na produkcie. Jeśli potrzebne są dodatkowe informacje dotyczące aktualnych wykazów zatwierdzeń według numeru części, należy skontaktować się z lokalnym przedstawicielem firmy Siemens.

Zatwierdzenie CE



System automatyki S7-1200 spełnia wymagania oraz zapewnia stopień bezpieczeństwa zgodnie z wymienionymi niżej dyrektywami EC (Wspólnoty Europejskiej), a także jest zgodny ze zharmonizowanymi normami europejskimi (EN) dotyczącymi sterowników programowalnych, wymienionymi w *Official Journals of the European Community*.

- EC Directive 2006/95/EC (dyrektywa niskonapięciowa) „Electrical Equipment Designed for Use within Certain Voltage Limits”
 - EN 61131-2:2007 Sterowniki programowalne – wymagania i testy sprzętu.
- EC Directive 2004/108/EC (dyrektywa EMC) „Electromagnetic Compatibility”
 - Norma emisyjności EN 61000-6-4:2007: Środowisko przemysłowe.
 - Norma odporności EN 61000-6-2:2005: Środowisko przemysłowe.
- EC Directive 94/9/EC (ATEX) „Equipment and Protective Systems Intended for Use in Potentially Explosive Atmosphere” – EN 60079-15:2005: Typ ochrony „n”. Deklaracja zgodności CE jest przechowywana i udostępniana właściwym władzom w:

Siemens AG
IA AS RD ST PLC Amberg
Werner-von-Siemens-Str. 50
D92224 Amberg
Germany

Zatwierdzenie cULus



System automatyki S7-1200 jest zgodny z normami:

- Underwriters Laboratories Inc.: UL 508 Listed (przemysłowe urządzenia sterujące).
- Canadian Standards Association: CSA C22,2 Number 142 (sprzęt sterowania procesami).

Uwaga

Seria SIMATIC S7-1200 spełnia normę CSA. Logo cULus oznacza, że S7-1200 był przebadany i jest certyfikowany przez Underwriters Laboratories (UL) na zgodność z normami UL 508 i CSA 22,2 No. 142.

Zatwierdzenie FM



Factory Mutual Research (FM):

Approval Standard Class Number 3600 i 3611.

Zatwierdzony do zastosowania zgodnie z:

Class I, Division 2, Gas Group A, B, C, D, Temperature Class T3C Ta = 60°C.

Class I, Zone 2, IIC, Temperature Class T3 Ta = 60°C.

Canadian Class I, Zone 2 Installation per CEC 18-150

WAŻNY WYJAŚTENIE: Liczbę wejść i wyjść, które mogą być włączone w tym samym czasie, można znaleźć w danych technicznych. Dla niektórych modeli ustalono Ta = 60°C.



Ostrzeżenie

Wymiana elementów może naruszyć zgodność z Class I, Division 2 oraz Zone 2.

Naprawy urządzeń powinny być wykonywane wyłącznie przez autoryzowane Centrum Serwisowe firmy Siemens.

Zatwierdzenie ATEX



Zatwierdzenie ATEX dotyczy tylko modeli DC. Zatwierdzenie ATEX nie dotyczy modeli AC i z przekaźnikiem.

EN 60079-0:2009: Środowisko wybuchowe – wymagania ogólne,

EN 60079-15:2010: Aparatura elektryczna do pracy w środowisku narażonym na zagrożenie wybuchem.

Typ zabezpieczenia „nA”

II 3 G Ex nA IIC T4 lub T3 Gc

Moduły muszą być instalowane w odpowiednich obudowach o stopniu zabezpieczenia co najmniej IP54 zgodnie z normą EN 60529, lub w lokalizacji zapewniającej równoważny stopień zabezpieczenia.

Dołączane kable i przewody powinny być dostosowane do rzeczywistej temperatury mierzonej w warunkach znamionowych.

W instalacji musi być zapewnione ograniczenie krótkotrwałych wahań napięcia do mniej niż 119 V. Patrz w tej części „Odporność na przepięcia”.

WAŻNY WYJĄTEK: Liczbę wejść i wyjść, które mogą być włączone w tym samym czasie, można znaleźć w danych technicznych. Dla niektórych modeli ustalono $T_a = 60^{\circ}\text{C}$.

Zatwierdzenie C-Tick



System automatyki S7-1200 spełnia wymagania norm AS/NZS 2064 (Class A).

Certyfikacja koreańska



System automatyki S7-1200 spełnia wymagania certyfikacji koreańskiej KC (Korean Certification). Został on sklasyfikowany jako sprzęt klasy A (Class A Equipment) przeznaczony do zastosowań przemysłowych, i nieprzewidziany do użytku osobistego.

Certyfikaty morskie

Urządzenia z rodziny S7-1200 są regularnie poddawane zatwierdzeniom przez różne instytucje specjalne, właściwe dla specyficznych rynków i aplikacji. W celu uzyskania dodatkowych informacji dotyczących dokładnej listy uzyskanych ostatnich zatwierdzeń sporządzonej według numerów części, należy skontaktować się z lokalnym przedstawicielem firmy Siemens.

Urzędy klasyfikacyjne:

- ABS (American Bureau of Shipping)
- BV (Bureau Veritas)
- DNV (Det Norske Veritas)

- GL (Germanischer Lloyd)
- LRS (Lloyds Register of Shipping)
- Class NK (Nippon Kaiji Kyokai)
- Korean Register of Shipping

Środowiska przemysłowe

System automatyki S7-1200 jest zaprojektowany do zastosowania w środowiskach przemysłowych.

Tabela A.1. Środowiska przemysłowe

Zakres zastosowania	Wymagania dotyczące emisyjności zakłóceń	Wymagania dotyczące odporności na zakłócenia
Przemysłowe	EN 61000-6-4:2007	EN 61000-6-2:2005

Kompatybilność elektromagnetyczna

Kompatybilność elektromagnetyczna (EMC) jest zdolnością urządzenia elektrycznego do zgodnej z założeniami pracy w środowisku elektromagnetycznym, bez emitowania takich poziomów zakłóceń elektromagnetycznych (EMI), które mogłyby zaburzyć pracę innych, pobliskich urządzeń elektromagnetycznych.

Tabela A.2. Odporność wg EN 61000-6-2

Kompatybilność elektromagnetyczna – odporność wg EN 61000-6-2	
EN 61000-4-2 Wyładowanie elektrostatyczne	8 kV wyładowanie przez powietrze do wszystkich powierzchni. 6 kV wyładowanie kontaktowe do odsłoniętych powierzchni przewodzących.
EN 61000-4-3 Promieniowane pole elektromagnetyczne	Od 80 do 100 MHz, 10 V/m, 80% AM przy 1 kHz. Od 1,4 do 2,0 GHz, 3 V/m, 80% AM przy 1 kHz. Od 2,0 do 2,7 GHz, 1 V/m, 80% AM przy 1 kHz.
EN 61000-4-4 Szybkie impulsy przejściowe	2 kV, 5 kHz z obwodem sprzęgającym do zasilania AC i DC systemu. 2 kV, 5 kHz z zaciskami sprzęgającymi do I/O.
EN 61000-4-5 Odporność na przepięcia	Systemy AC – 2 kV w trybie współbieżnym, 1 kV w trybie różnicowym. Systemy DC – 2 kV w trybie współbieżnym, 1 kV w trybie różnicowym. Dla systemów DC, patrz poniżej „Odporność na przepięcia”
EN 61000-4-6 Zakłócenia przewodzone	Od 150 kHz do 80 MHz, 10 V RMS, 80% AM przy 1 kHz
EN 61000-4-11 Spadki napięć	Systemy AC 0% dla 1 cyklu, 40% dla 12 cykli i 70% dla 30 cykli przy 60 Hz.

Odporność na przepięcia

Instalacje elektryczne narażone na przepięcia od sprzężeń powstających w wyniku uderzenia piorunem muszą mieć zewnętrzne zabezpieczenie. Specyfikacja do oceny ochrony przed przepięciami powstającymi w wyniku wyładowań atmosferycznych znajduje się w normie EN 61000-4-5, z pewnymi ograniczeniami eksploatacyjnymi określonymi w normie EN 61000-6-2. Jednostki CPU S7-1200 i moduły sygnałowe DC wymagają zewnętrznego zabezpieczenia w celu zapewnienia bezpiecznej pracy, gdy wystąpią napięcia udarowe określone w tej normie.

Poniżej wykazane są niektóre urządzenia spełniające wymagania ochrony przed przepięciami. Urządzenia te zapewniają ochronę tylko wtedy, gdy są zainstalowane poprawnie i zgodnie z zaleceniami producenta. Urządzenia produkowane przez innych producentów o tych samych lub lepszych parametrach mogą być również wykorzystywane:

- Napięcie zasilające, 24 VDC: DEHN, Inc., Typ BVT AD 24, Numer katalogowy 918 402
- Wejścia cyfrowe, 24 VDC: DEHN, Inc., Typ DCO RK E 24, Numer katalogowy 919 988
- Wyjścia cyfrowe, 24 VDC: DEHN, Inc., Typ DCO RK E 24, Numer katalogowy 919 988
- Wejścia analogowe: DEHN, Inc., Typ BXT ML4 BD12, Numer katalogowy 920 342
- Wyjścia analogowe: DEHN, Inc., Typ DCO RK E 12, Numer katalogowy 919 987
- Ethernet: DEHN, Inc., Typ DPA M CAT6 RJ45S 48, Numer katalogowy 929 100
- RS-232: DEHN, Inc., Typ DCO RK E 12, Numer katalogowy 919 987
- RS-485: DEHN, Inc., Typ BXT ML2 BE HFS 5, Numer katalogowy 920 270
- Wyjścia przekaźnikowe: nie wymagane

Tabela A.3. Emisyjność przewodzona i promieniowana wg EN 61000-6-4

Kompatybilność elektromagnetyczna – emisyjność przewodzona i promieniowana wg EN 61000-6-4		
Emisyjność przewodzona EN 55011, Class A, Group 11	Od 0,15 MHz do 0,5 MHz	<79 dB (µV) quasi-pik; <66 dB (µV) średnio
	Od 0,5 MHz do 5 MHz	<73 dB (µV) quasi-pik; <60 dB (µV) średnio
	Od 5 MHz do 30 MHz	<73 dB (µV) quasi-pik; <60 dB (µV) średnio
Emisyjność promieniowana EN 55011, Class A, Group 1	Od 30 MHz do 230 MHz	<40dB (µV/m) quasi-pik; pomiar w odległości 10 m
	Od 230 MHz do 1 GHz	<47dB (µV/m) quasi-pik; pomiar w odległości 10 m

Warunki środowiskowe

Tabela A.4. Transport i magazynowanie

Warunki środowiskowe – transport i magazynowanie	
EN60068-2-2, Test Bb, sucho i gorąco EN60068-2-1, Test Ab, zimno	Od -40°C do +70°C
EN60068-2-30, Test Db, wilgotno i gorąco	Od 25°C do 55°C, wilgotność 95%
EN60068-2-14, Test Na, szok temperaturowy	Od -40°C do +70°C czas przebywania 3 godziny, 2 cykle
EN 60068-2-32, swobodny upadek	0,3 m, 5 razy, produkt zapakowany
Ciśnienie atmosferyczne	Od 1080 do 660 hPa (odpowiadające wysokości od 1000 do 3500 m)

Tabela A.5. Warunki pracy

Warunki środowiskowe – praca	
Zakres temperatury otoczenia (wlot powietrza 25 mm poniżej urządzenia)	Od -20°C do 60°C montaż poziomy Od -20°C do 50°C montaż pionowy wilgotność bez kondensacji 95% o ile nie określono inaczej
Ciśnienie atmosferyczne	Od 1080 do 795 hPa (odpowiadające wysokości od 1000 do 2000 m)
Stężenie zanieczyszczeń	SO ₂ : <0,5 ppm; H ₂ S: <0,1 ppm; RH<60% bez kondensacji
EN60068-2-14, Test Nb, zmiana temperatury	Od 5°C do 55°C, 3°C/minutę
EN60068-2-27 udar mechaniczny	15 G, impuls 11 ms, 6 udarów w każdym z 3 kierunków
EN60068-2-6 wibracje sinusoidalne	Montaż na szynie DIN: 3,5 mm od 5 do 9 Hz; 1 G od 9 – 150 Hz Montaż na panelu: 7,00 mm od 5 do 9 Hz; 2 G od 9 do 150 Hz 10 odchyleń w każdej osi, 1 oktawa/minutę

Tabela A.6. Test izolacji wysokim napięciem

Test izolacji wysokim napięciem	
Obwody o napięciu znamionowym 24 VDC/5 VDC	520 VAC (test typu granic izolacji optycznej) 1500 VAC
Obwody 115 VAC / 230 VAC do uziemienia	1500 VAC
Obwody 115 VAC / 230 VAC do obwodów 115 VAC/230 VAC	1500 VAC (3000 VAC/4242 VDC test typu) 1500 VAC (tylko test typu)
Obwody 115 VAC/230 VAC do obwodów 24 VDC/5 VDC	
Port Ethernet do obwodów 24 VDC/5 VDC i uziemienia ¹	

¹ Izolacja portu Ethernet ma na celu ograniczenie zagrożeń począwszy od krótkich awarii sieciowych do niebezpiecznych napięć. Ona nie jest zgodna z wymaganiami bezpieczeństwa stawianymi standardowej izolacji linii napięcia zmiennego AC.

Klasa zabezpieczenia

- Protection Class II zgodnie z EN 61131-2 (Przewód zabezpieczający nie musi być podłączony).

Stopień zabezpieczenia

- Zabezpieczenie mechaniczne IP20, EN 60529.
- Zabezpieczenie przeciwko dotknięciu palcami wysokiego napięcia, zgodnie z testami standardową sondą. Wymagane jest zabezpieczenie zewnętrzne w przypadku kurzu, brudu, wody i ciał obcych o średnicy < 12,5 mm.

Napięcia znamionowe

Tab. A7. Napięcia znamionowe

Napięcie znamionowe	Tolerancja
24 VDC	Od 20,4 VDC do 28,8 VDC Od 22,0 VDC do 28,8 VDC dla temperatury z przedziału -20°C...0°C
120/230 VAC	Od 85 VAC do 264 VAC, od 47 do 63 Hz

Uwaga

Kiedy przełącznik mechaniczny załącza zasilanie CPU S7-1200 lub dowolnego cyfrowego modułu sygnałowego, wtedy przez okres około 50 mikrosekund na ich wyjściach cyfrowych ustawia się stan „1”. Użytkownik musi to uwzględnić, zwłaszcza gdy używa urządzeń reagujących na impulsy o krótkim czasie trwania.

Zabezpieczenie przed odwróceniem polaryzacji

Obwody zabezpieczenia przed odwróceniem polaryzacji występują na każdej parze zacisków napięcia zasilania +24 VDC, lub napięcia zasilania dla jednostek CPU, modułów sygnałowych (SM), oraz płytek sygnałowych (SB). Nadal jednak

możliwe jest uszkodzenie systemu przez podłączenie różnych par zacisków z odwróceniem polaryzacji.

Niektóre fragmenty systemu S7-1200 zasilane napięciem 24 VDC są ze sobą połączone poprzez wspólne łączenie wielu wyprowadzeń M. Na przykład, następujące układy są ze sobą połączone, gdy w karcie katalogowej są oznaczone jako „nieizolowane”: zasilacz wewnętrzny CPU 24 VDC, zasilacz czujników CPU, wejście zasilające uzwojenie przekaźnika SM, oraz zasilacz nieizolowanych wejść analogowych. Wszystkie nieizolowane zaciski M muszą być połączone do tego samego zewnętrznego potencjału odniesienia.

 **OSTRZEŻENIE**

Połączenie nieizolowanych wyprowadzeń M do różnych potencjałów odniesienia spowoduje nieplanowany przepływ prądów mogący doprowadzić do uszkodzenia lub nieprzewidywalnego zachowania PLC i podłączonych do niego urządzeń.

Nie zastosowanie się do podanych tu zaleceń może spowodować uszkodzenia i nieprzewidywalne działanie, które grozi śmiercią lub poważnym zranieniem personelu i/lub zniszczeniem mienia.

Koniecznym jest upewnić się, że nieizolowane zaciski M systemu S7-1200 są podłączone do tego samego potencjału odniesienia.

Wyjścia stałoprądowe (DC)

Obwód zabezpieczenia przeciwzwarceniowego nie występuje na wyjściach stałoprądowych (DC) jednostek CPU, modułów sygnałowych (SM) i płytek sygnalizacyjnych (SB).

Żywotność przekaźników elektrycznych

Typowe dane eksploatacyjne dla przekaźników, które zostały oszacowane na podstawie przykładowych testów, są przedstawione na rysunku poniżej. Rzeczywiste parametry mogą się zmieniać w zależności od konkretnej aplikacji. Zewnętrzny układ zabezpieczający, dostosowany do obciążenia wydłuża czas życia styków. Styki normalnie zwarte (NC) mają typową żywotność równą około jednej trzeciej żywotności styków normalnie rozwartych (NO) w warunkach obciążenia indukcyjnego i lampowego.

Zewnętrzny obwód ochronny zwiększa żywotność styków.

Tabela A.8. Typowe dane eksploatacyjne

Dane do wyboru urządzenia wykonawczego			
Ciągły prąd termiczny	2 A maks.		
Możliwości przełączania oraz żywotność styków			
Dla obciążenia rezystancyjnego	Napięcie	Prąd	Liczba cykli przełączania (typowo)
	24 VDC	2,0 A	0,1 million
	24 VDC	1,0 A	0,2 million
	24 VDC	0,5 A	1,0 million
	48 VAC	1,5 A	1,5 million
	60 VAC	1,5 A	1,5 million
	120 VAC	2,0 A	1,0 million
	120 VAC	1,0 A	1,5 million
	120 VAC	0,5 A	2,0 million
	230 VAC	2,0 A	1,0 million
230 VAC	1,0 A	1,5 million	
230 VAC	0,5 A	12,0 million	
Dla obciążenia indukcyjnego (zgodnie z IEC 947-5-1 DC13/AC15)	Napięcie	Prąd	Liczba cykli przełączania (typowo)
	24 VDC	2,0 A	0,05 million
	24 VDC	1,0 A	0,1 million
	24 VDC	0,5 A	0,5 million
	24 VAC	1,5 A	1,0 million
	48 VAC	1,5 A	1,0 million
	60 VAC	1,5 A	1,0 million
	120 VAC	2,0 A	0,7 million
	120 VAC	1,0 A	1,0 million
	120 VAC	0,5 A	1,5 million
230 VAC	2,0 A	0,7 million	
230 VAC	1,0 A	1,0 million	
230 VAC	0,5 A	1,5 million	
Aktywacja wejścia cyfrowego	Możliwe		
Częstotliwość przełączania			
Mechaniczna	Maks. 10 Hz		
Przy obciążeniu rezystancyjnym	Maks. 1 Hz		
Przy obciążeniu indukcyjnym (zgodnie z IEC 947-5-1 DC13 /AC15)	Maks. 0,5 Hz		
Przy obciążeniu lampowym	Maks. 1 Hz		

A.2. Specyfikacja CPU

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku systemu S7-1200 lub na internetowej stronie wsparcia klienta (<http://www.siemens.com/automation/>).

Tabela A.9. Ogólna specyfikacja

Ogólna specyfikacja		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Numer zamówiowy	• AC/DC/Przek	• 6ES7 211-1BE40-0XB0	• 6ES7 212-1BE40-0XB0	• 6ES7 214-1BG40-0XB0	• 6ES7 215-1BG40-0XB0	–
	• DC/DC/Przek	• 6ES7 211-1HE40-0XB0	• 6ES7 212-1HE40-0XB0	• 6ES7 214-1HG40-0XB0	• 6ES7 215-1HG40-0XB0	–
	• DC/DC/DC	• 6ES7 211-1AE40-0XB0	• 6ES7 212-1AE40-0XB0	• 6ES7 214-1AG40-0XB0	• 6ES7 215-1AG40-0XB0	• 6ES7 217-1AG40-0XB0
Wymiary W × H × D (mm)		90 × 100 × 75	90 × 100 × 75	110 × 100 × 75	130 × 100 × 70	150 × 100 × 75
Masa	<ul style="list-style-type: none"> • AC/DC/Przełącznik • DC/DC/Przełącznik • DC/DC/DC 	<ul style="list-style-type: none"> • 420 gramów • 380 gramów • 370 gramów 	<ul style="list-style-type: none"> • 425 gramów • 385 gramów • 370 gramów 	<ul style="list-style-type: none"> • 475 gramów • 435 gramów • 415 gramów 	<ul style="list-style-type: none"> • 530 gramów • 585 gramów • 520 gramów 	<ul style="list-style-type: none"> • – • – • 530 gramów
Pobór mocy	<ul style="list-style-type: none"> • AC/DC/Przełącznik • DC/DC/Przełącznik • DC/DC/DC 	<ul style="list-style-type: none"> • 10 W • 8 W • 8 W 	<ul style="list-style-type: none"> • 11 W • 9 W • 9 W 	<ul style="list-style-type: none"> • 14 W • 12 W • 12 W 	<ul style="list-style-type: none"> • 14 W • 12 W • 12 W 	<ul style="list-style-type: none"> • – • – • 12 W
Wydajność prądowa (5 VDC) magistrala CM lub SM		750 mA maks.	1000 mA maks.	1600 mA maks.	1600 mA maks.	1600 mA maks.
Wydajność prądowa (24 VDC) zasilanie czujników		300 mA maks.	300 mA maks.	400 mA maks.	400 mA maks.	400 mA maks.
Pobór prądu przez wejścia cyfrowe (24 VDC)		4 mA/ wykorzystane wejście	4 mA/ wykorzystane wejście	4 mA/ wykorzystane wejście	4 mA/ wykorzystane wejście	4 mA/wykorzystane wejście

Tabela A.10. Charakterystyka CPU

Charakterystyka CPU	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C	
Pamięć użytkownika • Pamięć robocza • Pamięć ładowania • Pamięć trwała	• 30 kB • 1 MB • 10 kB	• 50 kB • 1 MB • 10 kB	• 75 kB • 4 MB • 10 kB	• 100 kB • 4 MB • 10 kB	• 125 kB • 4 MB • 10 kB	
Wbudowane cyfrowe I/O (strona 298).	6 wejść 4 wyjścia	8 wejść 6 wyjść	14 wejść 10 wyjść	14 wejść 10 wyjść	14 wejść 10 wyjść	
Wbudowane analogowe I/O (strona 306).	2 wejścia	2 wejścia	2 wejścia	2 wejścia	2 wejścia 2 wyjścia	
Rozmiar obrazu procesu: • Wejść • Wyjść	• 1024 B • 1024 B	• 1024 B • 1024 B	• 1024 B • 1024 B	• 1024 B • 1024 B	• 1024 B • 1024 B	
Pamięć bitowa (M)	4096 B	4096 B	8192 B	8192 B	8192 B	
Pamięć tymczasowa (lokalna)	• 16 kB dla uruchomienia i cyklu programowego (w tym związanych FB i FC) • 6 kB dla każdego innego poziomu priorytetu przerwania (w tym FB i FC)					
Rozszerzające moduły sygnałowe (SM)	brak	2 SM maks.	8 SM maks.	8 SM maks.	8 SM maks.	
Rozszerzające SB, CB, lub BB	1 maks.	1 maks.	1 maks.	1 maks.	1 maks.	
Rozszerzające moduły komunikacyjne (CM)	3 maks.	3 maks.	3 maks.	3 maks.	3 maks.	
Szybkie liczniki	Łącznie	Do 6 konfigurowanych do użycia dowolnego wejścia wbudowanego lub SB				
	1 MHz	–	–	–	–	lb.2 do lb.5 (różnicowe)
	100/ ¹ 80 kHz	la.0 do la.5	la.0 do la.5	la.0 do la.5	la.0 do la.5	la.0 do la.5
	30/ ¹ 20 kHz	–	la.6 do la.7	la.6 do lb.5	la.6 do lb.5	la.6 do lb.1
Wyjścia impulsowe ²	Łącznie	Do 4 konfigurowanych do użycia dowolnego wejścia wbudowanego lub SB				
	1 MHz	–	–	–	–	Qa.0 do Qa.3 (różnicowe)
	100 kHz	Qa.0 do Qa.3	Qa.0 do Qa.3	Qa.0 do Qa.3	Qa.0 do Qa.3	Qa.4 do Qb.1
	30 kHz	–	Qa.4 do Qa.5	Qa.4 do Qb.1	Qa.4 do Qb.1	–
Wejścia rejestrujące impulsy	6	8	14	14	14	
Przerwania od opóźnienia	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	
Przerwania cykliczne	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	4 łącznie z rozdzielczością 1 ms	
Przerwania od zboczy Z opcjonalną SB	6 dla zboczy narastających i 6 dla zboczy opadających 10 dla zboczy narastających i 10 dla zboczy opadających	8 dla zboczy narastających i 8 dla zboczy opadających 12 dla zboczy narastających i 12 dla zboczy opadających	12 dla zboczy narastających i 12 dla zboczy opadających 16 dla zboczy narastających i 16 dla zboczy opadających	12 dla zboczy narastających i 12 dla zboczy opadających 16 dla zboczy narastających i 16 dla zboczy opadających	12 dla zboczy narastających i 12 dla zboczy opadających 16 dla zboczy narastających i 16 dla zboczy opadających	

Charakterystyka CPU	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Zegar czasu rzeczywistego <ul style="list-style-type: none"> • Dokładność • Czas podtrzymywania (bezobsługowy Super-kondensator) 	<ul style="list-style-type: none"> • +/- 60 sekund/miesiąc • 20 dni typ./12 dni min. przy 40°C 	<ul style="list-style-type: none"> • +/- 60 sekund/miesiąc • 20 dni typ./12 dni min. przy 40°C 	<ul style="list-style-type: none"> • +/- 60 sekund/miesiąc • 20 dni typ./12 dni min. przy 40°C 	<ul style="list-style-type: none"> • +/- 60 sekund/miesiąc • 20 dni typ./12 dni min. przy 40°C 	<ul style="list-style-type: none"> • +/- 60 sekund/miesiąc • 20 dni typ./12 dni min. przy 40°C
Szybkość wykonywania					
<ul style="list-style-type: none"> • operacji boolowskich 	• 0,08 µs/instrukcję	• 0,08 µs/instrukcję	• 0,08 µs/instrukcję	• 0,08 µs/instrukcję	• 0,08 µs/instrukcję
<ul style="list-style-type: none"> • operacji Move Word 	• 1,7 µs/instrukcję	• 1,7 µs/instrukcję	• 1,7 µs/instrukcję	• 1,7 µs/instrukcję	• 1,7 µs/instrukcję
<ul style="list-style-type: none"> • operacji Real Math 	• 2,3 µs/instrukcję	• 2,3 µs/instrukcję	• 2,3 µs/instrukcję	• 2,3 µs/instrukcję	• 2,3 µs/instrukcję

¹ Gdy HSC jest skonfigurowany w trybie kwadraturowym A/B, to ma zastosowanie szybkość mniejsza.

² Dla CPU z wyjściami przekaźnikowymi użytkownik musi zainstalować cyfrową płytkę sygnałową (SB), aby móc używać wyjść impulsowych.

Tabela A.11. Komunikacja

Dane techniczne	CPU 1211C, CPU 1212C, CPU 1214C	CPU 1215C, CPU 1217C
Komunikacja <ul style="list-style-type: none"> • Szybkość przesyłu danych • Izolacja (sygnału zewnętrznego od logiki PLC) • Typ kabla 	1 port Ethernet <ul style="list-style-type: none"> • 10/100 Mb/s • izolacja transformatorowa, 1500 VDC • CAT5e ekranowany 	2 porty Ethernet <ul style="list-style-type: none"> • 10/100 Mb/s • Izolacja transformatorowa, 1500 VDC • CAT5e ekranowany
Urządzenia	<ul style="list-style-type: none"> • 4 HMI • 1 PG 	<ul style="list-style-type: none"> • 4 HMI • 1 PG
Połączenia Ethernet ²	8 (aktywne lub pasywne)	8 (aktywne lub pasywne)
Połączenia S7 CPU-CPU (GET/PUT)	<ul style="list-style-type: none"> • 8 (klient) • 3 (serwer) 	<ul style="list-style-type: none"> • 8 (klient) • 3 (serwer)

¹ Otwieranie połączenia komunikacyjnego użytkownika (aktywne lub pasywne): TSEND_C, TRCV_C, TCON, TDISCON, TSEND oraz TRCV

Tabela A.12. Schemat podłączenia CPU1214C AC/DC/Przełącznik

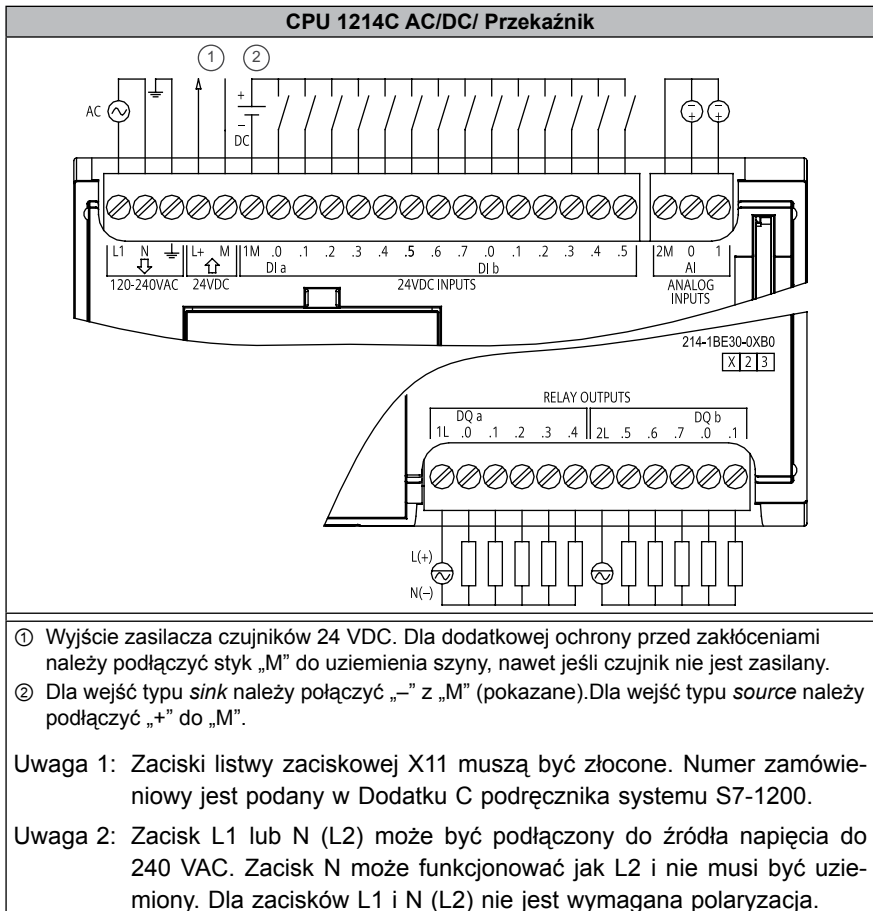
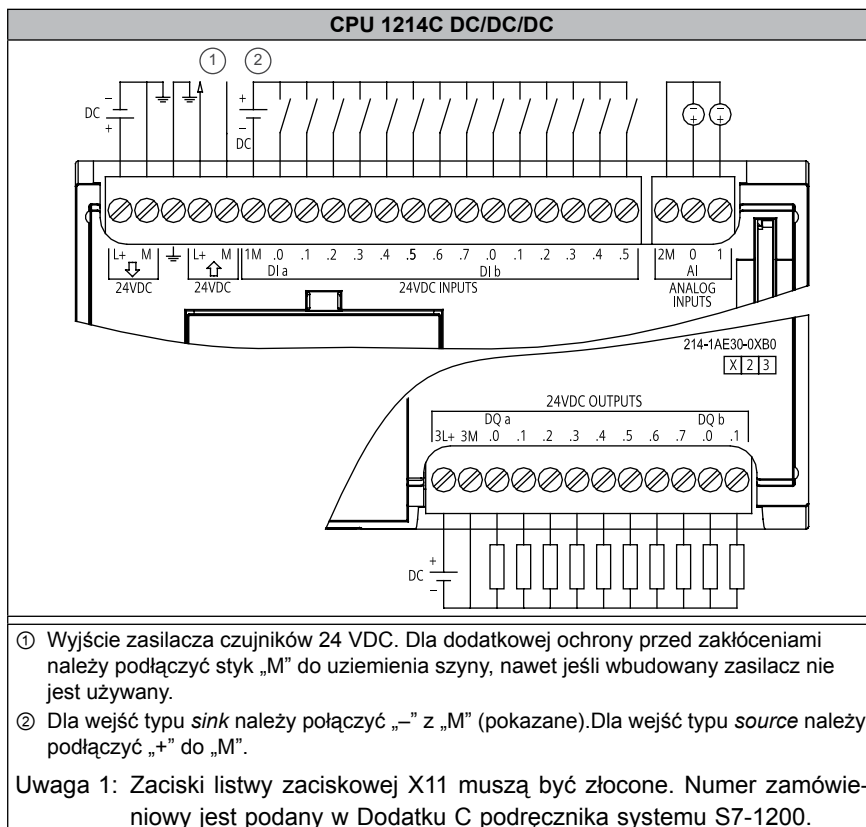


Tabela A.13. Schemat podłączenia CPU1214C DC/DC/DC



A.3. Moduły cyfrowych I/O

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku systemu S7-1200 lub na internetowej stronie wsparcia klienta (<http://www.siemens.com/automation/>).

A.3.1. Płytki sygnałowe SB 1221, SB 1222, oraz SB 1223 – cyfrowe wejścia/wyjścia (DI, DQ, oraz DI/DQ)

Tabela A.14. Moduły SB 1221 – cyfrowe wejścia (DI) i SB 1222 – cyfrowe wyjścia (DQ)

Ogólne	SB 1221 4 DI (200 kHz)	SB 1222 4 DQ (200 kHz)
Nr zamówieniowy	<ul style="list-style-type: none"> ● 24 VDC: 6ES7 221-3BD30-0XB0 ● 5 VDC: 6ES7 221-3AD30-0XB0 	<ul style="list-style-type: none"> ● 24 VDC: 6ES7 222-1BD30-0XB0 ● 5 VDC: 6ES7 222-1AD30-0XB0
Wymiary W × H × D (mm)	38 × 62 × 21 (mm)	38 × 62 × 21 (mm)
Masa	35 gramów	35 gramów
Pobór mocy	<ul style="list-style-type: none"> ● 24 VDC: 1,5 W ● 5 VDC: 1,0 W 	0,5 W

Ogólne		SB 1221 4 DI (200 kHz)	SB 1222 4 DQ (200 kHz)
Pobór prądu	magistrala SM	40 mA	35 mA
	24 VDC	<ul style="list-style-type: none"> • 24 VDC: 7 mA/wykorzystane wejście + 20 mA • 5 VDC: 15 mA/wykorzystane wejście + 15 mA 	15 mA
Wejścia/wyjścia		4 wejścia (typ: <i>source</i>)	4 wyjścia (półprzewodnik – MOSFET)

Tabela A.15. Moduł wejść/wyjść cyfrowych (DI/DQ) SM 1223

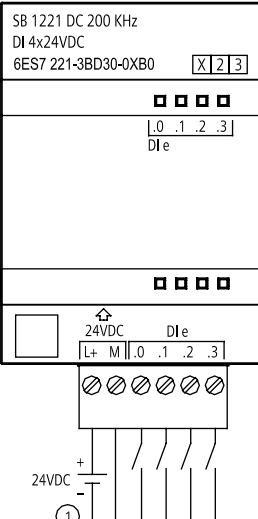
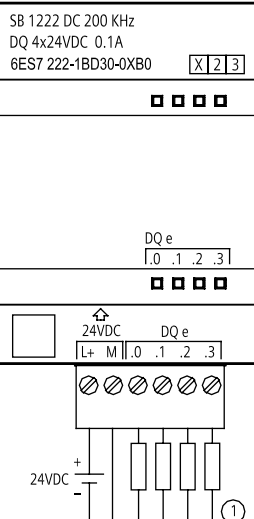
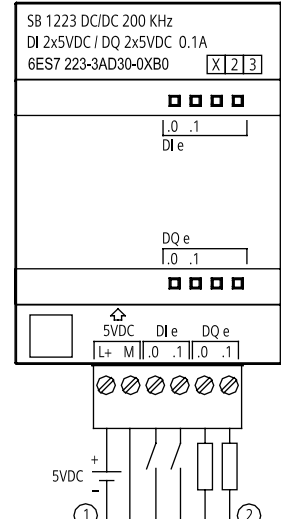
Ogólne		SB 1223 DI/DQ (200 kHz)	SB 1222 2 DI/2 DQ
Nr zamówieniowy		<ul style="list-style-type: none"> • 24 VDC: 6ES7 223-3BD30-0XB0 • 5 VDC: 6ES7 223-3AD30-0XB0 	24 VDC: 6ES7 223-0BD30-0XB0
Wymiary W × H × D (mm)		38 × 62 × 21 (mm)	38 × 62 × 21 (mm)
Masa		35 gramów	40 gramów
Pobór mocy		<ul style="list-style-type: none"> • 24 VDC: 1,0 W • 5 VDC: 0,5 W 	24 VDC: 1,0 W
Pobór prądu	magistrala SM	40 mA	35 mA
	24 VDC	<ul style="list-style-type: none"> • 24 VDC: 7 mA/wykorzystane wejście + 20 mA • 5 VDC: 15 mA/wykorzystane wejście + 15 mA 	15 mA
Wejścia/wyjścia		2 wejścia (typ: <i>source</i>) 2 wyjścia (półprzewodnik – MOSFET)	2 wejścia (IEC Type 1 sink) 2 wyjścia (półprzewodnik – MOSFET)

Uwaga

Szybkie (200 kHz) SB używa wejść typu *source*. Standardowa płytką SB (20 kHz) używa wejść typu *sink*. Więcej informacji znajduje się w specyfikacji cyfrowych wejść i wyjść (strona 298).

Szybkie (200 kHz) wyjścia (SB 1222 i SB 1223) mogą być typu *sink* lub *source*. Dla wyjść *source* należy połączyć „Load” z „-” (pokazane). Dla wyjść *sink* należy połączyć „Load” z „+”. Ponieważ obie konfiguracje pobierania prądu i dostarczania prądu są obsługiwane przez ten sam obwód, to aktywny stan obciążenia pobierającego prąd jest odwrotny do stanu obciążenia dostarczającego prąd. Wyjścia typu *source* wykazują logikę dodatnią (bit Q oraz LED są w stanie ON, wtedy gdy występuje przepływ prądu), podczas gdy wyjścia typu *sink* wykazują logikę ujemną (bit Q oraz LED są w stanie OFF, wtedy gdy występuje przepływ prądu). Jeżeli moduł jest podłączony bez programu użytkownika, domyślnym napięciem dla tego modułu jest 0 V, co oznacza, że obciążenie typu *sink* będzie w stanie ON.

Tabela A.16. Schemat podłączenia cyfrowej płytki SB

Moduł wejść SB 1221	Moduł wyjść SB 1222	Moduł wejść/wyjść SB 1223
<p>SB 1221 DI 4 (200 kHz)</p> <p>SB 1221 DC 200 KHz DI 4x24VDC 6ES7 221-3BD30-0XB0 [X 2 3]</p>  <p>① Obsługują tylko wejścia typu source.</p>	<p>SB 1222 DQ 4 (200 kHz)</p> <p>SB 1222 DC 200 KHz DQ 4x24VDC 0.1A 6ES7 222-1BD30-0XB0 [X 2 3]</p>  <p>① Dla wyjść source należy podłączyć Load do „-” (na rysunku). Dla wyjść sink należy podłączyć Load do „+”.</p> <p>Ponieważ obie konfiguracje pobierania prądu i dostarczania prądu są obsługiwane przez ten sam obwód, to aktywny stan obciążenia pobierającego prąd jest odwrotny do stanu obciążenia dostarczającego prąd. Wyjścia typu source wykazują logikę dodatnią (bit Q oraz LED są w stanie ON, wtedy gdy występuje przepływ prądu), podczas gdy wyjścia typu sink wykazują logikę ujemną (bit Q oraz LED są w stanie OFF, wtedy gdy występuje przepływ prądu). Jeżeli moduł jest podłączony bez programu użytkownika, domyślnym napięciem dla tego modułu jest 0V, co oznacza, że obciążenie typu „sink” będzie w stanie ON.</p>	<p>SB 1223 DI 2/DQ2 (200 kHz)</p> <p>SB 1223 DC/DC 200 KHz DI 2x5VDC / DQ 2x5VDC 0.1A 6ES7 223-3AD30-0XB0 [X 2 3]</p>  <p>① Obsługują tylko wejścia typu source.</p> <p>② Dla wyjść source należy podłączyć Load do „-” (na rysunku). Dla wyjść sink należy podłączyć Load do „+”.</p> <p>Ponieważ obie konfiguracje pobierania prądu i dostarczania prądu są obsługiwane przez ten sam obwód, to aktywny stan obciążenia pobierającego prąd jest odwrotny do stanu obciążenia dostarczającego prąd. Wyjścia typu source wykazują logikę dodatnią (bit Q oraz LED są w stanie ON, wtedy gdy występuje przepływ prądu), podczas gdy wyjścia typu sink wykazują logikę ujemną (bit Q oraz LED są w stanie OFF, wtedy gdy występuje przepływ prądu). Jeżeli moduł jest podłączony bez programu użytkownika, domyślnym napięciem dla tego modułu jest 0V, co oznacza, że obciążenie typu „sink” będzie w stanie ON.</p>

Uwaga

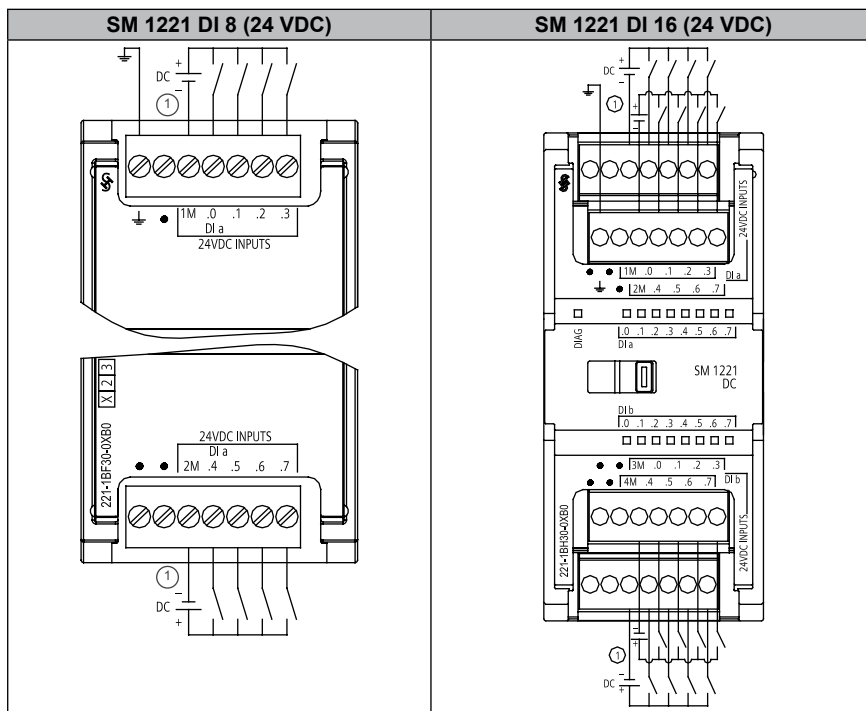
Szybkie (200 kHz) SB (SB 1221 i SB 1223) używa wejść tylko typu *sink*. Standardowa płytką SB 1233 używa styków tylko typu *source*.

Szybkie (200 kHz) wyjścia (SB 1222 i SB 1223) mogą być typu *sink* lub *source*. Dla wyjść *source* należy połączyć *Load* z „-” (na rysunku). Dla wyjść *sink* należy połączyć *Load* z „+”. Ponieważ obie konfiguracje pobierania prądu i dostarczania prądu są obsługiwane przez ten sam obwód, to aktywny stan obciążenia pobierającego prąd jest odwrotny do stanu obciążenia dostarczającego prąd. Wyjścia typu *source* wykazują logikę dodatnią (bit Q oraz LED są w stanie ON, wtedy gdy występuje przepływ prądu), podczas gdy wyjścia typu *sink* wykazują logikę ujemną (bit Q oraz LED są w stanie OFF, wtedy gdy występuje przepływ prądu). Jeżeli moduł jest podłączony bez programu użytkownika, domyślnym napięciem dla tego modułu jest 0 V, co oznacza, że obciążenie typu *sink* będzie w stanie ON.

A.3.2. Moduł SM 1221 – cyfrowe wejścia (DI)**Tabela A.17.** SM 1221 – cyfrowe wejścia (DI)

Dane techniczne		SM 1221 DI 8 (24 VDC)	SM 1221 DI 16 (24 VDC)
Nr zamówieniowy		6ES7 221-1BF32-0XB0	6ES7 221-1BH32-0XB0
Liczba wejść (DI) (Specyfikacja na stronie 283).		8	16
Wymiary W × H × D (mm)		45 × 100 × 75	45 × 100 × 75
Masa		170 gramów	210 gramów
Pobór mocy		1,5 W	2,5 W
Pobór prądu	magistrala SM	105 mA	130 mA
	24 VDC	4 mA/wykorzystane wejście	4 mA/wykorzystane wejście

Tabela A.18. Schemat podłączenia dla modułów cyfrowych wejść (DI) dla SM 1221



① Dla wejść typu *sink* należy podłączyć „-” do „M” (na rysunku). Dla wejść typu *source* należy podłączyć „+” do „M”.

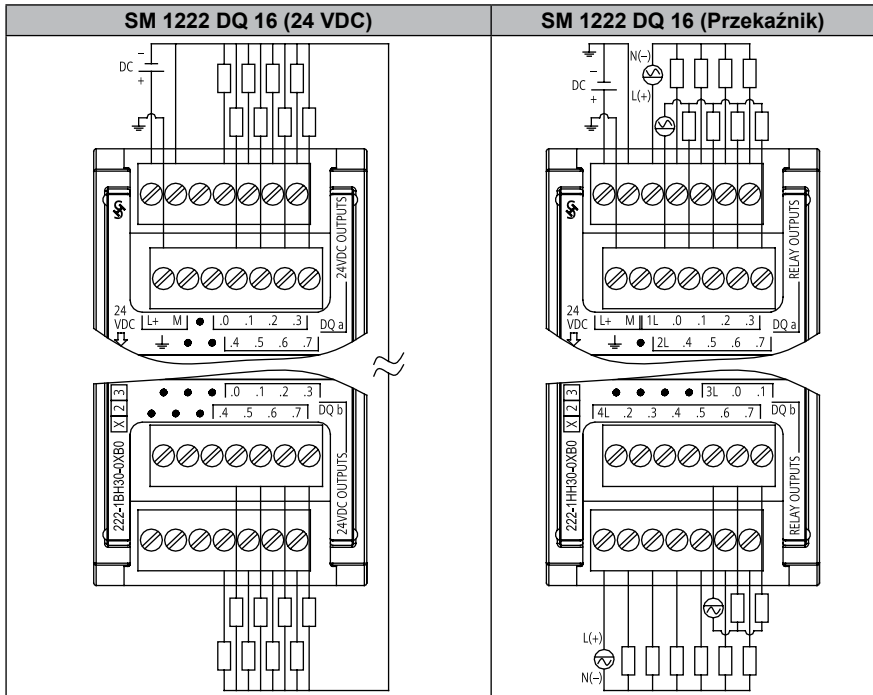
A.3.3. Moduł SM 1222 – cyfrowe wyjścia (DQ)

Tabela A.19. SM 1222 – cyfrowe wyjścia (DQ)

Dane techniczne	SM 1222 DQ (przełącznik)	SM1222 DQ (24VDC)
Nr zamówieniowy	<ul style="list-style-type: none"> • DQ 8: 6ES7 222-1HF32-0XB0 • DQ 8: przełączne: 6ES7 222- 1XF32-0XB0 • DQ 16: 6ES7 222-1HH32-0XB0 	<ul style="list-style-type: none"> • DQ 8: 6ES7 222-1BF32-0XB0 • DQ 16: 6ES7 222-1BH32-0XB0
Liczba wyjść (DQ) (Specyfikacja na stronie ???)	<ul style="list-style-type: none"> • 8 (DQ 8 i DQ 8 przełączne) • 16 (DQ 16) 	8 (DQ 8) lub 16 (DQ 16)
Wymiary W × H × D (mm)	<ul style="list-style-type: none"> • DQ 8 oraz DQ 16: 45 × 100 × 75 • DQ 8 przełączne: 70 × 100 × 75 	45 × 100 × 75
Masa	<ul style="list-style-type: none"> • DQ 8: 190 gramów • DQ 8 zamiennik: 310 grams • DQ 16: 260 gramów 	<ul style="list-style-type: none"> • DQ 8: 180 gramów • DQ 16: 220 gramów
Pobór mocy	<ul style="list-style-type: none"> • DQ 8: 4,5 W ••DQ 8 przełączne: 5 W • DQ 16: 8,5 W 	<ul style="list-style-type: none"> • DQ 8: 1,5 W • DQ 16: 2,5 W

Dane techniczne		SM 1222 DQ (przełącznik)	SM1222 DQ (24VDC)
Pobór prądu	magistrala SM	<ul style="list-style-type: none"> • DQ 8: 120 mA • DQ 8 przełączne: 140 mA • DQ 16: 135 mA 	<ul style="list-style-type: none"> • DQ 8: 120 mA • DQ 16: 140 mA
	24 VDC	<ul style="list-style-type: none"> • DQ 8 i DQ 16: 11 mA/wykorzystaną cewkę przekaźnika • DQ 8 przełączne: 16,7 mA/wykorzystaną cewkę przekaźnika 	<ul style="list-style-type: none"> • DQ 8: -- • DQ 16: --

Tabela A.20. Schemat podłączenia dla modułów cyfrowych wyjść (DQ) dla SM 1222



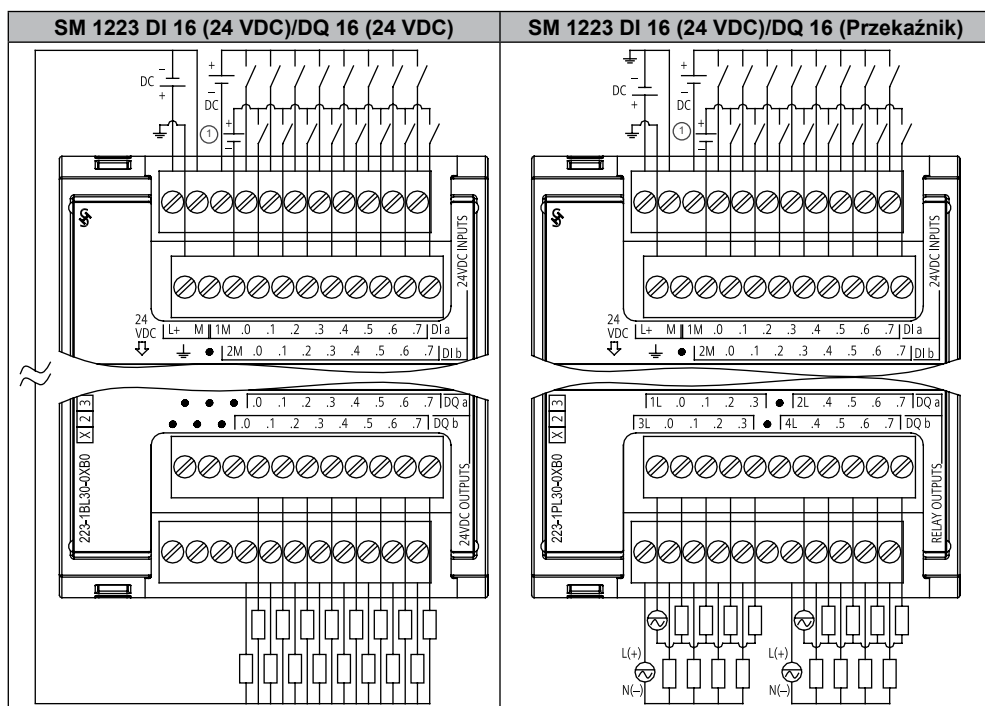
A.3.4. Moduł SM 1223 cyfrowe wejścia/wyjścia VDC (DI/DQ)

Tabela A.21. SM 1223 – cyfrowe wejścia wyjścia (DI/DQ)

Dane techniczne	SM 1223 DI (24 VDC)/DQ (Przełącznik)	SM 1223 DI (24 VDC)/DQ (24 VDC)
Nr zamówieniowy	DI 8/DQ 8: 6ES7 223-1PH32-0XB0 DI 16/DQ 16: 6ES7 223-1PL32-0XB0	DI 8/DQ 8: 6ES7 223-1BH32-0XB0 DI 8/DQ 8: 6ES7 223-1BL32-0XB0
Liczba wejść/wyjść (DI/DQ) (Specyfikacja na stronie 283).	<ul style="list-style-type: none"> • Wejść: 8 lub 16 (24 VDC) • Wyjść: 8 lub 16 (przełącznik) 	<ul style="list-style-type: none"> • Wejść: 8 lub 16 (24 VDC) • Wyjść: 8 lub 16 (24 VDC)
Wymiary W × H × D (mm)	<ul style="list-style-type: none"> • DI 8/DQ 8: 45 × 100 × 75 • DI 16/DQ 16: 70 × 100 × 75 	<ul style="list-style-type: none"> • DI 8/DQ 8: 45 × 100 × 75 • DI 16/DQ 16: 70 × 100 × 75

Dane techniczne		SM 1223 DI (24 VDC)/DQ (Przełącznik)	SM 1223 DI (24 VDC)/DQ (24 VDC)
Masa		<ul style="list-style-type: none"> DI 8/DQ 8: 230 gramów DI 16/DQ 16: 350 gramów 	<ul style="list-style-type: none"> DI 8/DQ 8: 210 gramów DI 16/DQ 16: 310 gramów
Pobór mocy		<ul style="list-style-type: none"> DI 8/DQ 8: 5,5 W DI 16/DQ 16: 10 W 	<ul style="list-style-type: none"> DI 8/DQ 8: 2,5 W DI 16/DQ 16: 4,5 W
Pobór prądu	magistrala SM	<ul style="list-style-type: none"> DI 8/DQ 8: 145 mA DI 16/DQ 16: 180 mA 	<ul style="list-style-type: none"> DI 8/DQ 8: 145 mA DI 16/DQ 16: 185 mA
	24 VDC	4 mA/wykorzystane wejście 11 mA/wykorzystaną cewkę przekaźnika	4 mA/wykorzystane wejście

Tabela A.22. Schemat podłączenia dla modułów cyfrowych wejść/wyjść (DI/DQ) dla SM 1223



Ⓛ Dla wejść typu *sink* należy podłączyć „-” do „M” (na rysunku). Dla wejść typu *source* należy podłączyć „+” do „M”.

A.3.5. Moduł SM 1223 120/Wejścia 230 VAC/Wyjścia przekaźnikowe

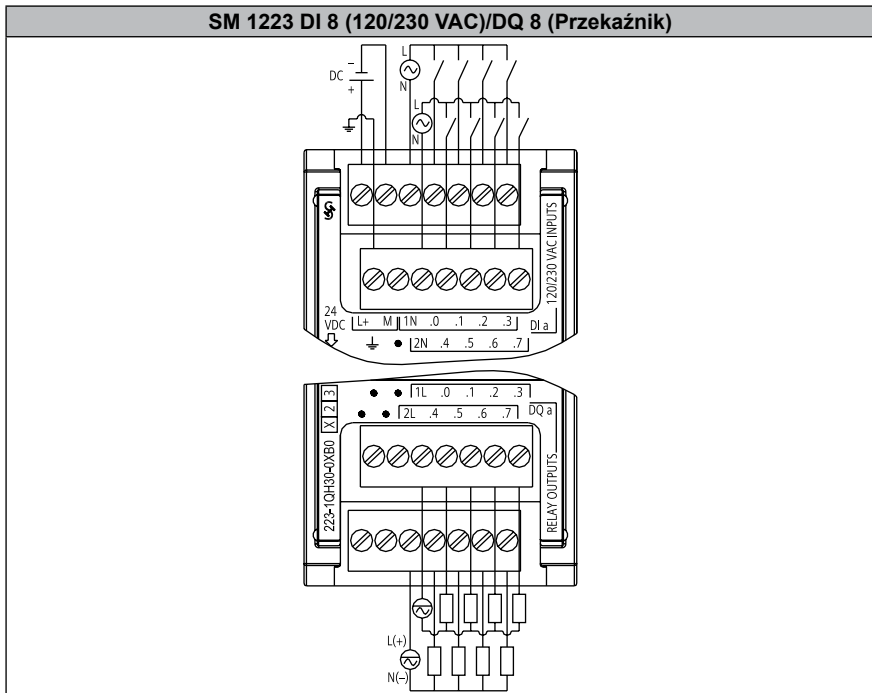
Tabela A.23. Moduł 1223 – cyfrowe wejścia VAC/wyjścia (DI/DQ)

Dane techniczne		SM 1223 DI (120/230 VAC)/DQ (Przełącznik)
Nr zamówieniowy		DI 8/DQ 8: 6ES7 223-1QH32-0XB0
Liczba wejść/wyjść (DI/DQ)		Wejścia: 8 (120/230 VAC) Specyfikacja dla wejść 120/230 VAC (strona 299). Wyjścia: 8 (przełącznik) Specyfikacja dla cyfrowych wyjść (strona 300)
Wymiary W × H × D (mm)		45 × 100 × 75
Masa		190 gramów
Pobór mocy		7,5 W
Pobór prądu	magistrala SM	120 mA
	24 VDC	11 mA/wykorzystaną cewkę przekaźnika

Uwaga

Moduł sygnałowy SM 1223 DI 8 x 120/230 VAC, DQ 8 x Przełącznik (6ES7 223-1QH32-0XB0) jest zatwierdzony do użytku w Class 1, Division 2, Gas Group A, B, C, D, Temperature Class T4 Ta = 40°C.

Tabela A.24. Schemat podłączenia dla modułów SM 1223 DI 8 (120/230 VAC)/DQ (Przełącznik)



A.4. Specyfikacja cyfrowych wejść i wyjść

A.4.1. Cyfrowe wyjścia 24 VDC (DI)

Tabela A.25. Specyfikacja cyfrowych wejść (DI)

Dane techniczne	CPU, SM oraz SB	Szybkie SB (200 kHz)
Typ	<ul style="list-style-type: none"> CPU oraz SM: typ sink/source (IEC Type 1 sink) SB 1223: tylko sink (IEC Type 1 sink) 	SB 1221 200 KHz oraz SB 1223 200 KHz: Typ: Source
Napięcie	24 VDC przy 4 mA, nominalnie	24 VDC SB: 24 VDC przy 7 mA, nominalnie 5 VDC SB: 5 VDC przy 15 mA, nominalnie
Ciągłe dopuszczalne napięcie	30 VDC, maks.	24 VDC SB: 28,8 VDC 5 VDC SB: 6 VDC
Udar napięciowy	35 VDC przez 0,5 s	24 VDC SB: 35 VDC przez 0,5 s 5 VDC SB: 6 V
Sygnał logiczny 1 (min.)	15 VDC przy 2,5 mA	24 VDC SB: L+ minus 10 VDC przy 2,9 mA 5 VDC SB: L+ minus 2,0 VDC przy 5,1 mA
Sygnał logiczny 0 (maks.)	5 VDC przy 1 mA	24 VDC SB: L+ minus 5 VDC przy 1,4 mA 5 VDC SB: L+ minus 1,0 VDC przy 2,2 mA
Izolacja (od strony wyjściowej do logiki)	500 VAC przez 1 minutę	500 VAC przez 1 minutę
Grupy izolacji	<ul style="list-style-type: none"> CPU: 1 SM 1221 DI 8: 2 SM 1221 DI 16: 4 SB 1223 DI 2: 1 SM 1223: 2 	<ul style="list-style-type: none"> SB 1221 DI 4: 1 SB 1223 DI 2: 1
Czasy filtru	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 lub 12,8 ms (wybierane w grupach po 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 lub 12,8 ms (wybierane w grupach po 4)
Liczba wejść znajdujących się jednocześnie w stanie włączonym	<ul style="list-style-type: none"> SM 1221 oraz SM 1223 DI 8: 8 SM 1221 oraz SM 1223 DI 16: 16 SB 1223 DI 2: 2 	<ul style="list-style-type: none"> SB 1221 DI 4: 4 SB 1223 DI 2: 2
Długość kabla (w metrach)	<ul style="list-style-type: none"> 500 ekranowany, 300 nieekranowany CPU: 50 ekranowany dla HSC 	50 m ekranowana skrętka

Uwaga

Przy częstotliwości przełączania powyżej 20 kHz cyfrowe wejścia muszą otrzymywać sygnał prostokątny. W celu poprawienia jakości sygnału należy rozważyć następujące zabiegi:

- Zmniejszenie długości okablowania.
- Zmianę sterownika ze sterownika typu sink na sterownik typu sink/source.
- Wymianę okablowania na kable wyższej jakości.
- Zmniejszenie napięcia obwodu/komponentów z 24 V do 5 V.
- Dodanie zewnętrznego obciążenia na wyjściu.

Tabela A.26. Taktowanie wejść HSC (maks.)

Dane techniczne	Jednofazowe	Kwadraturowe
CPU 1211C	100 kHz	80 kHz
CPU 1212C	100 kHz (od Ia.0 do Ia.5) oraz 30 kHz (od Ia.6 do Ia.7)	80 kHz (od Ia.0 do Ia.5) oraz 20 kHz (od Ia.6 do Ia.7)
CPU 1214C, CPU 1215C	100 kHz (od Ia.0 do Ia.5) oraz 30 kHz (od Ia.6 do Ib.5)	80 kHz (od Ia.0 do Ia.5) oraz 20 kHz (od Ia.6 do Ib.5)
CPU 1217C	1 MHz (Dlb.2 do Dlb.5)	1 MHz (Dlb.2 do Dlb.5)
Szybkie (200 kHz) SB	200 kHz	160 kHz
Standardowe (20 kHz) SB	30 kHz	20 kHz

¹ Sygnał logiczny 1 = 15 do 26 VDC

A.4.2. Cyfrowe wyjścia 120/230 VAC

Tabela A.27. Cyfrowe wyjścia 120/230 VAC

Dane techniczne		SM
Typ		IEC Type 1
Napięcie		120 VAC przy 6 mA, 230 VAC przy 9 mA
Ciągłe dopuszczalne napięcie		264 VAC
Udar napięciowy		N/A
Sygnał logiczny 1 (min.)		79 VAC przy 2,5 mA
Sygnał logiczny 0 (maks.)		20 VAC przy 1 mA
Prąd upływu (maks.)		1 mA
Izolacja (od strony wyjściowej do logiki)		1500 VAC przez 1 minutę
Grupy izolacji ¹		4
Zwłoka załączenia wejścia		<ul style="list-style-type: none"> • Zwykle: 0,2 do 12,8 ms, wybierane przez użytkownika • Maksimum: --
Podłączenie 2-przewodowego czujnika zbliżeniowego (Bero) (maks.)		1 mA
Długość kabla	Nieekranowany	300 metrów
	Ekranowany	500 metrów
Liczba wejść znajdujących się jednocześnie w stanie włączonym		8

¹ Kanały w jednej grupie muszą być w tej samej fazie.

A.4.3. Cyfrowe wyjścia (DQ)

Tabela A.28. Specyfikacja cyfrowych wyjść (DQ)

Dane techniczne	Przełącznik (CPU oraz SM)	24V DC (CPU, SM, oraz SB)	200 KHZ 24V DC (SB)
Typ	przełącznik, styki suche	półprzewodnik – MOSFET (Source)	półprzewodnik – MOSFET (Sink/ Source)
Zakres napięć	5 do 30 VDC oraz 5 do 250 VAC	20,4 do 28,8 VDC	20,4 do 28,8 VDC ¹ 4,25 do 6,0 VDC ²
Sygnal logiczny 1 przy maks. prądzie	N/A	20 VDC min.	L+ minus 1,5 V ¹ L+ minus 0,7 V ²
Sygnal logiczny 0 przy obciążeniu 10 kΩ	N/A	CPU: 20 VDC min., 0,1 VDC maks. SB: 0,1 VDC maks. SM DC: 0,1 VDC maks.	1,0 VDC, maks. ¹ 0,2 VDC, maks. ²
Prąd (maks.)	2,0 A	0,5 A	0,1 A
Obciążenie żarówką	30 W DC/200 W AC	SB: 5 W	N/A
Rezystancja w stanie ON	0,2 Ω maks. w stanie nowości	0,6 Ω maks.	11 Ω maks. ¹ lub 7 Ω maks. ²
Rezystancja w stanie OFF	N/A	N/A	6 Ω maks. ¹ lub 0,2 Ω maks. ²
Prąd upływu na jeden punkt	N/A	10 μA maks.	N/A
Częstotliwość PTO	CPU: N/A ³	CPU: 100 kHz maks., 2 Hz min. ⁴ SB: 20 kHz maks., 2 Hz min. ⁵	200 kHz maks., 2 Hz min.
Udar prądowy	7 A z zamkniętymi stykami	CPU: 8 A przez 100 ms maks. SB: 5 A przez 100 ms maks. SM: 8 A przez 100 ms maks.	0,11 A
Zabezpieczenie przed przeciążeniem	Brak	Brak	Brak
Izolacja (sygnału zewnętrznego od logiki)	Cewka do styku: 1500 VAC przez 1 minutę Cewka do logiki: brak	500 VAC przez 1 minutę	500 VAC przez 1 minutę
Grupy izolacji	<ul style="list-style-type: none"> • CPU 1211C: 1 • CPU 1212C: 2 • CPU 1214C: 2 • CPU 1215C: 2 • SM DQ 8: 2 • SM DQ 8 przełączne: 8 • SM DQ 16: 4 	<ul style="list-style-type: none"> • CPU: 1 • SB: 1 • SM (DQ 8): 1 • SM (DQ 16): 1 	1 ⁶
Rezystancja izolacji	100 MΩ min. w stanie nowości	N/A	N/A
Izolacja między otwartymi stykami	750 VAC przez 1 minutę	N/A	N/A

Dane techniczne	Przełącznik (CPU oraz SM)	24V DC (CPU, SM, oraz SB)	200 KHZ 24V DC (SB)
Prąd wspólny	CPU: SM przełącznik: • SM 1222: 10 A (DQ 8 oraz DQ 16) • SM 1223 DI 8/ DQ 8 przełącznik: 10 A • SM 1223 DI 16/DQ 16 przełącznik: 8 A	CPU: • SB: 1 A • SM DQ 8: 4 A • SM DQ 16: 8 A	0,4 A
Ograniczanie przepięć indukcyjnych	N/A	L+ minus 48 V, 1 W mocy strat	Brak
Maksymalna częstotliwość przełączania przełącznika	1 Hz	N/A	N/A
Opóźnienie przełączania	10 ms maks.	CPU: • Qa.0 do Qa.3: 1,0 μs maks., OFF-do-ON 3,0 μs maks., ON-do-OFF • Qa.4 do Qb.1: 50 μs maks., OFF-do-ON 200 μs maks., ON-do-OFF SB: 2 μs maks. OFF-do-ON; 10 μs maks. ON-do-OFF SM: 50 μs maks. OFF-do-ON 200 μs maks. ON-do-OFF	1,5 μs + 300 s narastanie ¹ 1,5 μs + 300 ns opadanie ¹ 200 ns + 300 ns narastanie ² 200 ns + 300 ns opadanie ²
Trwałość mechaniczna (bez obciążenia)	Przełącznik: 10 mln cykli załącz/wyłącz	N/A	N/A
Trwałość styków przy nominalnym obciążeniu	Przełącznik: 100,000 cykli załącz/wyłącz	N/A	N/A
Zachowanie przy przejściu z RUN do STOP	Ostatnia wartość lub wartość zastępcza (domyślnie 0)	Ostatnia wartość lub wartość zastępcza (domyślnie 0)	Ostatnia wartość lub wartość zastępcza (domyślnie 0)
Długość kabla (w metrach)	500 m ekranowany, 150 m nieekranowany	500 m ekranowany, 150 m nieekranowany	50 m ekranowana skrętka

¹ SB 24 VDC 200 kHz.² SB 5 VDC 200 kHz.³ Dla CPU wyposażonych w wyjścia przełącznikowe użytkownik musi zainstalować płytkę sygnałową (SB), aby móc używać wyjść impulsowych.⁴ W zależności od rodzaju urządzenia odbierającego impulsy oraz okablowania użytkownik musi dołączyć do obwodu dodatkowe obciążenie (co najmniej 10% prądu), które może polepszyć jakość sygnału i zwiększyć odporność układu na zakłócenia.⁵ W zależności od rodzaju urządzenia odbierającego impulsy oraz okablowania użytkownik musi dołączyć do obwodu dodatkowe obciążenie (co najmniej 10% prądu), które może polepszyć jakość sygnału i zwiększyć odporność układu na zakłócenia.⁶ SB 1223 200 kHz DI 2/DQ 2: Brak izolacji wejść.

A.5. Moduły analogowych I/O

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku systemu S7-1200 lub na internetowej stronie wsparcia klienta (<http://www.siemens.com/automation/>).

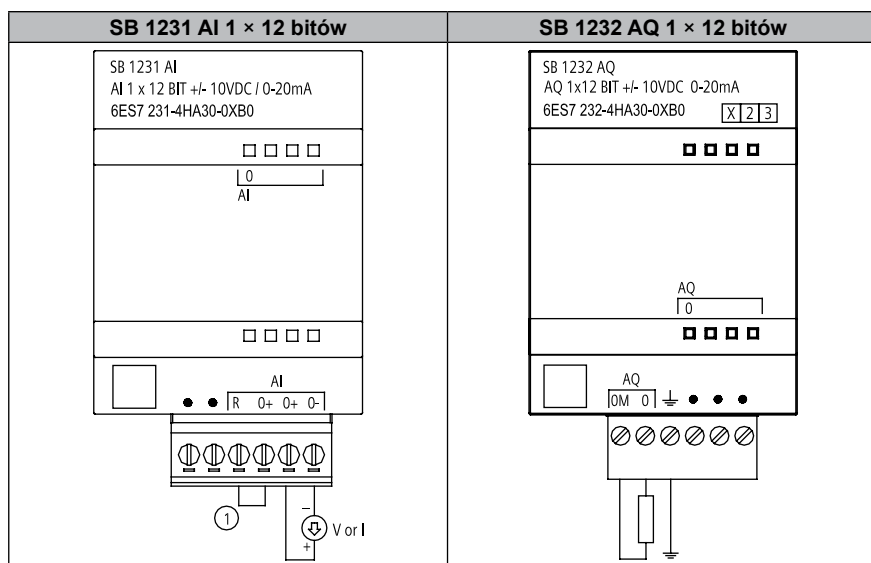
A.5.1. Płytki sygnałowe analogowych wejść (AI) – SB 1231 i analogowych wyjść (AQ) – SB 1232

Tabela A.29. Ogólna specyfikacja

Dane techniczne	SB 1231 AI 1 × 12 bitów ¹	SB 1232 AQ 1 × 12 bitów
Nr zamówieniowy	6ES7 231-4HA30-0XB0	6ES7 232-4HA30-0XB0
Wymiary W × H × D (mm)	38 × 62 × 21 mm	38 × 62 × 21 mm
Masa	35 gramów	40 gramów
Pobór mocy	0,4 W	1,5 W
Pobór prądu (magistrala SM)	55 mA	15 mA
Pobór prądu (24 VDC)	Brak	40 mA (bez obciążenia)
Liczba wejść/wyjść	1	1
Typ	Napięcie lub prąd (wejście różnicowe)	Napięcie lub prąd

¹ Oprogramowanie sprzętowe CPU musi być w wersji V2.0 lub wyższej, aby było możliwe korzystanie z SB 1231 AI 1

Tabela A.30. Schematy podłączeń dla analogowych SB



① Dla obsługi prądowego wejścia/wyjścia należy połączyć styk „R” oraz „0+”.

A.5.2. SM 1231 – analogowe wejścia (AI)**Tabela A.31.** SM 1231 – analogowe wejścia (AI)

Dane techniczne	SM 1231 AI 4 × 13 bitów	SM 1231 AI 8 × 13 bitów	SM 1231 AI 4 × 16 bitów
Nr zamówieniowy (MLFB)	6ES7 231-4HD32-0XB0	6ES7 231-4HF32-0XB0	6ES7 231-5ND32-0XB0
Liczba wejść	4 wejścia (AI)	8 wyjść (AI)	4 wejścia
Typ	Napięcie lub prąd (wejście różnicowe), typ wejścia określany dla 2 wejść analogowych (parami)	Napięcie lub prąd (wejście różnicowe), typ wejścia określany dla 2 wejść analogowych (parami)	Napięcie lub prąd (różnicowe)
Wymiary W × H × D (mm)	45 × 100 × 75	45 × 100 × 75	45 × 100 × 75
Masa	180 gramów	180 gramów	180 gramów
Pobór mocy	1,5 W	1,5 W	1,8 W
Pobór prądu (magistrala SM)	80 mA	90 mA	80 mA
Pobór prądu (24 VDC)	45 mA	45 mA	65 mA

A.5.3. SM 1232 – analogowe wyjścia (AQ)**Tabela A.32.** SM 1232 – analogowe wyjścia (AQ)

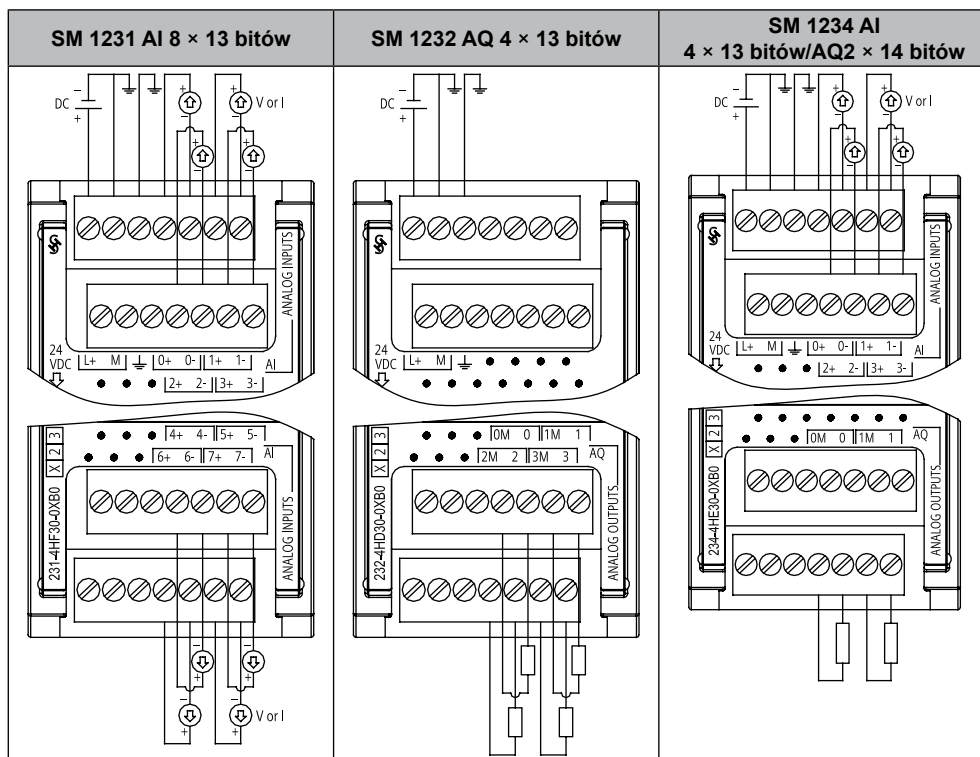
Dane techniczne	SM 1232 AQ 2 × 14 bitów	SM 1232 AQ 4 × 14 bitów
Nr zamówieniowy (MLFB)	6ES7 232-4HB32-0XB0	6ES7 232-4HD32-0XB0
Liczba i typ wyjść	2 wyjścia (AQ)	4 wyjścia (AQ)
Wymiary W × H × D (mm)	45 × 100 × 75	45 × 100 × 75
Masa	180 gramów	180 gramów
Pobór mocy	1,5 W	1,5 W
Pobór prądu (magistrala SM)	80 mA	80 mA
Pobór prądu (24 VDC)	45 mA (bez obciążenia)	45 mA (bez obciążenia)

A.5.4. SM 1234 – analogowe wejścia/wyjścia (AI/AQ)**Tabela A.33.** SM 1234 – analogowe wejścia/wyjścia (AI/AQ)

Dane techniczne	SM 1234 AI 4 × 13 bitów/AQ 2 × 14 bitów
Nr zamówieniowy (MLFB)	6ES7 234-4HE32-0XB0
Liczba wejść	4 wejścia (AI)
Typ	Napięcie lub prąd (wejście różnicowe), typ wejścia określany dla 2 wejść analogowych (parami)
Liczba wyjść	2 wyjścia (AQ)
Typ	Napięcie lub prąd (wyjście różnicowe)
Wymiary W × H × D (mm)	45 × 100 × 75
Masa	220 gramów
Pobór mocy	2,0 W
Pobór prądu (magistrala SM)	80 mA
Pobór prądu (24 VDC)	60 mA (bez obciążenia)

A.5.5. Schematy połączeń dla SM 1231 (AI), SM 1232 (AQ) oraz SM 1234 (AI/AQ)

Tabela A.34. Schematy połączeń analogowych SM



Uwaga

Nie używane wejścia napięciowe powinny być zwarte.

Nie używane wejścia prądowe powinny być ustawione na zakres od 0 do 20 mA i/lub z wyłączonym raportowaniem błędów przerwania przewodu.

Wejścia skonfigurowane w trybie prądowym nie będą przewodzić prądu pętli, dopóki moduł nie będzie zasilany i skonfigurowany.

Wejścia prądowe nie będą aktywne, dopóki do nadajnika nie zostanie dołączone zewnętrzne źródło zasilania.

A.6. Płytką bateryjną BB 1297

Płytką bateryjną BB 1297

Tabela A.34. Dane techniczne ogólne

Dane techniczne	Płytką bateryjną BB 1297
Numer zamówienia	6ES7 297-0AX30-0XA0
Wymiary szer. × wys. × głęb. (mm)	38 × 62 × 21
Waga	28 gramów
Czas podtrzymania pracy zegara	Okolo 1 roku
Typ baterii	CR1025 ¹
Dioda LED „Maint” w CPU	Sygnalizuje konieczność wymiany baterii
Program użytkownika	Aplikacja/system może oszacować stan naładowania baterii

¹ Informacje na temat instalowania BB 1297 lub wymiany baterii w BB znajdują się w rozdziale 2 podręcznika systemu S7-1200.

Płytką bateryjną BB 1297 jest używana w aplikacjach, w których wymagany czas podtrzymania pracy zegara czasu rzeczywistego przekracza jeden miesiąc. Właściwości BB 1297 przedstawiono poniżej:

- Podtrzymuje pracę zegara czasu rzeczywistego po wyłączeniu zasilania układów PLC. CPU S7-1200 w połączeniu z BB 1297 zapewnia podtrzymanie pracy zegara czasu rzeczywistego po wyłączeniu zasilania na okres do jednego roku.
- Tylko jedna płytką BB 1297 lub SB może być używana w danym czasie.
- Podłączanie lub wymiana przy włączonym zasilaniu nie jest dozwolone. Płytkę BB 1297 można podłączać lub wymieniać tylko przy wyłączonym zasilaniu CPU. Gdy przy wyłączonym zasilaniu CPU, płytką BB 1297 zostanie wyjęta w celu wymiany baterii, to podtrzymanie pracy zegara czasu rzeczywistego zapewnia wewnętrzny super-kondensator.
- Dioda LED „Maint” w CPU służy do sygnalizowania konieczności wymiany baterii na nową.
- Program użytkownika pozwala na monitorowanie i sprawdzanie stanu baterii i płytki bateryjnej i informowanie użytkownika za pomocą komunikatów wyświetlanych na panelu HMI lub na stronie Web.

A.7. Specyfikacja analogowych wejść/wyjść

A.7.1. Specyfikacja analogowych wejść (CPU, SM oraz SB)

Tabela A.36. Specyfikacja analogowych wejść (AI)

Dane techniczne	CPU	SB	SM
Typ	Napięcie (wejście niesymetryczne)	Napięcie lub prąd (wejście różnicowe)	Napięcie lub prąd (wejście różnicowe), typ określany parami
Zakres	0 do 10 V	± 10 V, ± 5 V, $\pm 2,5$, 0 do 20 mA, lub 4 mA do 20 mA	± 10 V, ± 5 V, $\pm 2,5$ V, 0 do 20 mA, lub 4 mA do 20 mA
Rozdzielczość	10 bitów	11 bitów + bit znaku	12 bitów + bit znaku
Zakres pomiarowy (słowo danych)	0 do 27648	-27648 do 27648	-2,648 do 27648
Dokładność (25°C/-20 do 60°C)	3,0%/3,5% pełnego zakresu	$\pm 0,3\%/ \pm 0,6\%$ pełnego zakresu	$\pm 0,1\%/ \pm 0,2\%$ pełnego zakresu
Zakres przerzutu górnego i dolnego (słowo danych) (Przypis 1)	Napięcie: Od 27,649 do 32,511	Napięcie: Od 32,511 do 27,649/ Od -27,649 do -32,512	Napięcie: Od 32,511 do 27,649/od -27,649 do -32,512
	Prąd: N/A	Prąd: Od 32,511 do 27,649/ od 0 do -4864	Prąd: Od 32,511 do 27,649/od 0 do -4864
Przepiętnienie górne i dolne (słowo danych) (Przypis 1)	Napięcie: Od 32,512 do 32,767	Napięcie: Od 32,767 do 32,512/ od -32,513 do -32,768	Napięcie: Od 32,767 do 32,512/od -32,513 do -32,768
	Prąd: N/A	Prąd: Od 32,767 do 32,512/ od -4865 do -32,768	Prąd: Od 32,767 do 32,512/od -4865 do -32,768
Maksymalne bezpieczne napięcie/prąd	35 VDC (napięcie)	± 35 V/ ± 40 mA	± 35 V/ ± 40 mA
Wygładzanie (Przypis 2)	None (brak), Weak (słabe), Medium (średnie) lub Strong (mocne)	None (brak), Weak (słabe), Medium (średnie) lub Strong (mocne)	None (brak), Weak (słabe), Medium (średnie) lub Strong (mocne)
Tłumienie zakłóceń (Przypis 2)	10, 50, lub 60 Hz	400, 60, 50, lub 10 Hz	400, 60, 50, lub 10 Hz
Zasada pomiaru	Konwersja aktualnej wartości	Konwersja aktualnej wartości	Konwersja aktualnej wartości
Tłumienie sygnału sumacyjnego	40 dB, DC do 60 Hz	40 dB, DC do 60 Hz	40 dB, DC do 60 Hz
Zakres operacyjny sygnału (sygnał plus napięcie sumacyjne)	Mniejsze niż +12 V i większe niż -12 V	Mniejsze niż +35 V i większe niż -35 V	Mniejsze niż +12 V i większe niż -12 V
Impedancja obciążenia	Wej. niesymetryczne: ≥ 100 k Ω	Różnicowe: 220 k Ω (napięcie), 250 Ω (prąd) Tryb sumacyjny: 55 k Ω (napięcie), 55 k Ω (prąd)	Różnicowe: 9 M Ω (napięcie), 250 Ω (prąd) Tryb sumacyjny: 4,5 M Ω (napięcie), 4,5 M Ω (prąd)

Dane techniczne	CPU	SB	SM
Izolacja (sygnału zewnętrznego od logiki)	Brak	Brak	Brak
Długość kabla (w metrach)	100 m, ekranowana para skręconych przewodów	100 m, ekranowana skrętka	100 m ekranowana skrętka
Diagnostyka	Przepiętnie górne i dolne	Przepiętnie górne i dolne	Przepiętnie górne i dolne Za niskie napięcie 24 VDC

Przypis 1: Zobacz zakresy pomiarowe wejść analogowych dla napięcia i prądu (strona 307), aby określić zakres przzerwotu górnego i dolnego oraz zakres przepiętnia górnego i dolnego.

Przypis 2: Zobacz czasy odpowiedzi skokowej (strona 308) aby określić wartości wygładzenia i tłumienia zakłóceń.

A.7.2. Napięciowe i prądowe zakresy pomiarowe dla analogowych wejść (AI)

Tabela A.37. Napięciowa reprezentacja wejścia analogowego

System		Zakres pomiarowy napięcia				
Dziesiętnie	Heksadecymalnie	±10 V	±5 V	±2,5 V	1,25 V	
32767	7FFF	11,851 V	5,926 V	2,963 V	1,481 V	Przepiętnie górne
32512	7F00					
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Zakres przzerwotu od góry
27649	6C01					
27648	6C00	10 V	5 V	2,5 V	1,250 V	Zakres nominalny
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V	
1	1	361,7 μV	180,8 μV	90,4 μV	45,2 μV	
0	0	0 V	0 V	0 V	0 V	
-1	FFFF					
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V	
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V	Zakres przzerwotu od dołu
-27649	93FF					
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	Przepiętnie dolne
-32513	80FF					
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	

Tabela A.38. Prądowa reprezentacja wejścia analogowego (SB oraz SM)

System		Zakres pomiarowy prądu		
Dziesiętnie	Heksadecymalnie	od 0 mA do 20 mA	4 mA do 20 mA	
32767	7FFF	23,70 mA	22,96 mA	Przepiętnie górne
32512	7F00			
32511	7EFF	23,52 mA	22,81 mA	Zakres przzerwotu od góry
27649	6C01			

System		Zakres pomiarowy prądu		
27648	6C00	20 mA	20 mA	Zakres nominalny
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			Zakres przerzutu od dołu
-4864	ED00	-3,52 mA	1,185 mA	
-4865	ECFF			Przepełnienie dolne
-32768	8000			

Tabela A.39. Napięciowa reprezentacja wejścia analogowego (CPU 1215C oraz CPU 1217C)

System		Zakres pomiarowy napięcia		
Dziesiętnie	Heksadecymalnie	0 do 10 V		
32767	7FFF	11,851 V		Przepełnienie górne
32512	7F00			
32511	7EFF	11,759 V		Zakres przerzutu od góry
27649	6C01			
27648	6C00	10 V		Zakres nominalny
20736	5100	7,5 V		
34	22	12 mV		
0	0	0 V		
Wartości ujemne		Wartości ujemne nie są obsługiwane		

A.7.3. Odpowiedź skokowa analogowych wejść (AI)

Poniższa tabela pokazuje czasy odpowiedzi skokowej wejścia analogowego dla CPU, SB oraz SM.

Tabela A.40. Odpowiedź skokowa (ms) wejścia analogowego

Wybór wygładzania (uśrednianie próbek)		Wybór czasu całkowania ¹			
		400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
None (1 cykl): Brak uśredniania	CPU	N/A	63	65	130
	SB	4,5	18,7	22,0	102
	SM	4	18	22	100
Weak (4 cykle): 4 próbki	CPU	N/A	84	93	340
	SB	10,6	59,3	70,8	346
	SM	9	52	63	320
Medium (16 cykli): 16 próbek	CPU	N/A	221	258	1210
	SB	33,0	208	250	1240
	SM	32	203	241	1200

Wybór wygładzania (uśrednianie próbek)		Wybór czasu całkowania ¹			
		400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Strong (32 cykle): 32 próbki	CPU	N/A	424	499	2410
	SB	63,0	408	490	2440
	SM	61	400	483	2410
Sample Rate (częstość próbek)	CPU	N/A	4,17	5	25
	SB	0,156	1,042	1,250	6,250
	SM • (4 kanały) • (8 kanałów)	• 0,625 • 1,25	• 4,17 • 4,17	• 5 • 5	• 25 • 25

¹ Skok od 0 V do 10 V, pomiar dla 95% końcowej wartości (CPU oraz SB), 0 pełnej wartości, pomiar dla 95% końcowej wartości (SM).

A.7.4. Czas próbkowania i czas odświeżania dla analogowych wejść

Tabela A.41. Czas próbkowania i czas odświeżania dla SM i CPU

Czas odświeżania dla wszystkich kanałów (czas całkowania)	Czas próbkowania	Tłumiona częstotliwość		
		4-kanałowy SM	8-kanałowy SM	CPU AI
400 Hz (2,5 ms)	0,625 ms ¹	2,5 ms	10 ms	N/A ms
60 Hz (16,6 ms)	4,17 ms	4,17 ms	4,17 ms	4,17 ms
50 Hz (20 ms)	5,0 ms	5 ms	5 ms	5 ms
10 Hz (100 ms)	25,0 ms	25 ms	25 ms	25 ms

¹ Czas próbkowania dla 8-kanałowego SM wynosi 1,250 ms.

Tabela A.42. Czas próbkowania i czas odświeżania dla SB

Tłumiona częstotliwość (czas całkowania)	Czas próbkowania	Czas odświeżania dla SB
400 Hz (2,5 ms)	0,156 ms	0,156 ms
60 Hz (16,6 ms)	1,042 ms	1,042 ms
50 Hz (20 ms)	1,250 ms	1,25 ms
10 Hz (100 ms)	6,250 ms	6,25 ms

A.7.5. Specyfikacja analogowych wyjść (SM oraz SB)

Tabela A.43. Specyfikacja analogowych wyjść (SB oraz SM)

Dane techniczne	SB	SM
Typ	Napięcie lub prąd	Napięcie lub prąd
Zakres	±10 V, 0 do 20 mA, lub 4 do 20 mA	±10 V, 0 do 20 mA, lub 4 do 20 mA
Rozdzielczość	Napięcie: 12 bitów Prąd: 11 bitów	Napięcie: 14 bitów Prąd: 13 bitów
Zakres pomiarowy (słowo danych) (Przypis 1)	Napięcie: od -27,648 do 27,648 Prąd: od 0 do 27,648	Napięcie: od -27,648 do 27,648 Prąd: od 0 do 27,648
Dokładność (25°C/-20 do 60°C)	±0,5%/±1% pełnego zakresu	±0,3%/±0,6% pełnego zakresu

Dane techniczne	SB	SM
Czas ustalania (do 95% nowej wartości)	Napięcie: 300 μ S (R), 750 μ S (1 uF) Prąd: 600 μ S (1 mH), 2 ms (10 mH)	Napięcie: 300 μ S (R), 750 μ S (1 uF) Prąd: 600 μ S (1 mH), 2 ms (10 mH)
Impedancja obciążenia	Napięcie: \geq 1000 Ω Prąd: \leq 600 Ω	Napięcie: \geq 1000 Ω Prąd: \leq 600 Ω
Zachowanie przy przejściu z RUN do STOP	Ostatnia wartość lub wartość zastępcza (domyślnie 0)	Ostatnia wartość lub wartość zastępcza (domyślnie 0)
Izolacja (sygnału zewnętrznego od logiki)	brak	brak
Długość kabla (w metrach)	100 m, ekranowana skrętka	100 m, ekranowana skrętka
Diagnostyka	<ul style="list-style-type: none"> Przepiętnie górne i dolne Zwarcie do uziemienia (tylko tryb napięciowy) Przerwa przewodu (tylko tryb prądowy) 	<ul style="list-style-type: none"> Przepiętnie górne i dolne Zwarcie do uziemienia (tylko tryb napięciowy) Przerwa przewodu (tylko tryb prądowy) Za niskie napięcie 24 VDC
Przypis 1: Zobacz zakresy prądowe i napięciowe dla analogowych wyjść (strona 310) dla określenia pełnego zakresu.		

A.7.6. Napięciowe i prądowe zakresy pomiarowe dla analogowych wyjść (AQ)

Tabela A.44. Napięciowa reprezentacja wyjścia analogowego (SB oraz SM)

System		Zakres pomiarowy napięcia	
Dziesiętnie	Heksadecymalnie	\pm 10 V	
32767	7FFF	Przypis 1	Przepiętnie górne
32512	7F00	Przypis 1	
32511	7EFF	11,76 V	Zakres przerzutu od góry
27649	6C01		
27648	6C00	10 V	Zakres nominalny
20736	5100	7,5 V	
1	1	361,7 μ V	
0	0	0 V	
-1	FFFF	-361,7 μ V	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		Zakres przerzutu od dołu
-32512	8100	-11,76 V	
-32513	80FF	Przypis 1	Przepiętnie dolne
-32768	8000	Przypis 1	

¹ W przypadku wystąpienia przepiętnia górnego lub dolnego, wartości na wyjściach analogowych zostaną zastąpione wartościami dla trybu STOP.

Tabela A.45. Prądowa reprezentacja wyjścia analogowego

System		Zakres pomiarowy prądu	
Dziesiętnie	Heksadecymalnie	Od 0 mA do 20 mA	
32767	7FFF	Przypis 1	Przepełnienie górne
32512	7F00	Przypis 1	
32511	7EFF	23,52 mA	Zakres przerzutu od góry
27649	6C01		
27648	6C00	20 mA	Zakres nominalny
20736	5100	15 mA	
1	1	723,4 nA	
0	0	0 mA	

¹ W przypadku wystąpienia przepełnienia górnego lub dolnego, wartości na wyjściach analogowych zostaną zastąpione wartościami dla trybu STOP.

Tabela A.46. Prądowa reprezentacja wyjścia analogowego (CPU 1215C oraz CPU 1217C)

System		Zakres pomiarowy prądu	
Dziesiętnie	Heksadecymalnie	0 mA do 20 mA	
32767	7FFF	Przypis 1	Przepełnienie górne
32512	7F00	Przypis 1	
32511	7EFF	23,52 mA	Zakres przerzutu od góry
27649	6C01		
27648	6C00	20 mA	Zakres nominalny
20736	5100	15 mA	
34	22	12 mV	
0	0	0 mA	
Wartości ujemne		Wartości ujemne nie są obsługiwane	

¹ W przypadku wystąpienia przepełnienia górnego, wyjścia analogowe będą się zachowywać zgodnie z ustawionymi właściwościami konfiguracji urządzenia. Dla parametru *Reaction to CPU STOP* (Reakcja na przejście CPU do trybu STOP), można wybrać opcję albo *Use substitute value* (Użyj wartości zastępczej), albo *Keep last value* (Zachowaj ostatnią wartość).

A.8. Moduły termopar i RTD

Moduły termopar (TC: SB 1231 TC oraz SM 1231 TC) mierzą wartość napięcia podłączonego do analogowych wejść. Wartość ta może być wyrażona zarówno w Voltach jak i w stopniach Celsjusza.

- Jeśli wartość jest wyrażona w Voltach, to zakres nominalny wartości będzie wynosił 27 648 (dziesiętnie).
- Jeśli odczytywana jest temperatura, to jej wartość będzie zgłaszana w stopniach jako dziesięciokrotność wartości zmierzonej (np. 25,3 stopni będzie wyrażone jako 253 (dziesiętnie)).

Moduły RTD (SB 1231 RTD oraz SM 1231 RTD) mierzą wartość rezystancji podłączonej do analogowych wejść. Wartość ta może być wyrażona zarówno w Voltach jak i w stopniach Celsjusza.

- Jeśli wartość jest wyrażona w Voltach, to zakres nominalny wartości będzie wynosił 27648 (dziesiętnie).
- Jeśli odczytywana jest temperatura, to jej wartość będzie zgłaszana w stopniach jako dziesięciokrotność wartości zmierzonej (np. 25,3 stopni będzie wyrażone jako 253 (dziesiętnie)).

RTD umożliwia pomiary z termometrem oporowym wykonanym w technologii 2-, 3-, oraz 4-przewodowej.

Uwaga

Jeśli do aktywnego kanału modułu termopary lub RTD nie został podłączony żaden czujnik, to moduł zgłosi na tym kanale wartość 32 767. Jeśli załączone jest także wykrywanie przerwy w obwodzie, moduł zapali odpowiednią czerwoną diodę LED.

Jeśli użyto RTD o zakresach 500 Ω oraz 1000 Ω z rezystorami o niższych opornościach, błąd pomiarowy może być do dwóch razy większy niż określony w specyfikacji RTD. Największa dokładność zostanie osiągnięta dla zakresu RTD wynoszącego 10 Ω , jeśli użyto połączenia 4-przewodowego.

Oporność połączenia w trybie 2-przewodowym spowoduje błąd odczytu z czujnika, więc duża dokładność może nie zostać osiągnięta.

Uwaga

Po podłączeniu zasilania moduł przeprowadza wstępną kalibrację przetwornika analogowo-cyfrowego. Podczas tej kalibracji moduł zgłasza wartość 32 767 na każdym kanale. Dzieje się tak aż do momentu, w którym właściwe dane będą dla niego dostępne. Program użytkownika musi uwzględnić czas potrzebny na kalibrację. Ponieważ konfiguracja modułu może zmieniać długość czasu inicjalizacji, należy sprawdzić zachowanie modułu w czasie kalibracji. Użytkownik może dodać odpowiednią logikę do swojego programu w celu uwzględnienia czasu inicjalizacji modułu.

A.8.1. Specyfikacja modułów SB 1231 RTD oraz SB 1231 TC

Uwaga

Oprogramowanie sprzętowe CPU musi być w wersji V2.0 lub wyższej, aby było możliwe korzystanie z płytek sygnałowych TC i RTD.

Tabela A.47. Ogólna specyfikacja

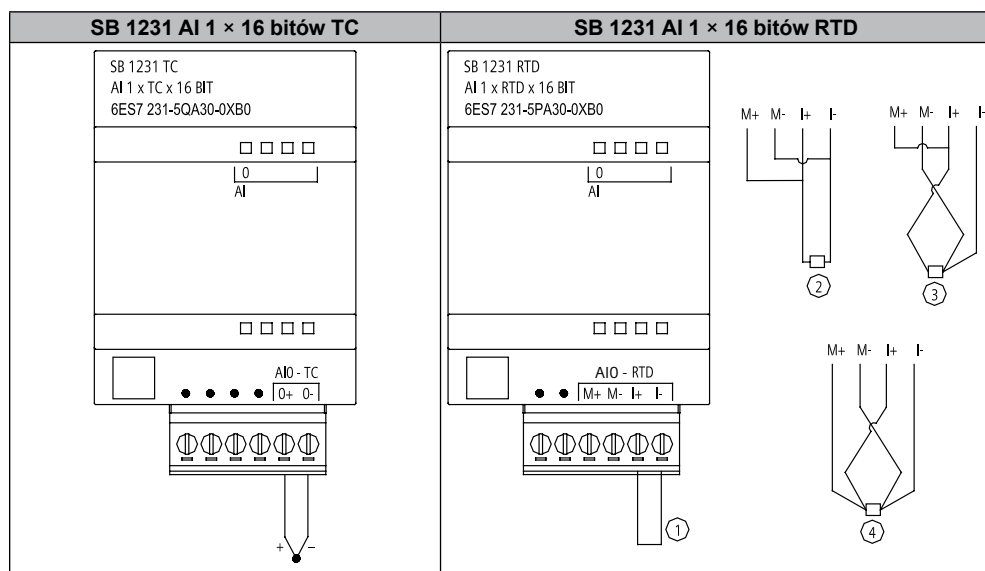
Dane techniczne	SB 1231 AI 1 × 16 bitów TC	SB 1231 AI 1 × 16 bitów RTD
Nr zamówieniowy	6ES7 231-5QA30-0XB0	6ES7 231-5PA30-0XB0
Wymiary W × H × D (mm)	38 × 62 × 21 mm	38 × 62 × 21 mm
Masa	35 gramów	35 gramów
Pobór mocy	0,5 W	0,7 W
Pobór prądu (magistrala SM)	5 mA	5 mA
Pobór prądu (24 VDC)	20 mA	25 mA
Liczba wejść (strona 316)	1	1
Typ	Nieziemiona TC/mV	Odniesione do modułu RTD/Ω
Diagnostyka	<ul style="list-style-type: none"> Przepełnienie górne i dolne^{1,2} Przerwanie obwodu³ 	<ul style="list-style-type: none"> Przepełnienie górne i dolne^{1,2} Przerwanie obwodu³

¹ Informacja o alarmie diagnostycznym z powodu przepełnienia górnego lub dolnego będzie zgłoszona w wartościach danych analogowych nawet jeśli alarmy zostały wyłączone podczas konfiguracji modułu.

² RTD: Wykrywanie przepełnienia dolnego nie jest nigdy aktywne dla zakresów rezystancyjnych.

³ Jeśli zgłaszanie alarmu z powodu przerwy w obwodzie jest wyłączone, a w okablowaniu czujnika występuje przerwa, to moduł może zgłaszać na poszczególnych kanałach losowe wartości.

Tabela A.48. Schemat podłączenia płytki SB 1231 TC/RTD



① Pętla zwrotna nieużywanego wejścia RTD

② 2-przewodowa RTD ③ 3- przewodowa RTD ④ 4- przewodowa RTD

A.8.2. Specyfikacja modułu SM 1231 RTD

Tabela A.49. Ogólna specyfikacja

Dane techniczne	SM 1231 AI 4 × RTD × 16 bit	SM 1231 AI 8 × RTD × 16 bit
Nr zamówieniowy	6ES7 231-5PD32-0XB0	6ES7 231-5PF32-0XB0
Wymiary W × H × D (mm)	45 × 100 × 75	70 × 100 × 75
Masa	220 gramów	270 gramów
Pobór mocy	1,5 W	1,5 W
Pobór prądu (magistrala SM)	80 mA	90 mA
Pobór prądu ¹ (24 VDC)	40 mA	40 mA
Liczba wejść (strona 316) Typ	4 Odniesione do modułu RTD/Ω	8 Odniesione do modułu RTD/Ω
Diagnostyka	<ul style="list-style-type: none"> Przepełnienie górne i dolne^{1, 2} Za niskie napięcie 24 VDC² Przerwanie obwodu (tylko tryb prądowy)⁴ 	<ul style="list-style-type: none"> Przepełnienie górne i dolne^{1, 2} Za niskie napięcie 24 VDC² Przerwanie obwodu (tylko tryb prądowy)⁴

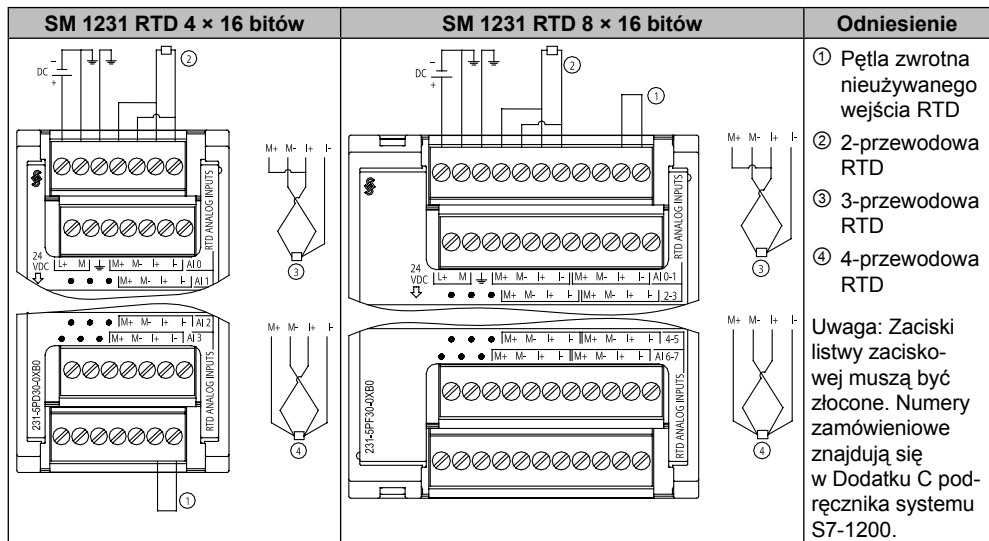
¹ 20,4 do 28,8 VDC (Class 2, ograniczone zasilanie lub zasilanie czujnika z CPU).

² Informacja o alarmie diagnostycznym z powodu przepełnienia górnego lub dolnego będzie zgłoszona w wartościach danych analogowych nawet jeśli alarmy zostały wyłączone podczas konfiguracji modułu.

³ Wykrywanie przepełnienia dolnego nie jest nigdy aktywne dla zakresów rezystancyjnych.

⁴ Jeśli zgłaszanie alarmu z powodu przerwy w obwodzie jest wyłączone, a w okablowaniu czujnika występuje przerwa, to moduł może zgłaszać na poszczególnych kanałach losowe wartości.

Tabela A.50. Schemat podłączeń modułu rozszerzeń RTD



A.8.3. Specyfikacja modułu SM 1231 TC

Tabela A.51. Ogólna specyfikacja

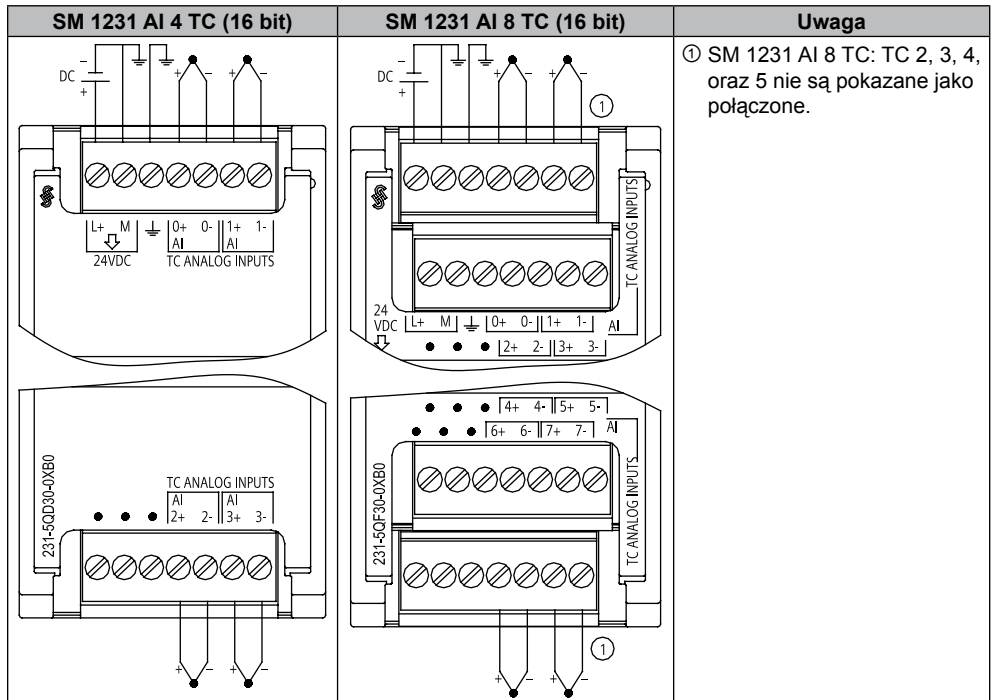
Dane techniczne	SM 1231 AI4 × 16 bit TC	SM 1231 AI8 × 16 bit TC
Nr zamówieniowy	6ES7 231-5QD32-0XB0	6ES7 231-5QF32-0XB0
Wymiary W × H × D (mm)	45 × 100 × 75	45 × 100 × 75
Masa	180 gramów	xxx gramów
Pobór mocy	1,5 W	1,5 W
Pobór prądu (magistrala SM)	80 mA	80 mA
Pobór prądu ¹ (24 VDC)	40 mA	40 mA
Liczba wejść (strona 316)	4	8
Typ	Nieuziemiona TC/mV	Nieuziemiona TC/mV
Diagnostyka	<ul style="list-style-type: none"> Przepiętnie górne i dolne ² Za niskie napięcie 24 VDC ² Przerwanie obwodu (tylko tryb prądowy)³ 	<ul style="list-style-type: none"> Przepiętnie górne i dolne ² Za niskie napięcie 24 VDC ² Przerwanie obwodu (tylko tryb prądowy)³

¹ Od 20,4 do 28,8 VDC (Class 2, ograniczone zasilanie lub zasilanie czujnika z CPU).

² Informacja o alarmie diagnostycznym z powodu przepiętnia górnego lub dolnego będzie zgłoszona w wartościach danych analogowych nawet jeśli alarmy zostały wyłączone podczas konfiguracji modułu.

³ Jeśli zgłaszanie alarmu z powodu przerwy w obwodzie jest wyłączone, a w okablowaniu czujnika występuje przerwa, to moduł może zgłaszać na poszczególnych kanałach losowe wartości.

Tabela A.52. Schemat połączeń modułu rozszerzeń TC



A.8.4. Specyfikacja analogowego wejścia dla RTD oraz TC (SM i SB)

Tabela A.53. Analogowe wejścia dla modułów RTD oraz TC (SM i SB)

Dane techniczne		RTD i Termopary (TC)
Liczba wejść		1 (SB), 4 lub 8 (SM)
Typ		<ul style="list-style-type: none"> RTD: Odniesione do modułu RTD/Ω TC: Nieuziemiona TC/mV
Zakres <ul style="list-style-type: none"> Zakres nominalny (słowo danych) Zakres przerzutu od góry i od dołu (słowo danych) Przepelnienie górne i dolne (słowo danych) 		Zobacz tabele typów RTD/TC: <ul style="list-style-type: none"> RTD (strona 318) TC (strona 317)
Rozdzielczość	Temperatura	0,1°C/0,1°F
	Rezystancja/napięcie	15 bitów plus bit znaku
Maksymalne bezpieczne napięcie		± 35 V
Tłumienie zakłóceń		85 dB dla wybranych ustawień filtra (10 Hz, 50 Hz, 60 Hz lub 400 Hz)
Tłumienie sygnału sumacyjnego		> 120 dB przy 120 VAC
Impedancja		≥ 10 M Ω
Izolacja	Sygnału zewnętrznego od logiki	500 VAC
	Poła od 24 VDC	SM RTD oraz SM TC: 500 VAC (nie stosuje się dla SB RTD i SB TC)
	24 VDC od logiki	SM RTD oraz SM TC: 500 VAC (nie stosuje się dla SB RTD i SB TC)
Izolacja pomiędzy kanałami		<ul style="list-style-type: none"> SM RTD: Brak (nie stosuje się dla SB RTD) SM TC: 120 VAC (nie stosuje się dla SB TC)
Dokładność (od 25°C/-20 do 60°C)		Zobacz tabele typów RTD/TC: <ul style="list-style-type: none"> RTD (strona 318) TC (strona 317)
Powtarzalność		$\pm 0,05\%$ FS
Maksymalny pobór mocy przez czujnik		<ul style="list-style-type: none"> RTD: 0,5 mW TC: Nie dotyczy
Zasada pomiaru		Całkowanie
Czas odświeżania modułu		Zobacz tabele wyboru filtrów RTD/TC: <ul style="list-style-type: none"> RTD (strona 319) TC (strona 317)
Błąd styku odniesienie		<ul style="list-style-type: none"> RTD: Nie dotyczy TC: $\pm 1,5^\circ\text{C}$
Długość kabla (w metrach)		Maks. 100 m. do czujnika
Opór przewodu		<ul style="list-style-type: none"> RTD: 20 Ω, maks. 2,7 Ω dla 10 Ω RTD. TC: maks. 100 Ω.

A.8.5. Typy termopar

Tabela A.54. Typy termopar (zakresy i dokładność)

Typ	Min. dolny zakres ¹	Nominalny dolny zakres	Nominalny górny zakres	Maks. górny zakres ²	Dokładność przy normalnym zakresie ^{3,4} przy 25°C	Dokładność przy normalnym zakresie ^{3,4} od -20°C do 60°C
J	-210,0°C	-150,0°C	1200,0°C	1450,0°C	±0,3°C	±0,6°C
K	-270,0°C	-200,0°C	1372,0°C	1622,0°C	±0,4°C	±1,0°C
T	-270,0°C	-200,0°C	400,0°C	540,0°C	±0,5°C	±1,0°C
E	-270,0°C	-200,0°C	1000,0°C	1200,0°C	±0,3°C	±0,6°C
R & S	-50,0°C	100,0°C	1768,0°C	2019,0°C	±1,0°C	±2,5°C
N	-270,0°C	-200,0°C	1300,0°C	1550,0°C	±1,0°C	±1,6°C
C	0,0°C	100,0°C	2315,0°C	2500,0°C	±0,7°C	±2,7°C
TXK/XK(L)	-200,0°C	-150,0°C	800,0°C	1050,0°C	±0,6°C	±1,2°C
Napięcie	-32512	-27648 -80 mV	27648 80 mV	32 511	±0,05%	±0,1%

¹ Wartości z termopar poniżej minimalnego dolnego zakresu zgłaszane są jako -32 768.

² Wartości z termopar powyżej maksymalnego górnego zakresu zgłaszane są jako 32 767,

³ Wewnętrzny błąd styku odniesienia wynosi ±1,5°C dla wszystkich zakresów. Wartość ta jest dodawana do błędów ujętych w powyższej tabeli. Moduł wymaga przynajmniej 30 minut nagrzewania, aby mógł spełnić te warunki.

⁴ Tylko dla 4-kanalowego SM TC: Dokładność wyników może spaść w przypadku obecności w pobliżu czujników nadajników częstotliwości radiowych (częstotliwości 970–990 MHz).

Uwaga

Kanał termopary

Dla każdego kanału modułu SM termopary można skonfigurować inny typ termopary (do wyboru w programie podczas konfiguracji modułu).

A.8.6. Wybór filtra termopary i czasy odświeżania

Do pomiarów termoparami sugerowane jest użycie czasu całkowania wynoszącego 100 ms. Użycie mniejszych wartości zwiększy błąd powtarzalności odczytu temperatury.

Tabela A.55. Wybór filtra termopary i czasy odświeżania

Tłumiona częstotliwość [Hz]	Czas całkowania [ms]	Czas odświeżania [s]		
		1-kanalowa SB	4-kanalowy SM	8-kanalowy SM
10	100	0,301	1,225	2,450
50	20	0,061	0,263	0,525
60	16,67	0,051	0,223	0,445
400 ¹	10	0,031	0,143	0,285

¹ Aby zachować rozdzielczość i dokładność modułu, jeśli tłumiona jest częstotliwość 400 Hz, należy użyć czasu całkowania wynoszącego 10 ms. Taka wartość pozwoli również tłumić szumy o częstotliwości 100 Hz oraz 200 Hz.

A.8.7. Tabela wyboru typu czujnika RTD

Tabela A.56. Zakresy pomiarowe i dokładność różnych czujników obsługiwanych przez moduły RTD

Współczynnik temperaturowy	Typ RTD	Min. dolny zakres ¹	Nominalny dolny zakres	Nominalny górny zakres	Maks. górny zakres ²	Dokładność przy normalnym zakresie przy 25°C	Dokładność od -20°C do 60°C Zakres normalny
Pt 0,003850 ITS90 DIN EN 60751	Pt 100 climatic	-145,00°C	-120,00°C	145,00°C	155,00°C	±0,20°C	±0,40°C
	Pt 10	-243,0°C	-200,0°C	850,0°C	1000,0°C	±1,0°C	±2,0°C
	Pt 50	-243,0°C	-200,0°C	850,0°C	1000,0°C	±0,5°C	±1,0°C
	Pt 100						
	Pt 200						
	Pt 500						
Pt 1000							
Pt 0,003902 Pt 0,003916 Pt 0,003920	Pt 100	-243,0°C	-200,0°C	850,0°C	1000,0°C	±0,5°C	±1,0°C
	Pt 200	-243,0°C	-200,0°C	850,0°C	1000,0°C	±0,5°C	±1,0°C
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2°C	-240,0°C	1100,0°C	1295°C	±1,0°C	±2,0°C
	Pt 50	-273,2°C	-240,0°C	1100,0°C	1295°C	±0,8°C	±1,6°C
	Pt 100						
	Pt 500						
Ni 0,006720 Ni 0,006180	Ni 100	-105,0°C	-60,0°C	250,0°C	295,0°C	±0,5°C	±1,0°C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0,005000	LG-Ni 1000	-105,0°C	-60,0°C	250,0°C	295,0°C	±0,5°C	±1,0°C
Ni 0,006170	Ni 100	-105,0°C	-60,0°C	180,0°C	212,4°C	±0,5°C	±1,0°C
Cu 0,004270	Cu 10	-240,0°C	-200,0°C	260,0°C	312,0°C	±1,0°C	±2,0°C
		-60,0°C	-50,0°C	200,0°C	240,0°C	±1,0°C	±2,0°C
		-60,0°C	-50,0°C	200,0°C	240,0°C	±0,6°C	±1,2°C
Cu 0,004280	Cu 10	-240,0°C	-200,0°C	200,0°C	240,0°C	±1,0°C	±2,0°C
		-240,0°C	-200,0°C	200,0°C	240,0°C	±0,7°C	±1,4°C
		-240,0°C	-200,0°C	200,0°C	240,0°C	±0,7°C	±1,4°C

¹ Wartości z RTD poniżej minimalnego dolnego zakresu zgłaszane są jako -32 768.² Wartości RTD powyżej maksymalnej wartości nadzakresu są podawane jako +32 767.

Tabela A.57. Rezystancje

Zakres	Min. dolny zakres	Nominalny dolny zakres	Nominalny górny zakres	Maks. górny zakres ¹	Dokładność przy normalnym zakresie przy 25°C	Dokładność od -20°C do 60°C Zakres normalny
150 Ω	n/a	0 (0 Ω)	27 648 (150 Ω)	176,383 Ω	±0,05%	±0,1%
300 Ω	n/a	0 (0 Ω)	27 648 (300 Ω)	352,767 Ω	±0,05%	±0,1%
600 Ω	n/a	0 (0 Ω)	27 648 (600 Ω)	705,534 Ω	±0,05%	±0,1%

¹ Wartości rezystancji powyżej maksymalnej wartości górnego zakresu są podawane jako +32 767.

A.8.8. Wybór filtra RTD i czasy odświeżania

Tabela A.58. Wybór filtra RTD i czasy odświeżania

Tłumiona częstotliwość [Hz]	Czas całkowania [ms]	Czas odświeżania [s]		
		1-kanałowa SB	4-kanałowy SM	8-kanałowy SM
10	100	4-/2-przewodowa: 0,301 3-przewodowa: 0,601	4-/2-przewodowa: 1,222 3-przewodowa: 2,445	4-/2-przewodowa: 2,445 3-przewodowa: 4,845
50	20	4-/2-przewodowa: 0,061 3-przewodowa: 0,121	4-/2-przewodowa: 0,262 3-przewodowa: .505	4-/2-przewodowa: 0,525 3-przewodowa: 1,015
60	16,67	4-/2-przewodowa: 0,051 3-przewodowa: 0,101	4-/2-przewodowa: 0,222 3-przewodowa: 0,424	4-/2-przewodowa: 0,445 3-przewodowa: 0,845
400 ¹	10	4-/2-przewodowa: 0,031 3-przewodowa: 0,061	4-/2-przewodowa: 0,142 3-przewodowa: 0,264	4-/2-przewodowa: 0,285 3-przewodowa: 0,525

¹ Aby zachować rozdzielczość i dokładność modułu, jeśli tłumiona jest częstotliwość 400 Hz, należy użyć czasu całkowania wynoszącego 10 ms. Taka wartość pozwoli również tłumić szumy o częstotliwości 100 Hz oraz 200 Hz.

Uwaga

Na każdym włączonym kanale bez podłączonego czujnika moduł podaje wartość 32767. Jeśli włączone jest wykrywanie rozwartego przewodu, to na module migają odpowiednie czerwone diody LED.

Najlepszą dokładność dla zakresu RTD 10 Ω zapewnia połączenie 4-przewodowe.

Rezystancja przewodów połączeniowych w trybie 2-przewodowym spowoduje błąd w pomiarze czujnika i dlatego dokładność nie jest gwarantowana.

A.9. Interfejsy komunikacyjne

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku systemu S7-1200 lub na internetowej stronie wsparcia klienta (<http://www.siemens.com/automation/>).

A.9.1. PROFIBUS master/slave

A.9.1.1. Moduł CM 1242-5 PROFIBUS slave

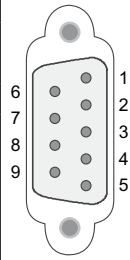
Tabela A.59. Specyfikacja techniczna modułu CM 1242-5

Dane techniczne	
Nr zamówieniowy	6GK7 242-5DX30-0XE0
Interfejsy	
Połączenie z PROFIBUS	9-pinowe złącze żeńskie D-sub
Maksymalny pobór prądu przez interfejs PROFIBUS, podczas gdy komponenty sieciowe są połączone (np. komponenty sieci optycznej)	15 mA przy 5 V (tylko zakończenia magistrali) ^{*)}
Dane techniczne	
Nr zamówieniowy	6GK7 242-5DX30-0XE0
Dopuszczalne warunki otoczenia	
Temperatura otoczenia <ul style="list-style-type: none"> • Podczas magazynowania • Podczas transportu • Podczas pracy w pozycji pionowej (pozioma szyna DIN) • Podczas pracy w pozycji poziomej (pionowa szyna DIN) 	<ul style="list-style-type: none"> • od -40°C do 70°C • od -40°C do 70°C • od 0°C do 55°C • od 0°C do 45°C
Względna wilgotność przy temperaturze 25°C, podczas pracy urządzenia, bez kondensacji, maksymalnie	95%
Stopień ochrony	IP20
Zasilanie, pobór mocy i straty mocy	
Typ zasilania	DC
Zasilanie z tylniej magistrali	5 V
Pobór prądu (typowo)	150 mA
Efektywna strata mocy (typowo)	0,75 W
Wymiary i masa	
<ul style="list-style-type: none"> • Szerokość (W) • Wysokość (H) • Głębokość (D) 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm
Masa <ul style="list-style-type: none"> • Masa netto • Masa wraz z opakowaniem 	<ul style="list-style-type: none"> • 115 g • 152 g

*) Prąd obciążenia zewnętrznego odbiornika połączonego pomiędzy VP (pin 6) oraz DGND (pin 5) nie może przekraczać 15 mA (zabezpieczenie przed zwarcim) dla zakończenia magistrali.

Interfejs PROFIBUS

Tabela A.60. Wyprowadzenia z gniazda D-sub

	Pin	Opis	Pin	Opis
	1	– Nieużywane –	6	P5V2: zasilanie +5 V
2	– Nieużywane –	7	– Nieużywane –	
3	RxD/TxD-P: Linia danych B	8	RxD/TxD-N: Linia danych A	
4	RTS	9	– Nieużywane –	
5	M5V2: Potencjał odniesienia danych (uziemiaenie DGND)	Obudowa	Masa obudowy	

A.9.1.2. Moduł CM 1243-5 PROFIBUS master

Tabela A.61. Specyfikacja techniczna modułu CM 1243-5

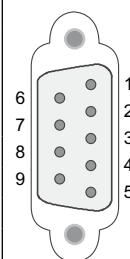
Dane techniczne	
Nr zamówieniowy	6GK7 243-5DX30-0XE0
Interfejsy	
Połączenie z PROFIBUS	9-pinowe złącze żeńskie D-sub
Maksymalny pobór prądu przez interfejs PROFIBUS, podczas gdy komponenty sieciowe są połączone (np. komponenty sieci optycznej)	15 mA przy 5 V (tylko zakończenia magistrali *)
Dopuszczalne warunki otoczenia	
Temperatura otoczenia <ul style="list-style-type: none"> • Podczas magazynowania • Podczas transportu • Podczas pracy w pozycji pionowej (pozioma szyna DIN) • Podczas pracy w pozycji poziomej (pionowa szyna DIN) 	<ul style="list-style-type: none"> • –40°C do 70°C • –40°C do 70°C • 0 C do 55 C • 0°C do 45°C
Względna wilgotność przy temperaturze 25 °C, podczas pracy urządzenia, bez kondensacji, maksymalnie	95 %
Stopień ochrony	IP20
Zasilanie, pobór mocy i straty mocy	
Typ zasilania	DC
Zasilanie/zewnętrzne <ul style="list-style-type: none"> • minimum • maksimum 	24 V <ul style="list-style-type: none"> • 19,2 V • 28,8 V
Pobór prądu (typowy) <ul style="list-style-type: none"> • z 24 V DC • z tylnej magistrali S7-1200 	<ul style="list-style-type: none"> • 100 mA • 0 mA
Efektywna strata mocy (typowo) <ul style="list-style-type: none"> • z 24 V DC • z tylnej magistrali S7-1200 	<ul style="list-style-type: none"> • 2,4 W • 0 W
Zasilanie 24 VDC/zewnętrzne <ul style="list-style-type: none"> • Min. przekrój przewodu • Maks. przekrój przewodu • Moment dokręcania zacisków terminali 	<ul style="list-style-type: none"> • min.: 0,14 mm² (AWG 25) • maks.: 1,5 mm² (AWG 15) • 0,45 Nm (4 lb-in)

Dane techniczne	
Wymiary i masa	
<ul style="list-style-type: none"> Szerokość(W) Wysokość (H) Głębokość (D) 	<ul style="list-style-type: none"> 30 mm 100 mm 75 mm
Masa	
<ul style="list-style-type: none"> Masa netto Masa wraz z opakowaniem 	<ul style="list-style-type: none"> 134 g 171 g

*) Prąd obciążenia zewnętrznego odbiornika połączonego pomiędzy VP (pin 6) oraz DGND (pin 5) nie może przekraczać 15 mA (zabezpieczenie przed zwarcie) dla zakończenia magistrali.

Interfejs PROFIBUS

Tabela A.62. Wyprowadzenia z gniazda D-sub

	Pin	Opis	Pin	Opis
	1	1	– Nieużywane –	6
2	2	– Nieużywane –	7	– Nieużywane –
3	3	RxD/TxD-P: Linia danych B	8	RxD/TxD-N: Linia danych A
4	4	CNTR-P: RTS	9	– Nieużywane –
5	5	DGND: masa dla sygnałów z danymi VP	Obudowa	Masa obudowy

Przewód PROFIBUS

Uwaga
<p>Uziemienie ekranu przewodu PROFIBUS</p> <p>Ekran przewodu PROFIBUS musi być uziemiony.</p> <p>W tym celu należy ściągnąć izolację z końca przewodu i podłączyć ekran do uziemienia.</p>

A.9.2. Procesor komunikacyjny GPRS

Uwaga

Procesor komunikacyjny GPRS CP nie jest dopuszczony do zastosowań morskich

Moduł wymieniony poniżej nie ma dopuszczenia morskiego:

- Moduł CP 1242-7 GPRS

Uwaga

Oprogramowanie sprzętowe CPU musi być w wersji V2.0 lub wyższej, aby było możliwe korzystanie z tych modułów.

A.9.2.1. Specyfikacja techniczna CP 1242-7

Tabela A.63. Specyfikacja techniczna CP 1242-7

Dane techniczne	
Nr zamówieniowy	6GK7 242-7KX30-0XE0
Interfejs bezprzewodowy	
Złącze antenowe	Złącze SMA
Nominalna impedancja	50 Ω
Połączenie bezprzewodowe	
Maksymalna moc nadawania	<ul style="list-style-type: none"> • GSM 850, klasa 4: +33 dBm ±2dBm • GSM 900, klasa 4: +33 dBm ±2dBm • GSM 1800, klasa 1: +30 dBm ±2dBm • GSM 1900, klasa 1: +30 dBm ±2dBm
GPRS	Klasa Multislot 10 klasa urządzenia B system kodowania 1...4 (GMSK)
SMS	Tryb połączeń wychodzących: MO Usługa: point-to-point
Dopuszczalne warunki otoczenia	
Temperatura otoczenia <ul style="list-style-type: none"> • Podczas magazynowania • Podczas transportu • Podczas pracy w pozycji pionowej (pozioma szyna DIN) • Podczas pracy w pozycji poziomej (pionowa szyna DIN) 	<ul style="list-style-type: none"> • od -40°C do 70°C • od -40°C do 70°C • od 0 C do 55 C • od 0°C do 45°C
Względna wilgotność przy temperaturze 25°C, podczas pracy urządzenia, bez kondensacji, maksymalnie	95%
Stopień ochrony	IP20
Zasilanie, pobór mocy i straty mocy	
Typ zasilania	DC
Zasilanie/zewnętrzne <ul style="list-style-type: none"> • minimum • maksimum 	24 V <ul style="list-style-type: none"> • 19,2 V • 28,8 V
Pobór prądu (typowy) <ul style="list-style-type: none"> • z 24 V DC • z tylniej magistrali S7-1200 	<ul style="list-style-type: none"> • 100 mA • 0 mA
Efektywna strata mocy (typowo) <ul style="list-style-type: none"> • z 24 V DC • z tylniej magistrali S7-1200 	<ul style="list-style-type: none"> • 2,4 W • 0 W
Zasilanie 24 VDC/zewnętrzne <ul style="list-style-type: none"> • Min. przekrój przewodu • Maks. przekrój przewodu • Moment dokręcania zacisków terminali 	<ul style="list-style-type: none"> • min.: 0,14 mm² (AWG 25) • maks.: 1,5 mm² (AWG 15) • 0,45 Nm (4 lb-in)
Izolacja Zasilacz do obwodu wewnętrznego	710 VDC przez 1 minutę

Dane techniczne	
Wymiary i masa	
<ul style="list-style-type: none"> • Szerokość(W) • Wysokość (H) • Głębokość (D) 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm
Masa	
<ul style="list-style-type: none"> • Masa netto • Masa wraz z opakowaniem 	<ul style="list-style-type: none"> • 133 g • 170 g

Specyfikacja techniczna anteny ANT794-4MR GSM/GPRS

Dane techniczne	
Nr zamówieniowy	6NH9860-1AA00
Mobilne sieci bezprzewodowe	GSM/GPRS
Zakresy częstotliwości	<ul style="list-style-type: none"> • 824 do 960 MHz (GSM 850, 900) • 1710 do 1880 MHz (GSM 1 800) • 1900 do 2200 MHz (GSM/UMTS)
Charakterystyki	wielokierunkowe
Wzmocnienie anteny	0 dB
Impedancja	50 Ω
SWR	< 2,0
Maksymalna moc	20 W
Biegunowość	Liniowa pionowa
Złącze	SMA
Długość kabla antenowego	5 m
Materiały obudowy	Twarde PVC, odporne na promienie UV
Stopień ochrony	IP20
Dopuszczalne warunki otoczenia	
<ul style="list-style-type: none"> • Temperatura pracy • Temperatura podczas transportu/ magazynowania • Względna wilgotność 	<ul style="list-style-type: none"> • od -40°C do +70°C • od -40°C do +70°C • 100%
Konstrukcja	Antena z przymocowanym przewodem(5 m) i złączem SMA
Wymiary (D × H) w mm	25 × 193
Waga	
<ul style="list-style-type: none"> • Antena z przewodem • Śrubunek 	<ul style="list-style-type: none"> • 310 g • 54 g
Instalacja	Za pomocą dołączonych zatrzasków

Specyfikacja techniczna anteny ANT794-3M

Nr zamówieniowy	6NH9870-1AA00	
Mobilne sieci bezprzewodowe	GSM 900	GSM 1800/1900
Zakresy częstotliwości	890-960 MHz	1710-1990 MHz
VSWR	≤ 2:1	≤ 1,5:1
Tłumienie odbić (Tx)	≈ 10 dB	≈ 14 dB
Wzmocnienie anteny	0 dB	

Nr zamówieniowy	6NH9870-1AA00
Impedancja	50 Ω
Maksymalna moc	10 W
Kabel antenowy	HF przewód RG 174 (stały) z męskim złączem SMA
Długość przewodu	1,2 m
Stopień ochrony	IP64
Dozwolony zakres temperatur	od -40°C do $+75^{\circ}\text{C}$
Łatwopalność	UL 94 V2
Materiały obudowy	ABS Polylac PA-765, jasnoszary (RAL 7035)
Wymiary (W \times L \times H) w mm	70,5 \times 146,5 \times 20,5
Masa	130 g

A.9.3. TeleService (TS)

Następujące podręczniki zawierają specyfikację techniczną dla urządzenia TS Adapter IE Basic oraz modularnego urządzenia TS Adapter:

- Industrial Software Engineering Tools Modular TS Adapter
- Industrial Software Engineering Tools TS Adapter IE Basic

Więcej informacji o tym produkcie oraz jego dokumentacja znajduje się na internetowej stronie katalogów produktów dla urządzenia TS Adapter.

A.9.4. Komunikacja poprzez RS485 i RS232

A.9.4.1. Specyfikacja CB 1241 RS485

Uwaga

Oprogramowanie sprzętowe CPU musi być w wersji V2.0 lub wyższej, aby było możliwe korzystanie z tej CB.

Tabela A.64. Specyfikacja ogólna

Dane techniczne	CB 1241 RS485
Nr zamówieniowy	6ES7 241-1CH30-1XB0
Wymiary W \times H \times D	38 \times 62 \times 21
Masa	40 gramów

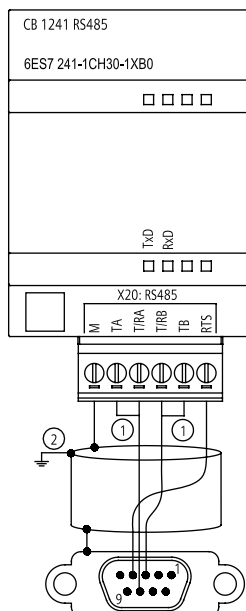
Tabela A.65. Urządzenia nadawcze i odbiorcze

Dane techniczne	CB 1241 RS485
Typ	RS485 (2-przewodowy półdupleks)
Zakres napięcia w trybie wspólnym	od -7 V do +12 V, 1 sekunda, 3 VRMS ciągle
Napięcie różnicowe wyjścia urządzenia nadawczego	2 V min. przy $R_L = 100 \Omega$ 1,5 V min. przy $R_L = 54 \Omega$
Zakończenie i obciążenie	od 10 K do +5 V na B, RS485 Pin 3 od 10 K do GND na A, RS485 Pin 4
Opcjonalne zakończenie	Krótki Pin TB do Pin T/RB, efektywna impedancja zakończenia wynosi 127Ω , połączenie do RS485 Pin 3 Krótki Pin TA do Pin T/RA, efektywna impedancja zakończenia wynosi 127Ω , połączenie do RS485 Pin 4
Impedancja wejścia odbiornika	5,4 k Ω min. włączając zakończenie
Odbiornik: próg/czułość	+/- 0,2 V min., histereza 60 mV
Izolacja Sygnał RS485 do uziemienia szyny Sygnał RS485 do wspólnej logiki CPU	500 VAC, 1 minuta
Długość przewodu, ekranowany	1000 m maks.
Szybkość transmisji	300 b/s, 600 b/s, 1,2 kb/s, 2,4 kb/s, 4,8 kb/s, 9,6 kb/s (domyślnie), 19,2 kb/s, 38,4 kb/s, 57,6 kb/s, 76,8 kb/s, 115,2 kb/s.
Parzystość	Bez parzystości (domyślnie), parzysty, nieparzysty, Mark (bit parzystości ustawiony zawsze na 1), Space (bit parzystości ustawiony zawsze na 0)
Liczba bitów stopu	1 (domyślnie), 2
Sterowanie strumieniem danych	Nie obsługuje
Czas oczekiwania	od 0 do 65 535 ms

Tabela A.66. Zasilanie

Dane techniczne	CB 1241 RS485
Pobór mocy	1,5 W
Pobór prądu (magistrala SM), maks.	50 mA
Pobór prądu (24 VDC) maks.	80 mA

CB 1241 RS485 (6ES7 241-1CH30-1XB0)



- ① Połącz „TA” oraz TB” jak pokazano na rysunku, aby zakończyć sieć (zakończa tylko końcowe urządzenia sieci RS485)
- ② Użyj skrętki ekranowanej i podłącz jej ekran do uziemienia

Można zamknąć sieć RS485 tylko na jej dwóch końcach. Urządzenia pomiędzy dwoma końcami sieci nie są terminowane ani obciążone. Więcej informacji znajduje się w podrozdziale „Polaryzacja i zakończenie złącza sieciowego RS485” podręcznika systemu S7-1200.

Tabela A.67. Zaciski listew zaciskowych CB 1241 (6ES7 241-1CH30-1XB0)

Pin	Złącze 9-pinowe	X20
1	RS485/Masa logiczna GND	–
2	RS485/Nie podłączony	–
3	RS485/TxD+	3 – T/RB
4	RS485/RTS	1 – RTS
5	RS485/Masa logiczna GND	–
6	RS485/Zasilanie 5 V	–
7	RS485/Nie podłączony	–
8	RS485/TxD-	4 – T/RA
9	RS485 / Nie podłączony	–
Shell		7 – M

A.9.4.2. Specyfikacja modułów CM 1241 RS422/RS485

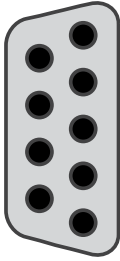
Tabela A.68. Specyfikacja ogólna

Dane techniczne	CM 1241 RS422/RS485
Numer zamówieniowy	6ES7 241-1CH32-0XB0
Wymiary (W × H × D)	30 × 100 × 75 mm
Masa	150 gramów

Tabela A.69. Nadajnik i odbiornik

Dane techniczne	CM 1241 RS422/RS485
Typ	RS422 lub RS485, 9-pinowe złącze żeńskie D-Sub
Zakres sygnału współbieżnego	Od -7 V do +12 V, przez 1 sekundę 3 V RMS (wartość skuteczna) w sposób ciągły
Wyjściowe napięcie różnicowe nadajnika	2 V min. przy RL = 100 Ω 1,5 V min. przy RL = 54 Ω
Zakończenie i polaryzacja	10 kΩ do +5 V na linii B, pin 3 PROFIBUS 10 kΩ do GND na linii A, pin 8 PROFIBUS Zapewnione opcje wewnętrznej polaryzacji, lub bez wewnętrznej polaryzacji. We wszystkich przypadkach wymagane jest zewnętrzne zakończenie. Dodatkowe informacje znajdują się w podrozdziałach „Polaryzacja i zakończenie złącza sieciowego RS485” oraz „Konfiguracja RS422 oraz RS485” podręcznika systemu S7-1200.
Impedancja wejściowa odbiornika	5,4 kΩ min. łącznie z zakończeniem
Próg/czułość odbiornika	+/- 0,2 V min., typowa histereza 60 mV
Izolacja Sygnał RS485 do masy obudowy Sygnał RS485 do masy logiki CPU	500 VAC, przez 1 minutę
Długość kabla ekranowanego	1000 m maks. (w zależności od szybkości transmisji)
Szybkość transmisji	300 bodów, 600 bodów, 1,2 kb/s, 2,4 kb/s, 4,8 kb/s, 9,6 kb/s (default), 19,2 kb/s, 38,4 kb/s, 57,6 kb/s, 76,8 kb/s, 115,2 kb/s
Parzystość	Brak parzystości (domyślnie), parzystość dodatnia, parzystość ujemna, znak (bit parzystości zawsze równy 1), spacja (bit parzystości zawsze równy 0)
Liczba bitów stop	1 (wartość domyślna), 2
Sterowanie przepływem	XON/XOFF obsługiwane w trybie RS422
Czas oczekiwania	od 0 do 65 535 ms

Tabela A.70. Wtyczka RS232 dla CM 1241 i okablowanie

Pin	Opis	Wtyczka (żeńska)	Pin	Opis		
1	Masa logiki lub sygnałów komunikacyjnych		6 PWR	+5 V z szeregowym rezystorem 100 Ω: Wyjście		
2 TxD+ ¹	Podłączony z RS422 Nie używany z RS485: Wyjście		5	7	Niepodłączony	
3 TxD+	Sygnał B (RxD/TxD+): Wejście/Wyjście		4			
4 RTS ²	Żądanie wysłania (poziom TTL): Wyjście		8	3	8 TXD-	Sygnał A (RxD/TxD-): Wejście/Wyjście
5 GND	Masa logiki lub sygnałów komunikacyjnych		7	2	9 TXD- ¹	Podłączony z RS422 Nie używany z RS485: Wyjście
		6	1	SHELL	Masa obudowy	

¹ Piny 2 i 9 są używane tylko dla sygnałów wysyłanych dla RS422.

² RTS jest sygnałem na poziomie TTL i może być używany do sterowania innego urządzenia półdupleksowego, które będzie bazować na tym sygnale. Jest on włączony tylko podczas nadawania.

A.9.4.3. Specyfikacja CM 1241 RS232

Tabela A.72. Specyfikacja ogólna

Dane techniczne	CM 1241 RS232
Numer zamówienia	6ES7 241-1AH32-0XB0
Wymiary W × H × D (mm)	30 × 100 × 75 mm
Masa	150 gramów

Tabela A.73. Nadajnik i odbiornik

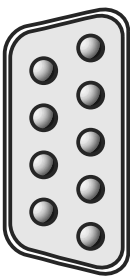
Dane techniczne	CM 1241 RS232
Typ	RS232 (pełny duplex)
Napięcie wyjściowe nadajnika	+/- 5 V min. przy RL = 3 kΩ
Napięcie wyjściowe nadawania	+/- 15 VDC maks.
Impedancja wejściowa odbiornika	3 kΩ min.
Próg/czułość odbiornika	0,8 V min. niski, 2,4 maks. wysoki Typowa histereza 0,5 V
Napięcie wejściowe odbiornika	+/- 30 VDC maks.
Izolacja	
Sygnał RS232 do masy obudowy	500 VAC, przez 1 minutę
Sygnał RS232 do masy logiki CPU	
Długość kabla ekranowanego	10 m maks.
Szybkość transmisji	300 bodów, 600 bodów, 1,2 kb/s, 2,4 kb/s, 4,8 kb/s, 9,6 kb/s (wartość domyślna), 19,2 kb/s, 38,4 kb/s, 57,6 kb/s, 76,8 kb/s, 115,2 kb/s

Dane techniczne	Opis
Parzystość	Brak parzystości (domyślnie), parzystość dodatnia, parzystość ujemna, znak (bit parzystości zawsze równy 1), spacja (bit parzystości zawsze równy 0)
Liczba bitów stopu	1 (wartość domyślna), 2
Sterowanie przepływem	Sprzętowe, programowe
Czas oczekiwania	od 0 do 65 535 ms

Tabela A.74. Zasilanie

Dane techniczne	CM 1241 RS232
Pobór mocy	1 W
Pobór prądu z +5 VDC	200 mA

Tabela A.75. Wtyczka RS485 lub RS422 (żeńska)

Pin	Opis	Wtyczka (męska)	Pin	Opis
1 DCD	Wykryty sygnał nośnej: Wejście		6 DSR	Dane gotowe do wysłania: Wejście
2 RxD	Dane otrzymywane z DCE: Wejście		7 RTS	Żądanie wysłania: Wyjście
3 TxD	Dane wysyłane do DCE: Wyjście		8 CTS	Gotowość do wysłania: Wejście
4 DTR	Gotowość terminala danych: Wyjście		9 RI	Sygnał dzwonka (nieużywany)
5 GND	Masa sygnałowa		SHELL	Masa obudowy

A.10. Moduły technologiczne

A.10.1. SM 1278 4xIO-Link Master

A.10.1.1. Specyfikacja modułu SM 1278 4xIO-Link Master

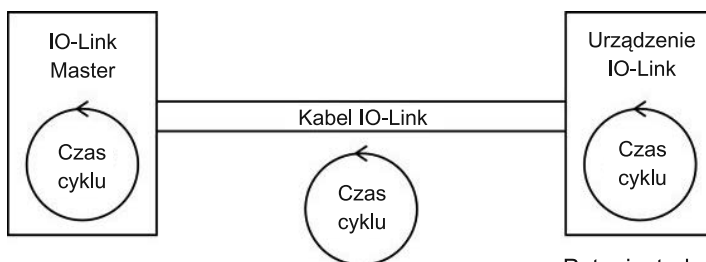
Tabela A.76. Specyfikacja ogólna

Dane techniczne	Moduł SM 1278 4xIO-Link Master
Numer zamówieniowy	6ES7 278-4BD32-0XB0
Wymiary W x H x D (mm)	45 x 100 x 75
Masa	150 gramów
Informacje ogólne	
Dane I&M	Tak; od IM0 do IM3
Napięcie zasilania	
Napięcie znamionowe (DC)	24 VDC
Dolny limit poprawnego zakresu (DC)	19,2 V; 20,5 V jeśli używany IO-Link (Napięcie zasilania dla urządzeń IO-Link Master musi mieć wartość co najmniej 20 V)
Górny limit poprawnego zakresu (DC)	28,8 VDC

Dane techniczne	Moduł SM 1278 4xIO-Link Master
Zabezpieczenie przed odwróceniem polaryzacji	Tak
Prąd wejściowy	
Pobór prądu	65 mA; bez obciążenia
Zasilanie enkodera	
Liczba wyjść	4
Prąd wyjściowy, wartość znamionowa	200 mA
Pobór mocy	
Pobór mocy, typowo	1 W, bez obciążenia portów
Wejścia/wyjścia cyfrowe	
Długość kabla (metry)	20 m, nieekranowany, maks.
SDLC	
Długość kabla (metry)	20 m, nieekranowany, maks.
IO-Link	
Liczba portów	4
Liczba portów, które mogą być sterowane jednocześnie	4
Protokół IO-Link 1.0	Tak
IO-Link protocol 1.1	Tak
Tryb pracy	
IO-Link	Tak
DI	Tak
DQ	Tak; maks. 100 mA
Połączenie urządzeń IO-Link	
Typ portu A	Tak
Szybkość transmisji danych	4,8 kBd (COM1)
	38,4 kBd (COM2)
	230,4 kBd (COM3)
Czas cyklu, min.	2 ms, dynamicznie, zależnie od długości danych użytkownika
Rozmiar danych procesowych, na wejściu portu	32 bajty; maks.
Rozmiar danych procesowych, na wejściu modułu	32 bajty
Rozmiar danych procesowych, na wyjściu portu	32 bajty; maks.
Rozmiar danych procesowych, na wyjściu modułu	32 bajty
Wielkość pamięci dla parametrów urządzenia	2 kilobajty
Długość kabla nieekranowanego, maks. (metry)	20 m Przerwania/diagnostyka/informacje o stanie
Wskaźnik stanu	Tak
Przerwania	
Przerwanie diagnostyczne	Tak; port diagnostyczny jest dostępny tylko w trybie IO-Link

Dane techniczne	Moduł SM 1278 4xIO-Link Master
Alarmy diagnostyczne	
Diagnostyka	
Monitorowanie napięcia zasilającego	Tak
Zabezpieczenie przeciwzwarciowe	Tak
Diagnostyczny wskaźnik LED	
Monitorowanie napięcia zasilającego	Tak; migająca czerwona dioda LED DIAG
Wyświetlanie statusu kanałów	Tak; jedna zielona dioda LED na kanał dla statusu kanału Qn (tryb SIO) i statusu portu Cn (tryb IO-Link)
Do diagnostyki kanału	Tak; czerwona dioda LED Fn
Do diagnostyki modułu	Tak; zielona/czerwona dioda LED DIAG
Separacja galwaniczna	
Separacja galwaniczna kanałów	
Pomiędzy kanałami	Nie
Pomiędzy kanałami i magistralą montażową	Tak
Dopuszczalna różnica potencjałów	
Między różnymi obwodami	75 VDC/60 VAC (izolacja podstawowa)
Izolacja	
Izolacja testowana z	707 VDC (test typu)
Warunki otoczenia	
Temperatura pracy	
Min.	-20°C
Maks.	60°C
Montaż poziomy, min.	-20°C
Montaż poziomy, maks.	60°C
Montaż pionowy, min.	-20°C
Montaż pionowy, maks.	50°C

Czas odpowiedzi



Czas cyklu jest ustalany pomiędzy IO-Link Master i urządzeniem IO-Link.

Patrz instrukcja obsługi urządzenia IO-Link.

Uzgodniony czas odpowiada minimalnemu czasowi cyklu IO-Link dla IO-Link Master.

A.10.1.2. Schematy połączeń modułu SM 1278 4xIO-Link Master

Tabela A.77. Schemat połączeń modułu SM 1278 IO-Link Master

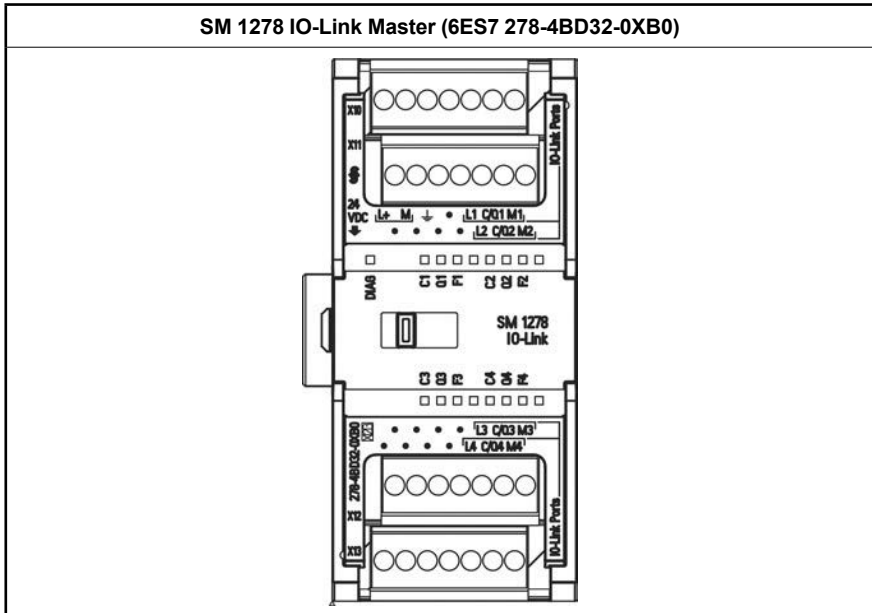


Tabela A.78. Zaciski listew zaciskowych SM 1278 IO-Link Master (6ES7 278-4BD32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 VDC	Brak połączenia	Brak połączenia	Brak połączenia
2	M / 24 VDC	Brak połączenia	Brak połączenia	Brak połączenia
3	Uziemienie funkcjonalne	Brak połączenia	Brak połączenia	Brak połączenia
4	Brak połączenia	Brak połączenia	Brak połączenia	Brak połączenia
5	L1	L2	L3	L4
6	C/Q1	C/QL2	C/Q3	C/QL4
7	ML1	ML2	M3	ML4

A.11. Urządzenia towarzyszące

A.11.1. Zasilacz PM1207

PM 1207 jest modułem zasilającym dla S7-1200. Ma on następujące funkcje:

- wejście 120/230 VAC, wyjście VDC/2,5A
- numer zamówieniowy 6ESP 332-1SH71-4AA0

Więcej informacji o tym produkcie oraz jego dokumentacja znajduje się na internetowej stronie katalogów produktów dla PM 1207.

Patrz również

Internetowa strona wsparcia klienta (<http://www.siemens.com/automation/>).

A.11.2. Kompaktowy switch CSM 1277

CSM 1277 jest kompaktowym modulem przełączającym. Może być użyty do zwielokrotnienia interfejsu Ethernet dla S7-1200, co pozwala na równoczesną komunikację z panelami operatorskimi, urządzeniami programującymi i innymi sterownikami. Posiada on następujące funkcje:

- 4 gniazda Ethernet do włączenia się w przemysłową sieć.
- 3-biegunowe wtyczki na listwie terminalu do podłączenia zasilania na górze obudowy sterownika.
- Diody LED do wyświetlania statusu portów przemysłowego Ethernetu.
- Nr zamówieniowy 6GK7 277-1AA00-0AA0.

Więcej informacji o tym produkcie oraz jego dokumentacja znajduje się na internetowej stronie katalogów produktów dla CSM 1277.

Patrz również

Internetowa strona wsparcia klienta (<http://www.siemens.com/automation/>).

A.11.3. Moduł CM CANopen

Moduł komunikacyjny (CM) CANopen jest wkładanym modulem podłączanym pomiędzy sterownikiem PLC rodziny SIMATIC S7-1200 a dowolnym urządzeniem działającym w oparciu o protokół komunikacyjny CANopen. Moduł CM CANopen może być skonfigurowany zarówno jako urządzenie nadrzędne (Master), jak i podrzędne (Slave). Występują dwa moduły CM CANopen: standardowy moduł CANopen (numer zamówieniowy 021620-B) oraz przeznaczony dla większych obciążeń moduł CANopen (*Ruggedized*) (numer zamówieniowy 021730-B).

Moduł CANopen zapewnia następujące funkcje:

- możliwość podłączenia do 3 modułów do jednego sterownika PLC,
- łączność z maksymalnie 16 węzłami CANopen typu Slave,
- obsługa 256-bajtowego wejścia i 256-bajtowego wyjścia,
- 3 diody LED dostarczają informacje diagnostyczne dotyczące modułu, sieci oraz stanów wejść/wyjść,
- umożliwia przechowywanie ustawień konfiguracji sieci CANopen w PLC,
- moduł jest integrowalny z katalogiem sprzętowym środowiska projektowego TIA Portal,

- konfiguracja CANopen poprzez dołączone oprogramowanie CANopen Configuration Studio (w zestawie) lub inne narzędzia programowe do konfiguracji CANopen,
- zgodność z profilami komunikacji CANopen CiA 301 rev. 4.2 oraz CiA 302 rev. 4.1,
- obsługa „przeźroczystego” standardu CAN 2.0A umożliwia korzystanie z niestandardowych protokołów,
- predefiniowane bloki funkcyjne są dostępne przy każdym programowaniu sterownika PLC w TIA Portal,
- w zestawie z modułami CM CANopen są dostarczane: złącza D-Sub z zaciskami śrubowymi do łączenia podsieci, płyta CD z oprogramowaniem CANopen Configuration Studio oraz kabel USB do przeprowadzenia konfiguracji.

Więcej informacji o tym produkcie oraz jego dokumentacja znajduje się na internetowej stronie katalogów produktów dla CM CANopen.

Wymiana CPU V3.0 na CPU V4.0

B

B.1. Wymiana CPU V3.0 na CPU V4.0

Można wymienić CPU w wersji V3.0 na CPU w wersji V4.0 (strona 71) i użyć istniejącego projektu STEP 7 utworzonego dla CPU V3.0. Można również sprawdzić dostępność plików aktualizacji oprogramowania sprzętowego (firmware) dla podłączonych modułów i przeprowadzić aktualizację firmware (strona 265).

Uwaga

Wymiana urządzeń z wersji V4.0 na wersję V3.0 nie jest możliwa

Można wymienić CPU V3.0 na CPU V4.0, ale po załadowaniu ustawień konfiguracji nie można z powrotem wymienić CPU v4.0 na CPU V3.0. Aby przeglądać lub w inny sposób używać istniejącego projektu STEP 7 w wersji V3.0, to przed wymianą urządzeń ten projekt należy zarchiwizować.

Jeśli ustawienia konfiguracji dla wymienionego urządzenia nie zostaną załadowane, to można je cofnąć. Jednak po ich załadowaniu, nie można cofnąć zmiany wersji V3.0 na V4.0.

Należy zdawać sobie sprawę z niektórych zmian konfiguracji i zmian operacyjnych pomiędzy dwoma wersjami CPU:

Bloki organizacyjne

W przypadku CPU V4.0 można skonfigurować, aby wykonywanie OB było przerywane lub nieprzerywane (strona 53). W projektach poprzedniego CPU V3.0, STEP 7 ustawiał domyślnie wszystkie bloki jako nieprzerywane.

STEP 7 ustawia wszystkie priorytety OB (strona 53) na wartości, które były w projekcie STEP 7 CPU V3.0.

Następnie, w razie potrzeby ustawienia priorytetów lub przerywalności mogą być zmieniane.

Informacje uruchomienia OB przerwania błędu diagnostycznego odnoszą się do danego modułu, jeśli nie występuje zdarzenie diagnostyczne.

Ochrona CPU za pomocą hasła

STEP 7 ustawia dla CPU V4.0 poziom ochrony hasłem (strona 78) równoważny do ustawionego dla CPU V3.0 i przypisuje hasło CPU V3.0 do hasła CPU V4.0 „Pełny dostęp (brak ochrony)”:

Poziom ochrony CPU V3.0	Poziom dostępu do CPU V4.0
Brak ochrony	Pełny dostęp (brak ochrony)
Ochrona przed zapisem	Dostęp do odczytu
Ochrona przed zapisem / odczytem	Dostęp do HMI

Należy pamiętać, że poziom dostępu V4.0 „Brak dostępu (pełna ochrona)” nie istnieje dla V3.0.

Web serwer

Jeśli CPU V3.0 zostanie zastąpiony przez CPU V4.0, to ustawienia projektu web serwera do aktywacji web serwera, i określające, czy nie jest wymagany dostęp HTTPS, będą takie same, jak były w wersji V3.0. Następnie, można skonfigurować użytkowników, uprawnienia, hasła (strona 186), oraz w razie potrzeby języki, aby korzystać z web serwera. Jeśli użytkownik nie zostanie skonfigurowany z dodatkowymi uprawnieniami, to będzie ograniczony do tego, co można zobaczyć od standardowych stronach sieci Web (strona 187). CPU V4.0 S7-1200 nie obsługuje poprzedniego prekonfigurowanego użytkownika i hasła „admin”.

Poprzednio, strona web serwera logów (dzienników) danych zapewniała funkcję *Download and Clear* (Pobierz i usuń). Na stronie przeglądarki plików web serwera V4.0 (strona 187), która umożliwiła dostęp do logów danych, tej funkcji już nie ma. Zamiast tego, web serwer umożliwi pobieranie, zmianę nazw i usuwanie plików logów danych.

Niezgodność karty transferowej

Nie można skorzystać z karty transferowej (transfer card) V3.0 (strona 57), aby przenieść program V3.0 do CPU V4.0. W tym celu należy otworzyć projekt V3.0 w STEP 7, zmienić urządzenie na CPU V4.0 (strona 71), i załadować projekt STEP 7 do CPU V4.0. Po zmianie projektu na projekt w wersji V4.0, można następnie utworzyć kartę transferową v4.0 dla kolejnych przeniesień programów.

Komunikacja za pomocą instrukcji GET i PUT

Domyślnie opcja komunikacji za pomocą instrukcji GET/PUT dla CPU V3.0 jest włączona. Po zastąpieniu CPU V3.0 przez CPU V4.0 (strona 71), w sekcji informacji dotyczących kompatybilności zostanie wyświetlony komunikat informujący, że instrukcje GET/PUT są aktywne.

Obsługa sterowania ruchem

Jednoski CPU V4.0 S7-1200 nie obsługują bibliotek sterowania ruchem V1.0 i V2.0. Jeśli dla projektu STEP 7 z bibliotekami sterowania ruchem V1.0 lub V2 zostanie przeprowadzona wymiana urządzeń, to po wymianie urządzeń podczas kompilacji biblioteki sterowania ruchem V1.0 lub V2.0 zostaną zastąpione kompatybilnymi instrukcjami sterowania ruchem w wersji V3.0 (strona 211).

Jeśli dla projektu STEP 7 z dwoma różnymi wersjami instrukcji sterowania ruchem (V3.0 i V4.0) zostanie przeprowadzona wymiana urządzeń, to po wymianie urządzeń podczas kompilacji, instrukcje te zostaną zastąpione kompatybilnymi instrukcjami sterowania ruchem V4.0 (strona 211).

Podczas wymiany urządzeń z V3.0 na V4.0, wersja obiektu technologicznego (TO) sterowania ruchem automatycznie zmienia się na wersję V4.0. Struktura obiektu TO różni między wersjami V3.0 i V4.0. Wszystkie powiązane bloki rów-

niez się zmieniają. Interfejsy bloków, tablice monitorujące, funkcje śledzenia zostaną zaktualizowane do nowej struktury V4.0. Różnice parametrów osi sterowania ruchem pomiędzy wersjami V3.0 i V4.0 są wykazane w dwóch podanych niżej tabelach:

V3.0	V4.0
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter. PulsesPerDriveRevolution
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults. EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel
Config.Homing.AutoReversal	Homing.AutoReversal
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.Sidelnput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.Sidelnput
Config.Homing.Offset	Sensor[1].ActiveHoming. HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity
MotionStatus.Distance	StatusPositioning.Distance
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

Jedynym parametrem *commandtable*, którego nazwa jest zmieniana, jest tablica poleceń:

V3.0	V4.0
Config.Command[]	Command[]

Uwaga: tablica „Command[]” jest typem danych zdefiniowanym przez użytkownika UDT (user defined data type): typu „TO_CmdTab_Config_Command” w wersji V3.0, i typu „TO_Struct_Command” w wersji V4.0.

Zmiany instrukcji

Poniższe instrukcje zmieniają parametry lub zachowanie:

- RDREC oraz WRREC (strona 127)
- CONV (strona 98)

Komunikacja z panelami HMI

Jeśli do modułu CPU V3.0 S7-1200 jest podłączonych więcej niż jeden panel HMI (strona 20), to komunikacja z CPU V4.0 S7-1200 zależy od typu używanej komunikacji i wersji oprogramowania sprzętowego (*firmware*) panelu HMI. Projekt należy ponownie skompilować i załadować do CPU i HMI oraz/lub zaktualizować firmware HMI.

Wymaganie ponownej kompilacji bloków programowych

Po wymianie CPU V3.0 na CPU V4.0, wszystkie bloki programowe przed załadowaniem do CPU V4.0 należy ponownie skompilować. Dodatkowo, jeśli któryś z bloków ma zabezpieczenie wiedzy (know-how) (strona 81) lub zabezpieczenie przed kopiowaniem przez powiązanie z numerem seryjnym PLC (strona 82), należy usunąć zabezpieczenie przed kompilacją i załadować bloki kodu. (Nie trzeba jednak dezaktywować zabezpieczenia przed kopiowaniem przez powiązanie z kartą pamięci.) Po załadowaniu bloków można ponownie skonfigurować zabezpieczenie wiedzy oraz/lub numer seryjny PLC do zabezpieczenia przed kopiowaniem. Należy pamiętać, że jeśli projekt zawiera jakiegokolwiek bloki kodu z zabezpieczeniem wiedzy, które zapewnił producent sprzętu oryginalnego OEM (Original Equipment Manufacturer), to aby tym blokom zapewnić wersję V4.0 należy skontaktować się z producentem OEM.

Ogólnie, po wymianie urządzeń (na CPU wyższej wersji) firma Siemens zaleca, aby konfigurację sprzętu i oprogramowanie ponownie skompilować w środowisku projektowym STEP 7 i załadować do wszystkich urządzeń projektu.

Szczegółowe informacje na temat funkcji urządzeń w wersji V4.0 znajdują się podręczniku systemu S7-1200.

Patrz również

Podrozdział „Zmiana urządzenia” (strona 71)

Zestaw 1:

SIMATIC S7-1200

CPU 1212C AC/DC/RLY

- Symulator wejść
- SIMATIC STEP 7 Basic (TIA Portal)
- IE TP Przewód 2 m
- Dokumentacja na CD

Nr zam.: 6ES7 212-1BD34-4YB0



Zestaw 2:

SIMATIC S7-1200 + KP300 Basic Panel

CPU 1212C AC/DC/RLY

- Symulator wejść
- KP300 Basic mono PN
- SIMATIC STEP 7 Basic (TIA Portal)
- IE TP Przewód 2 m
- Dokumentacja na CD

Nr zam.: 6AV6 651-7HA01-3AA4



Zestaw 3:

SIMATIC S7-1200 + KTP400 Basic Panel

CPU 1212C AC/DC/RLY

- Symulator wejść
- KTP400 Basic
- SIMATIC STEP 7 Basic (TIA Portal)
- IE TP Przewód 2 m
- Dokumentacja na CD

Nr zam.: 6AV6 651-7KA01-3AA4



Zestaw 4:

SIMATIC S7-1200 + KTP700 Basic Panel

CPU 1212C AC/DC/RLY

- Symulator wejść
- KTP700 Basic
- SIMATIC STEP 7 Basic (TIA Portal)
- IE TP Przewód 2 m
- Dokumentacja na CD

Nr zam.: 6AV6 651-7DA01-3AA4



Zestaw 5

SIMATIC S7-1200 Fail-safe

CPU 1212FC DC/DC/RLY

- Symulator wejść
- SM 1226 F-DI
- SM 1226 F-DQ
- SIMATIC STEP 7 Basic oraz STEP 7 Safety Basic (TIA Portal)
- IE PT Przewód 2 m
- Dokumentacja na CD

Nr zam.: 6ES7212-1HF41-4YB0



Zestawy dostępne u autoryzowanych dystrybutorów:
siemens.pl/simatic/dystrybutorzy

Biura sprzedaży:

Siemens Sp. z o.o.
 Factory Automation
 03-821 Warszawa
 ul. Żupnicza 11
 tel.: 22 870 8200
 fax: 22 870 9149

Regionalne biura sprzedaży:

80-300 Gdańsk
 Al. Grunwaldzka 413
 tel.: 22 870 8200
 fax: 58 764 6099

40-527 Katowice
 ul. Gawronów 22
 tel.: 22 870 8200
 fax: 32 208 4139

30-443 Kraków
 ul. Józefa Marcika 14B
 tel.: 22 870 8200
 fax: 12 363 8229

92-333 Łódź
 ul. Wydawnicza 1/3
 tel.: 22 870 8200
 fax: 42 677 1799

60-164 Poznań
 ul. Ziębicza 35
 tel.: 22 870 8200
 fax: 61 664 9864

87-100 Toruń
 ul. Gdańska 4A
 tel.: 22 870 8200
 fax: 56 656 4229

53-611 Wrocław
 ul. Strzegomska 52
 tel.: 22 870 8200
 fax: 71 777 5011