

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

Dane techniczne

A

04/2012

A5E02486774-05

Informacje prawne

System ostrzeżeń

Uwagi pojawiające się w tym podręczniku służąć mają 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

Ze znakiem ostrzegawczym 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 wzięcia pod uwagę odpowiednich informacji może wystąpić niezamierzony stan lub sytuacja.

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 zastrzeżone

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 właściwicieli.

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

System S7-1200 jest jednym z najnowszych opracowań firmy Siemens w rodzinie SIMATIC. 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 przy stosowania 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 jest integralną częścią nowatorskiej platformy programistycznej TIA – Totally Integrated Automation, w ramach której urządzenia z rodziny S7-1200 wraz z pakietem programów narzędziowych S7-1200 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 wchodzą sterowniki SIMATIC S7-1200 oraz panele operatorskie SIMATIC HMI Basic. Obydwa rodzaje urządzeń są programowane za pomocą pakietu narzędziowego SIMATIC STEP 7. Możliwość przygotowania programów dla całego systemu sterowania za pomocą jednego narzędzia radykalnie skracą czas ich przygotowywania i zmniejsza koszty.



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, dwa generatory impulsowe: PWM (Pulse Width Modulation) i PTO (Pulse Train Output) (strona 64) i do 6 szybkich liczników (strona 110).

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



Dodatkowe moduły wejść/wyjść (przekaźnikowe, tranzystorowe, analogowe) rozszerzają dostępną liczbę punktów wej./wyj. Innowacyjne płytki sygnałowe montowane na przedniej ścianie sterownika pozwalają go elastycznie dopasować do wielu aplikacji (strona 16).

Panele operatorskie SIMATIC HMI Basic (strona 18) 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 231).

W podręczniku opisano następujące wyroby:

- Pakiety narzędziowe STEP 7 V11 Basic i Professional
- Sterownik S7-1200 CPU z firmware wersja V3.0

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) oraz drukowanej. 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

Poza łatwo dostępną dokumentacją oferujemy także wsparcie techniczne przez Internet, dostępne poprzez stronę <http://www.siemens.com/automation/support-request>).

W sprawie wsparcia technicznego sugerujemy także bezpośredni kontakt z naszymi lokalnymi biurami sprzedaży i dystrybutormi, których przedstawiciele udzielają 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.

| | |
|--|-----------|
| Wprowadzenie | 3 |
| 1. Charakterystyka zaawansowanego sterownika S7-1200 | 13 |
| 1.1. Wprowadzenie do sterownika PLC S7-1200..... | 13 |
| 1.2. Możliwość rozbudowy CPU | 16 |
| 1.3. Moduły S7-1200..... | 18 |
| 1.4. Panele operatorskie Basic HMI | 18 |
| 1.5. Wymiary montażowe i wymagane tolerancje pasowania..... | 20 |
| 1.6. Nowe możliwości | 23 |
| 2. STEP 7 ułatwia pracę..... | 25 |
| 2.1. Łatwe wstawianie instrukcji do programu użytkownika | 26 |
| 2.2. Łatwy dostęp do ulubionych instrukcji z paska narzędzi | 26 |
| 2.3. Łatwe dodawanie wejść i wyjść do instrukcji w drabince lub bloku funkcyjnym..... | 27 |
| 2.4. Rozwijane instrukcje | 27 |
| 2.5. Łatwa zmiana trybu pracy CPU | 27 |
| 2.6. Łatwa zmiana wyglądu i konfiguracji dla STEP 7 | 28 |
| 2.7. Ułatwienia dostępu dla projektu i globalnych bibliotek | 28 |
| 2.8. Łatwy dobór wersji instrukcji..... | 29 |
| 2.9. Łatwe przeciaganie elementów pomiędzy edytorami..... | 29 |
| 2.10. Zmiana typu wywołania DB | 30 |
| 2.11. Tymczasowe odłączenie urządzeń z sieci..... | 31 |
| 2.12. Wirtualne „rozłączanie” modułów bez straty konfiguracji | 32 |
| 3. Rozpoczęcie pracy | 33 |
| 3.1. Tworzenie projektu..... | 33 |
| 3.2. Tworzenie zmiennych (tagów) dla obszaru I/O CPU | 34 |
| 3.3. Tworzenie prostej sieci w programie użytkownika | 35 |
| 3.4. Użycie zmiennych PLC z tablicy zmiennych do adresowania instrukcji | 37 |
| 3.5. Dodawanie pola instrukcji | 38 |
| 3.6. Użycie instrukcji CALCULATE dla złożonych równań matematycznych | 39 |
| 3.7. Dodawanie urządzenia HMI do projektu | 41 |
| 3.8. Tworzenie połączenia pomiędzy CPU i urządzeniem HMI | 42 |

| | | |
|-----------|--|-----------|
| 3.9. | Tworzenie połączenia urządzeń HMI do współdzielenia zmiennych | 42 |
| 3.10. | Tworzenie ekranu urządzenia HMI | 43 |
| 3.11. | Wybór zmiennych PLC dla elementów HMI | 44 |
| 4. | Prosty w obsłudze sterownik PLC | 45 |
| 4.1. | Zadania wykonywane w każdym cyklu..... | 45 |
| 4.2. | Tryby pracy CPU..... | 46 |
| 4.3. | Wykonanie programu użytkownika | 47 |
| 4.3.1. | Wykonywanie cyklu programu w trybie RUN..... | 48 |
| 4.3.2. | Zastosowanie bloków organizacyjnych OB porządkujące program użytkownika..... | 49 |
| 4.3.3. | Kolejkowanie i priorytety wykonania obsługi zdarzeń..... | 50 |
| 4.4. | Obszary pamięci, adresowanie i typy danych | 55 |
| 4.4.1. | Typy danych obsługiwane przez S7-1200 | 57 |
| 4.4.2. | Adresowanie obszarów pamięci..... | 59 |
| 4.4.3. | Dostęp do fragmentów zmiennych adresowanych symbolicznie | 62 |
| 4.4.4. | Dostęp do zmiennych z użyciem struktury AT | 63 |
| 4.5. | Wyjścia impulsowe..... | 64 |
| 5. | Konfiguracja sterownika..... | 66 |
| 5.1. | Rozpoznawanie konfiguracji dla nieokreślonego CPU..... | 67 |
| 5.2. | Dołączanie modułów..... | 68 |
| 5.3. | Dodawanie modułów do systemu..... | 69 |
| 5.4. | Konfiguracja parametrów pracy CPU i modułów..... | 70 |
| 5.4.1. | Pamięć systemu i zegara jako standardowa funkcjonalność | 71 |
| 5.5. | Konfiguracja adresu IP dla CPU..... | 74 |
| 5.6. | Ochrona dostępu do CPU i kodu programu..... | 75 |
| 5.6.1. | Zabezpieczenie „know-how” | 76 |
| 5.6.2. | Zabezpieczenie przed kopiowaniem | 77 |
| 6. | Koncepcja programowania | 79 |
| 6.1. | Projektowanie programu użytkownika | 79 |
| 6.1.1. | Użycie bloków OB do organizacji programu użytkownika | 81 |
| 6.1.2. | FB i FC upraszczają programowanie zadań modularnych | 82 |
| 6.1.3. | Bloki danych (DB) ułatwiają przechowywanie danych programu..... | 83 |

| | | |
|-----------|--|------------|
| 6.1.4. | Tworzenie nowego bloku kodu..... | 84 |
| 6.1.5. | Wywoływanie bloku kodu z innego bloku kodu | 84 |
| 6.2. | Języki programowania | 85 |
| 6.2.1. | Język drabinkowy (LAD)..... | 85 |
| 6.2.2. | Język programowania Function Block Diagram (FBD) | 86 |
| 6.2.3. | Przegląd SCL..... | 86 |
| 6.2.4. | Edytor programów SCL..... | 87 |
| 6.3. | Instrukcje programowania..... | 89 |
| 6.3.1. | Podstawowe instrukcje | 89 |
| 6.3.2. | Instrukcje porównania oraz move | 91 |
| 6.3.3. | Instrukcje konwersji | 92 |
| 6.3.4. | Operacje arytmetyczne z instrukcją Calculate | 94 |
| 6.3.5. | Układy czasowe – timery | 95 |
| 6.3.6. | Liczniki | 100 |
| 6.3.7. | Instrukcja modulacji szerokości impulsu (PWM)..... | 102 |
| 6.4. | Tworzenie dzienników zdarzeń w plikach .csv | 103 |
| 6.5. | Monitorowanie i testowanie programu..... | 105 |
| 6.5.1. | Tablice monitorujące i tablice wymuszeń | 105 |
| 6.5.2. | Odsyłacze pokazujące wykorzystanie elementów | 106 |
| 6.5.3. | Struktura wywołania do badania hierarchii połączeń..... | 107 |
| 6.5.4. | Instrukcje diagnostyczne do monitorowania sprzętu | 107 |
| 6.5.4.1. | Odczytywanie stanu diod LED na sterowniku..... | 107 |
| 6.5.4.2. | Instrukcje odczytujące stan diagnostyczny urządzeń | 108 |
| 6.6. | Szybkie liczniki (HSC)..... | 108 |
| 6.6.1. | Działanie HSC | 110 |
| 6.6.2. | Konfiguracja HSC | 114 |
| 7. | Komunikacja pomiędzy urządzeniami – standard PROFINET | 115 |
| 7.1. | Tworzenie połączenia sieciowego | 115 |
| 7.2. | Opcje komunikacji..... | 116 |
| 7.3. | Liczba asynchronicznych połączeń komunikacyjnych..... | 118 |
| 7.4. | Instrukcje do obsługi PROFINET i PROFIBUS | 118 |
| 7.5. | PROFINET | 119 |
| 7.5.1. | Otwarta komunikacja użytkownika | 119 |
| 7.5.1.1. | Tryb ad hoc..... | 120 |
| 7.5.1.3. | Parametry połączenia PROFINET..... | 124 |

| | | |
|-----------|---|------------|
| 7.5.2. | Konfigurowanie ścieżki połączenia..... | 126 |
| 7.6. | PROFIBUS..... | 129 |
| 7.6.1. | Przykłady konfiguracji PROFIBUS | 130 |
| 7.6.2. | Dodawanie modułów CM 1243-5 (DP master) oraz DP slave | 133 |
| 7.6.3. | Przypisanie adresów PROFIBUS do modułu CM 1243-5 i urządzeń DP slave | 134 |
| 7.7. | Interfejs AS-i..... | 136 |
| 7.7.1. | Dodawanie modułów CM 1243-2 AS-i master oraz AS-i slave | 136 |
| 7.7.2. | Przypisywanie adresu do urządzeń slave AS-i..... | 137 |
| 7.8. | Komunikacja S7 | 138 |
| 7.8.1. | Instrukcje GET i PUT..... | 138 |
| 7.8.2. | Tworzenie połączenia S7 | 139 |
| 7.8.3. | Przypisanie wartości parametrów połączeniu GET/PUT..... | 139 |
| 7.9. | GPRS | 140 |
| 7.9.1. | Połączenie z siecią GSM | 140 |
| 7.10. | Protokoły komunikacyjne PtP, USS oraz Modbus | 147 |
| 7.10.1. | Korzystanie z interfejsów komunikacji szeregowej | 147 |
| 7.10.2. | Instrukcje PtP..... | 148 |
| 7.10.3. | Instrukcje USS..... | 149 |
| 7.10.4. | Instrukcje obsługujące protokół Modbus | 150 |
| 8. | Łatwy w użyciu regulator PID | 152 |
| 8.1. | Wstawianie instrukcji PID i obiektów technologicznych | 154 |
| 8.2. | Instrukcja PID_Compact | 156 |
| 8.3. | Parametry ErrorBits instrukcji PID_Compact..... | 160 |
| 8.4. | Instrukcja PID_3Step | 161 |
| 8.5. | Parametry ErrorBits instrukcji PID_3STEP..... | 167 |
| 8.6. | Konfigurowanie regulatora PID..... | 169 |
| 8.7. | Uruchomienie regulatora PID | 172 |
| 9. | Wbudowany serwer WWW ułatwiający komunikację z Internetem ... | 173 |
| 9.1. | Łatwość użycia standardowych stron Web | 173 |
| 9.2. | Ograniczenia, które mogą mieć wpływ na korzystanie z serwera sieci Web | 175 |
| 9.2.1. | Ograniczenia związane z wyłączeniem obsługi JavaScript | 176 |

| | | |
|------------|--|------------|
| 9.2.2. | Funkcje ograniczone poprzez wyłączenie obsługi „ciasteczek” | 177 |
| 9.3. | Uproszczone tworzenie stron definiowanych przez użytkownika..... | 177 |
| 9.3.1. | Łatwe tworzenie dowolnych, zdefiniowanych przez użytkownika stron sieci Web | 177 |
| 9.3.2. | Ograniczenia związane ze stronami definiowanymi przez użytkownika | 178 |
| 9.3.3. | Konfiguracja zdefiniowanych przez użytkownika stron Web | 179 |
| 9.3.4. | Używanie instrukcji WWW..... | 180 |
| 10. | Łatwe sterowanie napędami | 181 |
| 10.1. | Konfiguracja osi napędu | 184 |
| 10.2. | Konfiguracja TO_CommandTable_PTO..... | 187 |
| 10.3. | Instrukcja MC_Power..... | 190 |
| 10.4. | Instrukcja MC_Reset..... | 193 |
| 10.5. | Instrukcja MC_Home | 194 |
| 10.6. | Instrukcja MC_Halt..... | 198 |
| 10.7. | Instrukcja MC_MoveAbsolute | 200 |
| 10.8. | Instrukcja MC_MoveRelative | 202 |
| 10.9. | Instrukcja MC_MoveVelocity | 204 |
| 10.10. | Instrukcja MC_MoveJog | 206 |
| 10.11. | Instrukcja MC_CommandTable..... | 208 |
| 10.12. | Instrukcja MC_ChangeDynamic | 211 |
| 11. | Łatwe w użyciu narzędzia online..... | 214 |
| 11.1. | Tryb online i połączenie z CPU | 214 |
| 11.2. | Interakcja z połączonym CPU | 215 |
| 11.3. | Połączenie online w celu monitorowania wartości w CPU..... | 216 |
| 11.4. | Wyświetlanie stanu programu użytkownika..... | 216 |
| 11.5. | Tablice monitorujące do monitorowania CPU..... | 217 |
| 11.6. | Użycie tablic wymuszeń..... | 218 |
| 11.7. | Przechwytywanie wartości online bloków DB do skasowania wartości początkowych..... | 221 |
| 11.8. | Kopiowanie elementów projektu | 222 |

| | | |
|-----------|---|------------|
| 11.9. | Porównywanie i synchronizacja CPU będących <i>online</i> i <i>offline</i> | 223 |
| 11.10. | Wyświetlanie zdarzeń diagnostycznych | 224 |
| 11.11. | Ustawianie adresu IP i czasu dnia | 224 |
| 11.12. | Ustawienia fabryczne..... | 225 |
| 11.13. | Załadowanie adresu IP do połączonego online CPU..... | 225 |
| 11.14. | Użycie „nieznanego CPU” do automatycznej detekcji | 227 |
| 11.15. | Ładowanie programu do pamięci sterownika w trybie pracy RUN..... | 228 |
| 11.15.1. | Modyfikacje programu w trybie RUN | 229 |
| A. | Dane techniczne | 231 |
| A.1. | Dane techniczne ogólne | 231 |
| A.2. | Specyfikacja CPU | 237 |
| A.3. | Moduły cyfrowych I/O | 241 |
| A.3.1. | Płytki sygnałowe SB 1221, SB 1222, oraz SB 1223 – cyfrowe wejścia/wyjścia (DI, DQ, oraz DI/DQ) | 241 |
| A.3.2. | Moduł SM 1221 – cyfrowe wejścia (DI) | 243 |
| A.3.3. | Moduł SM 1222 – cyfrowe wyjścia (DQ)..... | 244 |
| A.3.4. | Moduł SM 1223 cyfrowe wejścia/wyjścia VDC (DI/DQ)..... | 245 |
| A.3.5. | Moduł SM 1223 120/Wejścia 230 VAC/Wyjścia przekaźnikowe | 246 |
| A.4. | Specyfikacja cyfrowych wejść i wyjść..... | 247 |
| A.4.1. | Cyfrowe wyjścia 24 VDC (DI)..... | 247 |
| A.5. | Moduły analogowych I/O | 252 |
| A.5.1. | Płytki sygnałowe analogowych wejść (AI)- SB 1231 i analogowych wyjść (AQ) – SB 1232..... | 252 |
| A.5.2. | SM 1231 – analogowe wejścia (AI) | 253 |
| A.5.3. | SM 1232 – analogowe wyjścia (AQ)..... | 253 |
| A.5.4. | SM 1234 – analogowe wejścia/wyjścia (AI/AQ)..... | 253 |
| A.5.5. | Schematy podłączeń dla SM 1231 (AI), SM 1232 (AQ) oraz SM 1234 (AI/AQ)..... | 254 |
| A.6. | Moduł baterii BB1297 | 255 |
| A.7. | Specyfikacja analogowych wejść/wyjść..... | 255 |
| A.7.1. | Specyfikacja analogowych wejść (CPU, SM oraz SB) | 255 |
| A.7.2. | Napięciowe i prądowe zakresy pomiarowe dla analogowych wejść (AI)..... | 257 |

| | | |
|----------|--|-----|
| A.7.3. | Odpowiedź skokowa analogowych wejść (AI) | 258 |
| A.7.5. | Specyfikacja analogowych wyjść (SM oraz SB) | 259 |
| A.8. | Moduły termopar i RTD | 260 |
| A.8.1. | Specyfikacja modułów SB 1231 RTD oraz SB 1231 TC | 261 |
| A.8.2. | Specyfikacja modułu SM 1231 RTD | 263 |
| A.8.3. | Specyfikacja modułu SM 1231 TC | 264 |
| A.8.4. | Specyfikacja analogowego wejścia dla RTD oraz TC (SM i SB)..... | 265 |
| A.8.5. | Typy termopar..... | 266 |
| A.8.6. | Wybór filtra termopary i czasy odświeżania | 266 |
| A.8.7. | Tabela wyboru typu czujnika RTD..... | 267 |
| A.8.8. | Wybór filtra RTD i czasy odświeżania | 268 |
| A.9. | Interfejsy komunikacyjne..... | 268 |
| A.9.1. | PROFIBUS master/slave..... | 268 |
| A.9.1.1. | Moduł CM 1242-5 PROFIBUS slave..... | 268 |
| A.9.1.2. | Moduł CM 1243-5 PROFIBUS master | 270 |
| A.9.2. | Procesor komunikacyjny GPRS | 271 |
| A.9.2.1. | Specyfikacja techniczna CP 1242-7 | 272 |
| A.9.3. | TeleService (TS) | 274 |
| A.9.4. | Komunikacja poprzez RS485 i RS232 | 274 |
| A.9.4.1. | Specyfikacja CB 1241 RS485 | 274 |
| A.9.4.2. | Moduły CM 1241 RS485/RS232 | 277 |
| A.9.4.3. | Moduły CM 1241 RS422/485 | 278 |
| A.10. | Urządzenia towarzyszące | 279 |
| A.10.1. | Zasilacz PM1207 | 279 |
| A.10.2. | Kompaktowy switch CSM 1277 | 280 |

Charakterystyka zaawansowanego sterownika S7-1200

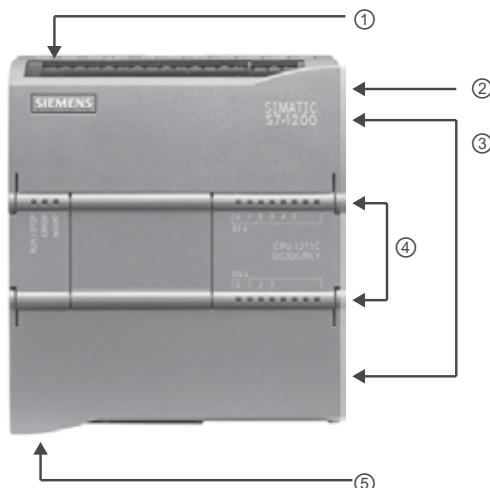
1

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 rozbudowywany 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 75), które umożliwia konfigurowanie dostępu do funkcji CPU.
- Możesz użyć ochrony typu „know-how protection” (strona 76) 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 |
|--|--|-------------------------------------|--|-------------------------------------|
| Wymiary fizyczne [mm] | 90 × 100 × 75 | 90 × 100 × 75 | 110 × 100 × 75 | 130 × 100 × 75 |
| Pamięć użytkownika | Robocza | 30 kb | 50 kb | 75 kb |
| | Ładowania | 1 Mb | 1 Mb | 4 Mb |
| | Nieulotna | 10 kb | 10 kb | 10 kb |
| Lokalne porty I/O | Cyfrowe | 6 wejść/4 wyjść | 8 wejść/6 wyjść | 14 wejść/10 wyjść |
| | Analogowe | 2 wejścia | 2 wejścia | 2 wejścia/2 wyjścia |
| Rozmiar obrazu procesu | Wejścia (I) | 1024 bitów | 1024 bitów | 1024 bitów |
| | Wyjścia (Q) | 1024 bitów | 1024 bitów | 1024 bitów |
| Pamięć bitów (M) | 4096 bitów | 4096 bitów | 8192 bitów | 8192 bitów |
| Moduł rozszerzeń sygnałów (SM) | Brak | 2 | 8 | 8 |
| Płytki sygnałowa (SB) moduł baterii lub płytka komunikacyjna (CB) | 1 | 1 | 1 | 1 |
| Moduł komunikacyjny (CM) (lewostronne rozszerzenie) | 3 | 3 | 3 | 3 |
| Szybkie liczniki | Razem | Wbudowane 3 linie I/O, 5 z SB | Wbudowane 4 linie I/O, 6 z SB | 6 |
| | Jedno-fazowe | 3 przy 100 kHz SB: 2 przy 30 kHz | 3 przy 100 kHz 1 przy 30 kHz SB: 2 przy 30 kHz | 3 przy 100 kHz 3 przy 30 kHz |
| | Kwadra-turowe | 3 przy 80 kHz SB: 2 przy 20 kHz | 3 przy 80 kHz 1 przy 20 kHz SB: 2 przy 20 kHz | 3 przy 80 kHz 3 przy 20 kHz |
| Wyjścia impulsowe ¹ | 4 | 4 | 4 | 4 |
| Karta pamięci | Karta pamięci SIMATIC (opcjonalnie) | | | |
| Czas podtrzymania zegara czasu rzeczywistego | 20 dni, typowo/12 dni minimum przy +40°C (wbudowany bezobsługowy superkondensator) | | | |
| PROFINET | 1 port komunikacyjny typu Ethernet | | | 2 porty komunikacyjne typu Ethernet |
| Rzeczywisty czas wykonania instrukcji matematycznych | 2,3 µs/instrukcję | | | |
| Szybkość wykonania operacji Boolowskiej | 0,08 µs/instrukcję | | | |

¹ Dla modeli CPU z wyjściami przekaźnikowymi należy zainstalować płytę sygnałów cyfrowych (SB), aby móc skorzystać z wyjść impulsowych.

Do każdego CPU można podłączyć do 3 paneli operatorskich HMI. Łączna liczba interfejsów HMI jest zależna od typu paneli HMI użytych w aplikacji. Przykładowo, do jednego CPU można przyłączyć do trzech paneli SIMATIC Basic lub do dwóch paneli SIMATIC Comfort i dodatkowo jeden panel SIMATIC Basic.

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 231).

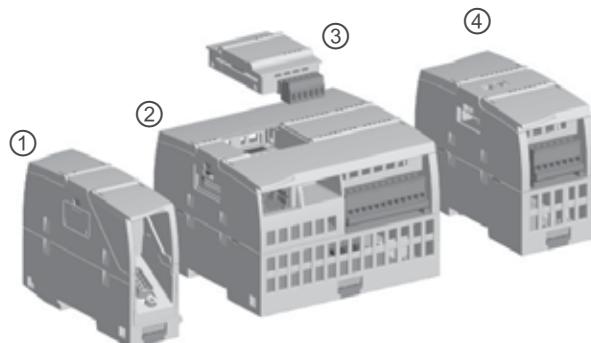
Tabela 1.2. Bloki, timery i liczniki obsługiwane przez S7-1200

| Element | Opis |
|-----------------------|--|
| Blok | Typ OB, FB, FC, DB |
| | Rozmiar 30 kb (CPU 1211C) 50 kb (CPU 1212C) 64 kb (CPU 1214C oraz CPU 1215C) |
| | Liczba do 1024 bloków razem (OB + FB + FC + DB) |
| | Zakres adresów dla bloków FB, FC, i DB od 1 do 65 535 (np. od FB 1 do FB 65 535) |
| | Głębokość zagnieźdzenia 16 z cyklu programu lub bloku startowego OB; 4 z przerwania opóźnienia, przerwania cyklicznego, przerwania sprzętowego, przerwania czasowego lub przerwania od obsługi błędu w OB |
| | Monitorowanie status bloków dwustanowych może być monitorowany w trybie ciągłym |
| Blok organizacyjny OB | Cykl programu Wielokrotny: OB 1, od OB 200 do OB 65 535 |
| | Startowy Wielokrotne: OB 100, od OB 200 do OB 65 535 |
| | Przerwania opóźnienia i przerwania cykliczne 4 ¹ (1 na zdarzenie): od OB 200 do OB 65 535 |
| | Przerwania sprzętowe (zbocza i HSC) 50 (1 na zdarzenie): od OB 200 do OB 65 535 |
| | Przerwania od błędu czasowego 1: OB 80 |
| | Przerwania od błędu diagnostycznego 1: OB 82 |
| Timery | Typ IEC |
| | Ilość Ograniczona tylko przez rozmiar pamięci |
| | Magazynowanie Struktura w DB, 16 bitów na timer |
| Liczni | 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 Opóźnienie przerwania i przerwania cykliczne używają tych samych zasobów pamięci CPU. W związku z tym można uzyskać w sumie tylko 4 tego typu przerwania. Nie można uzyskać równocześnie 4 przerwań czasowych i 4 przerwań cyklicznych.

1.2. Możliwość rozbudowy CPU

Rodzina sterowników S7-1200 zawiera wiele typów modułów rozszerzeń i płyt 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 231).



- ① Moduł komunikacyjny (CM), procesor komunikacyjny (CP), lub TS adapter
- ② CPU
- ③ Płytki sygnałowa (SB), płytki komunikacyjna (CB) lub moduł baterii (BB – *Battery Board*)
- 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 × 24 V DC, 200 kHz | 2 × 24 V DC wejście/2 × 24 V DC wyjście |
| | 4 × 5 V DC, 200 kHz | 4 × 5 V DC, 200 kHz | 2 × 24 V DC wejście/2 × 24 V DC wyjście 200 kHz |
| ④ cyfrowe SM | 8 × 24 V DC | 8 × 24 V DC 8 × wyjście przekaźnikowe | 2 × 5 V DC wejście/2 × 5 V DC wyjście, 200 kHz |
| | | | 8 × 24 V DC wejście/8 × 24 V DC wyjście 8 × 24 V DC wejście/8 × wyjście przekaźnikowe 8 × 120/230 V AC wejście/8 × wyjście przekaźnikowe |
| | 16 × 24 V DC | 16 × 24 V DC wyjście 16 × wyjście przekaźnikowe | 16 × 24 V DC wejście/16 × 24VDC wyjście 16 × 24 V DC wejście/16 × wyjście przekaźnikowe |

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 × wejścia 16-bitowe analogowe 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 |
| ① Procesor komunikacyjny (CP) | Łączność przez modem | GPRS |
| ① Płytki komunikacyjne (CB) | RS485 | Pół-duplex |
| ① Zdalne połączenie (TeleService) | TS Adapter IE Basic ¹ | Połączenie z CPU |
| | TS Adapter GSM | GSM/GPRS |
| | TS Modem | Modem |
| | TS Adapter ISDN | ISDN |
| | TS Adapter RS232 | RS232 |

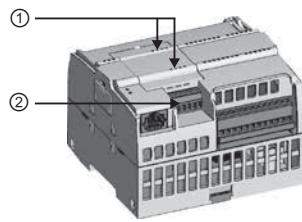
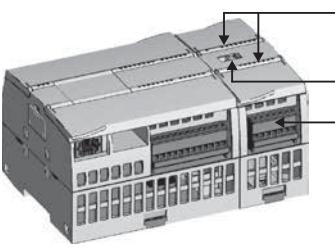
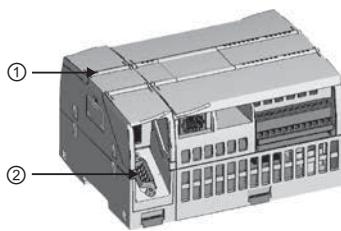
¹ TS adaptery pozwalają na połączenie różnych interfejsów komunikacyjnych z portem PROFINET jednostki centralnej CPU. TS adapter należy przyłączyć do CPU po lewej stronie, a następnie do niego można podłączyć do trzech różnych adapterów modułowych.

Tab. 1.6. Inne moduły

| Moduł | Opis |
|-----------------|---|
| ③ Battery Board | Moduł montowany w gnieździe rozszerzeń znajdującym się na płycie głównej sterownika. Zapewnia długoterminowe podtrzymwanie pracy zegara RTC |

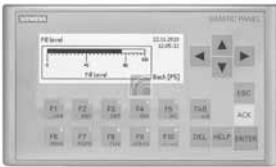
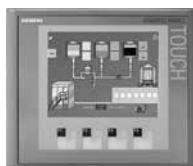
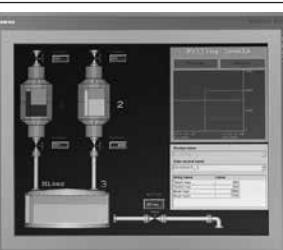
1.3. Moduły S7-1200

Tabela 1.7. Moduły rozszerzeń S7-1200

| Typ modułu | Opis |
|--|--|
| CPU obsługuje jedną płytę rozszerzającą: <ul style="list-style-type: none"> • Płytki sygnałowe (SB) pozwalają dodawać do CPU porty I/O. SB jest dopasowana od strony frontowej CPU • Płytki komunikacyjne (CB) pozwalają na dodanie kolejnego portu komunikacyjnego do CPU • Moduł baterii BB zapewnia długoterminowe podtrzymwanie pracy zegara RTC |  <p>① Diody LED statusu SB ② Wyjmowalna listwa zaciskowa</p> |
| Moduły rozszerzeń (SM) służą do zwiększania funkcjonalności CPU. Są dopasowane z prawej strony CPU. <ul style="list-style-type: none"> • Cyfrowe I/O • Analogowe I/O • RTD i termopary |  <p>① Diody LED statusu portów I/O modułu rozszerzeń ② Gniazdo magistrali ③ Wyjmowalna listwa zaciskowa</p> |
| Moduły komunikacyjne (CM) oraz procesory komunikacyjne (CP) zwiększają funkcjonalność komunikacyjną CPU o takie protokoły jak PROFIBUS lub łączność po RS232/RS485 (dla PtP, Modbusa lub USS). CP zapewnia kompatybilność z innymi typami standardów połączeń, takimi jak sieć GPRS. <ul style="list-style-type: none"> • CPU obsługuje do 3 modułów komunikacyjnych (CM lub CP) • Każdy moduł komunikacyjny jest dopasowany z lewej strony CPU (lub z lewej strony innego CM lub CP, który jest już dopasowany do CPU) |  <p>① Diody LED statusu modułu komunikacyjnego ② Złącze komunikacyjne</p> |

1.4. Panele operatorskie Basic HMI

W miarę jak systemy wizualizacji stają się standardowym elementem wielu maszyn i urządzeń, SIMATIC HMI Basic Panels 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.

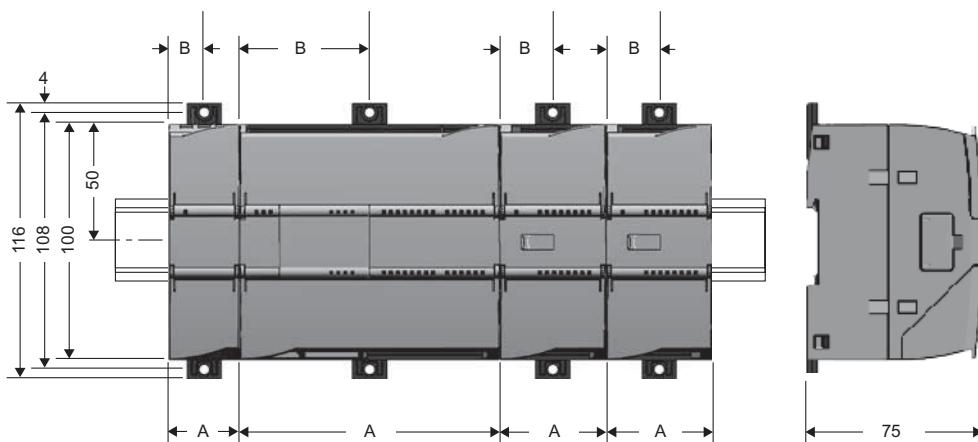
| Panel Basic HMI | Opis | Dane techniczne |
|--|---|---|
|  KP 300 Basic PN | <p>Klawiatura membranowa o przekątnej 3,6 cala z 10 swobodnie programowanymi przyciskami.</p> <ul style="list-style-type: none"> Wyświetlacz monochromatyczny (STN, czarno-biały). Wymiary 87 mm × 31 mm (3,6") Programowany kolor podświetlenia (biały, zielony, żółty lub czerwony). Rozdzielcość: 240 × 80 | <ul style="list-style-type: none"> 250 tagów 50 ekranów procesu 200 alarmów 25 krzywych trendu 32 kB dedykowanej pamięci 5 receptor, 20 rekordów danych, 20 zmiennych |
|  KTP 400 Basic PN | <p>4" ekran dotykowy z 4 dotykovymi przyciskami funkcyjnymi</p> <ul style="list-style-type: none"> Wyświetlacz monochromatyczny (STN, skala szarości) Wymiary 6,79 mm × 57,59 mm (przekątna 3,8") Wertykalnie lub horyzontalnie Rozdzielcość: 320 × 240 | <ul style="list-style-type: none"> 250 tagów 50 ekranów procesu 200 alarmów 25 krzywych trendu 32 kB dedykowanej pamięci 5 receptor, 20 rekordów danych, 20 zmiennych |
|  KTP 600 Basic PN | <p>6" ekran dotykowy z 6 dotykovymi przyciskami funkcyjnymi</p> <ul style="list-style-type: none"> Wyświetlacz kolorowy (TFT, 256 kolorów) lub monochromatyczny (STN, skala szarości) Wymiary 115,2 mm × 86,4 mm (przekątna 5,7") Wertykalnie lub horyzontalnie Rozdzielcość: 320 × 240 | <ul style="list-style-type: none"> 500 tagów 50 ekranów procesu 200 alarmów 25 krzywych trendu 32 kB dedykowanej pamięci 5 receptor, 20 rekordów danych, 20 zmiennych |
|  KTP 1000 Basic PN | <p>10" ekran dotykowy z 8 dotykovymi przyciskami funkcyjnymi</p> <ul style="list-style-type: none"> Wyświetlacz kolorowy (TFT, 256 kolorów) Wymiary 211,2 mm × 158,4 mm (przekątna 10,4") Rozdzielcość: 640 × 480 | <ul style="list-style-type: none"> 500 tagów 50 ekranów procesu 200 alarmów 25 krzywych trendu 32 kB dedykowanej pamięci 5 receptor, 20 rekordów danych, 20 zmiennych |
|  TP 1500 Basic PN | <p>15" ekran dotykowy</p> <ul style="list-style-type: none"> Wyświetlacz kolorowy (TFT, 256 kolorów) Wymiary 304,1 mm × 228,1 mm (przekątna 15,1") Rozdzielcość: 1024 × 768 | <ul style="list-style-type: none"> 500 tagów 50 ekranów procesu 200 alarmów 25 krzywych trendu 32 kB dedykowanej pamięci (zintegrowana pamięć Flash) 5 receptor, 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.

Każde urządzenie CPU, SM, CM, oraz CP montowana są na standardowej szynie lub na płycie. W celu montażu na szynie stosuje się zaciski szynowe dla zabezpieczenia zawieszonego urządzenia. Zaciski te mogą także być zatrzaśnięte w szerszej pozycji tak, aby umożliwić przykręcenie jednostki bezpośrednio do płyty montażowej. Wewnętrzna średnica otworów montażowych zacisków wynosi 4,3 mm.

CPU 1211C, CPU 1212C, CPU 1214C



CPU 1215C

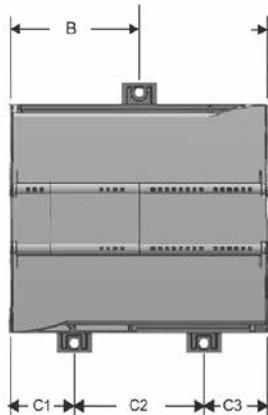


Tabela 1.8. Wymiary montażowe [mm]

| Urządzenia S7-1200 | | Szerokość A | Szerokość B | Szerokość C |
|------------------------------|---|--------------------|-------------|--|
| CPU | CPU 1211C i CPU 1212C | 90 mm | 45 mm | – |
| | CPU 1214C | 110 mm | 55 mm | – |
| | CPU 1215C | 130 mm | 65 (góra) | Dół: C1: 32,5, C2: 65, 33: 32,5 |
| Moduły rozszerzeń | Cyfrowe: 8 i 16 punktów; analogowe: 2, 4 i 8 punktów, termopary: 4 i 8 punktów, RTD: 4 punkty | 45 mm | 22,5 mm | – |
| | Cyfrowe wyjścia przekaźnikowe (8 × DQ, przełączane) | 70 mm | 22,5 mm | – |
| | Analogowe: 16 punktów, RTD: 8 punktów | 70 mm | 35 mm | – |
| Moduły komunikacyjne | CM 1241 RS232, CM 1241 RS485 oraz CM 1241 RS422/485 | 30 mm | 15 mm | – |
| | CM 1243-5 PROFIBUS dla urządzenia master, | | | – |
| | CM 1242-5 PROFIBUS dla urządzenia slave | | | – |
| | CM 1242-2 AS-i Master | | | – |
| | CP 1242-7 GPRS | | | – |
| Teleservice Adapter IE Basic | | 60 ¹ mm | 15 mm | – |

¹ Ponieważ konieczne jest zainstalowanie modułu TS Adapter całkowita szerokość (wymiar „A”) wynosi 60 mm.

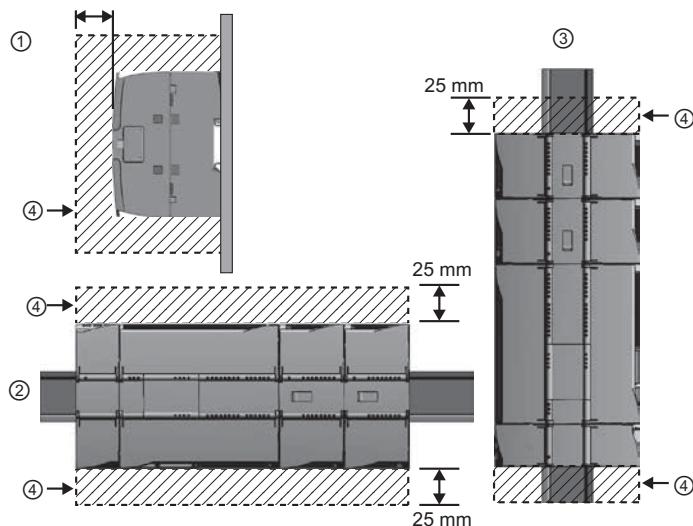
Każde urządzenie CPU, SM, CM oraz CP jest przystosowane do montażu na szynie DIN lub panelu. Podczas montażu urządzeń na szynie DIN należy stosować klipsy zabezpieczające, chroniące urządzenia po zamontowaniu. Rozsuwanie klipsów zabezpieczających powoduje powstanie uchwytów pozwalających na montaż urządzeń w panelu za pomocą śrub mocujących. Średnica otworów mocujących znajdujących się na klipsach DIN wynosi 4,3 mm.

Podczas montażu urządzeń należy pamiętać o zachowaniu odstępów 25 mm powyżej i poniżej obudowy w celu zapewnienia odpowiedniej cyrkulacji powietrza.

Przy projektowaniu instalacji należy zawsze zwrócić uwagę na następujące wytyczne:

- Należy odseparować urządzenia od ciepła, wysokiego napięcia i zakłóceń elektrycznych.
- Należy zapewnić odpowiednią przestrzeń dla chłodzenia i okablowania. Dla zapewnienia naturalnego konwekcyjnego przepływu powietrza należy zapewnić 25 mm strefę termiczną nad i pod urządzeniem.

Dokładne wytyczne i wymagania dla instalacji znajdują się w *S7-1200 System Manual*.



① Widok z boku

③ Montaż pionowy

② Montaż poziomy

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 urządzeń.

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

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

Nowe możliwości sterownika:

- Aktualizacja firmware'u odbywa się za pomocą strony internetowej generowanej przez CPU.
- Możliwość współpracy z trzema modułami master PROFIBUS DP CM1243-5 lub trzema modułami master AS-i CM1243-2.

Uwaga

Użycie trzech modułów master AS-i wymaga aktualizacji ich firmware'u.

Nowe moduły sterownika S7-1200

Różnorodność nowych modułów rozszerza możliwości sterownika S7-1200 w zakresie zaspokojenia potrzeb automatyzacji:

- Nowe wersje i modyfikacje CPU:
 - Nowe moduły CPU1215C DC/DC/DC, CPU1215C DC/DC/Relay oraz CPU1215C AC/DC/Relay wyposażono w pamięć roboczą o pojemności 100 kB, dwa kanały Ethernet oraz wyjścia analogowe;
 - Nowe i zmodyfikowane wersje CPU1211C, CPU1212C oraz CPU1214C szybciej wykonują programy, mogą być wyposażone w 4 kanały PTO (CPU1211C wymaga zastosowania modułu sygnałowego), mają powiększoną do 10 kB pamięć z podtrzymaniem, wydłużono także do 20 dni czas podtrzymania zegara czasu rzeczywistego.
- Nowy moduł sygnałowy I/O SM1231AI, wyposażony w cztery wejścia o rozdzielczości 16-bitowej, charakteryzuje się zwiększoną częstotliwością próbkowania.
- Nowy moduł płytka sygnałowej baterii (BB1297) zapewnia długi czas pracy zegara czasu rzeczywistego. Moduł BB1297 jest przystosowany do montażu w gnieździe modułu płytka sygnałowych sterowników S7-1200 (wymagany firmware 3.0 lub nowszy).

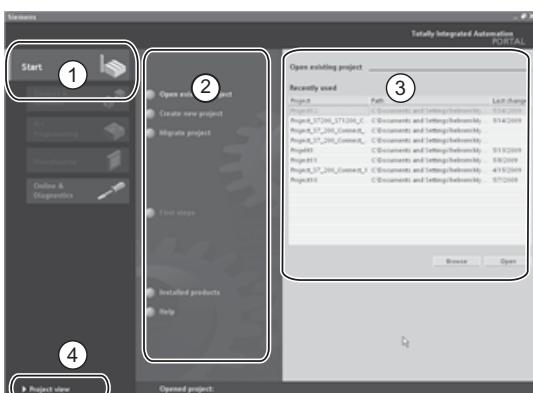
Aby użyć nowych modułów wymagana jest odpowiednia wersja oprogramowania inżynierskiego STEP7 V11 SP2 Update 3 lub nowsza (Basic lub Professional). Konieczne jest pobranie i zainstalowanie pakietów HSP (*Hardware Support Package*), do czego służy w menu STEP7 opcja *Options ->Support Packages*.

Patrz także

Międzynarodowe wsparcie techniczne (<http://support.automation.siemens.com>).

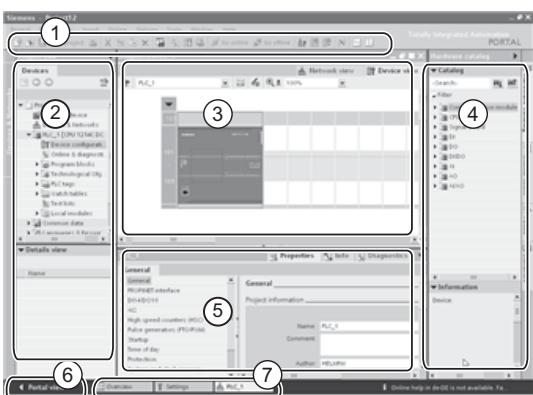
STEP 7 ułatwia pracę

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 przełączania się pomiędzy obydwooma 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 wiadomości diagnostyczne i inne wiadomości.

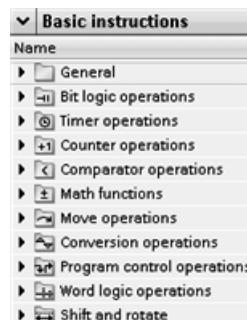
Pokazując wszystkie otwarte edytory, pasek edytora pozwala na szybszą i wydajniejszą pracę. Wystarczy kliknąć na dany edytor, aby przełączyć się pomiędzy poszczególnymi otwartymi edytorami. Istnieje także możliwość równoczesnego wyświetlania dwóch edytorów w pozycji poziomej lub pionowej. Pozwala to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

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.

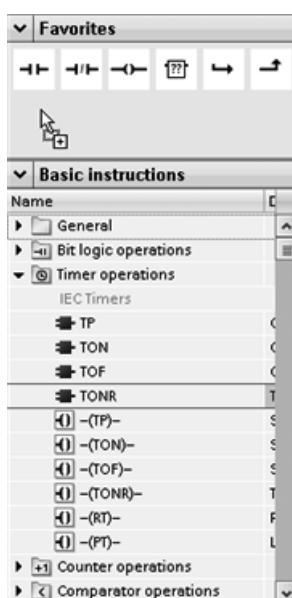


2.2. Łatwy dostęp do ulubionych instrukcji z paska narzędzi

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



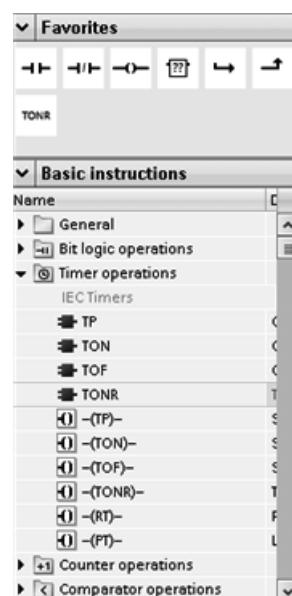
(*Favorites* w drzewie instrukcji uruchamiane jest poprzez podwójne kliknięcie ikony).



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

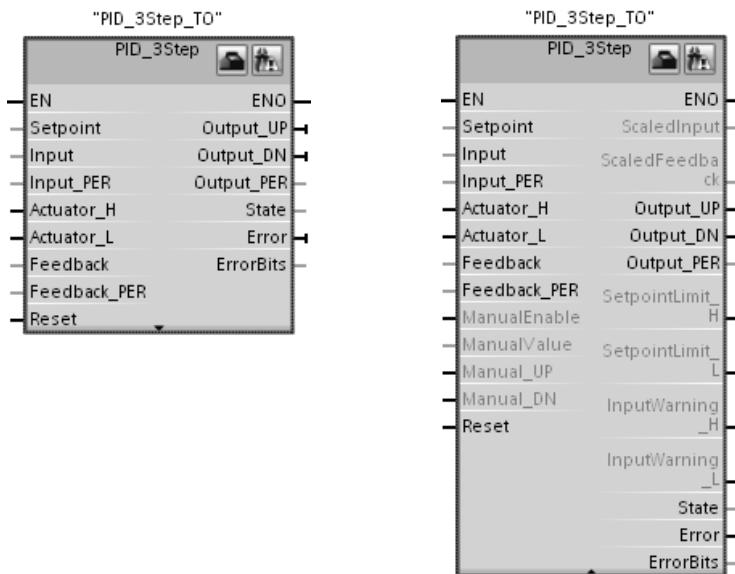


Niektóre instrukcje pozwalają na tworzenie dodatkowych wejść i wyjść.

- 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

Niekotóre bardziej złożone instrukcje wyświetlają tylko kluczowe wejścia i wyjścia. Można rozwinąć widok instrukcji poprzez naciśnięcie strzałki 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 70).

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 narzędzi online *Online Tools*

wyświetla panel operacyjny, który pokazuje tryb pracy podłączonego CPU. Panel operacyjny pozwala także na zmianę trybu pracy podłączonego CPU.

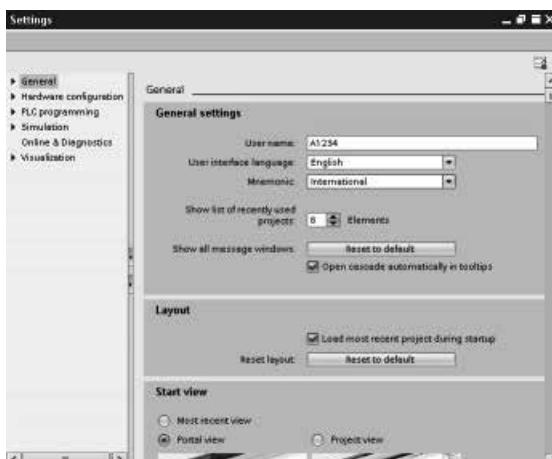


Użyj przycisku w panelu operacyjnym do zmiany trybu pracy (STOP lub RUN). Panel operacyjny ma także przycisk MRES do czyszczenia pamięci.

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

Konfigurację domyślnych ustawień sterownika po wyłączeniu zasilania szczegółowo opisano w rozdziale „Tryby pracy sterownika” publikacji „Podręcznik systemowy S7-1200”.

2.6. Łatwa zmiana wyglądu i konfiguracji dla STEP 7

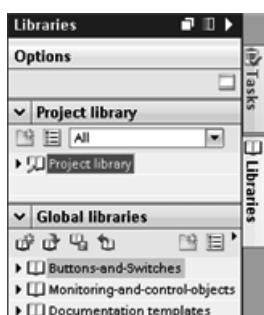


Dostępne jest szeroka gama ustawień, takich jak wygląd interfejsu, język lub też zmiana katalogu do zapisu projektu użytkownika.

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

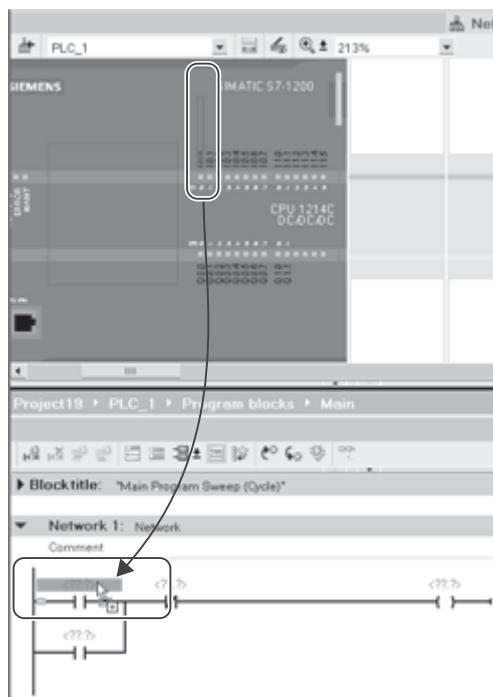
Rozwój i ciągłe tworzenie oraz doskonalenie niektórych instrukcji (takich jak Modbus, PID i instrukcje sterujące napędami) spowodował powstanie wielu wersji tych instrukcji. Dla zapewnienia kompatybilności ze starszymi projektami, STEP 7 pozwala użytkownikowi na wybór wersji instrukcji użytej w programie.



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

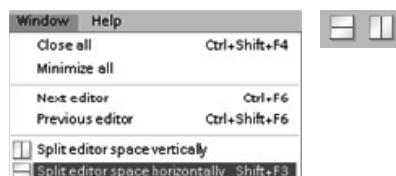


W celu przyspieszenia realizacji zadań STEP 7 pozwala na przeciąganie i upuszczanie elementów pomiędzy poszczególnymi edytorami. Przykładowo można chwycić adres wejścia z CPU i przenieść go do instrukcji w programie użytkownika.

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.

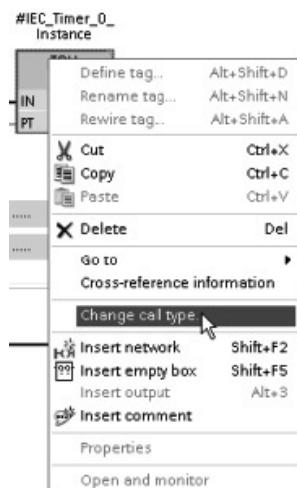
Do wyświetlenia dwóch edytorów jednocześnie użyj komendy *Split editor* z menu wybieralnego lub przycisków na pasku narzędzi.



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



2.10. Zmiana typu wywołania DB



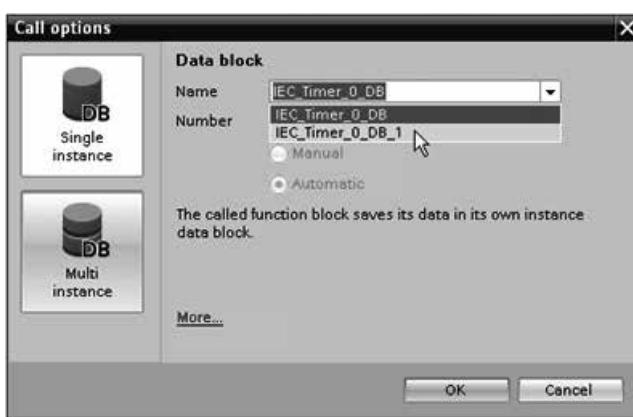
STEP 7 pozwala na łatwą zmianę lub stworzenie powiązania DB (bloku danych) dla instrukcji lub FB (bloku funkcji) znajdującym się w FB.

Możesz przełączać powiązania pomiędzy różnymi DB.

Możesz zmieniać powiązania pomiędzy pojedynczą instancją DB i DB z wielomainstancjami.

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

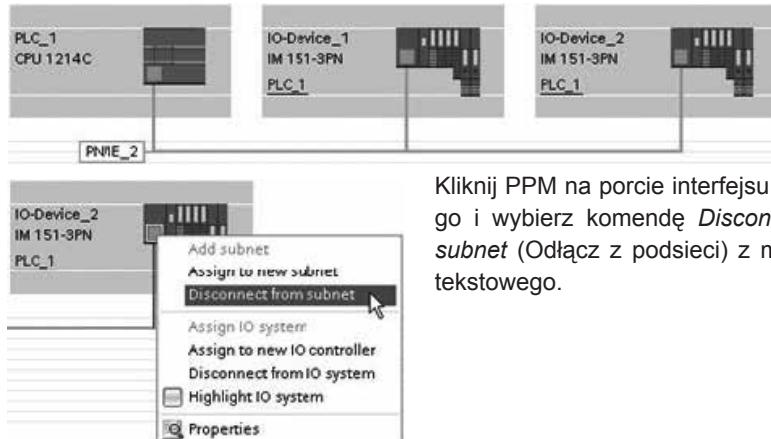
Możesz uzyskać dostęp do komendy *Change call state* (Zmiana typu wywołania) PPM na instrukcję lub FB w edytorze programu lub wybierając komendę *Block call* (Wywołanie bloku) z menu *Options* (Opcje).



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

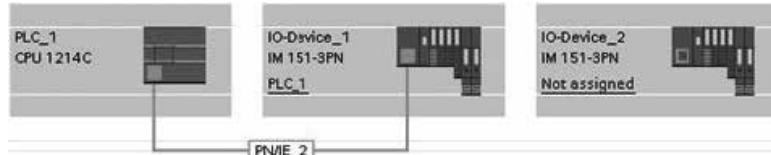
2.11. Tymczasowe odłączenie urządzeń z sieci

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.

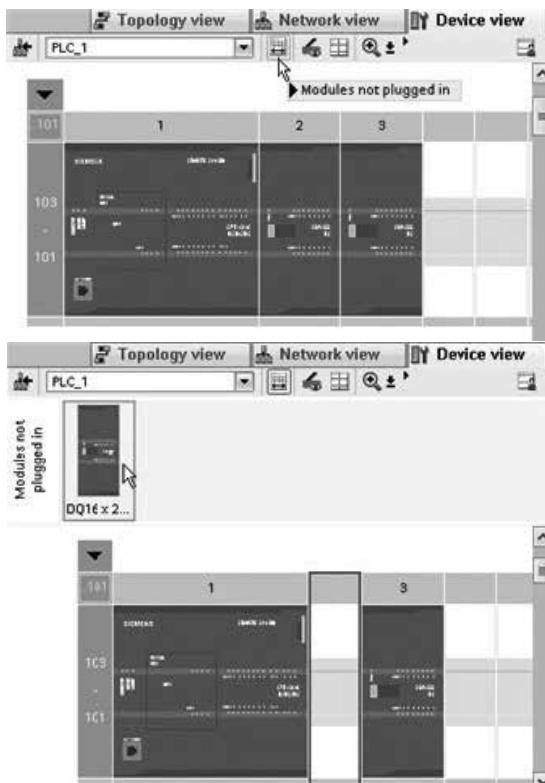
STEP 7 rekonfiguruje połączenia sieciowe, ale nie usuwa rozłączonego urządzenia z projektu. Podczas gdy połączenie sieciowe danego urządzenia jest usunięte, adresy jego interfejsu nie są zmienione.



Podczas ładowania nowych połączeń sieciowych, CPU przełącza się w tryb pracy STOP.

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

2.12. Wirtualne „rozłączanie” modułów bez straty konfiguracji



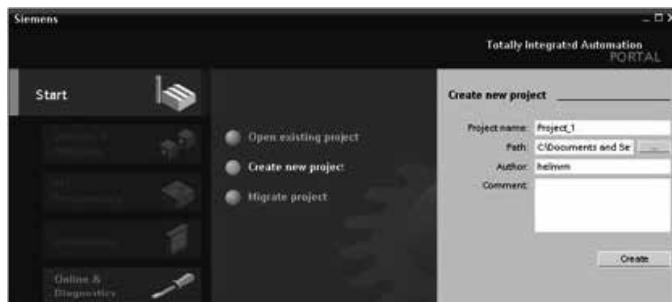
STEP 7 ma obszar magazynowania dla rozłączonych modułów. Możesz łatwo przemieścić moduł z wirtualnej listwy montażowej, aby zapisać konfigurację tego modułu. Te rozłączone moduły są zapisywane razem z projektem pozwalać na powtórne podłączenie modułu w przyszłości bez konieczności rekonfiguracji jego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



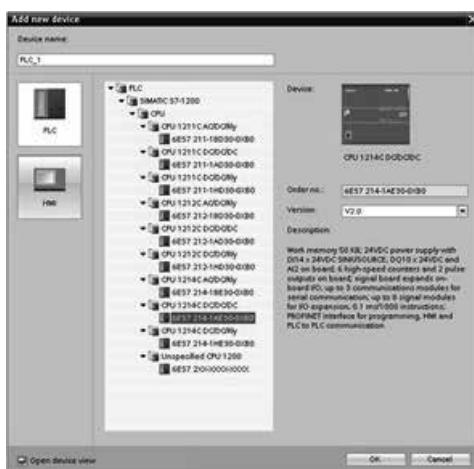
W portalu startowym kliknij *Create new project* (Stwórz nowy projekt).

Wprowadź nazwę projektu i kliknij przycisk *Create*.



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

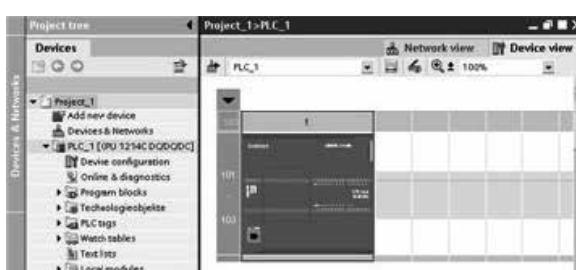
Kliknij przycisk *Add new device* (Dodaj nowe urządzenie).



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).

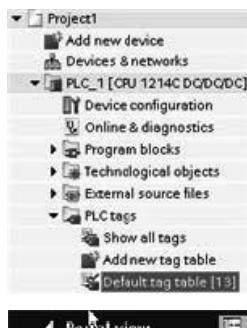
Zauważ, że zaznaczona jest opcja *Open device view* (Otwórz widok urządzenia). Klikając *Dodaj* z tą opcją otwierasz *Device configuration* (Konfigurację urządzenia) w widoku Projektu.



Widok Urządzenia wyświetla dodane CPU.

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

PLC tags (zmienne 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.



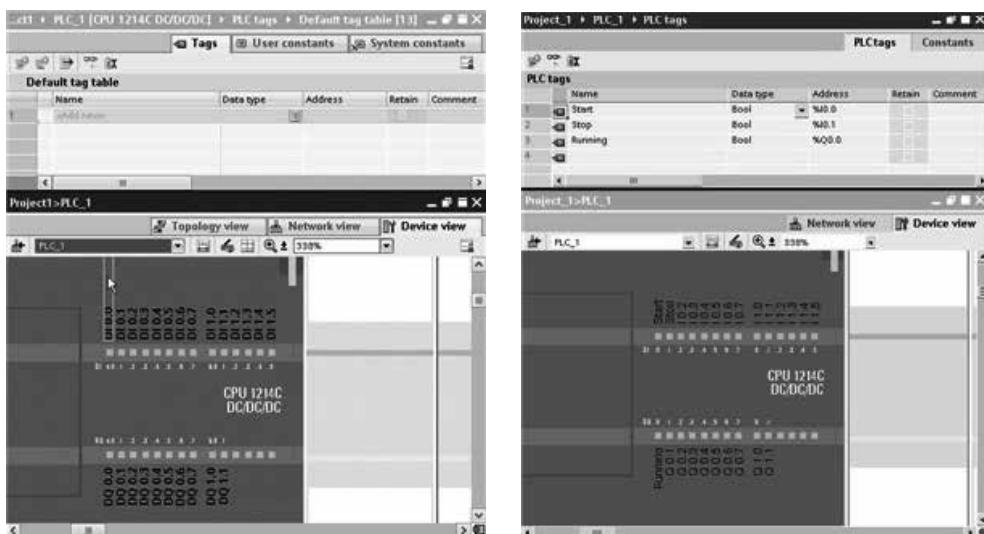
Na pasku narzędzi kliknij przycisk *Split editor space horizontally* (Rozłoż obszar edytorów horyzontalnie).



STEP 7 wyświetla obydwa 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:

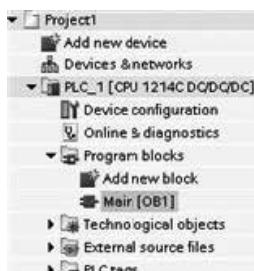
1. Wybierz I0.0 i przeciągnij je do pierwszego wiersza tablicy zmiennych.
2. Zmień nazwę zmiennej z „I0.0” na „Start”.
3. Przeciągnij I0.1 do tablicy zmiennych i zmień nazwę na „Stop”.
4. Przeciągnij Q0.0 (dół CPU) do tablicy zmiennych i zmień jego nazwę na *Running* (Uruchomiony).



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

3.3. Tworzenie prostej sieci w programie użytkownika

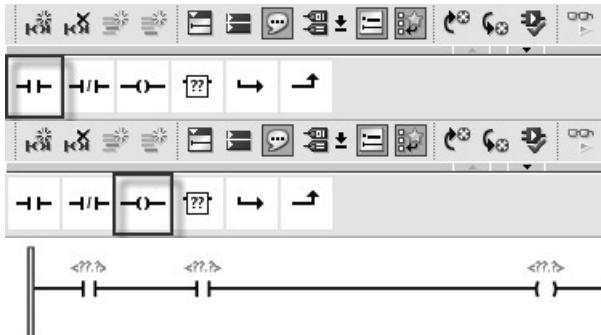
Kod programu użytkownika składa się z instrukcji, które CPU wykonuje w sekwencji. W tym przykładzie użyj języka drabinkowego (LAD) do stworzenia kodu programu. Program w LAD jest sekwencją sieci, która przypomina szczeble drabiny.



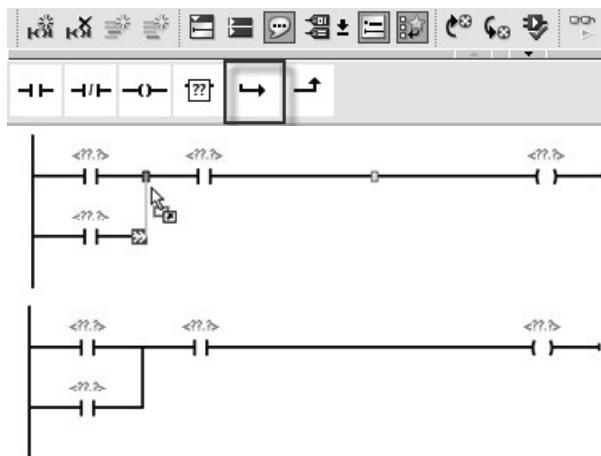
Do otwarcia programu należy:

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.



Zakładka *Ulubione* posiada także przycisk tworzenia gałęzi.



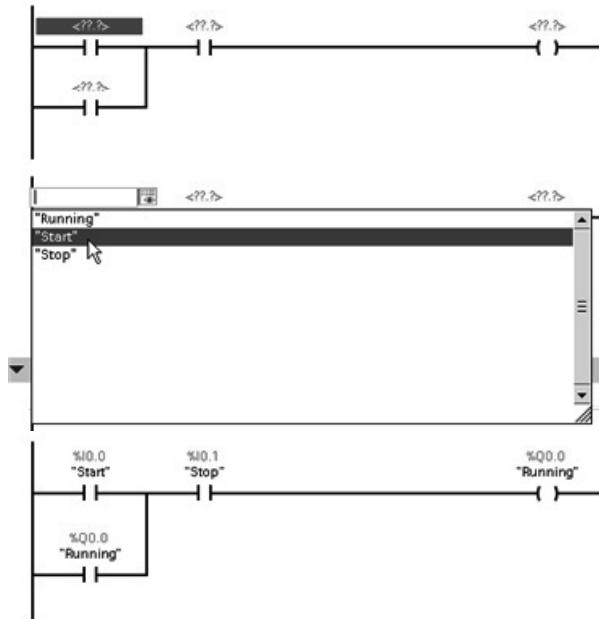
1. Kliknij *Normally open contact* (Styk normalnie otwarty) w *Ulubione*, aby dodać styk do programu
2. Dodaj drugi styk
3. Kliknij *Output coil* (Cewka wyjściowa), aby dodać do programu cewkę.

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 szczeblu) pomiędzy dwoma stykami w pierwszej linii.

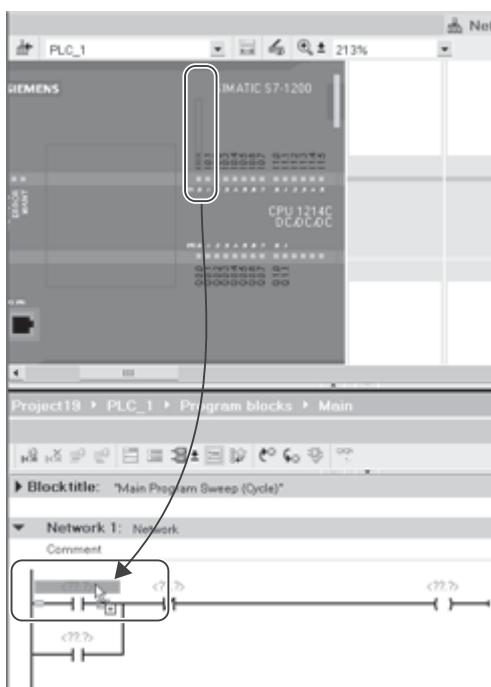
Kliknij przycisk *Save project* na pasku narzędzi, aby zapisać projekt. Zauważ, że edycja szczebla nie musi być zakończona przed zapisaniem. Możesz teraz powiązać nazwy zmiennych z tymi instrukcjami.

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 zmienne 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 29).

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 umożliwia 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*.



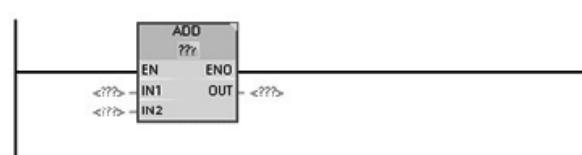
Ogólne pole instrukcji posiada różne instrukcje. W tym przypadku użyj instrukcji ADD:



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



- Przewiń w dół listę i zaznacz instrukcję ADD.
- Kliknij w żółtym narożniku „?” aby wybrać typ danych dla wejścia i wyjścia.

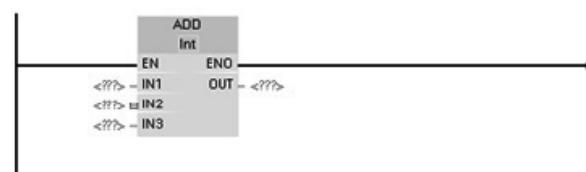


Możesz teraz wpisać zmienne (lub adresy pamięci) dla wartości używanych przez instrukcję ADD.



Możesz stworzyć także dodatkowe wejścia dla pewnych instrukcji:

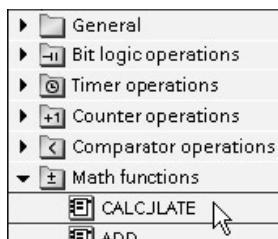
- Naciśnij jedno z wejść znajdujących się wewnętrz instrukcji w polu instrukcji.
- PPM dla wyświetlenia menu kontekstowego i zaznaczenia komendy *Insert input* (Dodaj wejście).



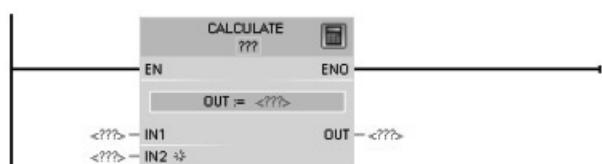
Instrukcja ADD używa teraz trzech wejść.

3.6. Użycie instrukcji CALCULATE dla złożonych równań matematycznych

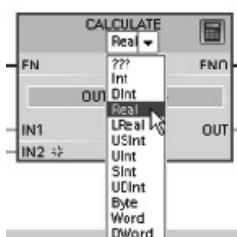
Instrukcja CALCULATE pozwala na stworzenie matematycznej funkcji, która operuje na wielu parametrach wejściowych i generuje wyniki zgodne z zdefiniowanym równaniem.



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



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

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

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

gdzie:

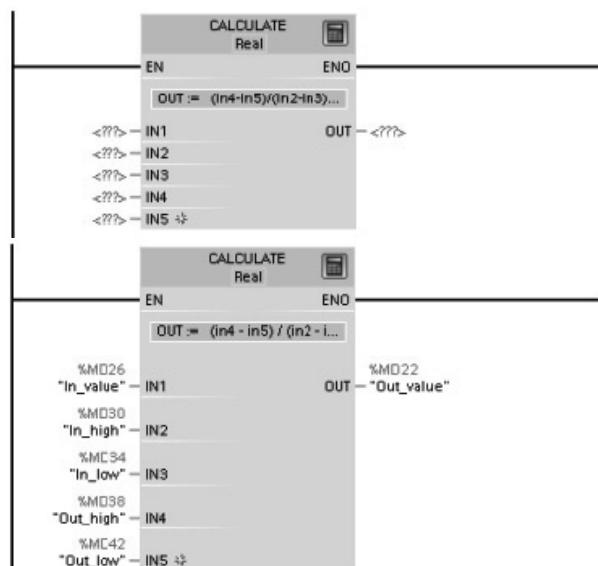
| | | |
|-----------------------------|-------|--|
| $\text{Out}_{\text{value}}$ | (Out) | Przeskalowane wartości wyjściowe |
| In_{value} | (In1) | Wartości analogowego wejścia |
| In_{high} | (In2) | Górna granica dla przeskalowanej wartości wejściowej |
| In_{low} | (In3) | Dolna granica dla przeskalowanej 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:

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

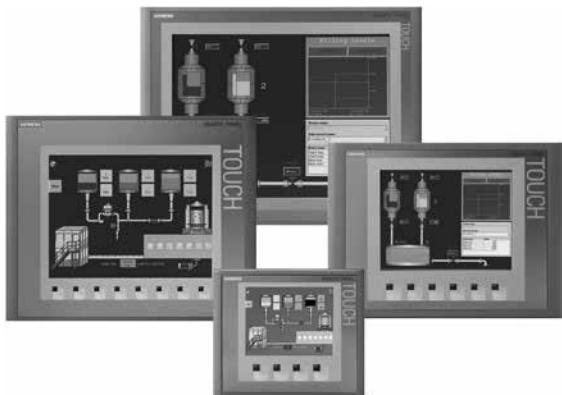


Po naciśnięciu **OK**, instrukcja *Calculate* tworzy wejścia wymagane do jej obsługi.

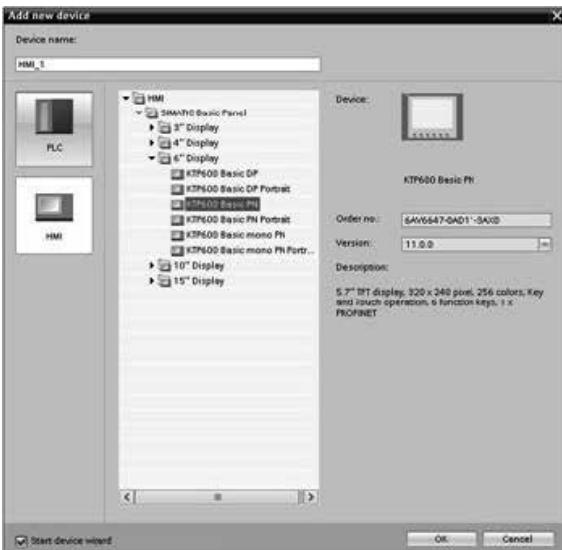


Wprowadź nazwy zmiennych dla wartości poszczególnych parametrów.

3.7. Dodawanie urządzenia HMI do projektu

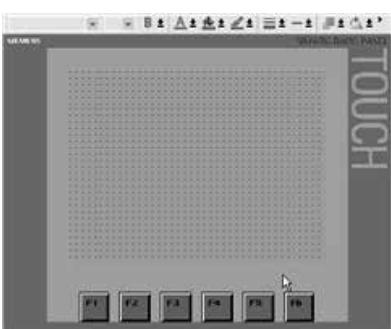


Dodawanie urządzenia HMI do projektu jest bardzo łatwe!



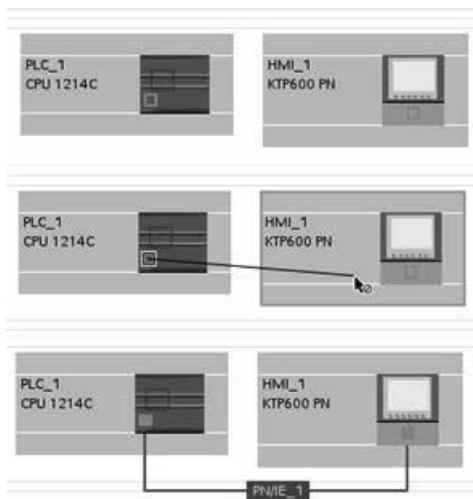
1. Kliknij dwukrotnie na ikonę *Add new device* (Dodaj nowe urządzenie).
 2. Naciśnij przycisk *SIMATIC HMI* w oknie dialogowym *Add new device*.
 3. Wybierz podłączone urządzenie HMI z listy.
- Możesz uruchomić kreatora HMI (HMI wizard), który pomoże Ci w skonfigurowaniu ekranów urządzenia HMI.
4. Naciśnij *OK* dla dodania urządzenia HMI do twojego projektu.

Urządzenie HMI zostało dodane do twojego projektu. STEP 7 ma kreator HMI, który pomaga w konfiguracji wszystkich ekranów i struktur urządzenia HMI.



Jeśli kreator HMI nie zostanie uruchomiony, STEP 7 stworzy prosty, domyślny ekran HMI.

3.8. Tworzenie połączenia pomiędzy CPU i urządzeniem HMI

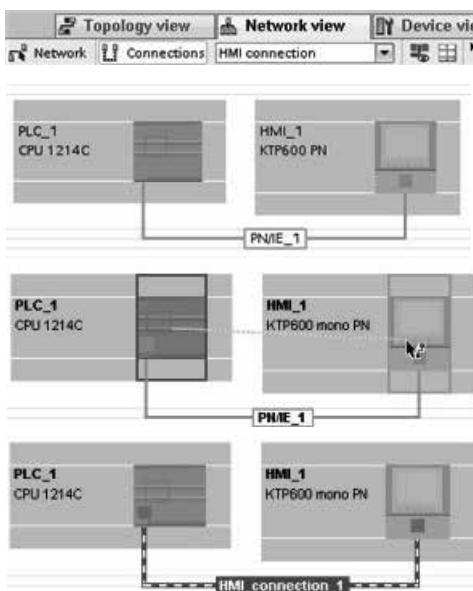


Tworzenie sieci jest proste!

- 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.

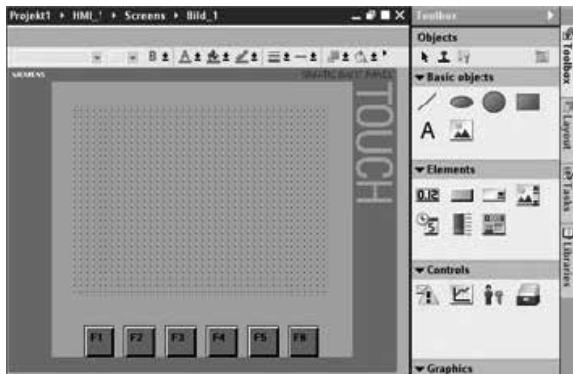
- Po zaznaczeniu połączenia sieciowego naciśnij przycisk *Connections* i wybierz z menu rozwijalnego *HMI connection*.
- Połączenie HMI zmieni kolor obydwu urządzeń na niebieski.
- Zaznacz urządzenie CPU i przeciągnij linię do urządzenia HMI.
- Połączenie HMI pozwoli na konfigurację zmiennych HMI poprzez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

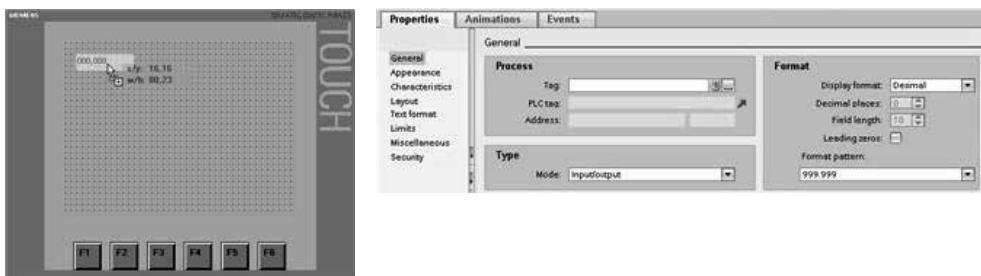
3.10. Tworzenie ekranu urządzenia HMI

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.

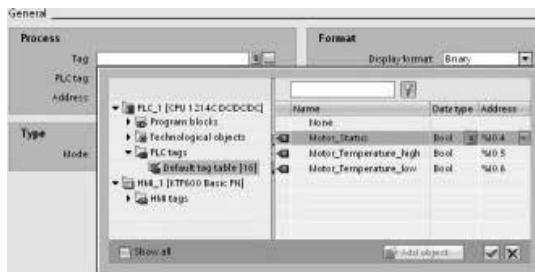
Aby dodać element, wystarczy przeciągnąć i upuścić jeden z elementów na ekranie. Użyj właściwości elementu (w oknie *Inspector*), aby skonfigurować wygląd i zachowanie elementu.



Można również tworzyć elementy na ekranie przez przeciągnięcie i upuszczenie zmiennej PLC z drzewa projektu lub edytora programu do ekranu HMI. Zmiana 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

4.1. Zadania wykonywane w każdym cyklu

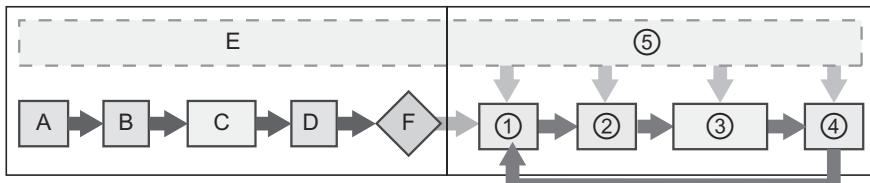
Każdy cykl programu obejmuje ustawienie wyjść, odczytanie stanu wejść, wykonanie instrukcji z programu użytkownika i wykonanie obsługi systemu lub przetwarzanie w tle.



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 zwany 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

- A Czyści obszar pamięci I.
- B Inicjalizuje wyjścia z ostatnimi wartościami lub zastępczymi wartościami.
- C Wykonuje startowe OB.
- D Kopiuje stan fizycznych wejść do pamięci I.
- E Zapisuje zdarzenia przerwań do kolejki oczekującej na wykonanie w trybie RUN.
- F Uaktywnia zapis zawartości pamięci Q do fizycznych wyjść.

RUN

- ① Zapisuje zawartość pamięci Q do wyjść fizycznych.
- ② Kopiuje stan fizycznych wejść do pamięci I.
- ③ Wykonuje cykliczne OB.
- ④ Wykonuje autodiagnostykę.
- ⑤ Obsługuje przerwania w dowolnej części cyklu programu.

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.
- W trybie STARTUP wykonywany jest jednokrotnie startowy OB (jeśli istnieje). W fazie startowej trybu RUN nie są obsługiwane przerwania.
- W trybie RUN regularnie jest powtarzany cykl programu. Mogą się pojawiać przerwania i są przetwarzane w dowolnym miejscu cyklu programu.

Przechodzenie do trybu RUN CPU wykonuje metodą gorącego restartu. Gorący restart nie obejmuje kasowania pamięci, ale pamięć może zostać skasowana odpowiednim rozkazem umieszczonym w programie. Kasowanie pamięci czyści całą pamięć roboczą, obszary pamięci trwałej i nietrwałej oraz kopiuje zawartość pamięci ładowania do pamięci roboczej. Kasowanie pamięci nie czyści zawartości bufora diagnostycznego a także na stałe zapisanej wartości adresu IP. Podczas gorącego restartu inicjalizowane są wszystkie nietrwałe elementy systemu oraz dane użytkownika.

Za pomocą oprogramowania można wyspecyfikować tryb włączania zasilania CPU wraz z metodą restartu. Ta pozycja konfiguracji jest dostępna w menu *Startup* dla CPU pod nagłówkiem *Device Configuration*. Po włączeniu zasilania CPU wykonuje sekwencję testów diagnostycznych i inicjalizuje system. Następnie wchodzi w odpowiedni tryb włączania zasilania. Wykrycie pewnych błędów uniemożliwia CPU wejście w tryb RUN. CPU może pracować w następujących trybach włączania zasilania: w trybie STOP, przejścia do trybu RUN po wykonaniu gorącego restartu oraz przejścia do poprzedniego trybu pracy po wykonaniu gorącego restartu.

Ostrzeżenie

Moduł CPU może samoczynnie przełączać się w tryb STOP w przypadku wystąpienia awarii typu: uszkodzenie lub wymiana modułu sygnałowego lub zakłóceń w linii zasilającej.

Jeżeli moduł CPU skonfigurowano w tryb „Gorący restart w chwili wyłączenia zasilania”, nie przełączy się on automatycznie w tryb RUN do chwili odebrania polecenia RUN z pakietu STEP7. Do momentu odebrania tego polecenia moduł będzie znajdował się w trybie STOP.

Moduły CPU mające pracować niezależnie od pakietu STEP7 powinny być skonfigurowane w tryb „Gorący restart – RUN” powodujący, że po ustąpieniu zakłóceń uniemożliwiających pracę sterownik po ponownym wyłączeniu załączeniu zasilania automatycznie przechodzi w tryb RUN.



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 50).
- 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 13). 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.

System gwarantuje wykonanie kompletnego cyklu programu w czasie nazywanym maksymalnym czasem cyklu; w przeciwnym wypadku generowane jest zderzenie 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 najwyższym 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 czynność początkowa, taka jak przerwanie diagnostyczne lub interwał czasowy może uruchomić wykonywanie OB. Inny blok kodu nie może wywołać 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 26 (należą do niej przerwania od błędu czasowego).

OB sterują następującymi operacjami:

- OB cyklu programu są wykonywane cykliczne jeśli CPU jest w trybie RUN. Główny blok kodu jest cyklicznym OB. To w nim umieszcza się instrukcje, które sterują programem użytkownika i stąd wywołuje się dodatkowe bloki użytkownika. Dopuszcza się użycie wielu cyklicznych OB. I są wykonywane w kolejności zgodnej z ich numeracją. Domyślnym jest OB 1. Pozostałe muszą nosić nazwy OB 200 lub wyższe.
- OB startowy (rozruchowy) jest wykonywany jednorazowo wtedy, kiedy stan CPU zmienia się ze STOP na RUN, wliczając w to włączenie zasilania w trybie RUN i nakazane rozkazem przejście STOP – RUN. Po zakończeniu wykonania startowego OB, jest uruchamiany główny OB „cyklu programu”. Dopuszcza się użycie wielu startowych OB. Domyślnym jest OB 100. Pozostałe muszą nosić nazwy OB 200 lub wyższe.
- Przerwanie cykliczne OB jest wykonywane w określonym interwale czasowym. Przerwanie cykliczne OB przerywa cykliczne wykonanie programu po zdefiniowanym przez użytkownika czasie, na przykład co 2 sekundy. Można skonfigurować do czterech zdarzeń opóźnionych i zdarzeń cyklicznych dla dowolnej chwili, przy czym dozwolony jest jeden OB dla każdego skonfigurowanego zdarzenia opóźnionego i cyklicznego. Nazwa OB musi być OB 200 lub większa.
- OB przerwania sprzętowego jest wykonywany wtedy, kiedy wystąpi istotne zdarzenie, włączając w to zbocza opadające lub narastające na wbudowanych wejściach cyfrowych i zdarzenia HSC (dotyczące szybkiego licznika). OB przerwania sprzętowego przerywa wykonywanie normalnego programu cyklicznego w odpowiedzi na sygnał o zdarzeniu sprzętowym. Użytkownik definiuje w konfiguracji sprzętowej jakie to są zdarzenia. Dozwolony jest jeden

OB dla każdego skonfigurowanego zdarzenia sprzętowego. Nazwa OB musi być OB 200 lub większa.

- OB przerwania błędu czasowego jest wykonywany wtedy, kiedy zostanie wykryty błąd czasowy. OB przerwania błędu czasowego przerywa wykonywanie normalnego programu cyklicznego. OB 80 jest jedynym numerem OB obsługującym zdarzenie błędu czasowego. Użytkownik może określić jaka akcja CPU jest podejmowana w razie, gdy OB 80 nie istnieje: CPU pozostaje w RUN (błąd jest ignorowany) lub następuje zmiana do trybu STOP. Istnieją dwa typy przerwań od błędu czasowego obsługiwane przez OB 80:
 - Przekroczenie maksymalnego czasu cyklu: można skonfigurować maksymalny czas wykonania cyklu programu we właściwościach CPU. Ustawienia domyślne CPU dla przekroczenia tego czasu powodują przejście CPU w STOP;
 - Inne błędy czasowe, takie jak wywołanie drugiego cyklicznego przerwania przed zakończeniem wykonania przez CPU pierwszego. Domyślna akcja dla CPU w tym wypadku to pozostanie w trybie RUN.
- OB przerwania błędu diagnostycznego jest wykonywany wtedy, kiedy zostaje wykryty i zgłoszony błąd diagnostyczny. OB przerwania błędu diagnostycznego przerywa wykonywanie normalnego programu cyklicznego wtedy, kiedy moduł zdolny do wykonywania diagnostyki rozpoznaje błąd (o ile wystawienie sygnału przerwania w odpowiedzi na błąd diagnostyczny zostało w tym module odblokowane). OB 82 jest jedynym numerem OB obsługującym zdarzenie błędu diagnostycznego. Po włożeniu instrukcji STP (przełączca CPU w tryb STOP) do bloku OB 82 sterownik przełącza się w tryb STOP po wystąpieniu tego typu błędu. Jeśli nie ma diagnostycznego OB w programie, CPU pozostaje w trybie RUN i ignoruje błąd.

4.3.3. Kolejkowanie 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 jednakowo, 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. Istnieją również zdarzenia jak błąd diagnostyczny, bądź błąd czasowy, które powstają tylko w przypadku wystąpienia błędu. Do określenia porządku wykonywania procedur obsługi zdarzeń stosuje się priorytety, grupy priorytetów i kolejki.

Zdarzenie cyklu programu wykonywane jest przy każdym cyklu programu (lub skanu). Podczas jednego cyklu CPU ustawia wyjścia, odczytuje wejścia i wykonuje OB cyklu programu. Zdarzenie cyklu programu jest konieczne i zawsze włączone. W programie można nie mieć bloków OB z cyklem programu lub mieć wiele bloków OB wybranych do obsługi zdarzenia cyklu programu. W momencie uruchomienia zdarzenia cyklu programu zostanie wykonane zdarzenie cyklu programu najniżej

numerowanego bloku OB (zwykle jest to OB 1). Zdarzenia cyklu programu innych bloków OB są uruchamiane sekwencyjnie, w kolejności numerycznej.

Zdarzenie przerwania cyklicznego pozwala na konfigurację wykonania przerwania OB w określonym czasie skanu. Wstępny czas skanu jest konfigurowany, gdy dany OB jest tworzony i wybrany do cyklicznych przerwań. Zdarzenie cykliczne przerywa cykl programu i wykonuje OB cyklicznego przerwania (zdarzenie cykliczne ma wyższy priorytet niż zdarzenie cyklu programu).

Tylko jeden OB obsługujący cykliczne przerwania może być dołączony do zdarzeń cyklicznych.

Każdemu zdarzeniu cyklicznemu może być przypisane przesunięcie fazowe tak, aby realizacja cyklicznych przerwań w tym samym czasie skanowania mogła być przesunięta pomiędzy sobą o to przesunięcie fazowe. Jego domyślna wartość wynosi 0. Aby zmienić wartość przesunięcia fazowego należy zmienić wstępny czas skanowania dla cyklicznych zdarzeń klikając PPM na blok OB w drzewie projektu, kliknąć *Properties* i następnie *Cyclic interrupt*. W tym polu należy wpisać nową wartość. Można zbadać oraz zmienić czas skanu i przesunięcie czasowe z poziomu programu używając instrukcji badania cyklicznego przerwania (QRY_CINT) a także ustawienia cyklicznego przerwania (SET_CINT). Czas skanu i czas przesunięcia fazowego ustawione przez instrukcję SET_CINT powrócią do wartości początkowych po wyłączeniu i włączeniu zasilania lub przy przejściu do trybu STOP. CPU obsługuje do czterech zdarzeń przerwań cyklicznych i od opóźnienia.

Zdarzenie rozruchu wykonywane jest raz przy przejściu z trybu STOP do RUN i powoduje uruchomienie startowego OB. Wiele OB może być użytych do zdarzenia rozruchowego. Startowe OB są wykonywane w kolejności numerycznej.

Zdarzenia opóźnienia przerwania pozwalają na konfigurację wykonania OB przerwań po minięciu określonego czasu. Czas opóźnienia jest określony przez instrukcję SRT_DINT. OB opóźnienia przerywa wykonywanie normalnego programu cyklicznego wtedy, kiedy upłynie ustalony czas opóźnienia. Można skonfigurować do czterech zdarzeń opóźnionych dla dowolnej chwili, przy czym dozwolony jest tylko jeden OB dla każdego skonfigurowanego zdarzenia opóźnionego.

OB przerwania sprzętowego jest wykonywany wtedy, kiedy wystąpi zdarzenie sprzętowe takie jak zbocza opadające lub narastające na wejściach cyfrowych lub zdarzenia HSC (dotyczące szybkiego licznika). Dozwolony jest jeden OB dla każdego skonfigurowanego zdarzenia sprzętowego. Zdarzenia sprzętowe są włączane w konfiguracji urządzenia (*Device configuration*). Bloki OB są określane w konfiguracji urządzenia lub z poziomu programu za pomocą instrukcji ATTACH. CPU obsługuje kilka sprzętowych przerwań. Ich dokładna liczba zależy od rodzaju CPU i liczby punktów wejściowych.

Zdarzenia błędu czasu i zdarzenie błędów diagnostyki uruchamiane są gdy CPU wykryje błąd. Te zdarzenia mają wyższy priorytet niż inne zdarzenia przerwań i mogą przerwać wykonanie zdarzeń opóźnienia czasowego, przerwań cyklicznych i sprzętowych. Jeden blok OB przerwań może być określony dla zdarzenia błędu czasowego i jeden dla zdarzenia błędu diagnostycznego.

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 przepełnienia kolejek jest dostępnych w kolejnym punkcie „Zdarzenia błędów czasu”.

Każde zdarzenie CPU ma przyznany priorytet. Użytkownik nie może zmienić przypisanego dla OB priorytetu. Ogólnie, zdarzenia są obsługiwane zgodnie z ich priorytetem (najpierw najwyższy priorytet). Zdarzenia mające ten sam priorytet są obsługiwane zgodnie z zasadą „pierwszy się pojawił, pierwszy jest obsłużony”.

Tabela 4.1. OB zdarzeń

| Zdarzenie | Numer OB | Liczba zdarzeń | Zdarzenie inicjujące | Priorytet |
|-------------------|--------------------------------------|--|---|-----------|
| Cykl programu | OB 1, od OB 200 do OB 65 535 | 1 zdarzenie cykl programu Dозволенных wiele OB | <ul style="list-style-type: none"> • Koniec OB rozruchowego. • Koniec ostatniego OB obsługującego cykl programu. | 1 |
| Rozruch | OB 100, od OB 200 do OB 65 535 | 1 zdarzenie rozruchu ^{1, 2} Dозволенных wiele OB | Zmiana trybu ze STOP do RUN | 1 |
| Czasowe | Od OB 200 do OB 65 535 | 4 zdarzenia czasowe ³ 1 OB na zdarzenie | Zaplanowane opóźnienie czasowe | 3 |
| | | | Zaplanowany cykliczny blok OB | 7 |
| Zdarzenia procesu | Od OB 200 do OB 65 535 | Do 50 zdarzeń ⁴ 1 OB na zdarzenie | Zbocza: <ul style="list-style-type: none"> • Zdarzenia zbocza narastającego: 16 maksymalnie. • Zdarzenia zbocza opadającego: 16 maksymalnie. | 5 |
| | | | Dla HSC: <ul style="list-style-type: none"> • CV=PV: 6 maksymalnie. • Zmiana kierunku: 6 maksymalnie. • Zewnętrzny reset: 6 maksymalnie. | 6 |
| Błąd diagnostyki | OB 82 | 1 zdarzenie (tylko jeśli załadowano OB 82) | Błąd transmisji modułowej | 9 |
| Błąd czasu | OB 80 | 1 zdarzenie (tylko jeśli OB 80 został załadowany) ⁵ | <ul style="list-style-type: none"> • Przekroczono maksymalny czas cyklu. • Drugie przerwanie czasowe (cykliczne lub od opóźnienia) rozpoczęło się przed zakończeniem przez CPU pierwszego przerwania. | 26 |

¹ Zdarzenie rozruchowe i zdarzenie cyku programu nigdy nie występują w tym samym czasie, ponieważ zdarzenie rozruchu kończy się zanim rozpoczyna się zdarzenie cyku programu (jest to kontrolowane przez system operacyjny).

² Zdarzenie rozruch może być przerwane wyłącznie przez zdarzenie błąd diagnostyki (skojarzone z OB 82) . Wszystkie inne zdarzenia są ustawiane w kolejkę do późniejszej obsługi po zakończeniu rozruchu.

³ CPU obsługuje maksymalnie do 4 zdarzeń czasowych współdzielonych przez OB opóźnień czasowych i cykliczne OB. Liczba OB opóźnień czasowych i cyklicznych OB nie może przekroczyć 4.

⁴ Można użyć więcej niż 50 zdarzeń procesu, jeśli zostały użyte instrukcje ATTACH oraz DETACH.

⁵ Można skonfigurować CPU tak, aby pozostał on w trybie RUN nawet jeśli maksymalny czas cyklu został przekroczony. Istnieje także możliwość użycia instrukcji RE_TRIGGER do zresetowania czasu cyklu. CPU przejdzie jednak w tryb STOP jeśli maksymalny czas wykonania cyklu zostanie drugi raz przekroczony w jednym cyklu skanu.

Po rozpoczęciu wykonywania OB z priorytetem od 2 do 25, jego przetwarzanie nie może zostać przerwane przez obecność innego zdarzenia, poza OB 80 (przerwanie od błędu czasowego, które ma priorytet 26). Wszystkie inne zdarzenia zostaną zakolejkowanie do późniejszego przetwarzania pozwalając na zakończenie bieżącego OB.

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.

Zdarzenia błędu czasu

Spełnienie jednego z kilku różnych warunków określających błąd czasu generuje w efekcie zdarzenie błędu czasu. Mogą wystąpić następujące błędy czasu:

- Przekroczenie maksymalnego czasu cyklu.
- Nie może wystartować wywołany OB.
- Wystąpienie przepełnienia kolejki.

Warunek przekroczenia maksymalnego czasu cyklu występuje wtedy, kiedy cykl nie zostaje zakończony w wyspecyfikowanym maksymalnym czasie cyklu programu. Więcej na temat informacji związanych z maksymalnym czasem cyklu można uzyskać w książce „Podręcznik systemowy S7-1200”.

Warunek braku możliwości uruchomienia wywoływanego OB występuje wtedy, kiedy OB jest wywoływany przerwaniem cyklicznym lub przerwaniem opóźnienia czasu, a ten wywoływany OB jest już uruchomiony.

Warunek zdarzenia czasu dnia (time of day – TOD), które zostało pominięte lub powtórzone występuje gdy jedno lub więcej przerwań czasowych zostało pominięte ze względu na zmianę ustawień zegara TOD lub przez CPU będące w trybie STOP.

Warunek przepełnienia kolejki występuje wtedy, kiedy przerwania pojawiają się szybciej niż mogą być obsługiwane. Liczba oczekujących (kolejkowanych) zdarzeń jest ograniczona przez zastosowanie różnych kolejek dla zdarzeń każdego typu. Kiedy jakieś zdarzenie wystąpi, gdy jego kolejka jest pełna, to generowane jest zdarzenie błędu czasu.

Wszystkie zdarzenia błędu czasu uruchamiają OB 80 (jeśli istnieje). Jeśli OB 80 nie istnieje w programie użytkownika, to ustawienia konfiguracji urządzenia determinują reakcję CPU na wystąpienie błędu czasu:

- Domyślna konfiguracja dla błędów czasu, takich jak drugie cykliczne przerwanie przed zakończeniem przez CPU wykonania pierwszego, pozostawia CPU w trybie RUN.
- Domyślna konfiguracja dla przekroczenia maksymalnego czasu powoduje przełączenie CPU w tryb STOP.

Dostępna jest instrukcja RE_TRIGR (ponownie rozpoczęcie monitorowania czasu cyklu), która kasuje timer mierzący czas cyklu. Jeśli jednak nastąpią dwa przekroczenia maksymalnego czasu cyklu w jednym cyklu programu, to CPU przejdzie w tryb STOP, niezależnie od istnienia OB 80. Więcej informacji znajduje się w sekcji „Monitorowanie czasu cyklu” w publikacji „Podręcznik systemowy S7-1200”.

OB 80 zawiera informacje rozruchowe pomagające użytkownikowi określić, które zdarzenie i OB wygenerowało błąd czasu. Wewnątrz OB 80 można wstawić program badający te wartości rozruchowe i podejmujący odpowiednią akcję.

Tabela 4.2. Informacje rozruchowe dla OB 80

| Wejście | Typ danych | Opis |
|----------|------------|--|
| fault_id | BYTE | 16#01 – przekroczony maksymalny czas cyklu 16#02 – nie można uruchomić wywoływanego OB 16#07 i 16#09 – wystąpiło przepelnienie kolejki |
| csg_OBnr | OB_ANY | Numer OB wykonywanego podczas wystąpienia błędu |
| csg_prio | UINT | Priorytet OB powodującego błąd |

W nowo utworzonym projekcie nie ma OB 80 obsługującego przerwanie błędu czasu. Jeśli jest to konieczne OB 80 można dodać do projektu poprzez podwójne kliknięcie *Add new block* w menu drzewa *Program blocks*, a następnie wybranie *Organization block*, i na koniec *Time error interrupt*.

Zdarzenia błędów diagnostyki

Urządzenia analogowe (lokalne), urządzenia PROFINET oraz PROFIBUS mają możliwość wykrywania i raportowania błędów diagnostycznych. Wystąpienie lub usunięcie jakiegokolwiek spośród różnych błędów diagnostycznych powodują wystąpienie zdarzenia błędów diagnostyki. Mogą wystąpić następujące błędy diagnostyki:

- brak zasilania,
- przepłynięcie,
- niedomiar,
- przerwa w obwodzie,
- zwarcie.

Wszystkie zdarzenia błędu diagnostyki uruchamiają OB 82 (jeśli istnieje). Jeśli OB 82 nie istnieje, to CPU ignoruje ten błąd. W nowo utworzonym projekcie nie ma OB 82 obsługującego przerwanie błędu diagnostyki. Jeśli jest to konieczne OB 82 można dodać do projektu poprzez podwójne kliknięcie *Add new block* w menu drzewa *Program blocks*, a następnie wybranie *Organization block*, i na koniec *Diagnostic error interrupt*.

Uwaga

Błędy diagnostyczne dla wielokanałowych, lokalnych urządzeń analogowych (I/O, RTD i termopary).

Przerwanie od błędu diagnostycznego w OB 82 może raportować błąd diagnostyczny dla tylko jednego kanału równocześnie.

Jeśli dwa kanały wielokanałowego urządzenia zgłaszą błąd, to drugi błąd uruchamia OB 82 przy następujących warunkach: błąd pierwszego kanału został skasowany, wykonanie OB 82 wywołanego przez pierwszy błąd zostaje dokonane, a drugi błąd nie został skasowany.

OB 82 zawiera informacje rozruchowe pomagające użytkownikowi określić, czy zdarzenie powstało na skutek pojawiienia się bądź skasowania błędu oraz pozwala określić urządzenie i kanał, który zgłasza ten błąd. Wewnątrz OB 82 można wstawić program badający te wartości rozruchowe i podejmujący odpowiednią akcję:

Tabela 4.3. Informacje rozruchowe dla OB 82

| Wejście | Typ danych | Opis |
|-------------|------------|---|
| IState | WORD | Stan IO urządzenia <ul style="list-style-type: none"> Bit 0 = 1 jeśli konfiguracja jest poprawna; i = 0 jeśli konfiguracja nie jest już poprawna. Bit 4 = 1 jeśli obecny jest błąd (jak np. przerwa w obwodzie). (Bit 4 = 0 gdy nie ma błędu). Bit 5 = 1 jeśli konfiguracja nie jest poprawna; i = 0 gdy przywrócono poprawną konfigurację. Bit 6 = 1 jeśli nastąpił błąd dostępu do obszaru I/O. Odniesienie do laddr dla sprzętowego rozpoznania I/O z błędem dostępu. (Bit 6 = 0 gdy nie ma błędu). |
| laddr | HW_ANY | Identyfikacja urządzenia które zgłosiło błąd ¹ |
| channel | UINT | Numer kanału |
| multiterror | BOOL | TRUE jeśli wystąpił więcej niż jeden błąd |

¹ Wejście laddr zawiera sprzętowe rozpoznanie urządzenia lub funkcjonału jednostki, który zwrócił błąd. Sprzętowe rozpoznanie jest przypisane automatycznie w momencie dołączenia komponentu do urządzenia lub sieci i ujawnia się w tablicy Constant znaczników PLC. Nazwa dla sprzętowego identyfikatora jest także przyznawana automatycznie. Wpisy w tablicy Constant w znacznikach PLC nie mogą być zmienione.

Zobacz także

„Połączenie online w celu monitorowania wartości w CPU” (strona 216).

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. CPU jest celowo zaprojektowane w taki sposób, by w przypadku utraty zasilania posiadało dostatecznie dużo czasu na przepisanie do pamięci trwałej ograniczonej liczby wybranych wartości. Po przywróceniu zasilania te zachowane wartości są odtwarzane do oryginalnych pozycji.



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:

- 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 103). Karta programu zapewnia więcej pamięci niż w wewnętrznej pamięci CPU. Funkcja Web server (strona 173) pozwala na ściagnięcie plików dzienników zdarzeń do komputera.

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 4.4. 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 65 535 (16#FFFF), co oznacza 31 grudnia 2168. Wszystkie możliwe wartości tej zmiennej są poprawne. 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 tygodnia (USInt): 1 (niedziela) do 7 (sobota), 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 zawierającą 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 danej 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 zagnieżdzona 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 poredni 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 poredni 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 do 999).
- BCD32 jest reprezentowany na 32 bitach (od -9 999 999 do 9 999 999).

4.4.2. Adresowanie obszarów pamięci

Step 7 ułatwia symboliczne programowanie. Można stworzyć symboliczne nazwy lub „zmienne” dla adresów danych, niezależnie czy zmienne (tagi) są związane z adresami pamięci i punktami I/O, czy też z lokalnymi zmiennymi użytymi w kodzie bloku. Zmienne 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 jest możliwe tylko dla wartości 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 218).

Tabela 4.5. Obszary pamięci

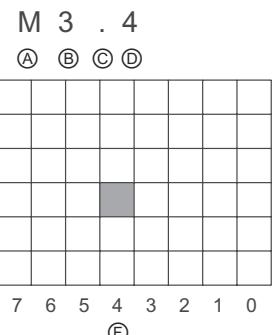
| Obszar pamięci | Opis | Wymuszony | Trwały |
|--|--|-----------|----------------------|
| I obraz procesu – wejście I_.P ¹ (fizyczne wejście) | Skopiowany na początek cyklu programu stan wejść fizycznych. | Nie | Nie |
| | Bezpośredni odczyt wejściowych punktów fizycznych CPU, SB, SM. | Tak | Nie |
| Q obraz procesu – wyjście Q_.P ¹ (fizyczne wyjście) | Stan skopiowany na początek 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).

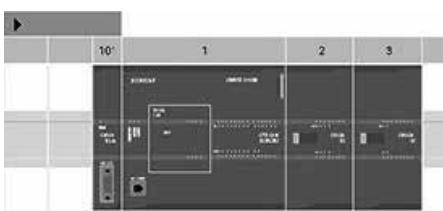


Bezwzględny adres obszaru pamięci:

- 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 ($M = \text{bit obszaru pamięci } M; \text{ cyfra } 3 = \text{bajt } 3$) podany jest, oddzielony kropką („.”), adres bitu (bit 4).

Konfigurowanie I/O w CPU i modułów I/O



| Device overview | | | | | |
|-----------------|------|-----------|-----------|-----------------------|-------|
| Module | Slot | I address | Q address | Type | Order |
| | 103 | | | | |
| | 102 | | | | |
| RS485_1 | 101 | | | OM 1341 (RS485) 6ES7 | |
| ▼ FLC_1 | 1 | | | CPU 1214C DDCO 6ES7 | |
| DI14/DD10 | 1.1 | 0..1 | 0..1 | DI14/DD10 | |
| AI2 | 1.2 | 64..67 | | AI2 | |
| AO1 x 12bit | 1.3 | 80..81 | | AO1 signal board 6ES7 | |
| HSC_1 | 1.16 | 1000.... | | High speed counter | |
| HSC_2 | 1.17 | | | High speed counter | |
| HSC_3 | 1.18 | | | High speed counter | |
| HSC_4 | 1.19 | | | High speed counter | |
| HSC_5 | 1.20 | | | High speed counter | |
| HSC_6 | 1.21 | | | High speed counter | |
| Pulse_1 | 1.32 | | | Pulse generator (P) | |
| Pulse_2 | 1.33 | | | Pulse generator (P) | |
| ► PROFINET L_X1 | | | | PROFINET interface | |
| DI8 x 24VDC | 2 | | | SM 1221 DI8 x 24 6ES7 | |
| | | | 8 | | |

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 na ekranie konfiguracyjnym 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).

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. Dostęp do fragmentów zmiennych adresowanych symbolicznie

Dostęp do zmiennych PLC oraz bloków danych jest możliwy na poziomie bitów, bajtów lub słów w zależności od ich rozmiaru. Dostęp do wszystkich formatów danych jest możliwy w następujący sposób:

- „<PLC tag name>”.xn (dostęp na poziomie bitów)
- „<PLC tag name>”.bn (dostęp na poziomie bajtów)
- „<PLC tag name>”.wn (dostęp na poziomie słowa)
- „<Data block name>”.<tag name>.xn (dostęp na poziomie bitów)
- „<Data block name>”.<tag name>.bn (dostęp na poziomie bajtów)
- „<Data block name>”.<tag name>.wn (dostęp na poziomie słowa)

Do fragmentów danych przechowywanych w formacie podwójnego słowa można sięgać poprzez bity 0–31, bajty 0–3 lub słowa 0–1. Do fragmentów danych przechowywanych w formacie pojedynczego słowa można sięgać poprzez bity 0–15, bajty 0–1 lub słowa 0. Do danych przechowywanych w formacie bajtu można sięgać poprzez bity 0–7 lub bajt 0.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 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

Dopuszczalne typy danych dla dostępu częściowego to Byte, Char, Conn_Any, Date, DIInt, 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, UDInt, UInt, USInt oraz Word. Dane typu Real są także dostępne w taki sposób, nie dotyczy to zmiennych bloków danych typu Real.

Przykłady

W tabeli zmiennych PLC „DW” ma zadeklarowany typ DWORD. Poniższe przykłady pokazują dostęp częściowy na poziomie bitów, bajtów i słów.

| | LAD | FBD | SCL |
|---------------------------------|--|--|--|
| Dostęp bitowy | "DW".x11 — — — | "DW".x11 — — — & | IF "DW".x11 THEN ... END_IF; |
| Dostęp bajtowy | "DW".b2 == Byte "DW".b3 | == Byte "DW".b2 — IN1 "DW".b3 — IN2 | IF "DW".b2 = "DW".b3 THEN ... END_IF; |
| Dostęp na poziomie słowa | AND Word EN — ENO — "DW".w0 — IN1 "DW".w1 — IN2 & | AND Word ... — EN — OUT — "DW".w0 — IN1 "DW".w1 — IN2 & ENO | out := "DW".w0 AND "DW".w1; |

4.4.4. Dostęp do zmiennych z użyciem struktury AT

Mechanizm zastępowania zmiennych AT umożliwia dostęp do zmiennych wcześniejszych zadeklarowanych za pomocą deklaracji innych typów danych. Za pomocą tego mechanizmu można przykładowo adresować pojedyncze bity zmiennej Byte, Word lub DWord za pomocą tablicy Array of Bool.

Deklaracje

Żeby zastąpić wcześniej zadeklarowany typ zmiennej należy po jej zadeklarowaniu utworzyć kolejny parametr i wybrać dla niego typ danych „AT”. Edytor automatycznie tworzy zastępstwo, dla którego należy wybrać typ danych, strukturę lub tablicę, które będą używane zastępco.

Przykład

Niniejszy przykład pokazuje parametry wejściowe standardowego FB. Zmienna bajtu B1 została w przykładzie zastąpiona tablicą Bool:

| | | |
|---------|---------|----------------------|
| ■ 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 4.6. 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] |

Kolejny przykład ilustruje zastąpienie typu DWord typem Struct:



Zmienne zastępujące mogą być adresowane wprost z programu:

| LAD | FBD | SCL |
|--|-----|--|
| #AT[1] | | IF #AT[1] THEN ... END_IF; |
| #DW1_Struct.S1 == Word W#16#000C | | IF (#DW1_Struct.S1 = W#16#000C) THEN ... END_IF; |
| | | out1 := #DW1_Struct.S2; |

Reguły

- Zastępstwo typów zmiennych jest możliwe wyłącznie w standardowych blokach FB i FC.
- Mogą zastąpić parametry wszystkich typów bloków i wszystkich sekcji deklaracji.
- Parametr zastępczy może być używany jak każdy inny parametr bloku.
- Nie jest możliwe tworzenie zastępstw parametrami typu VARIANT.
- Rozmiar parametru zastępującego musi być mniejszy lub równy parametrowi zastępowanemu.

4.5. Wyjścia impulsowe

W CPU bądź w płytce sygnałowej (SB) dostępne są dwa 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 (Q0.0 do Q0.3, oraz Q4.0 do Q4.3) 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 na wejściach!

Jak zostało to opisane w *S7-1200 System Manual*, maksymalna częstotliwość impulsowa dla generatorów wyjścia impulsowego wynosi 100 kHz (cyfrowe wyjścia CPU), 20 kHz (dla standardowej płytki sygnałowej SB) i 200 kHz (dla cyfrowych wyjść szybkiej SB).

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 przekroczeno maksymalnej częstotliwości dla danego urządzenia.

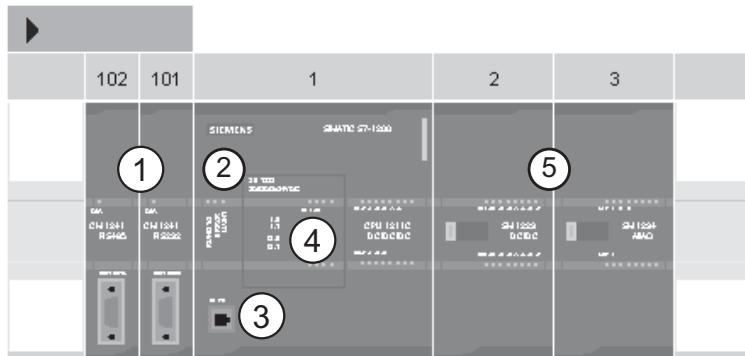
Oba generatory impulsów są mapowane do określonych wyjść cyfrowych, tak jak to pokazano w poniższej tabeli. Można wykorzystać w tym celu zarówno wewnętrzne wyjścia CPU, jak i opcjonalne wyjścia płytki sygnałowej. Numery punktów wyjściowych są pokazane w poniższej tabeli (przy założeniu domyślnej konfiguracji 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 dodatkowej 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.

Tabela 4.6. Przypisanie wyjść dla generatorów impulsów

| Opis | Domyślne przypisanie wyjść | Impuls | Kierunek |
|-------|----------------------------|--------|----------|
| PTO 1 | Wyjścia CPU | Q0.0 | Q0.1 |
| | Wyjścia płytki sygnałowej | Q4.0 | Q4.1 |
| PWM 1 | Wyjścia CPU | Q0.0 | – |
| | Wyjścia płytki sygnałowej | Q4.0 | – |
| PTO 2 | Wyjścia CPU | Q0.2 | Q0.3 |
| | Wyjścia płytki sygnałowej | Q4.2 | Q4.3 |
| PWM 2 | Wyjścia CPU | Q0.2 | – |
| | Wyjścia płytki sygnałowej | Q4.2 | – |

Konfiguracja sterownika

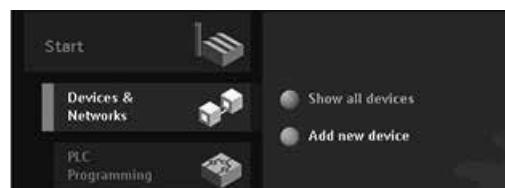
Konfiguracji systemu PLC dokonuje się dodając do projektu CPU i dodatkowe moduły.



- ① Moduł komunikacyjny (CM) lub procesor komunikacyjny CP: do 3 modułów, instalowane na pozycjach 101, 102 i 103.
- ② CPU: na pozycji 1.
- ③ Port ethernetowy CPU.
- ④ Płytkę sygnałową (SB), moduł komunikacyjny (CB) lub moduł baterii (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 oraz CPU 1215C obsługuje 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*.



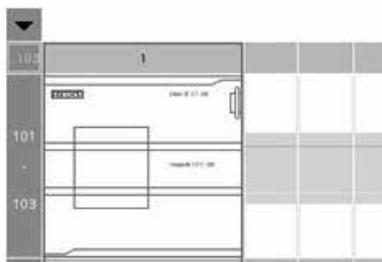
5.1. Rozpoznawanie konfiguracji dla nieokreślonego CPU



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.

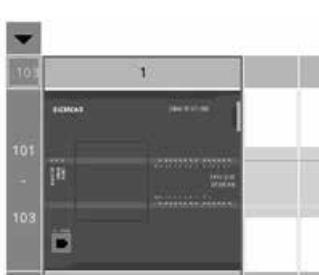
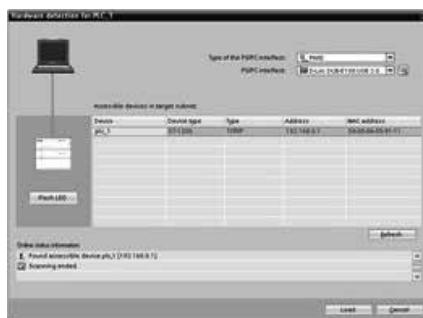
Z poziomu edytora programu należy wybrać komendę *Hardware detection* z menu *Online*.

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

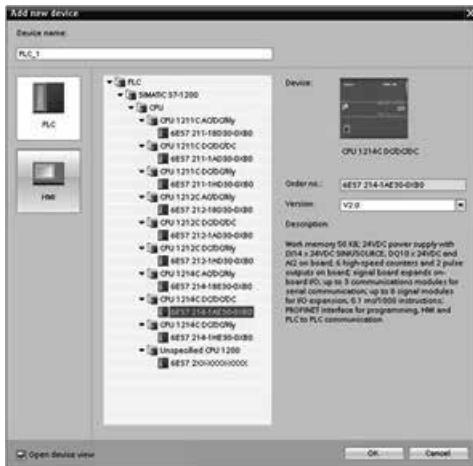


The device is not specified.
→ Please use the [hardware catalog](#) to specify the CPU.
→ or [detect](#) the configuration of the connected device.

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 70).



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.

PLC_1 [CPU 1214C-DIO/DIC]

General

- General
- PROFINET interface
 - General
 - Ethernet addresses
- Advanced
- Time synchronization
- DI/DO/Q/I/O
- General
- Digital inputs
- Digital outputs
- IP addresses
- Hardware identifier

A12

► High speed counters (HSC)

► Pulse generators (PTO/PWM)

Startup

Cycle

Okno *Device view* konfiguracji sprzętowej i wirtualnej listwy.

Wybór CPU w oknie *Device view* powoduje wyświetlenie w oknie inspekcjonijnym właściwości CPU. Użyj tych właściwości do konfiguracji parametrów pracy CPU (strona 70).

Uwaga

CPU nie ma wstępnie ustalonego 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 strownika.

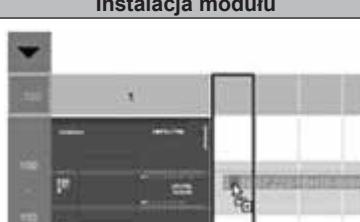
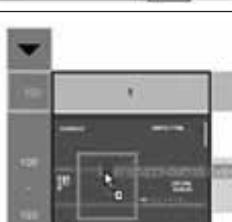
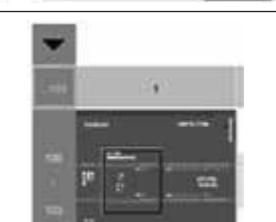
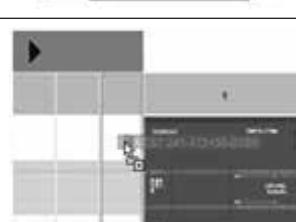
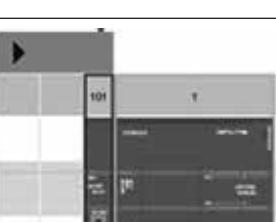
5.3. Dodawanie modułów do systemu

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.
- Moduł baterii (BB1297) zapewnia długotrwałe funkcjonowanie zegara czasu rzeczywistego. Moduł BB1297 jest montowany w przedniej części obudowy 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.

Żeby uwzględnić w konfiguracji programowej zamontowany moduł dodatkowy, należy wybrać go w katalogu sprzętu w pakiecie STEP7 i dwukrotnie w niego kliknąć lub przeciągnąć do podświetlonego widoku gniazda. Żeby moduł dodatkowy funkcjonował prawidłowo konieczne jest dodanie go do konfiguracji sterownika w STEP7 i załadowanie jej do pamięci CPU.

Tabela 5.1. Dodawanie modułu do projektu

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

5.4. 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 inspekcjonego.



- 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.

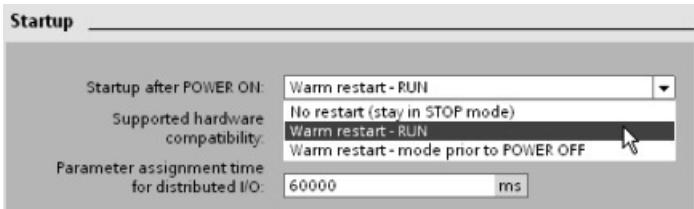
Czynności wykonywane przez CPU przy przejściu z trybu pracy STOP do RUN

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 w trybie STARTUP bieżących stanów wejść fizycznych 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重启 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.

5.4.1. Pamięć systemu i zegara jako standardowa funkcjonalność

Bajty „pamięci systemu” i „pamięci zegara” uaktywniają się we właściwościach CPU. Program użytkownika ma dostęp do pojedynczych bitów tych funkcji.

- Użytkownik może przeznaczyć jeden bajt pamięci M na pamięć systemu. Bajt pamięci systemu zawiera cztery bity, do których ma dostęp program użytkownika:
- Pierwszy cykl: (znacznik „First Scan”) bit jest ustawiony w stan 1 na czas trwania pierwszego cyklu po zakończeniu startu OB. (Po zakończeniu wykonywania pierwszego cyklu bit „First Scan” jest ustawiany w stan 0).
- Zmiana statusu diagnostycznego: (znacznik „DiagStatusUpdate”) bit jest ustawiany w stan 1 na czas trwania jednego cyklu skanowania po wystąpieniu zdarzenia diagnostycznego. Ponieważ CPU nie może ustawiać bitu „Diagnostic Graph Changed” do chwili zakończenia pierwszego cyklu programowego OB, program użytkownika nie może wykryć wystąpienia zmiany statusu diagnostycznego zarówno podczas startu OB jak i podczas trwania pierwszego cyklu programowego OB.

- Bit „Always 1” (wysoki): (znacznik „Always TRUE”) jest zawsze ustawiony na 1.
- Bit „Always 0” (niski): (znacznik „Always False”) 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 zboczy, 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

Nadpisywanie zawartości pamięci systemowej lub bitów pamięci zegara może uszkodzić dane przechowywane w pamięci i – w konsekwencji – spowodować nieprawidłowe działanie programu, co w konsekwencji może doprowadzić do uszkodzenia sprzętu i wywołać zagrożenie dla zdrowia lub życia personelu.

Ponieważ obydwa obszary pamięci (zegarowej i systemowej) nie są chronione w obszarze pamięci M, możliwe jest uszkodzenie przechowywanych w nich danych za pomocą instrukcji użytych w programie.

Należy unikać zapisywania danych do niechronionych obszarów pamięci, co zapewni prawidłową pracę sterownika. W programie sterującym jego pracą należy zawsze zaimplementować możliwość awaryjnego zatrzymania działania maszyny lub wstrzymania procesu obsługiwanej przez sterownik.

| System memory bits | |
|--|--------------------------|
| <input checked="" type="checkbox"/> Enable the use of system memory byte | |
| Address of system memory byte (MBx): | 1 |
| First cycle: | %M1.0 (FirstScan) |
| Diagnostics status changed: | %M1.1 (DiagStatusUpdate) |
| Always 1 (high): | %M1.2 (AlwaysTRUE) |
| Always 0 (low): | %M1.3 (AlwaysFALSE) |

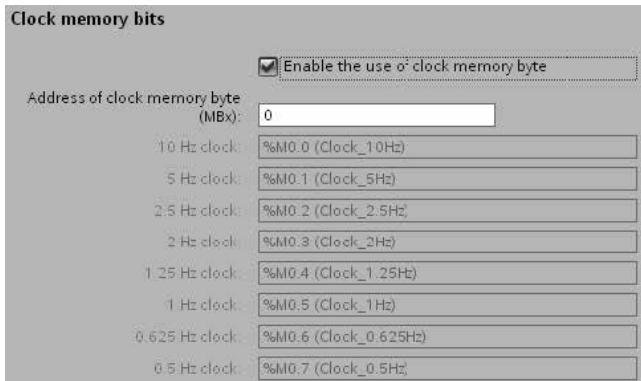
Pamięć systemowa konfiguruje bity w bajcie, które są aktywne (wartość = 1) dla określonych zdarzeń.

Tab. 5.2. Pamięć systemowa

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------------------|---|---|---|----------------------------------|---------------------------------|---|---|
| Zarezerwowane Wartość 0 | | | | Zawsze wyłączone Wartość 0 | Zawsze włączone Wartość 1 | Status diagnostyczny • 1: zmieniony • 0: bez zmiany | Wskaźnik „First Scan” • 1: po startupie • 0: brak |

- *First scan*: Włączony dla rozruchowego OB i dla pierwszego cyklicznego OB po przejściu w tryb RUN, włączając w to ponowne włączenie zasilania.
- „Diagnostic graph changed”.

- Always 1 (wysoki): zawsze jest włączony.
- Always 0 (niski): zawsze jest wyłączony.



Pamięć zegara ma tak skonfigurowany bajt, że pojedyncze bity na przemian włączają się i wyłączają z ustaloną częstotliwością.

Każdy bit zegarowy generuje sygnał prostokątny na odpowiadającym mu biegu pamięci M. Bity te mogą być użyte jako bity sterujące, zwłaszcza jeśli są używane z instrukcjami synchronicznymi, pozwalającymi wyzwalać cykliczne fragmenty programu sterującego pracą sterownika.

Tab. 5.3. Pamięć zegarowa

| Numer bitu | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------------------|-----|-------|---|------|-----|-----|-----|-----|
| Nazwa znacznika | | | | | | | | |
| Okres [s] | 2 | 1,6 | 1 | 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 zegarowa działa asynchronicznie w stosunku do cykli CPU, jej status może zmieniać się kilkakrotnie podczas długich cykli.

Konfiguracja pracy modułów I/O oraz modułów komunikacyjnych

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



Moduły sygnałowe (SM) i płytki sygnałowe (SB)

- Digital I/O: Konfigurowanie indywidualnych wejść, takich jak wykrywanie zbocza i „pulse catch“ (decyduje o pozostaniu w stanie niskim lub wysokim na jeden cykl po chwilowym niskim lub wysokim impulsie). Konfigurowanie wyjść w celu uaktywniania 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.

5.5. 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 identyfikować. Każde urządzenie musi również mieć adres protokołu internetowego (IP).
- Podsieć (subnet) tworzą logicznie pogrupowane urządzenia podłączone do sieci. Maska (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.

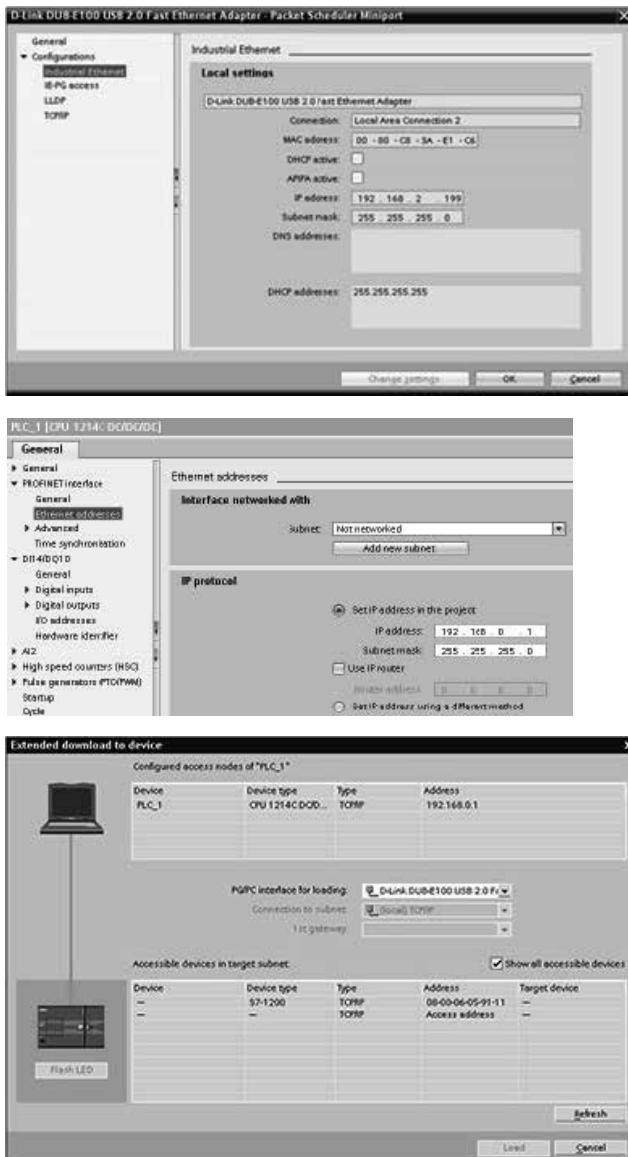
Przed załadowaniem adresu IP do CPU należy upewnić się, czy adres IP dla CPU jest kompatybilny z tym wpisany do urządzenia programującego.

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 podsieci 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

5.6. Ochrona dostępu do CPU i kodu programu

CPU zapewnia 3-poziomową ochronę przed niepozwolonym dostępem do pewnych funkcji. Użytkownik podczas konfigurowania poziomu bezpieczeństwa i haseł CPU, ogranicza funkcje i obszary pamięci, do których można mieć dostęp bez podania hasła.

Podczas wpisywania hasła jest rozróżniana 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.



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 jest związana z wykonywaniem programu ani funkcjami komunikacyjnymi. Wprowadzenie prawidłowego hasła zapewnia dostęp do wszystkich funkcji sterownika.

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 |
|---------------------------------------|--|
| Brak zabezpieczenia | Bez podania hasła dozwolony jest pełen dostęp. |
| Zabezpieczenie przez zapisem | 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). |
| Zabezpieczenie przed zapisem/odczytem | 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). |

5.6.1. 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*.



5.6.2. Zabezpieczenie przed kopiowaniem

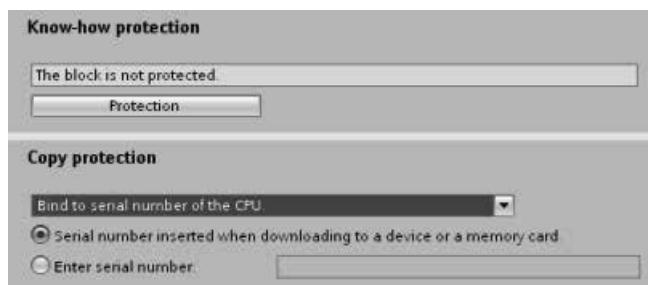
Dodatkowe zabezpieczenia umożliwiają powiązanie programu lub bloku kodu z określonym CPU lub kartą pamięci. Ta możliwość jest szczególnie użyteczna w ochronie własności intelektualnej użytkownika. Po związaniu programu lub bloku z określonym urządzeniem, możliwe jest jego uruchomienie tylko na określonym urządzeniu lub karcie pamięci. Pozwala to na elektroniczną dystrybucję programu lub bloku kodu (przez Internet lub e-mail) lub też poprzez przesłanie karty pamięci.

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.



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

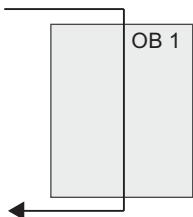
Podczas tworzenia programu użytkownika dla sterownika S7-1200 używa się instrukcji zamieszczanych w blokach kodu (OB., FB lub FC).

Wybór typu struktury programu użytkownika

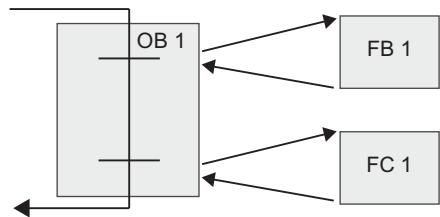
W oparciu o wymagania aplikacji użytkownik podczas tworzenia swojego programu może wybrać dla niego strukturę liniową lub modułową.

- Program liniowy wykonuje wszystkie instrukcje zadania automatyzacji po kolejności – 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 wywoływanie jednego bloku kodu z innego bloku.

Struktura liniowa:

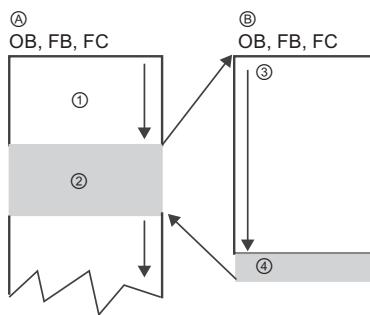


Struktura modularna:



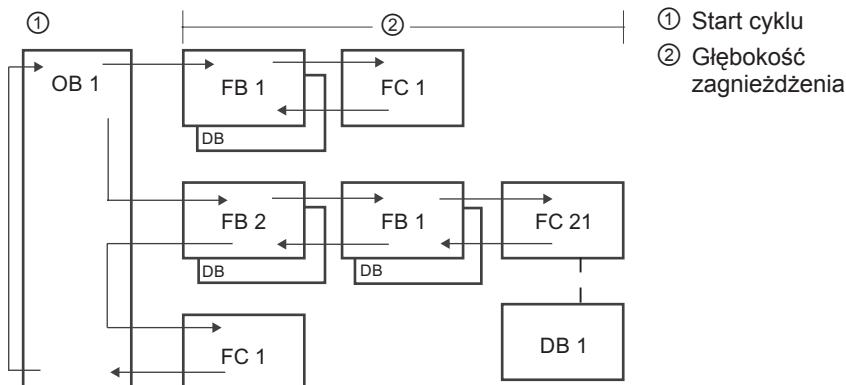
Poprzez utworzenie FB i FC służących do wykonywania ogólnych zadań użytkownik stwarza modularne bloki kodu. Jeżeli inne bloki kodu mogą wywoływać te moduły przewidziane do wielokrotnego użycia, to charakter programu staje się strukturalny. Bloki wywołujące przekazują do bloków wywoływanych parametry 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 przerwia 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.



- A Blok wywołujący (lub blok przerwania)
 B Blok wywoływany FB lub BC (lub blok przerwania OB)
 ① Wykonywanie programu
 ② Instrukcja (lub zdarzenie przerwania) wywołująca inny blok
 ③ Wykonywanie programu
 ④ Zakończenie bloku (powrót do bloku wywołującego)

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.



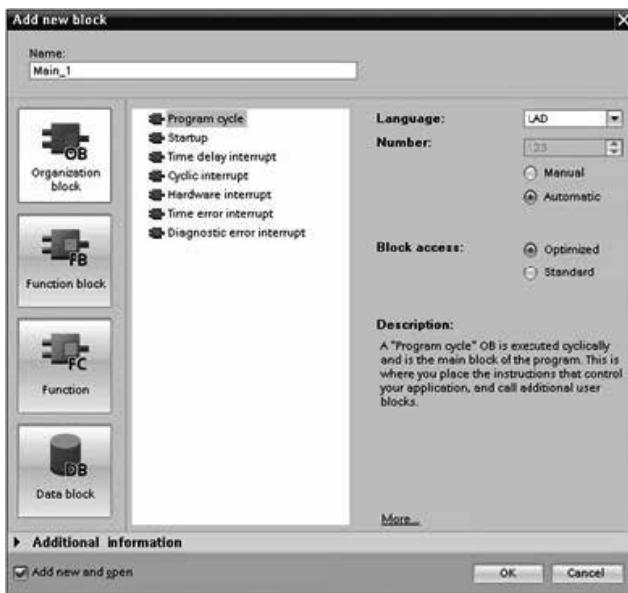
Poprzez stworzenie ogólnego bloku kodu, który może być wykorzystywany w programie użytkownika można uprościć projektowanie i implementację programu użytkownika.

- Do wykonywania 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 rozwiązańach.
- 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 funkcjonalnymi upraszcza i przyspiesza wdrażanie całej aplikacji.

6.1.1. Użycie bloków OB do organizacji programu użytkownika

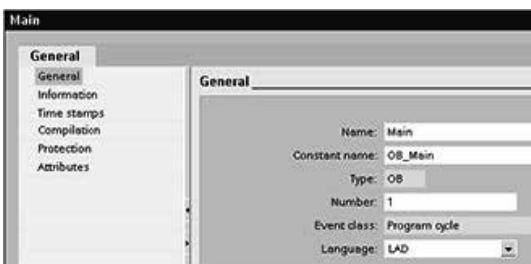
Bloki organizacyjne wprowadzają w programie pewną strukturę. Służą jako interfejs między systemem operacyjnym i programem użytkownika. OB są sterowane zdarzeniami. Zdarzenie, takie jak przerwanie diagnostyczne lub interwału czasowego, powoduje, że CPU wykonuje OB. Niektóre OB mają predefiniowane zdarzenia 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., jak np. w domyślnym cyklicznym OB1.



Użytkownik może utworzyć inne OB przewidziane do wykonywania specyficznych zadań, takich jak zadania rozruchowe, obsługa przerwań i błędów lub wykonywanie określonego kodu programu w stałych odstępach czasu. W celu utworzenia nowego OB w programie użytkownika należy skorzystać z okna dialogowego *Add new block*.

CPU określa kolejność wykonywania przerwań na podstawie priorytetu przypisanego do każdego OB (strona 50).



Użytkownik może zmodyfikować parametry operacyjne dla OB. Na przykład, może skonfigurować parametr czasowy opóźnionego OB. lub cyklicznego OB.

Tworzenie dodatkowego OB w istniejącej klasie OB: użytkownik może utworzyć wiele OB do wykorzystania w swoim programie, nawet w klasie cyklicznej i rozruchowej. W celu utworzenia nowego OB należy skorzystać z okna dialogowego *Add new block*. Należy podać nazwę OB oraz nadać mu numer OB większy od 200.

Jeśli w programie użytkownika zostanie utworzonych wiele cyklicznych OB, to CPU wykonuje wszystkie cykliczne OB w kolejności zgodnej z ich numerami, począwszy od głównego OB cyklu (domyślnie OB 1). Na przykład, po zakończeniu pierwszego cyklicznego OB (OB 1), CPU wykonuje drugi cykliczny OB (np. OB. 200).

6.1.2. FB i FC upraszczają programowanie zadań modularnych

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.

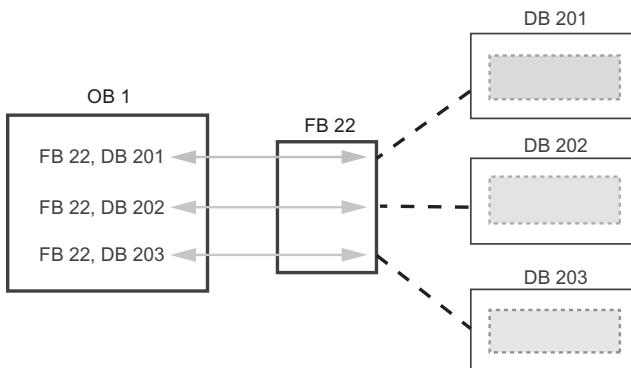
FB są zwykle używane do sterowania działaniem zadań lub urządzeń, które nie kończą swojej pracy w jednym cyklu programu. W celu przechowywania parametrów operacyjnych w taki sposób, by były szybko dostępne w kolejnych cyklach programu, każda FB w programie użytkownika ma jedną lub więcej instancji DB. 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 zo-

staną nadane wartości, to będą wykorzystane wartości pamietane 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 orazinstancję 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 rozpoczęcia 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 szyb-

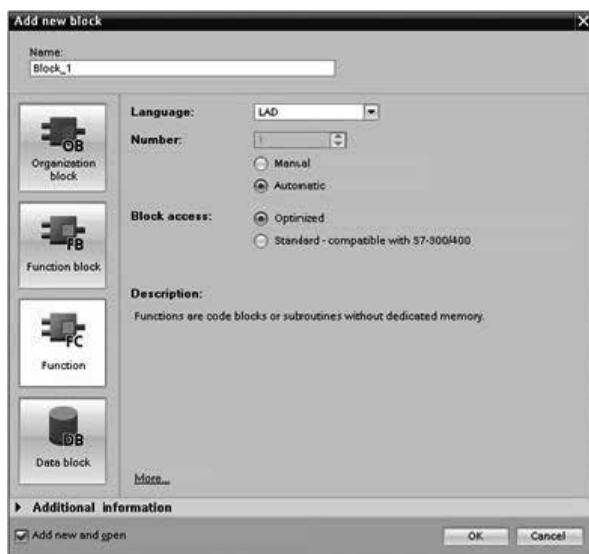
kiego dostępu do danych przechowywanych w samym programie. Użytkownik może nadać DB status „tylko do odczytu”.

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 nowego bloku kodu



1. Otwórz folder *Program blocks*.
2. Naciśnij dwukrotnie *Add new block*.
3. W oknie dialogowym *Add new block* naciśnij typ bloku, który ma zostać dodany. W tym przypadku będzie to ikona *Function* (FC) (dodaje funkcje FC).
4. Określ język programowania dla bloku kodu wybierając *LAD* z listy rozwijalnej.
5. Naciśnij *OK* aby dodać ten blok do projektu.

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

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 stworzyć instrukcję wywołania (*Call instruction*).

Uwaga

OB nie może zostać wywołane przez program użytkownika, ponieważ OB jest sterowane zdarzeniami (strona 50). 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 tekstowym językiem programowania wysokiego poziomu.

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 rozwarte 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 rozwarte 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.

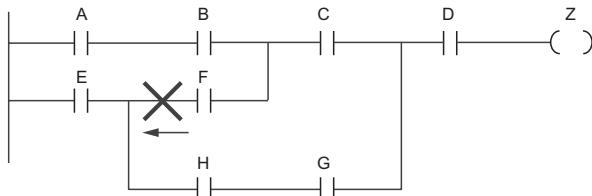
STEP 7 nie ma ograniczenia co do liczby instrukcji (wierszy i kolumn) w sieci programowanej w LAD.

Uwaga

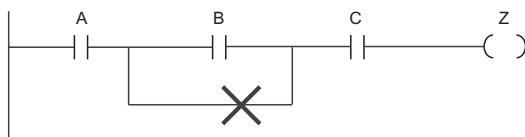
Każda sieć LAD musi kończyć się cewką lub instrukcją.

Podczas tworzenia sieci LAD, należy kierować się następującymi zasadami:

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

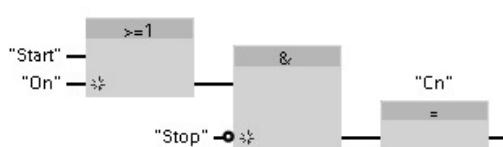


- Nie wolno tworzyć gałęzi, która mogłaby spowodować zwarcie.



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. Przegląd SCL

Język wysokiego poziomu SCL (*Structured Control Language*) bazuje na Pascalu, za jego pomocą można tworzyć programy dla jednostek CPU z rodziny SIMATIC S7. SCL obsługuje bloki strukturalne stosowane w pakiecie STEP 7. Możliwe jest także łączenie opisów bloków programowych przygotowanych w SCL z blokami napisanymi w językach LAD oraz FBD.

Instrukcje SCL wykorzystują standardowe operatory, jak na przykład operator przypisania ($:=$), czy funkcje arytmetyczne (+ dla dodawania, - dla odejmowania, * dla mnożenia oraz / dla dzielenia). W programach SCL można stosować standardowe konstrukcje sterujące jak na przykład IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO oraz RETURN. Podczas pisania programów w SCL można korzystać z projektów napisanych w Pascalu o dowolnej syntaktyce, wiele in-

strukcji SCL (na przykład związanych z obsługą timerów i liczników) ma swoje bezpośrednie odpowiedniki w językach LAD i FBD.

Ponieważ SCL, podobnie jak Pascal, zapewnia mechanizmy warunkowego wykonywania fragmentów programów, pętli i zagnieżdżania wywołań, za pomocą tego języka można tworzyć złożone algorytmy regulacji i sterowania w sposób łatwiejszy niż za pomocą języków LAD lub FBD.

Poniższe przykłady pokazują zastosowania różnych wyrażeń:

| | |
|------------------------------------|---|
| "C" := #A+#B; | przypisuje dwie zmienne lokalne do znacznika, |
| "Data_block_1".Tag := #A; | przypisanie do znacznika bloku danych, |
| IF #A > #B THEN "C" := #A; | zastosowanie warunku IF-THEN, |
| "C" := SQRT (SQR (#A) + SQR (#B)); | przypisanie parametrów do instrukcji SQRT. |

Podobnie do innych języków wysokiego poziomu, także SCL używa standardowych sposobów zapisu podstawowych zadań:

- przypisanie: :=
- funkcje arytmetyczne: +, -, * oraz /
- adresowanie zmiennych globalnych (znaczników): "<nazwa_zmiennej>"
- adresowanie zmiennych lokalnych: #<nazwa_zmiennej>

Operacje arytmetyczne mogą być wykonywane na wielu typach danych numerycznych. Wynikowy typ danych zależy od typu najbardziej znaczącej danej, spośród używanych do realizacji obliczeń. Przykładowo przy mnożeniu liczb typu INT oraz REAL wynik będzie zapisany w formacie REAL.

6.2.4. Edytor programów SCL

Za pomocą języka SCL można opisać dowolny typ bloku (OB, FB, FC). W pakiecie STEP 7 zastosowano edytor SCL wyposażony w:

- sekcję interfejsu umożliwiającą zdefiniowanie parametrów kodu bloku,
- sekcję kodu przeznaczoną na program realizowany przez blok,
- drzewo instrukcji zawierające instrukcje SCL obsługiwane przez CPU.

Kod SCL opisujący działanie bloku jest wprowadzany bezpośrednio do sekcji kodu. W przypadku instrukcji złożonych można przeciągać je bezpośrednio z drzewa instrukcji SCL do sekcji kodu. Możliwe jest także używanie dowolnego edytora tekstowego i importowanie gotowych programów SCL do pakietu STEP 7.

| Interface | | | |
|-----------|-----------------|-----------|---------|
| | Name | Data type | Comment |
| 1 | Input | | |
| 2 | StartStopSwitch | Bool | |
| 3 | Output | | |
| 4 | RunYesNo | Bool | |
| 5 | InOut | | |
| 6 | <Add new> | | |
| 7 | Temp | | |
| 8 | <Add new> | | |
| 9 | Return | | |
| 10 | Ret_Val | Void | |


```

IF... CASE... FOR... TO WHILE...
OF... DO... DO...
  
```

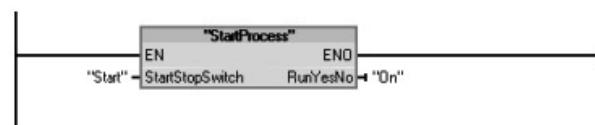
```

1 IF _condition_ THEN
2   // Statement section IF
3 ;
4 END_IF;
  
```

W sekcji kodu użytkownik może zadeklarować następujące typy parametrów:

- Input, Output, InOut oraz Ret_Val: te parametry definiują znaczniki wejścia i wyjścia, a także wartość zwracaną przez blok. Nadane przez użytkownika nazwy znaczników są używane lokalnie podczas wykonywania programu zapisanego w bloku. Nie zaleca się używania globalnych nazw znaczników w tabeli znaczników.
- Static (wyłącznie w FB, ilustracja powyżej dotyczy FC): znaczniki statyczne są używane do przechowywania wyników pośrednich w instancjach bloków danych. Dane statyczne są przechowywane do chwili ich modyfikacji, co może się zdarzyć po kilku cyklach programowych. Nazwy wywoływanych bloków są także przechowywane lokalnie jako dane statyczne.
- Temp: są w nich przechowywane chwilowe informacje o znacznikach, niezbędne podczas wykonywania programu zapisanego w bloku.

Jeżeli blok SCL zostanie wywołany z innego bloku kodu, parametry bloku zostaną pokazane jako wejściowe lub wyjściowe.



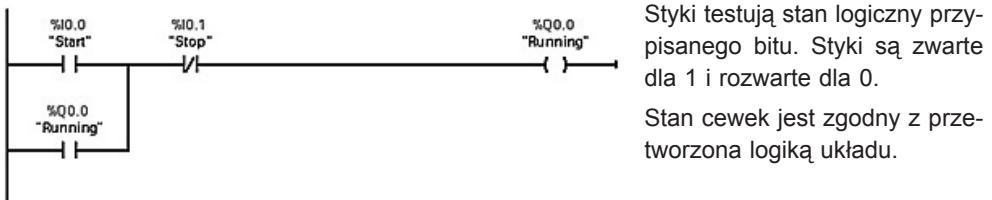
W przykładzie znaczniki „Start” oraz „On” (zapisane w tabeli znaczników projektu) odpowiadają znacznikom „StartStopSwitch” oraz „RunYesNo” w tabeli deklaracji programu SCL.

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.



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

Styki normalnie
rozowane



Styki normalnie
zwarte



Styki normalnie rozowane (*normally open*) są zwarte (ON) wtedy, kiedy wartość przypisana do bitu jest równa 1.

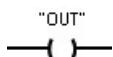
Styki normalnie zwarte (*normally closed*) są zwarte (ON) wtedy, kiedy wartość przypisana do bitu jest równa 0.

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



- 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 wej- ściem 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.

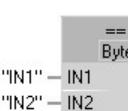
W języku programowania FBD, stykowe sieci LAD są reprezentowane przez operatory logiczne AND (&), OR (≥ 1) i XOR (x), w których użytkownik może wy-specyfikować wartości bitów wejściowych i wyjściowych bloków. Może również wykonać połączenia z innymi blokami logicznymi, co pozwala utworzyć własną logikę 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:  "IN1" == Byte "IN2" | <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 (\geq): Porównanie jest prawdą, jeśli IN1 jest większe od lub równe IN2. • Less than or equal to (\leq): 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:  "IN1" — IN1 "IN2" — IN2 | | |

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 tablicy zmiennych PLC.

Tabela 6.2. Instrukcje MOVE, MOVE_BLK i UMOVE_BLK

| LAD/FBD | SCL | Opis |
|---------|--|--|
| | out1 := in; | 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. |
| | MOVE_BLK(in:=_variant_in, count:=_uint_in, out=>_variant_out); | Przerywalna instrukcja MOVE, która kopiuje blok elementów danych pod nowy adres. |
| | UMOVE_BLK(in:=_variant_in, count:=_uint_in out=>_variant_out); | Nieprzerywalna instrukcja MOVE, która kopiuje blok elementów danych pod nowy adres. |

6.3.3. Instrukcje konwersji

Tabela 6.3. Instrukcje teorii

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

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: Tworzy instrukcję konwersji identyfikując typ danych parametrów: wejściowego (in) i wyjściowego (out). Przykładowo: DWORD_TO_REAL konwertuje wartość DWord do liczby rzeczywistej.

Tabela 6.4. Instrukcje zaokrąglania i skrócenia

| LAD/FBD | SCL | Opis |
|---------|--------------------|---|
| | out := ROUND (in); | 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: <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12 |
| | out := TRUNC(in); | TRUNC zamienia liczbę rzeczywistą (Real lub LReal) na całkowitą. Część ułamkowa liczby rzeczywistej jest skrócona do zera.(IEEE – round to zero). |

Tabela 6.5. Instrukcje wyznaczania najbliższych liczb całkowitych

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

Tabela 6.6. Instrukcje skalowania i normalizacji

| LAD/FBD | SCL | Opis |
|---------|--|---|
| | out := SCALE_X(min,:= _undef_in_ value:= _real_in_, max:= _undef_in_); | SCALE_X skaluje znormalizowany parametr VALUE (gdzie (0.0 <= VALUE <= 1.0)) do typu danej i zakresu wartości wyspecyfikowanych przez parametry MIN i MAX ¹ : OUT = VALUE (MAX – MIN) + MIN. |
| | out := NORM_X(min:= _undef_in_ value:= _undef_in_, max:= _undef_in_); | NORM_X normalizuje parametr VALUE wewnętrz zakresu wartości wyspecyfikowanych przez parametry MIN i MAX ² : OUT = (VALUE – MIN)/(MAX – MIN), gdzie (0.0 <= OUT <= 1.0). |

¹ Ekwiwalent SCL: out := value (max – min) + min.² Ekwiwalent SCL: out := (value – min)/(max – min).

6.3.4. Operacje arytmetyczne z instrukcją Calculate

Tabela 6.7. Instrukcja CALCULATE

| LAD/FBD | SCL | Opis |
|--|--|---|
| <pre> graph TD EN[EN] --> CALCULATE[CALCULATE] IN1[IN1] --> CALCULATE IN2[IN2] --> CALCULATE CALCULATE --> OUT["OUT := <???>"] CALCULATE --> OUT2[OUT] </pre> | <p>Obliczenia wykonuje się za pomocą standardowych wyrażeń arytmetycznych 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. |

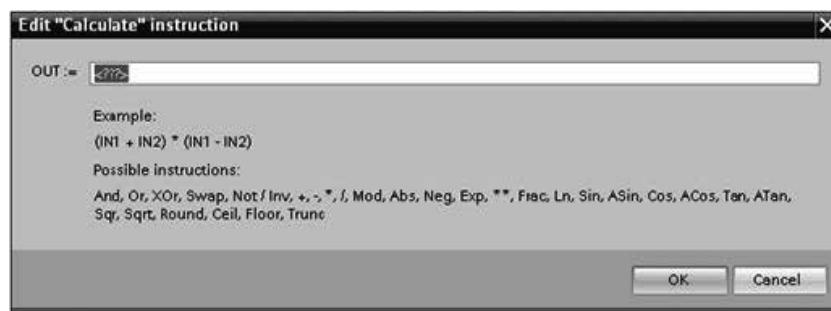
Tabela 6.8. Typy danych dla parametrów

| Parametr | Obsługiwany typ danych ¹ |
|-----------------|---|
| IN1, IN2, ..INn | SIInt, Int, DIInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord |
| OUT | SIInt, Int, DIInt, 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)

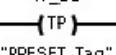
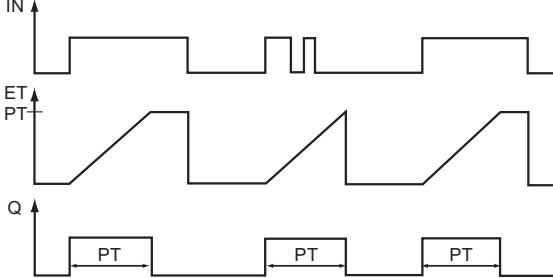
| LAD/FBD | SCL | Przebieg czasowy |
|--|---|--|
|  TP_DB  | <pre>"timer_db".TP(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre> |  |

Tabela 6.10. TON (ON-delay timer)

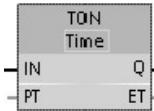
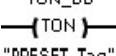
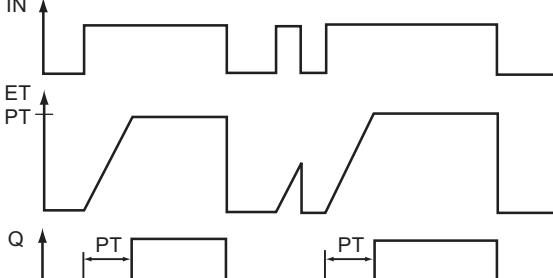
| LAD/FBD | SCL | Przebieg czasowy |
|---|--|--|
|  TON_DB  | <pre>"timer_db".TON(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre> |  |

Tabela 6.11. TOF (OFF-delay timer)

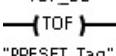
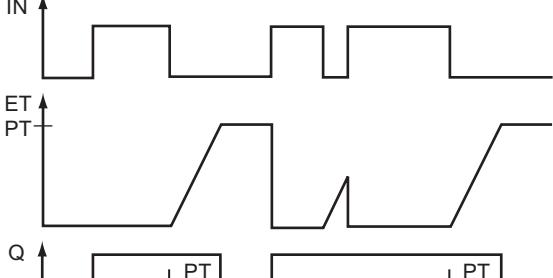
| LAD/FBD | SCL | Przebieg czasowy |
|---|--|--|
|  TOF_DB  | <pre>"timer_db".TOF(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre> |  |

Tabela 6.12. TONR (ON-delay Retentive timer)

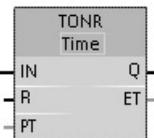
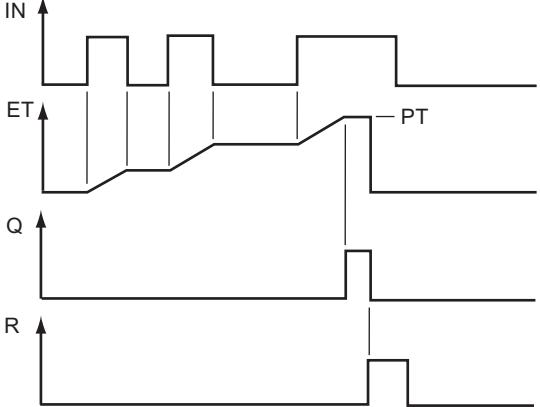
| LAD/FBD | SCL | Przebieg czasowy |
|--|--|--|
|   | <pre>"timer_db".TONR(IN:=_bool_in_, R:=_bool_in_ PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre> |  |

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

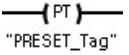
| LAD | Opis |
|---|---|
|  | Instrukcje-PT oraz RT jako cewki lub bloki. Cewki instrukcji mogą zostać wstawione w pozycji środkowej. Stan zasilania wyjścia cewki jest zawsze taki sam jak stan zasilania jej wejścia. <ul style="list-style-type: none"> Kiedy cewka PT jest aktywna, wartość elementu PRESET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

| Parametr | Typ danych | Opis |
|-------------------------------|------------|---|
| Con: IN Coil: Power flow | Bool | TP, TON oraz TONR: Box: 0=wylączanie timera, 1=włączanie timera. Coil: No power flow= wylączanie timera, Power flow=włączanie timera. TOF: Box: 0=wylączanie timera, 1=włączanie timera. Coil: No power flow= wylączanie timera, Power flow=włączanie timera |
| R | Bool | Tylko TONR: 0=brak zerowania, 1=zerowanie zmierzonego czasu oraz ustawienie bitu Q na 0. |
| Box: PT Coil: „PRESET_Tag” | Time | Wprowadzenie ustawianego czasu |
| Box: Q Coil: CBdata.Q | Bool | Timer box: wyprowadzenie Q box na wyjście lub bitu Q w danych DB timera. Timer coil: adresowanie wyłącznie bitu Q w danych DB timera |
| Box: ET Coil: DBdata.ET | Time | Timer box: czas ET (elapsed time) na wyjściu lub wartość ET w danych DB timera. Timer coil: adresowanie wyłącznie bitu Q 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 przygotowywania swojego programu dla sterownika należy brać pod uwagę następujące konsekwencje korzystania z timerów:

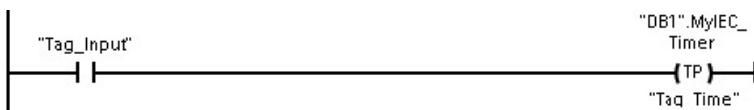
- Może wystąpić jednocześnie wiele modyfikacji stanu timera podczas jednego cyklu pracy sterownika. Stan timera jest modyfikowany zawsze gdy wykonywana jest dotycząca go instrukcja (TP, TON, TOF, TONR), każdorazowo także elementy przypisanej do timera struktury (ELAPSED oraz Q) są wykorzystywane jako parametry początkowe dla zmiany jego stanu. Jest to poważna zaleta gdy w programie są wykorzystywane dane zliczania czasu przez timer. Jeżeli w aplikacji jest konieczne utrzymanie stanu licznika przez czas trwania cyklu programowego, instrukcji obsługujących timer należy użyć przed wszystkimi instrukcjami korzystającymi z danych zliczonych przez timer. Do przekazywania danych należy wykorzystać znaczniki Q i ET zamiast elementów ELASPED i Q struktury DB timera.
- Mogą wystąpić cykle programowe, podczas których stany liczników nie są modyfikowane. Można na przykład uruchomić timer, następnie ponownie wy-

wołać tę funkcję podczas jednego lub więcej cykli programowych. Jeżeli nie są wykonywane kolejne instrukcje odwołujące się do elementów ELAPSED lub Q struktury DB timera, stan timera nie jest modyfikowany. Modyfikacja jego stanu nie nastąpi aż do chwili gdy nie zostanie wykonana instrukcja odwołująca się do timera lub elementów jego struktury.

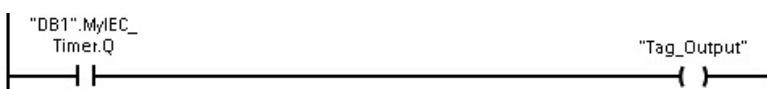
- Użytkownik może przypisać strukturę DB jednego timera do wielu instrukcji go obsługujących. Żeby uniknąć nieoczekiwanych interakcji należy używać pojedynczych instrukcji (TP, TON, TOF, TONR) dla każdej struktury DB.

Samoczynne zerowanie timerów jest przydatne podczas implementacji procesów występujących okresowo. Zazwyczaj taką konfigurację tworzy się dołączając przełącznik normalnie-zwarty sterowany przez bit pola zliczania odpowiedniej instrukcji timera. Ta sieć połączeń jest zazwyczaj ulokowana powyżej jednej lub większej liczby sieci zależnych, które używają bitu timera do wyzwalania procesów. Gdy timer zliczy zadany odcinek czasu, bit licznika jest przełączany w stan ON na czas trwania jednego cyklu wykonywania programu. W kolejnym cyklu styki przełącznika normalnie-zwartego są rozwierane, co powoduje wyzerowanie timera i jego bitu, następny cykl powoduje zwarcie styków przełącznika i restart timera. W takiej konfiguracji timera z samoczynnym zerowaniem nie należy korzystać w elemencie Q struktury DB. Zamiast niego należy użyć znaczników dołączonych do wyjścia „Q” timera. Powodem tego zabiegu jest samoczynne modyfikowanie stanu elementu Q struktury DB w chwili jego odczytu, co może spowodować przedwczesne wyzerowanie timera.

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 108) 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 wywoływanego 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 UDInt).

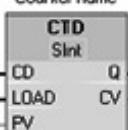
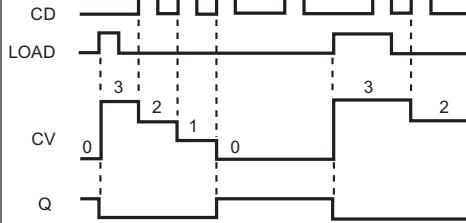
Tabela 6.17. Licznik CTU zliczający w górę

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

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.

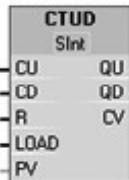
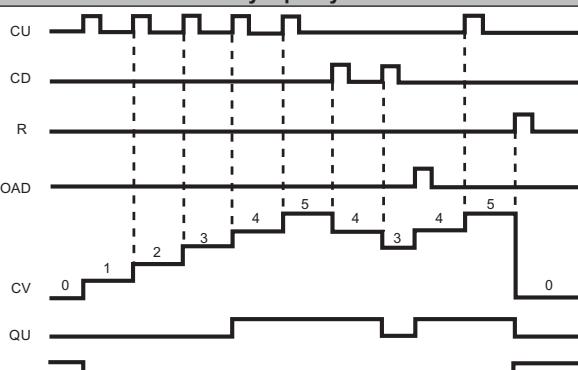
Tabela 6.18. Licznik CTD zliczający w dół

| LAD/FBD | SCL | Cykl pracy |
|--|--|--|
|  "Counter name" CTD Slink CD Q LOAD CV PV | <pre> "ctd_db".CTU(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 góre i w dół

| LAD/FBD | SCL | Cykl pracy |
|--|--|--|
|  "Counter name" CTUD Slink CU QU CD QD R CV LOAD PV | <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> |  |

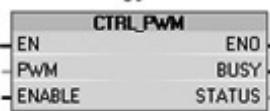
Na powyższym rysunku przedstawiono przebieg czasowy w przypadku licznika CTUD zliczającego liczby całkowite bez znaku (dla PV = 4).

- 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 |
|---|---|---|
|  "CTRL_PWM_DB" | <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 określona 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 pożąanej 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).



- ① Okres cyku
② Szerokość impulsu

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).

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 zamknięcia 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.

Użytkownik może użyć wbudowanego w PLC Web server do zarządzania plikami dziennika. Za pomocą standardowej strony Web można ściągnąć wszystkie zapisane rekordy, ostatnie rekordy, usunąć rekordy lub nawet całe pliki dzienników. Po przetransferowaniu plików dziennika do PC użytkownika można przeanalizować dane za pomocą standardowego arkusza kalkulacyjnego, takiego jak np. MS Excel.

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

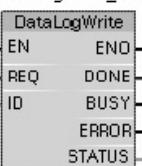
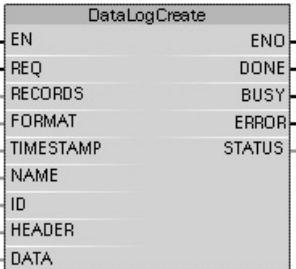
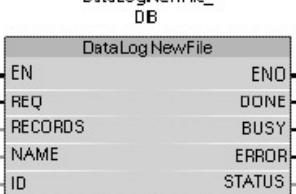
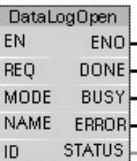
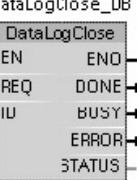
| LAD/FBD | SCL | Opis |
|---|--|--|
| DataLogWrite_DB  | <pre>"DataLogWrite_DB" req:= _bool_in_, done=> _bool_out_, busy=> _bool_out_, error=> _bool_out_, status=> _word_out_, ID:= _dword_inout_;</pre> | <p>DataLogWrite zapisuje rekordy danych w określonym dzienniku. Istniejący docelowy dziennik musi 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 |
|---|---|--|
| DataLogCreate_DB  | <pre>"DataLogCreate_DB" req:= _bool_in_, records:= _udint_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> |
| DataLogNewFile_DB  | <pre>"DataLogNewFile_DB" req:= _bool_in_, records:= _udint_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 |
|--|--|--|
|  DataLogOpen_DB <ul style="list-style-type: none"> - EN ENO - - REQ DONE - - MODE BUSY - - NAME ERROR - - ID STATUS - | <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> | 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. |
|  DataLogClose_DB <ul style="list-style-type: none"> - EN ENO - - REQ DONE - - IU BUSY - - ERROR - - STATUS - | <pre>„DataLogClose_DB”(req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre> | 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. Przejście sterownika w tryb STOP zamyka wszystkie pliki dziennika. |

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 ustawać 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 218).

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“).

6.5.2. Odsyłacze pokazujące wykorzystanie elementów

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

Uwaga

Nie trzeba zamykać edytorów, aby zobaczyć informacje o odsyłaczach.

Można sortować wpisy w odsyłaczach. Lista odsyłaczy 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 odsyłaczy 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 odsyłaczy

| Kolumna | Opis |
|----------|--|
| Object | Nazwa obiektu, który używa obiektów niższych poziomów lub jest używany przez obiekty niższego poziomu. |
| Quantity | Liczba użyć |
| 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 wywoływany 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 wywołyany 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 komplikację bloków kodu.
- W przypadku niepowodzenia komplikacji, 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 zmienne dla urządzenia HMI.

Tabela 6.25. Instrukcje LED

| LAD/FBD | SCL | Opis |
|---|---|--|
| <pre> LED - EN ENO - LADDR Ret_Val - LED </pre> | <pre> ret_val := #LED(laddr:=_hw_io_in_, LED:=_uint_in_); </pre> | RET_VAL zwraca następujące stany diod LED z CPU: <ul style="list-style-type: none"> • RUN/STOP: zielona lub czerwona • Error(błąd): czerwona • MAINT (podtrzymywanie): 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> GET_DIAG - EN ENO - MODE - LADDR RET_VAL - DIAG - DETAIL </pre> | <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> DeviceStates - EN ENO - LADDR - MODE - STATE </pre> | <pre> ret_val := DeviceStates(laddr:=hw_io_in_, mode:=_uint_in_, state:=_variant_inout_); </pre> | Instrukcja DeviceStates odczytuje informacje o stanie urządzeń PROFINET IO lub PROFIBUS DP. |
| <pre> ModuleStates - EN ENO - LADDR - MODE - STATE </pre> | <pre> ret_val := ModuleStates(laddr:=_hw_io_in_, mode:=_uint_in_, state:=_variant_inout); </pre> | Instrukcja ModuleStates odczytuje informacje o stanie urządzeń PROFINET IO. |

6.6. Szybkie liczniki (HSC)

Szybkie liczniki (*high-speed counters* – HSC) używane są do zliczania zdarzeń występujących częściej niż wynosi częstość wywoływanego 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 wywoływanego 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 |
|---|--|--|
|  "Counter_name" | <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 wyzwala 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 instrukcji CTRL_HSC.

Ustawienia:

- 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 HSC

Szybkie liczniki (*high-speed counters*) używane są do zliczania zdarzeń występujących częściej niż wynosi częstość wywoływanego cyklu programu. Jeśli zliczane zdarzenia występują wolniej niż częstość wykonywania OB, należy użyć liczników CTU, CTD, lub CTUD. Jeśli zdarzenia występują częściej niż wynosi częstość wywoływanego cyklu programu wykonywanego przez CPU, należy użyć HSC. Niektóre parametry szybkiego licznika mogą być modyfikowane pod kontrolą programu.

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 zostaje załadowany do programu użytkownika wraz z pierwszą z ustalonych konfiguracji, a wyjścia są aktywowane przez program użytkownika po okresie, w którym bieżąca wartość zliczania jest mniejsza niż bieżąca wartość nastawiona. Program użytkownika konfiguruje HSC do obsługi przerwania, gdy wartość zliczania jest równa z wartością referencyjną (CV = RV), gdy nastąpi kasowanie wartości lub gdy nastąpi zmiana kierunku.

Każdy warunek CV = RV wyzwala przerwanie, a program użytkownika wczytuje nową wartość referencyjną i ustawia następny stan wyjść wewnętrz przerwania OB. Kiedy nastąpi skasowanie zdarzenia przerwania, program użytkownika załadowuje pierwszą wartość referencyjną i ustawia pierwszy stan wyjść w zerującym przerwaniu OB, po czym cykl zostanie powtórzony.

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.

Tabela 6.28. Maksymalne częstotliwości [kHz]

| HSC | | Jednofazowe | Dwufazowe oraz AB kwadraturowe |
|------|------------|-------------|--------------------------------|
| HSC1 | CPU | 100 kHz | 80 kHz |
| | Szybkie SB | 200 kHz | 160 kHz |
| | SB | 30 kHz | 20 kHz |
| HSC2 | CPU | 100 kHz | 80 kHz |
| | Szybkie SB | 200 kHz | 160 kHz |
| | SB | 30 kHz | 20 kHz |
| HSC3 | CPU | 100 kHz | 80 kHz |
| HSC4 | CPU | 30 kHz | 20 kHz |
| HSC5 | CPU | 30 kHz | 20 kHz |
| | Szybkie SB | 200 kHz | 160 kHz |
| | SB | 30 kHz | 20 kHz |
| HSC6 | CPU | 30 kHz | 20 kHz |
| | Szybkie SB | 200 kHz | 160 kHz |
| | SB | 30 kHz | 20 kHz |

Wybór funkcjonalności HSC

Wszystkie HSC działają w ten sam sposób dla tego samego trybu zliczania. 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 określona 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. Przykładowo, jeśli HSC1 jest w trybie, który używa wbudowanych wejść, ale nie używa zewnętrznego zerowania (I0.3), wtedy I0.3 może zostać użyty do przerwania od zboiska lub dla HSC2.

Tabela 6.29. 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 |
| | | | Faza Z | Zliczanie |

Adresy wejściowe HSC

Uwaga

Punkty cyfrowych I/O używanych przez HSC są przypisywane podczas konfiguracji urządzenia. Kiedy adresy tych punktów zostaną przypisane, ich wartości nie mogą zostać wymuszone w tabeli *watch table*.

Podczas konfiguracji CPU istnieje możliwość włączenia i skonfigurowania każdego HSC. CPU automatycznie przypisuje adresy wejść dla każdego HSC zgodnie ze swoją konfiguracją.

(Niektóre HSC zezwalają na wybór pomiędzy użyciem wbudowanych wejść CPU lub wejść SB).

Uwaga

Jak pokazano w poniżej 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.

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

Dla przykładu, poniższa tabela przedstawia przypisanie wejść HSC dla wbudowanych I/O jednostki CPU 1212C oraz SB. (Jeśli SB posiada tylko 2 wejścia, dostępne są tylko wejścia 4.0 i 4.1).

- Dla jednofazowych: C to wejście zegarowe, [d] d jest opcjonalnym wejściem kierunkowym, a [R] jest opcjonalnym zewnętrznym wejściem kasującym (kasowanie jest możliwe tylko w trybie zliczania).
- Dla dwufazowych: Cu to zegar zliczający w góre, CD jest wejściem zegara zliczającego w dół, a [R] jest opcjonalnym zewnętrznym wejściem kasującym. (kasowanie jest możliwe tylko w trybie zliczania).
- Dla AB kwadraturowych: A jest wejściem zegara A, B jest wejściem zegara B, a [R] jest opcjonalnym zewnętrznym wejściem kasującym. (kasowanie jest możliwe tylko w trybie zliczania).

Tabela 6.30. Przypisanie wejść HSC dla CPU 1212C

| HSC | | Wejścia CPU (0.x) | | | | | | | Wejścia SB (4.x) ³ | | | | |
|--------------------|-----------------|-------------------|-----|----|-----|----|-----|----|-------------------------------|-----|-----|-----|-----|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |
| HSC 1 ¹ | 1-fazowe | C | [d] | | [R] | | | | | C | [d] | | [R] |
| | 2-fazowe | CU | CD | | [R] | | | | | CU | CD | | [R] |
| | AB-kwadraturowe | A | B | | [R] | | | | | A | B | | [R] |
| HSC 2 ¹ | 1-fazowe | | [R] | C | [d] | | | | | [R] | C | [d] | |
| | 2-fazowe | | [R] | CU | CD | | | | | [R] | CU | CD | |
| | AB-kwadraturowe | | [R] | A | B | | | | | [R] | A | B | |
| HSC 3 | 1-fazowe | | | | | C | [d] | | [R] | | | | |
| | 2-fazowe | | | | | CU | CD | | [R] | | | | |
| | AB-kwadraturowe | | | | | A | B | | [R] | | | | |
| HSC 4 | 1-fazowe | | | | | | [R] | C | [d] | | | | |
| | 2-fazowe | | | | | | [R] | CU | CD | | | | |
| | AB-kwadraturowe | | | | | | [R] | A | B | | | | |
| HSC 5 ² | 1-fazowe | | | | | | | | | C | [d] | | [R] |
| | 2-fazowe | | | | | | | | | CU | CD | | [R] |
| | AB-kwadraturowe | | | | | | | | | A | B | | [R] |
| HSC 6 ² | 1-fazowe | | | | | | | | | [R] | C | [d] | |
| | 2-fazowe | | | | | | | | | [R] | CU | CD | |
| | AB-kwadraturowe | | | | | | | | | [R] | A | B | |

¹ HSC 1 oraz HSC 2 mogą zostać skonfigurowane jako wbudowane wejścia lub jako wejścia SB.

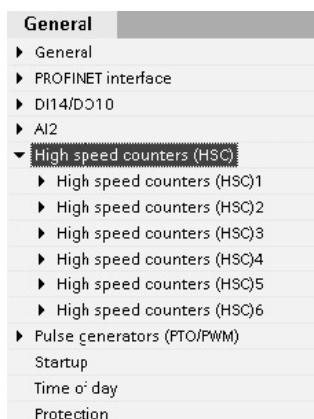
² HSC 5 oraz HSC 6 są dostępne tylko dla SB. HSC 6 jest dostępne tylko dla płytki SB z 4 wejściami.

³ Płytki SB z tylko 2 wejściami obsługują wejścia 4.0 i 4.1.

Dostęp do wartości bieżącej HSC

Jeżeli włączysz generator impulsów w trybie PTO, to powiązany z nim HSC zostanie do danego PTO przypisany. HSC1 jest przypisany do PTO1, a HSC2 jest przypisany do PTO2. Przypisany HSC jest zarezerwowany przez kanał PTO, I standartowe wyjście HSC jest wyłączone. Wartość HSC jest używana tylko do wewnętrznych potrzeb. Nie jest możliwe monitorowanie wartości bieżącej (na przykład w ID1000), w przypadku pojawienia się impulsów.

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 kontrolę działania HSC.

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

Enable

Enable this high speed counter for use

Uwaga

Aktywowanie szybkich liczników i dołączenie do nich linii wejściowych powoduje automatyczne skonfigurowanie filtrów wejściowych na 800 ns. Każde wejście ma przypisaną pojedynczą nastawę ważną dla wszystkich zastosowań: wejść procesowych, przerwań, zliczania impulsów oraz wejść szybkich liczników (HSC).

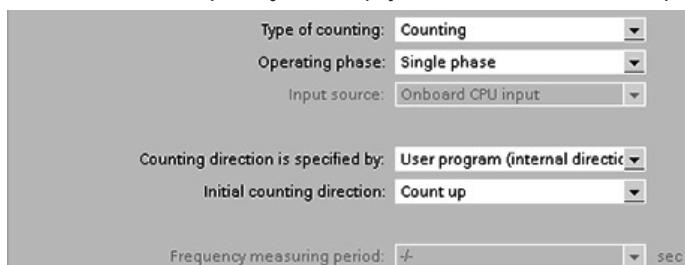
⚠️ OSTRZEŻENIE

W przypadku zmiany czasu reakcji filtra wejściowego na nową wartość konieczne jest podanie na to wejście stanu „0” o czasie trwania powyżej 20 ms, co spowoduje aktywowanie nowego ustawienia. Jeżeli to nie nastąpi, impulsy o czasie trwania krótszym niż 20 ms mogą nie być wykrywane lub zliczanie.

Zmiana czas reakcji filtra wejściowego bez jego aktywowania może spowodować poważne urazy lub uszkodzenia ciała personelu obsługującego urządzenia, a także uszkodzenia lub zniszczenie sprzętu.

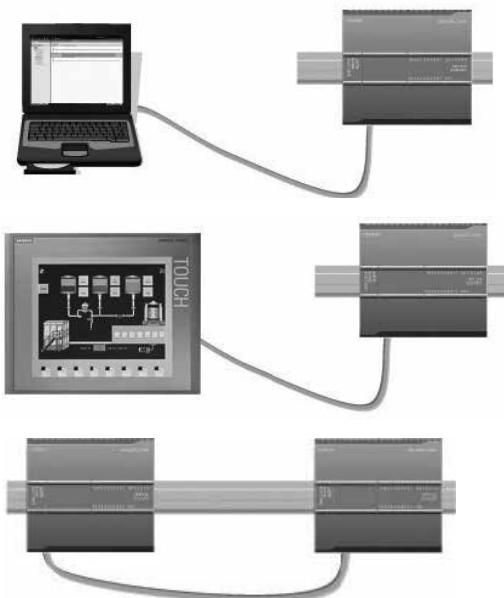
Żeby uzyskać pewność, że filtr wejściowy jest poprawnie skonfigurowany, zalecane jest wyłączenie i ponowne włączenie zasilania sterownika.

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



Więcej informacji o konfiguracji HSC znajduje się w sekcji poświęconej konfiguracji CPU (strona 70).

Komunikacja pomiędzy urządzeniami – standard PROFINET



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 obydwoma 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.

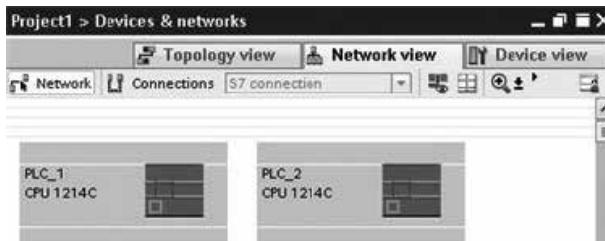
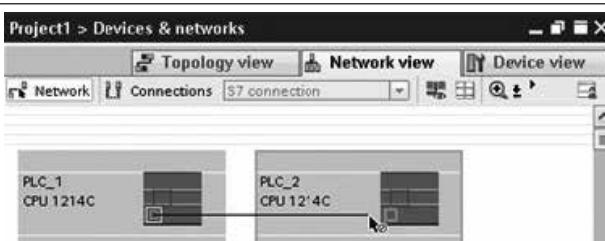
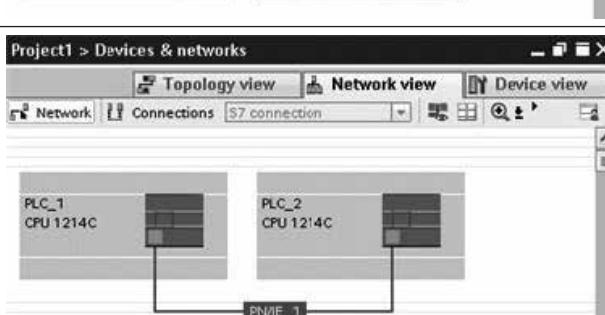
Dodając moduł PROFIBUS CM, CPU może pracować jako urządzenie typu master lub slave w sieci PROFIBUS.

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

7.2. Opcje komunikacji

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

PROFINET

PROFINET jest wykorzystywany do wymiany danych przez sterownik z innymi urządzeniami dołączonymi do sieci Ethernet:

- Moduły CPU zapewniają obsługę:
 - w wersji V3.0 możliwy jest dostęp poprzez PROFINET do 16 urządzeń I/O z 256 submodułami. PROFIBUS dopuszcza zastosowanie 3 niezależnych urządzeń master PROFIBUS DP, obsługujących (każde) do 32 urządzeń I/O z 512 submodułami.
 - w wersji V2.2 możliwy jest dostęp poprzez PROFINET do 8 urządzeń I/O ze 128 submodułami (w przypadku gdy skonfigurowano 8 lub mniej modułów slave lub submodułów). PROFIBUS dopuszcza zastosowanie do 16 urządzeń I/O z maksymalnie 256 submodułami na każdego мастera.

- Komunikację S7.
- Protokół UDP.
- Protokół ISO przez TCP (RFC 1006).
- Protokół TCP.

Kontroler PROFINET RT IO

Jeżeli urządzenie I/O wyposażono w interfejs PROFINET RT, moduł CPU może z nim współpracować po zastosowaniu w sieci PN. Dodatkowe informacje są dostępne w dokumentacji PROFIBUS i PROFINET (<http://www.us.profinet.com>):

- w wersji V3.0 sterowniki S7-1200 mogą komunikować się z maksymalnie 16 urządzeniami PN,
- w wersji V2.2 sterowniki S7-1200 mogą komunikować się z maksymalnie 8 urządzeniami PN.

PROFIBUS

PROFIBUS jest wykorzystywany do wymiany danych przez sterownik z innymi urządzeniami dołączonymi do sieci PROFIBUS:

- we współpracy z modułem CM1242-5 moduł CPU współpracuje z siecią PROFIBUS DP jako slave,
- we współpracy z modułem CM1243-5 moduł CPU współpracuje z siecią PROFIBUS DP jako master class 1,
- w wersji V3.0 urządzenia slave i master PROFIBUS DP, AS-i oraz PROFINET są logicznie rozdzielone,
- w wersji V2.2 moduł CPU zapewnia następujące wsparcie dla PROFINET i PROFIBUS:
 - obsługę do 16 urządzeń z 256 submodułami, w tym maksymalnie 16 urządzeń slave PROFIBUS DP z 256 submodułami (w przypadku gdy nie skonfigurowano urządzeń i submodułów PROFINET IO).

Uwaga

W wersji V2.2 łączna liczba 16 urządzeń PROFINET i PROFIBUS zawiera:

- moduł slave PROFIBUS DP dołączony do master PROFIBUS DP (CM1243-5),
- dowolne urządzenie slave PROFIBUS DP dołączone do CPU,
- dowolne urządzenie PROFINET dołączone do CPU przez port PROFINET.

Przykładowo: konfiguracja CPU z trzema modułami CM (jeden CM1243-5 – master oraz dwa CM1242-5 – slave) zmniejsza maksymalną dostępną liczbę możliwych do dołączenia modułów slave dostępnych dla mastera PROFIBUS DP (CM1243-5) do 14.

- AS-i – moduł S7-1200 CM 1243-2 (master AS-i) umożliwia dołączenie do sieci AS-i modułu CPU z serii S7-1200.
- Komunikacja S7 CPU – S7 CPU.

Komunikacja teleserwisowa

Komunikacja teleserwisowa stacji roboczej z zainstalowanym STEP7, poprzez modem GPRS lub Internet, jest możliwa z S7-1200 dzięki modułowi CP1242-7. Połączenie jest realizowane przez dołączony do Internetu serwer pośredniczący.

7.3. Liczba asynchronicznych połączeń komunikacyjnych

CPU pozwala na obsługę następującej liczby równoczesnych, asynchronicznych połączeń komunikacyjnych:

- 8 połączeń dla Otwartej Komunikacji Użytkownika (Open User Communications), aktywnych lub pasywnych: TSEND_C, TRCV_C, TCON, TDISCON, TSEND, i TRCV.
- 3 połączenia CPU –CPU dla serwera danych GET/PUT.
- 8 połączeń CPU –CPU dla klienta danych GET/PUT.

Uwaga

CPU jednostek S7-1200, S7-300 i S7-400 używają instrukcji GET oraz PUT do komunikacji CPU-CPU. CPU jednostki S7-200 używa instrukcji ETHx_XFER do komunikacji CPU-CPU.

- Podłączenie HMI: CPU wyposażono w interfejsy HMI umożliwiające dołączenie do trzech urządzeń HMI (użytkownik może podłączyć do 2 paneli typu SIMATIC Comfort). Maksymalna liczba paneli HMI zależy od ich typu. Przykładowo użytkownik może podłączyć do CPU do trzech paneli SIMATIC Basic lub do dwóch SIMATIC Comfort z jednym dodatkowym SIMATIC Basic.
- Połączenia PG: CPU umożliwia połączenie z 1 urządzeniem programującym (PG).
- Połączenia http: CPU umożliwia połączenia dla serwera WWW.

7.4 Instrukcje do obsługi PROFINET i PROFIBUS

Instrukcje do PROFINET

Instrukcje TSEND_C i TRCV_C ułatwiają komunikację 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 jedna 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 TUSEND 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 opisany 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 slave DP.
- W przypadku korzystania z interfejsu PROFIBUS do odczytu aktualnych danych diagnostycznych z urządzenia slave DP należy korzystać z instrukcji DPNRM_DG (dane odczytywane w formacie EN50170 Vol. 2).

7.5. PROFINET

7.5.1. Otwarta komunikacja użytkownika

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 65535.

Jeśli instrukcje TRCV_C lub TRCV nie są często wywoływanie 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. Połączenia i porty ID dla instrukcji PROFINET

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 port ID dla danego połączenia.

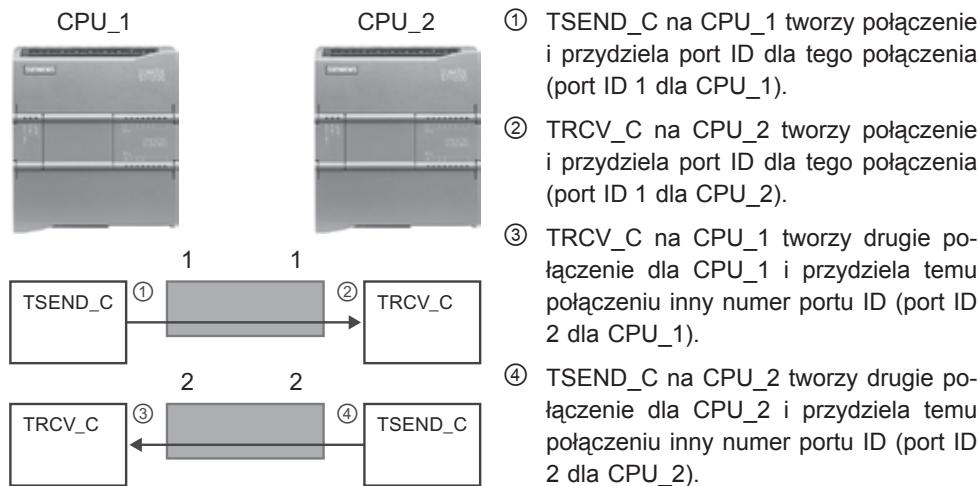
- ID połączenia musi być unikalne dla CPU. Każde utworzone połączenie musi mieć różny port ID oraz DB.
- Lokalne CPU i partnerskie CPU mogą używać tego samego portu ID dla tego samego połączenia, ale numery portów ID mogą się też nie pokrywać. Numer portu ID są odpowiednie tylko dla instrukcji PROFINET użytych w programie użytkownika dla poszczególnego CPU.
- Można używać każdego numeru portu ID dla CPU. Jednak konfigurowanie portów ID sekwencyjnie od numeru 1 pozwala na szybkie wyśledzenie liczby połączeń użytych dla poszczególnego CPU.

Uwaga

Każda instrukcja TSEND_C, TRCV_C oraz TCON użyta w programie użytkownika tworzy nowe połączenie. Ważne jest użycie poprawnego portu ID dla każdego połączenia.

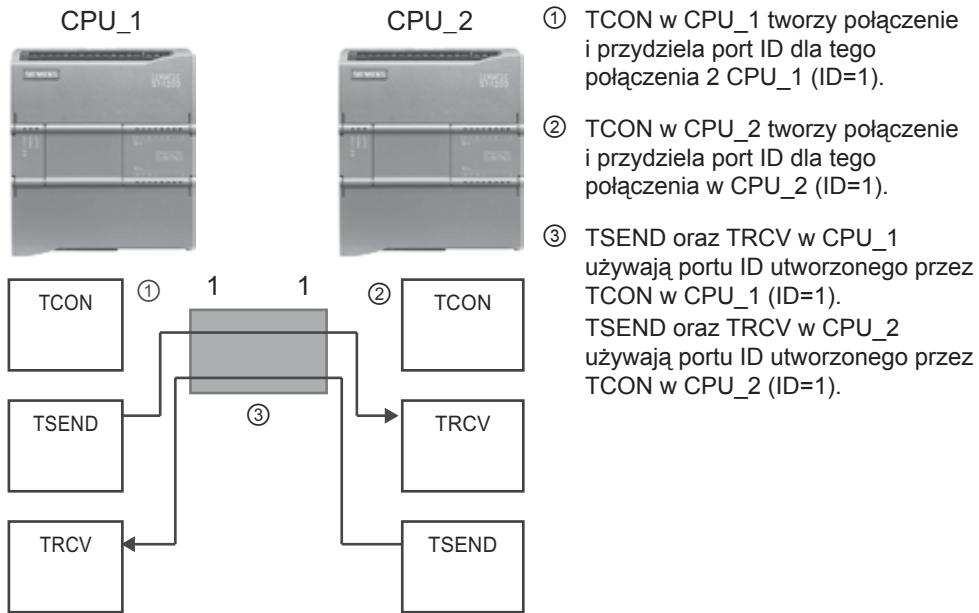
Poniższy przykład pokazuje komunikacje pomiędzy dwoma CPU, które wykorzystuje 2 oddzielne połączenia do wysyłania i odbierania danych.

- Instrukcja TSEND_C dla CPU_1 łączy się z instrukcją TRCV_V w CPU_2 przez pierwsze połączenie (port ID1 dla CPU_1 i CPU_2).
- Instrukcja TRCV_V dla CPU_1 łączy się z instrukcją TSEND_C w CPU_2 przez drugie połączenie (port ID2 dla CPU_1 i CPU_2).

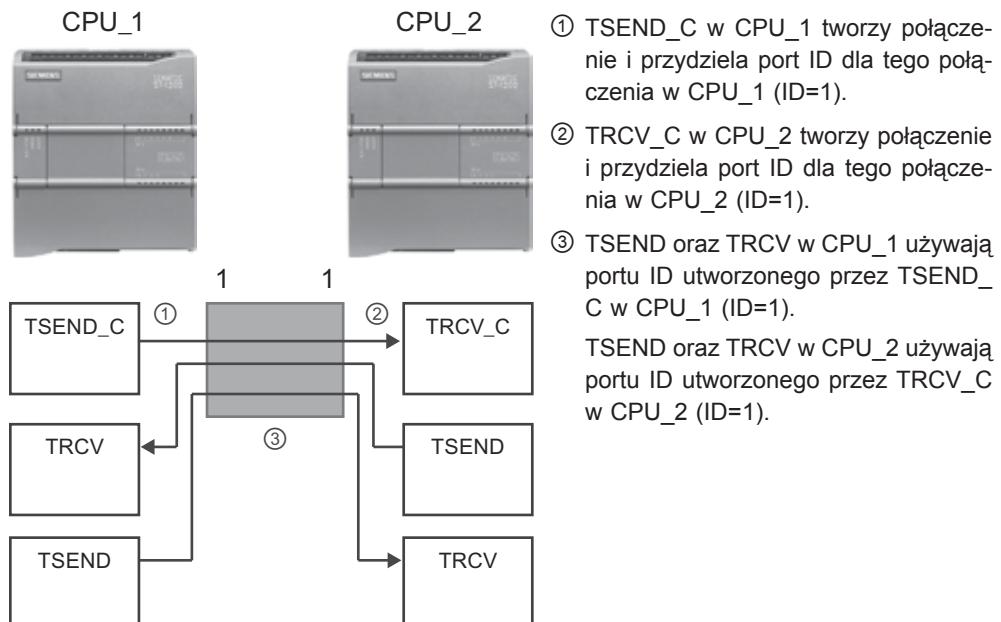


Poniższy przykład pokazuje komunikację pomiędzy dwoma CPU, które wykorzystują 1 połączenie do wysyłania i odbierania danych.

- Każde CPU używa instrukcji TCON do konfiguracji połączenia pomiędzy sobą.
- Instrukcja TSEND na CPU_1 łączy się z instrukcją TRCV w CPU_2 używając połączenia ID (port ID1), które zostało skonfigurowane poprzez instrukcję TCON na CPU_1. Instrukcja TRCV w CPU_2 łączy się z instrukcją TSEND na CPU_1 używając tego samego portu ID (port ID1), który został skonfigurowany za pomocą instrukcji TCON w CPU_2.
- Instrukcja TSEND w CPU_2 łączy się z instrukcją TRCV w CPU_1 używając połączenia ID (port ID1), które zostało skonfigurowane poprzez instrukcję TCON w CPU_2. Instrukcja TRCV w CPU_1 łączy się z instrukcją TSEND w CPU_2 używając tego samego portu ID (port ID1), który został skonfigurowany za pomocą instrukcji TCON w CPU_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 samodzielnie nowego połączenia, dlatego też muszą używać DB i portu ID, które zostały stworzone przez instrukcje TSEND_C, TRCV_C lub TCON.



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.

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 | USIn 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 | 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: rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = nieistotne |
| 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 | Zastrzezone: W#16#0000 |

7.5.2. Konfigurowanie ścieżki połączenia

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 |
|---|--|
| 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”. Rysunek przedstawia pole <i>Connection properties</i> dla zakładki konfiguracyjnej dla połączenia ISO-on-TCP. |  <p>The screenshot shows the 'ISO-on-TCP' configuration dialog. It has tabs for 'General' and 'Configuration'. Under 'Configuration', the 'Connection parameter' tab is selected. The 'General' section shows 'End point' fields for 'Local' (PLC_1) and 'Partner' (PLC_2). Below these are fields for 'Interface', 'Subnet', 'Address', 'Connection type' (set to 'ISO-on-TCP'), 'Connection ID' (set to 1), and 'Connection data' (set to 'PLC_1_Receive_DB' and 'PLC_2_Send_DB'). There are two radio buttons: 'Establish active connection' (selected) and 'Establish passive connection'. The 'Address details' section shows 'Local TSAP' (10.00) and 'Partner TSAP' (11.00), along with 'TSAP ID' values.</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 „nieokreślonym” CPU.

Użytkownik nadal musi wstawić instrukcję 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. | |

Więcej informacji oraz lista instrukcji komunikacyjnych znajduje się w dziale „Protokoły” (strona 119) w sekcji „PROFINET” lub w „Tworzenie połączenia S7” (strona 139) 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. Nazwa przypisana do partnera (odbierającego) CPU.</i> |
| | Rack/slot <i>Tylko dla komunikacji S7. Położenie gniazda i szyny.</i> |
| | Connection resource <i>Tylko dla komunikacji S7. Komponent TSAP używany do konfiguracji komunikacji S7 z CPU urządzeń S7-300 lub S7-400.</i> |
| | 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 określić zdalny adres IP i numer portu partnerskiego (pasywnego) połączenia CPU.

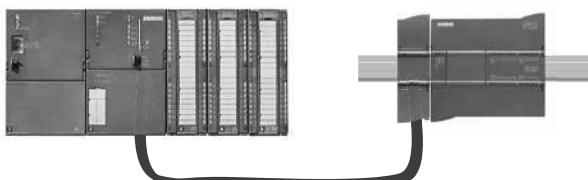
W sekcji *Address Details* okna dialogowego *Connection Parameters*, użytkownik definiuje porty, które będą wykorzystywane. Port połączenia w CPU jest wprowadzany w polu *Local Port*. Port przypisany do połączenia partnerskiego CPU jest wprowadzony w polu *Partner Port*.

7.6. PROFIBUS

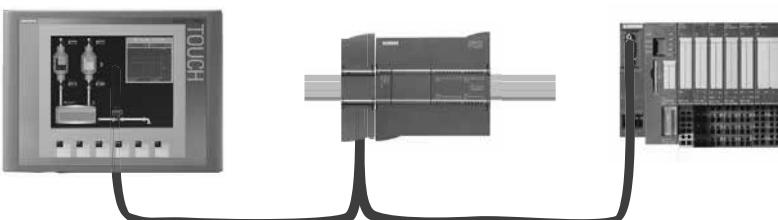
System PROFIBUS używa nadzędnej magistrali do odpytania urządzeń typu slave rozproszonych w sposób wielopunktowy na magistrali szeregowej RS485. Urządzeniem PROFIBUS slave jest każde urządzenie peryferyjne (przetwornik I/O, zawór, napęd lub inne urządzenie pomiarowe), które przetwarza informację i wysyła 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 wysyłać 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 konfigurujące, 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.

S7-1200 jest połączony z siecią PROFIBUS jako DP slave za pomocą modułu komunikacyjnego CM 1242-5. Moduł CM 1242-5 (DP slave) może być partnerem komunikacyjnym z urządzeniami DP V0/V1 master. 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 usytuowa-

nnych systemów DP master i jako urządzenie typu master dla niżej usytuowanych systemów DP master.

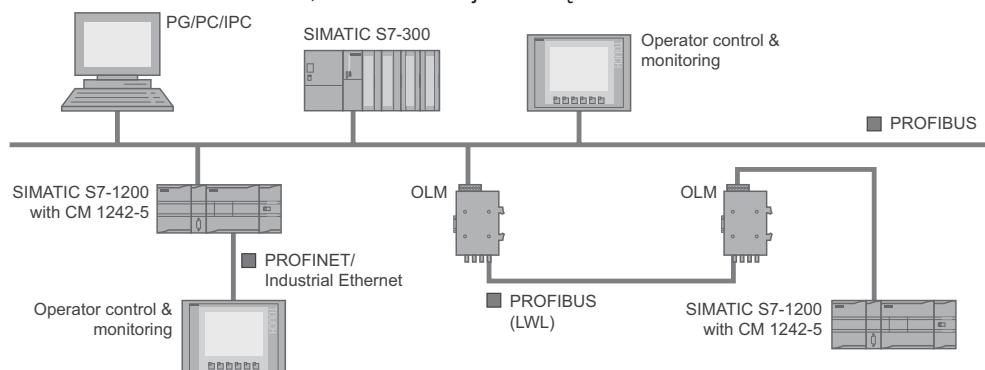


W wersji V3.0 można skonfigurować maksymalnie trzy moduły CM dla każdej stacji PROFIBUS, przy czym możliwe są dowolne kombinacje modułów CM slave i master. Każdy moduł master może obsługiwać do 32 moduły slave.

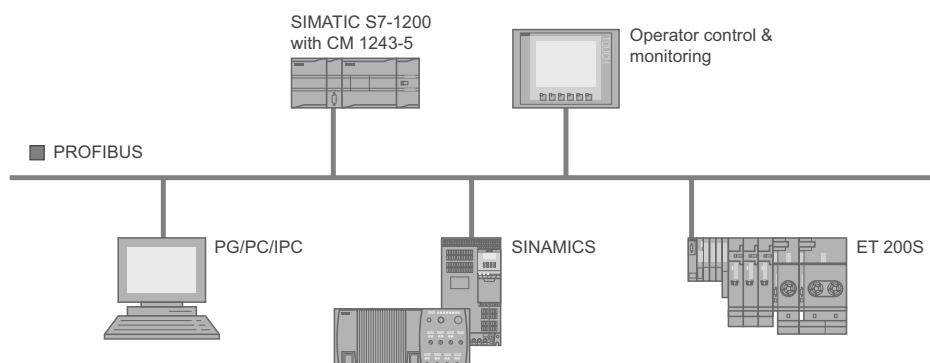
W wersji V2.2 można skonfigurować maksymalnie trzy moduły CM dla każdej stacji PROFIBUS, przy czym dopuszczalny jest jedynie jeden moduł master. Może on obsługiwać do 16 urządzeń slave.

7.6.1. Przykłady konfiguracji PROFIBUS

Poniżej znajdują się przykłady, w których CM 1242-5 jest użyty jako urządzenie PROFIBUS slave, a CM 1243-5 jako urządzenie PROFIBUS master.



Rys. 7.1. Przykład konfiguracji z modułem CM1242-5 użyтыm jako PROFIBUS slave



Rys. 7.2. Przykład konfiguracji z modułem CM1243-5 użyтыm jako PROFIBUS master

Połączenie S7-1200 z PROFIBUS DP

S7-1200 może zostać włączony w system magistrali PROFIBUS za pomocą następujących modułów komunikacyjnych:

- CM 1242-5
Pracuje jako DP slave
- CM 1243-5
Pracuje jako DP master klasy 1

Jeśli CM 1242-5 i CM 1243-5 są obecne równocześnie w systemie, to S7-1200 przeprowadza obydwa zadania równocześnie:

- Slave dla systemu DP master wyższego poziomu
oraz
- Master dla systemu DP niższego poziomu

Protokół magistrali

Moduły komunikacyjne PROFIBUS CM używają protokołu PROFIBUS DP-V1 zgodnego ze standardami:

- IEC 61158 (2004), typ 3
- IEC 61784-1 (2007), CPF-3/1

Partnerzy komunikacyjni sterownika S7-1200 w standardzie PROFIBUS

Dwa moduły komunikacyjne PROFIBUS CM pozwalają sterownikowi S7-1200 na wymianę danych z następującymi partnerami komunikacyjnymi:

- CM 1242-5
CM 1242-5 (DP slave) może być partnerem komunikacyjnym następujących urządzeń typu DP V0/V1 master:
 - SIMATIC S7-1200, S7-300, S7-400
 - Moduły DP master i rozproszone IO SIMATIC ET200
 - Stacje SIMATIC PC
 - SIMATIC NET IE/PB Link
 - Programowalne sterowniki różnych producentów
- CM 1243-5
CM 1242-5 (DP master) może być partnerem komunikacyjnym następujących urządzeń typu DP V0/V1 slave:
 - Napędy i silowniki różnych producentów
 - Czujniki różnych producentów
 - S7-1200 CPU z procesorem komunikacyjnym CP 1242-5 dla PROFIBUS DP
 - S7-200 CPU z modułem EM 277 dla PROFIBUS DP

- Moduły DP slave i rozproszone IO SIMATIC ET200
- CP 342-5
- S7-300/400 CPU z interfejsem PROFIBUS
- S7-300/400 CPU z procesorem komunikacyjnym PROFIBUS CP

Typy komunikacji z DP-V1

Następujące typy komunikacji z D-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 przerwał.
 - Instrukcje „RDREC“ oraz „WRREC“ służą do transferu konfiguracji i danych diagnostycznych.

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.

Mögliche Partner für die Kommunikation sind: HMI und PC-Komponenten mit WinCC Flexible oder SCADA, welche die S7-Kommunikation unterstützen.

Konfiguracja i zamiana modułów

Konfiguracja modułów, sieci i połączeń następuje w wersji V11.0 oprogramowania STEP 7.

Użytkownik może skonfigurować moduł w systemie opartym o pliki GSD. Dla modułu komunikacyjnego CM 1242-5 (DP slave) dostępny jest plik GSD dołączany

na płycie CD dostarczanej wraz z modułem. Dostępny jest on też na stronie Internetowej Siemens Automation w dziale wsparcia klienta.

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

Użytkownik może skonfigurować maksymalnie do trzech modułów PROFIBUS CM w jednej stacji, z których tylko jeden może pełnić rolę DP master.

Połączenie elektryczne

- Zasilanie
 - CM 1242-5 jest zasilane przez magistralę tylnią stacji SIMATIC.
 - CM 1243-5 wyposażono w oddzielne przyłącze zasilania 24 VDC
- PROFIBUS

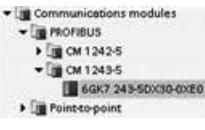
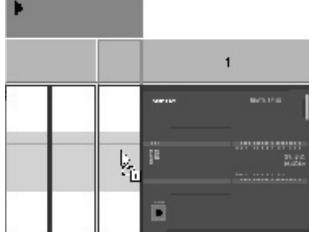
Wtyczka interfejsu RS-485 to 9-stykowa żeńska wtyczka typu D-sub.

Użytkownik może połączyć się z siecią optyczną PROFIBUS poprzez Terminal Optical Bus OBT lub moduł Optical Link OLM.

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

W portalu „Devices and networks” w celu dodania do CPU modułów PROFIBUS 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 |
|---------------------|-------------------|
| | |

7.6.3. 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.

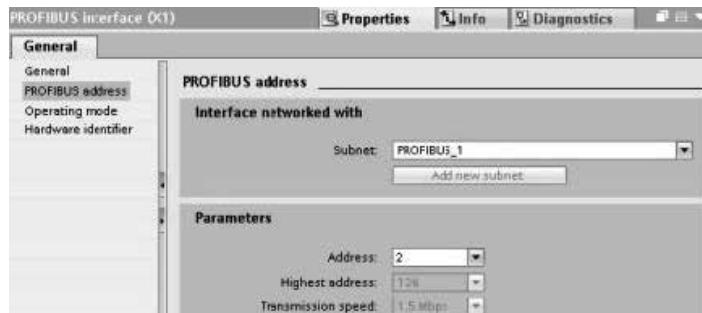


Tabela 7.10. Parametry adresów PROFIBUS

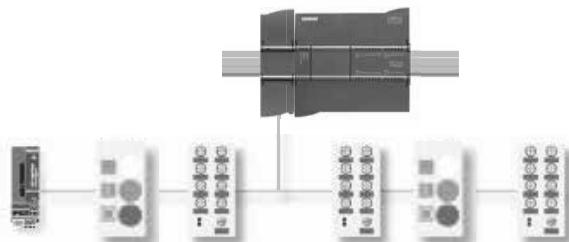
| Parametr | Opis | |
|----------|---|---|
| Podsieć | Należy nazwać podsieć, do której podłączone jest urządzenie. Należy nacisnąć przycisk <i>Add new subnet</i> , aby stworzyć nową podsieć. Jej stan domyślny to „Not connected” (niepołączona). Możliwe są dwa typy połączenia: <ul style="list-style-type: 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. Interfejs AS-i

Moduł S7-1200 CM 1243-2 umożliwia dołączenie sterownika do sieci AS-i jako urządzenia master.

System sieciowy AS-i jest przeznaczony do stosowania w sieciach przemysłowych, zapewniając komunikację na ich najniższym strukturalnie poziomie. Moduł CM 1243-2 spełnia rolę urządzenia master, w każdej sieci AS-i dopuszczalne jest tylko jedno urządzenie tego rodzaju. Urządzenia wchodzące w skład systemu regulacji i sterowania (czujniki, silowniki, styczniki itp.) mogą komunikować się ze sterownikiem za pomocą pojedynczego kabla sieciowego AS-i. Moduł CM 1243-2 obsługuje całą komunikację pomiędzy urządzeniami zewnętrznymi i sterownikiem, lokując się w przestrzeni I/O jednostki CPU. Dostęp do elementów peryferyjnych może być binarny lub analogowy, w zależności od rodzaju współpracującego urządzenia slave. Moduł CM 1243-2 obsługuje m.in. cyfrowe wejścia i wyjścia, z których w danej chwili obsługiwane może być jedno.

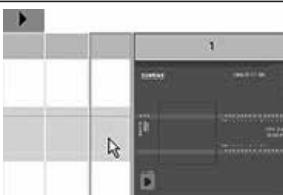
Na rysunku poniżej sterownik S7-1200 jest masterem sieci AS-i, współpracując poprzez sieć AS-i z panelem operatorskim oraz modułami cyfrowych i analogowych wejść i wyjść.



7.7.1. Dodawanie modułów CM 1243-2 AS-i master oraz AS-i slave

Moduły CM 1243-2, spełniające rolę AS-i master, są dołączane do CPU z lewej jej strony, można użyć maksymalnie trzech modułów AS-i master (CM1243-2). Dodanie moduły CM 1243-2 do projektowanego systemu wymaga wybrania w pakiecie programowym modułu w katalogu i przeciągnięcie go do podświetlonego slotu.

Tabela 7.11. Dodawanie modułu AS-i CM 1243-2 do konfigurowanego systemu

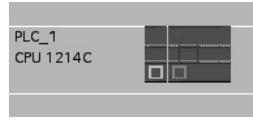
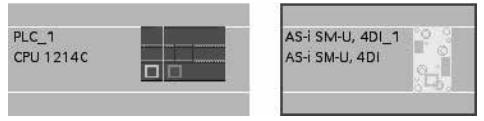
| Moduł | Sposób |
|-----------------------|--|
| CM 1243-2 AS-i Master |    |

W podobny sposób można dodać do projektu systemu urządzenia slave sieci AS-i. Przykładowo w katalogu „I/O module, compact, digital, input” rozwijamy gałęzie drzewa katalogowego o nazwach:

- Field devics,
- AS-interface slaves.

Następnie wybieramy z listy element oznaczony symbolem „3RG9 001-0AA00” (AD-i SM-U, 4DI) i dodajemy slave'a jak widać to na rysunkach poniżej.

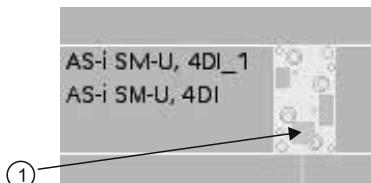
Tabela 7.12. Dodawanie urządzenia slave do sieci AS-i

| Wstawianie urządzenia AS-i slave | Wynik operacji |
|---|--|
|  |  |

7.7.2. Przypisywanie adresu do urządzeń slave AS-i

Konfiguracja interfejsu AS-i

Konfiguracja parametrów interfejsu AS-i wymaga kliknięcia w żółty prostokąt na widoku urządzenia slave AS-i, następnie wybranie zakładki *Properties* w wyświetlnym oknie.



① Interfejs AS-i

Przypisanie adresu AS-i

W sieci AS-i każde urządzenie ma przypisany indywidualny adres, mieszczący się w zakresie od 0 do 31. Domyślnie adres 0 jest przeznaczony dla nowych urządzeń slave. Adresy urządzeń slave mogą się więc mieścić w zakresie od 1 (A lub B) do 31 (A lub B), co pozwala zastosować w systemie do 62 takich urządzeń. Każdy adres z zakresu 1–31 może zostać przypisany dowolnie każdemu urządzeniu, nie ma konieczności utrzymywania kolejności adresowania i zaczynania jej od pierwszego wolnego adresu.

Wprowadź żądaną adres urządzenia slave AS-i.

**Tabela 7.13.** Parametry interfejsu AS-i

| Parametr | Opis |
|----------|---|
| Sieć | Nazwa sieci, do której dołączono urządzenie |
| Adresy | Przypisany przez użytkownika adres w sieci AS-i w zakresie od 1 (A lub B) do 31 (A lub B), co daje łącznie 62 różne urządzenia. |

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.

- Dostęp do danych w zdalnym CPU: jednostka CPU S7-1200 do adresowania zdalnych jednostek CPU (z rodzin S7-200/300/400/1200) może używać w polu danych polecenia ADDR_x wyłącznie adresów bezwzględnych.
- Dostęp do danych przechowywanych w standardowych DB: jednostka CPU S7-1200 do adresowania zmiennych DB lokowanych w standardowych DB zdalnych jednostek CPU S7 może używać w polu danych polecenia ADDR_x wyłącznie adresów bezwzględnych.
- Dostęp do danych przechowywanych w zoptymalizowanych DB: jednostka CPU S7-1200 nie może adresować zoptymalizowanych DB w zdalnych jednostkach CPU S7-1200.
- Dostęp do danych przechowywanych w lokalnych CPU: jednostka CPU S7-1200 może używać zarówno adresowania symbolicznego jak i bezwzględnego jako parametrów RD_x oraz SD_x instrukcji (odpowiednio) GET i PUT.

STEP 7 podczas wstawiania instrukcji automatycznie tworzy DB.

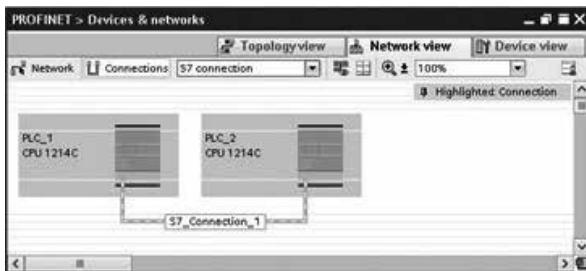
Uwaga

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 rozpoczęć następną operację odczytu/zapisu.

7.8.2. Tworzenie połączenia S7

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 o tworzeniu połączenia sieciowego znajduje się na stronie 113.

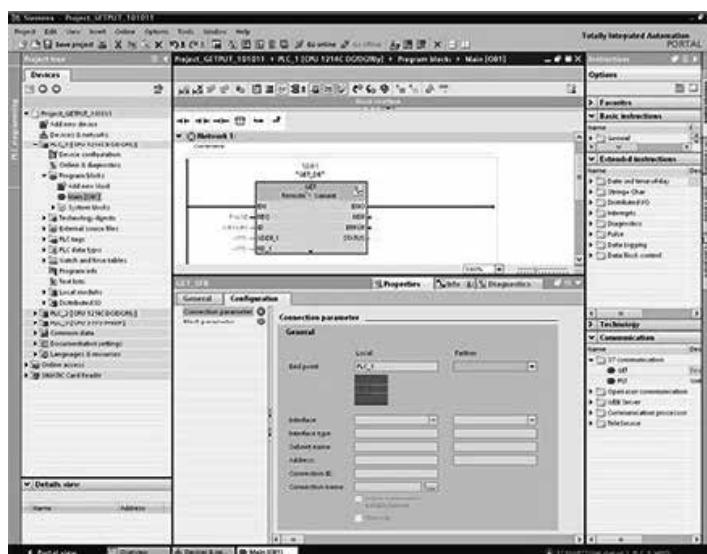


Naciśnij przycisk *Highlighted: Connection*, aby uzyskać dostęp do okna dialogowego *Properties* dla instrukcji komunikacyjnej.

7.8.3. Przypisanie wartości parametrów połączeniu GET/PUT

Instrukcje przypisywania parametrów połączeń GET/PUT pomagają użytkownikom konfigurować połączenia sterownika w komunikacji CPU-CPU.

Po wstawieniu w program bloku GET lub PUT automatycznie jest otwierane okno edycji parametrów instrukcji:



W oknie inspekcyjnym są wyświetlane właściwości połączenia niezależnie od zaznaczonej części bloku. Parametry połączenia są specyfikowane w zakładce *Configuration*, ulokowanej w *Properties* instrukcji obsługującej komunikację.

Po wstawieniu w program bloku GET lub PUT okno *Configuration* jest wyświetlane automatycznie, zostaje także wyświetlona strona *Connection parameters*. Za jej pomocą użytkownik może skonfigurować niezbędne połączenie, może także nadać mu indywidualną nazwę „*Connection ID*”, która jest jego nazwą referencyjną. Strona *Block parameters* umożliwia modyfikacje dodatkowych parametrów edytowanego bloku.

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.

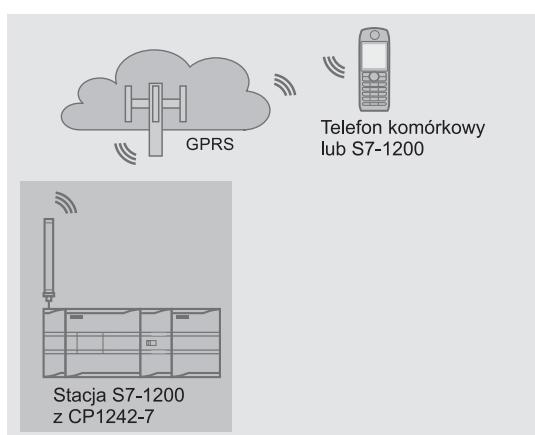
Jeśli używany jest GPRS, umowa musi zezwalać na jego użycie.

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.

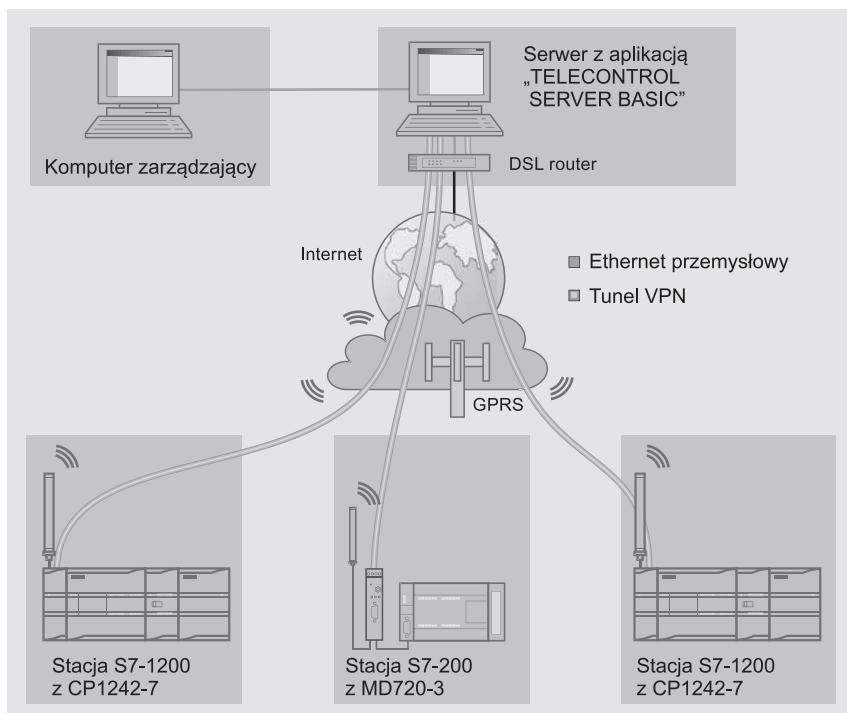
Wysyłanie wiadomości przez SMS



Rys. 7.3. Wysyłanie wiadomości przez SMS ze stacji S7-1200

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

Zdalna kontrola (teleservice) przez centrum sterujące



Rys. 7.4. 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.

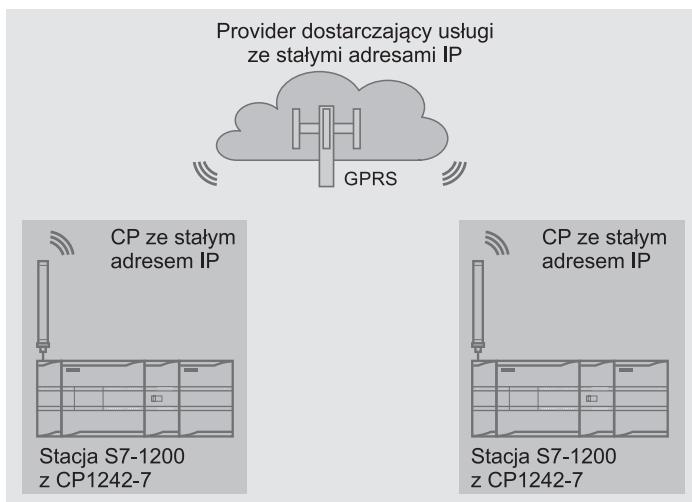
W pierwszym przypadku stacje komunikowały się z serwerem sterowania zdalnego. Używając serwera OPC z aplikacji TELECONTROL SERVER BASIC, serwer zdalnego sterowania wymienia dane z komputerem PC podłą-

czonym 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.

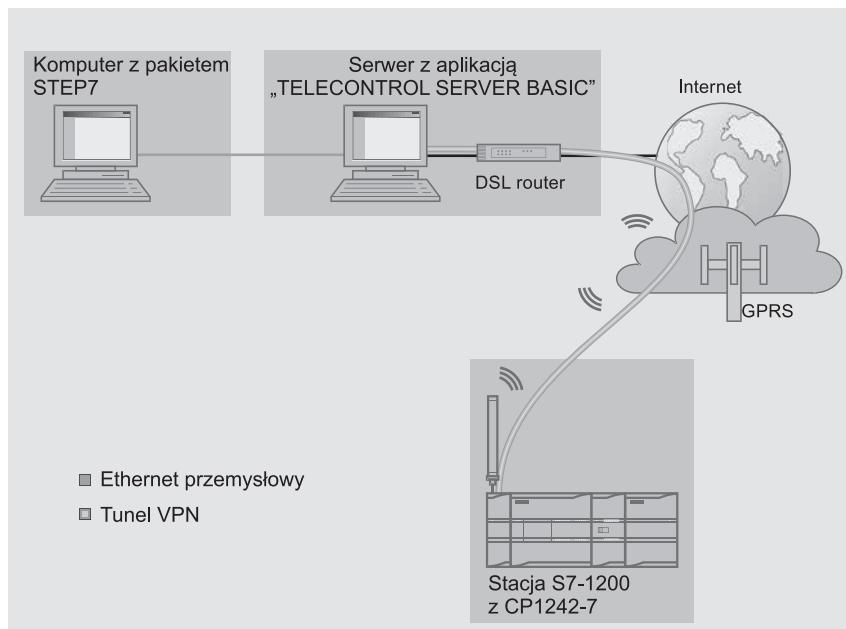
Bezpośrednia komunikacja pomiędzy stacjami



Rys. 7.5. 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. 7.6. TeleService poprzez GPRS

TeleService poprzez GPRS: stacja inżynieryjna, 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 połączony 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

CP 1242-7 są konfigurowane przez STEP 7 w wersji V11.0. 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

Procesor komunikacyjny wyposażono w oddzielne gniazdo na zasilanie 24 V DC

- Interfejs bezprzewodowy sieci GSM

Wymagana jest dodatkowa antena dla komunikacji w sieci GSM. Jest ona podłączana przez kieszeń SMA znajdująca 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. 7.7. 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



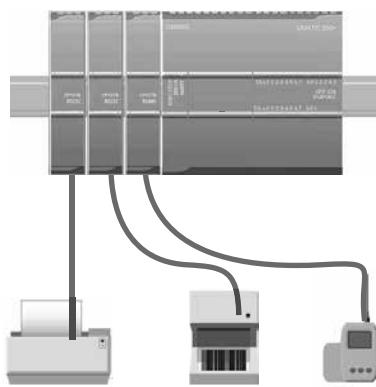
Rys. 7.8. Płaska antena ANT794-3M

| 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 samoprzylepnią, możliwe śruby montażowe |

Anteny muszą być zamawiane oddzielnie.

7.10. Protokoły komunikacyjne PtP, USS oraz Modbus

CPU obsługuje protokół PtP (*Point-to-Point*) dla znakowej transmisji szeregowej, podczas której aplikacja użytkownika całkowicie definiuje i implementuje wybrany protokół.



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 urządzeń innego typu.
- Pozwala wymieniać informację, wysyłać i odbierać dane, z innymi urządzeniami, takimi jak GPS, kamery i systemy wizyjne innych producentów, modemy radiowe.

PtP jest protokołem szeregowej komunikacji, która obsługuje różne szybkości transmisji i warianty parzystości. STEP 7 umożliwia użycie instrukcji obsługujących protokół USS sterowania napędami (tylko RS485), protokół Modbus RTU Master oraz protokół Modbus RTU Slave.

7.10.1. Korzystanie z interfejsów komunikacji szeregowej

Interfejs dla komunikacji PtP zapewniają dwa moduły komunikacyjne (CM) i jedna płytka komunikacyjna (CB):

- CM 1241 RS232,
- CM 1241 RS422/485,
- CB 1241 RS485.

Można podłączyć do trzech modułów komunikacyjnych CM (dowolnego typu) i jedną płytę komunikacyjną CB, co daje w sumie do 4 interfejsów komunikacyjnych. CM należy instalować z lewej strony CPU lub innego CM. Płyta komunikacyjna CB jest instalowana od przodu CPU. Szczegółowe informacje dotyczące instalacji i deinstalacji modułów są podane w rozdziale „Instalacja” w publikacji „Podręcznik systemowy S7-1200”.

Moduły komunikacyjne RS232 i RS485 mają następujące cechy:

- Izolowane porty.
- Obsługują protokół Point-to-Point.
- Są konfigurowane i programowane za pomocą rozszerzonych instrukcji oraz funkcji bibliotecznych.
- 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): Dioda świeci się na czerwono dopóki jest zaadresowana poprzez CPU. Po włączeniu CPU sprawdza ona obecność CM i adresuje je. Dioda zaczyna migać na zielono. Oznacza to, że CPU zaadresował CM, ale nie zostały one jeszcze skonfigurowane. CPU ładowuje konfiguracje do konfigurowanych CM w momencie, kiedy program jest ładowany do CPU. Po tym zdarzeniu dioda diagnostyczna na module komunikacyjnym powinna świecić 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.

7.10.2. 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 określona 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.

7.10.3. Instrukcje USS

S7-1200 obsługuje protokół USS i dostarcza instrukcje, które są zaprojektowane specjalnie do komunikacji z napędami przez port RS485 modułu CM lub płytka 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 przerwań.
- 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 wywoływanie 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 wywoływanie tylko z głównego OB.

Instancja DB zawiera tymczasowe magazynowanie i bufora 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 operacja komunikacji z jednym napędem.

Interwał USS_PORT jest to czas potrzebny do wykonania jednej operacji komunikacji. W tabeli 7.14 pokazano minimalny interwał USS_PORT dla każdej prędkości transmisji. Wywoływanie USS_PORT częściej niż wynosi interwał USS_PORT nie powoduje zwiększenia liczby operacji. Limit czasowy napędu jest to czas jaki może być przeznaczony na operację w warunkach, gdy błędy komunikacyjne spowodowały, że podjęte były 3 próby dokończenia operacji. Biblioteka protokołu USS domyślnie wykonuje dwie próby transmisji danych dla każdej transakcji.

Tabela 7.14. Czasy charakteryzujące transmisję szeregową

| Prędkość transmisji (b/s) | Wymagany interwał pomiędzy wywołaniami USS_PORT (ms) | Wymagane interwały pomiędzy transmisją wiadomości (ms) |
|---------------------------|--|--|
| 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 |

7.10.4. 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 tę 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 niepowtarzalnąinstancję 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 wykonywanie indywidualnie.

Uwaga

Protokół Modbus TCP pracuje poprawnie z oprogramowaniem sprzętowym 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. |
| Modbus TCP (PROFINET) | MB_CLIENT: Tworzy połączenie TCP typu klient-serwer, wysyła komunikaty polecenia, odbiera odpowiedzi i steruje odłączaniami 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 odpływać instrukcjami MB_MASTER/MB_SLAVE lub MB_Client/MB_Server abytrzymać 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.

Łatwy w użyciu regulator PID

8

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 regulatora PID i wprowadzone do pamięci sterownika będącego w trybie RUN nie zostaną uwzględnione podczas jego działania do chwili przełączenia sterownika z trybu STOP w tryb RUN.

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 cyku programu (takich jak OB 1).

Czas próbkowania algorytmu PID odpowiada czasowi pomiędzy obliczeniem dwóch kolejnych wartości wyjściowych (wartości sterowania). Wartości wyjściowe są obliczane podczas autodostrajania, a czas zaokrąglony do wielokrotności czasu cyku. Wszystkie inne instrukcje są wykonywane podczas każdego wywołania.

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 wstępniego rozruchu. War-

tość ta jest używana jako odniesienie dla funkcji nadzorujących w obliczeniach. Monitorowanie obejmuje pomiar bieżącego czasu pomiędzy dwoma wywołaniami i uśrednianie zdefiniowanego przez sterownik czasu próbkiowania.

Wartości wyjściowe regulatora PID składają się z trzech składników:

- P (proporcjonalny): wartość wyjściowa jest proporcjonalna do różnicy pomiędzy wartością zadaną, a wartością bieżącą (uchybą).
- 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 zadaną | s | Operator Laplace'a |
| K _p | Proporcjonalne wzmacnianie (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 zadaną | s | Operator Laplace'a |
| K _p | Proporcjonalne wzmacnianie (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 169). Należy określić tryb autodostrajania (*pretuning* podczas rozruchu lub ręczne dostrajanie – *fine tuning*), aby narucić tryb pracy regulatora PID (strona 172).

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 instrukcję PID. Każda instrukcja PID musi posiadać własną, niepowtarzaną 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 169).

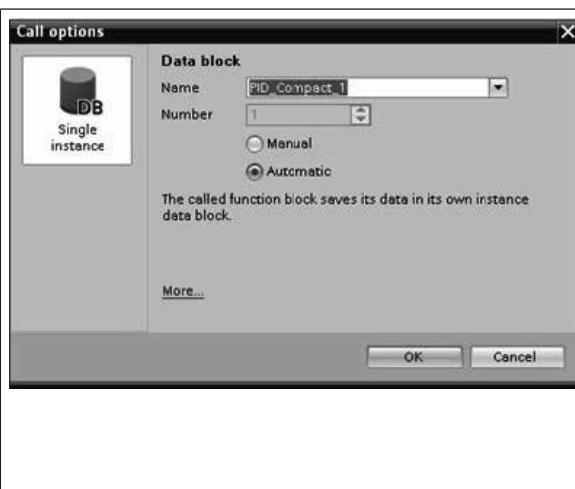
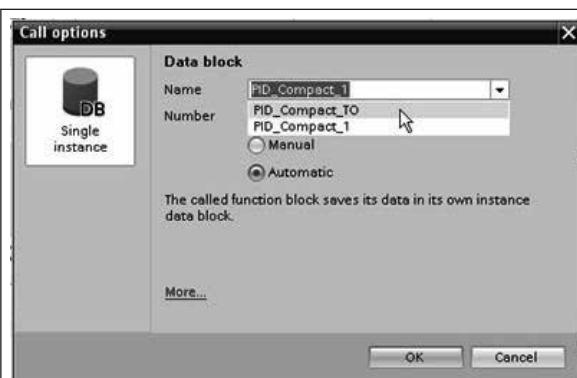
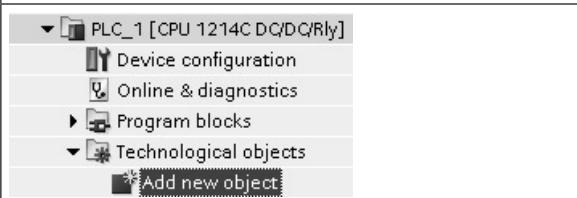


Tabela 8.2. Tworzenie projektu technologicznego z poziomu nawigatora projektu (opcjonalnie)

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.

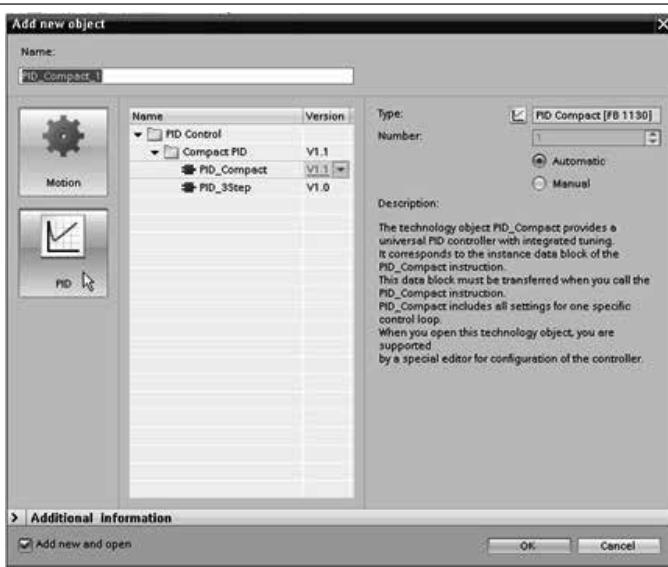


Aby stworzyć obiekt technologiczny, należy dwukrotnie kliknąć ikonę *Add new object* w nawigatorze projektu.



Należy kliknąć ikonę *Control*, aby wybrać obiekt technologiczny dla danego typu regulatora PID (PID_Compact lub PID_3Step). Użytkownik może stworzyć nazwę dla tego obiektu.

Naciśnij *OK*, aby stworzyć obiekt technologiczny.



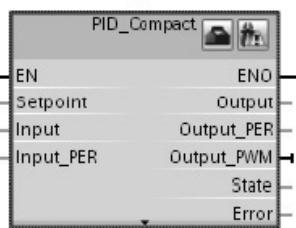
8.2. Instrukcja PID_Compact

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) |

Tabela 8.3. Instrukcja PID_Compact

| LAD/FBD | SCL | Opis |
|---|-----|---|
|  <pre>"PID_Compact_1" Setpoint:= _real_in_ Input:= _real_in_ Input_PER:= _word_in_ ManualEnable:= _bool_in_ ManualValue:= _real_in_ Reset:= _bool_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=> _dword_out_;</pre> | | PID_Compact umożliwia użycie regulatora PID z autodostrajaniem w trybie automatycznym i manualnym. PID_Compact jest regulatorem typu PIDT1 z mechanizmem zabezpieczenia przed „nawijaniem” (anti-windup) oraz ważeniem składników P oraz D. |

¹ Podczas wstawiania instrukcji PID do programu użytkownika STEP 7 automatycznie tworzy dla instrukcji obiekt technologiczny i instancję DB. Instancia DB zawiera wszystkie parametry obiektu technologicznego.

² W przykładzie SCL „PID_Compact_1” jest nazwą instancji DB.

Tabela 8.4. Typy danych dla określonych parametrów

| Parametr i jego typ | | Typ danych | Opis |
|-------------------------|-----|------------|---|
| Setpoint | IN | Real | Wartość zadana dla regulatora PID pracującego w trycie automatycznym. Wartość domyślna: 0.0. |
| Input | IN | Real | Wartość bieżąca procesu. Wartość domyślna: 0.0 Należy także ustawić: sPid_Cmpt.b_Input_PER_On = FALSE. |
| Input_PER | IN | Word | Wartość analogowa procesu (opcjonalnie). Wartość domyślna: W#16#0 Należy także ustawić: sPid_Cmpt.b_Input_PER_On = TRUE. |
| ManualEnable | IN | Bool | Włącza lub wyłącza manualny tryb pracy. Wartość domyślna: FALSE <ul style="list-style-type: none"> PID_Compact V1.0 i V2.0: gdy moduł CPU jest przełączany w tryb RUN i ManualEnable = TRUE to PID_Compact startuje w trybie ręcznym. Nie jest konieczne przełączanie PID_Compact w tryb ręczny podczas zmiany ustawienia FALSE na TRUE. PID_Compact V1.1: gdy moduł CPU jest przełączany w tryb RUN i ManualEnable = TRUE to PID_Compact startuje w trybie ostatnio ustalonym. Zmiana ustawienia z TRUE na FALSE oraz z FALSE na TRUE jest niezbędna, jeżeli PID_Compact jest przełączany w tryb ręczny. |
| ManualValue | IN | Real | Wartość bieżąca procesu dla ręcznego trybu pracy. Wartość domyślna: 0.0 |
| Reset | IN | Bool | Parametr Reset powoduje wyzerowanie sterownika. Jego wartość domyślna: FALSE. Zobacz diagramy umieszczone dalej w temacie „Odpowiedź na zerowanie”. |
| ScaledInput | OUT | Real | Przeskalowane wartości procesu Wartość domyślna: 0.0 |
| Output ¹ | OUT | Real | Wartość wyjściowa. 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 impulsu (PWM). Wartość domyślna: FALSE |
| SetpointLimit_H | OUT | Bool | Góra granica wartości zadanej. Wartość domyślna: FALSE Jeśli SetpointLimit_H = TRUE to bezwzględna wartość górnego limitu wartości zadanej została osiągnięta. Wartość domyślna: FALSE. |
| SetpointLimit_L | OUT | Bool | Dolina granica wartości zadanej. Wartość domyślna: FALSE Jeśli SetpointLimit_L = TRUE to bezwzględna wartość dolnego limitu wartości zadanej została osiągnięta. Wartość domyślna: FALSE. |
| InputWarning_H | OUT | Bool | Jeśli InputWarning_H = TRUE to wartość procesu osiągnęła lub przekroczyła wartość górnego ostrzeżenia. Wartość domyślna: FALSE. |
| InputWarning_L | OUT | Bool | Jeśli InputWarning_L = TRUE to wartość procesu osiągnęła lub przekroczyła wartość dolnego ostrzeżenia. Wartość domyślna: FALSE. |
| State | OUT | Int | Bieżący tryb pracy regulatora PID. Wartość domyślna: 0 Do zmiany trybu należy użyć sRet.i_Mode: <ul style="list-style-type: none"> State = 0: Inactive (nieaktywny) State = 1: Pretuning (dostrajanie podczas rozruchu) State = 2: Manual fine tuning (ręczne strojenie) State = 3: Automatic mode (tryb automatyczny) State = 4: Manual mode (tryb ręczny) |
| ErrorBits | OUT | DWord | Parametr ErrorBits instrukcji PID_Compact zawiera kod komunikatu o błędzie (strona 160). Wartość domyślna: DW#16#0000. |

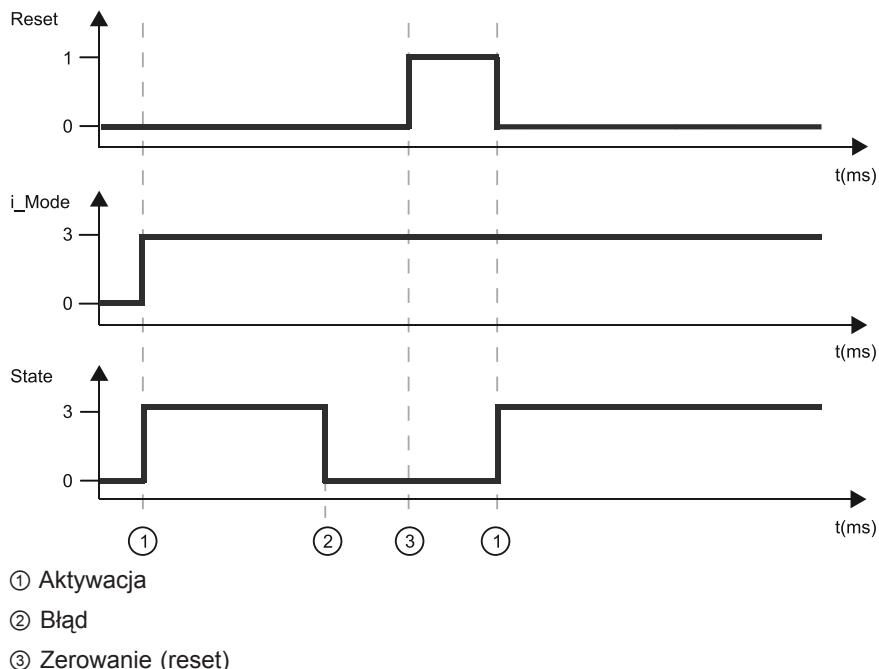
¹ Parametry wyjścia: Output, Output_PER oraz Output_PWM mogą być używane równolegle.

Odpowiedź na zerowanie

Odpowiedź na Reset=TRUE zależy od wersji instrukcji PID_Compact.

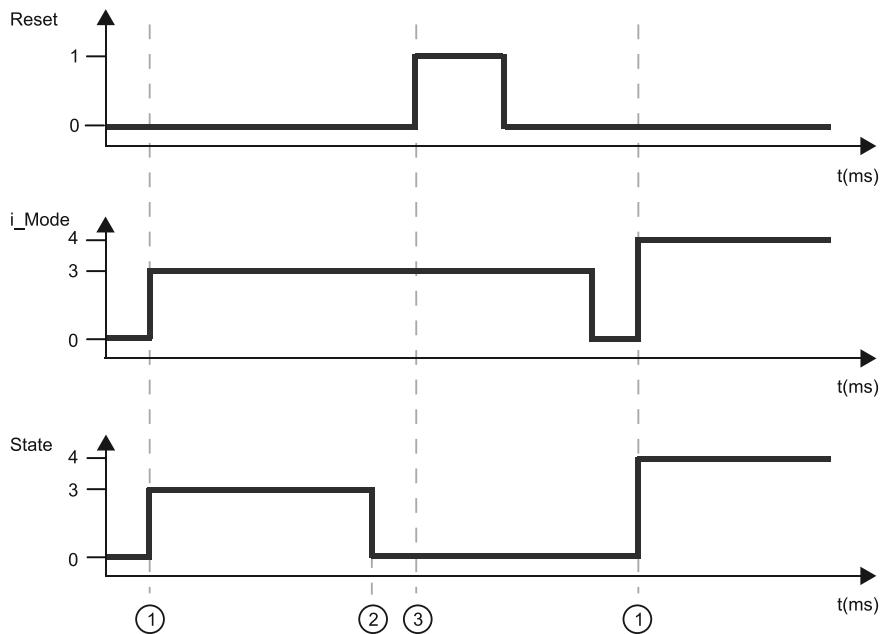
Instrukcja PID_Compact w wersji V1.1

Narastające zbocze na Reset zeruje komunikaty o błędach, ostrzeżenia oraz zatrzymuje działanie sterownika. Opadające zbocze na Reset powoduje powrót sterownika do ostatniej aktywności.



Instrukcja PID_Compact w wersji V1.0

Narastające zbocze na Reset zeruje komunikaty o błędach, ostrzeżenia oraz zatrzymuje działanie sterownika. Wraca on do normalnego działania dopiero po powojeniu się narastającego zbocza na i_Mode.

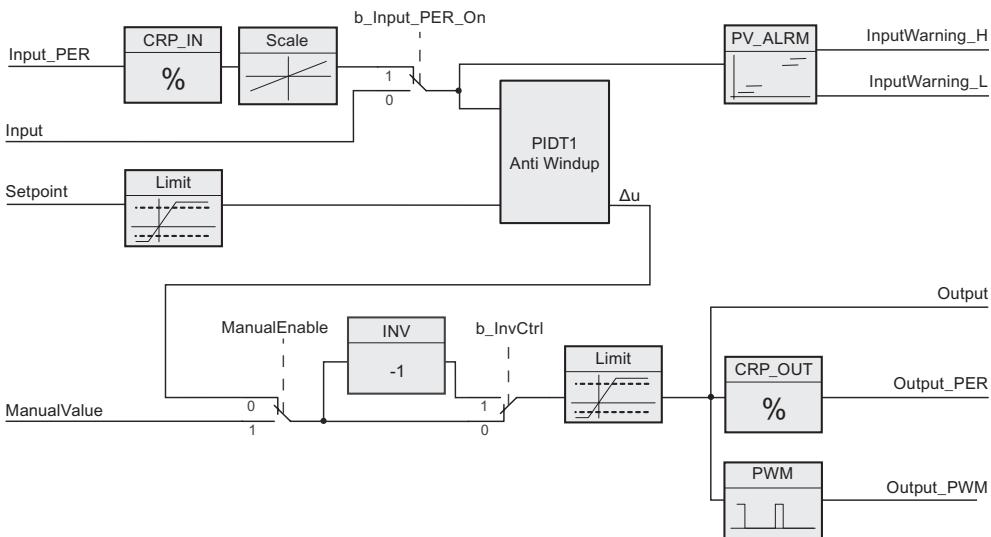


① Aktywacja

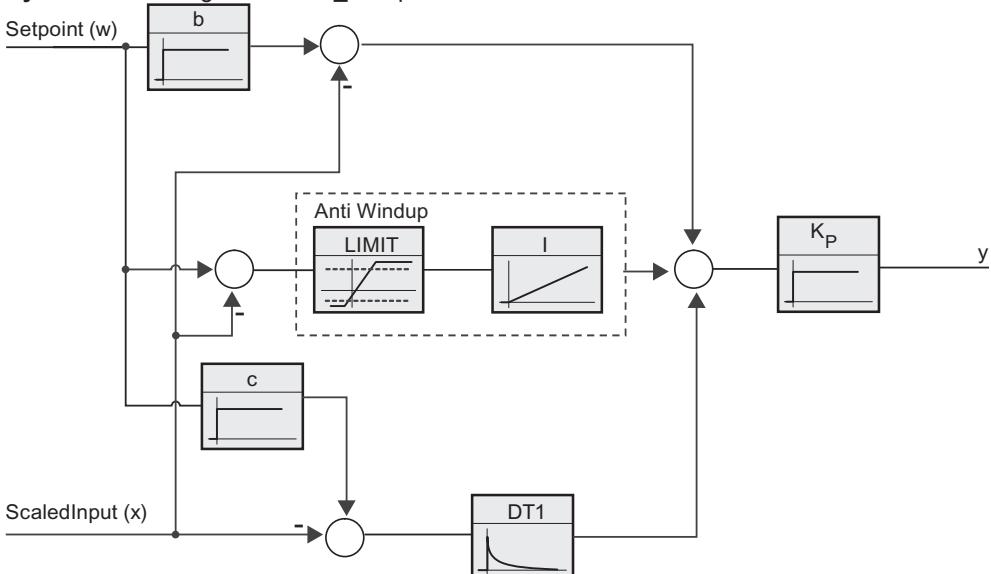
② Błąd

③ Zerowanie (reset)

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 ErrorBits instrukcji PID_Compact

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 8.5. Parametry ErrorBits instrukcji PID_Compact

| Wartość ErrorBit (DW#16#...) | Opis |
|---------------------------------|---|
| 0000 | Brak błędu |
| 0001 | Wartość parametru „Input” ma wartość spoza dopuszczalnego zakresu. Input>sPid_Cmpt.r_Pv_HImor Input< sPid_Cmpt.r_Pv_LIm |
| 0002 | Nieprawidłowa wartość parametru „Input_PER”. Konieczna weryfikacja czy błąd wystąpił na wejściu analogowym. |
| 0004 | Błąd wykryty podczas precyzyjnego dobierania wartości Oscillation – brak możliwości utrzymania wartości procesowej. |
| 0008 | Błąd wykryty na etapie strojenia wstępniego. Wartość procesowa zbyt bliska wartości zadanej. Start strojenia precyzyjnego. |
| 0010 | Wartość zadana zmieniała się podczas strojenia regulatora. |
| 0020 | Strojenie wstępne nie może być zrealizowane w trybie automatycznym ani podczas strojenia precyzyjnego. |
| 0040 | Błąd wykryty na etapie strojenia precyzyjnego. Wartość zadana jest zbyt bliska dopuszczalnym wartościom skrajnym. |
| 0080 | Nieprawidłowa konfiguracja zakresów wartości wyjściowej. Wymagane sprawdzenie czy wartości wyjściowe skonfigurowano prawidłowo i czy są zgodne z wymaganymi kierunkami regulacji. |
| 0100 | Błąd wykryty podczas strojenia regulatora wymusił nieprawidłowe wartości parametrów. |
| 0200 | Nieprawidłowa wartość parametru „Input”: format numeryczny jest nieprawidłowy. |
| 0400 | Błąd obliczania wartości wyjściowej. Wymagane sprawdzenie parametrów PID. |
| 0800 | Błąd czasu próbkowania: instrukcja PID_Compact nie została wywołana w czasie cyklicznego wywołania przerwania OB. |
| 1000 | Nieprawidłowa wartość parametru „Setpoint”: format numeryczny jest nieprawidłowy. |

8.4. Instrukcja PID_3Step

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 wzmacnienie
(Składnik P)

a Współczynnik opóźnienia całkowania
(Składnik D)

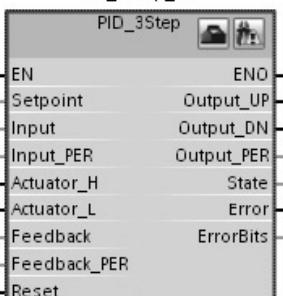
T_I Czas całkowania – czas zdwojenia
(Składnik I)

b Wagi działania proporcjonalnego
(Składnik P)

T_D Czas różniczkowania czas wyprzedzenia
(Składnik D)

c Wagi działania różniczkującego
(Składnik D)

Tabela 8.6. Instrukcja PID_3Step

| LAD/FBD | SCL | Opis |
|--|--|---|
|  <p>"PID_3Step_TO"</p> <p>EN ENO</p> <p>Setpoint Output_UP</p> <p>Input Output_DN</p> <p>Input_PER Output_PER</p> <p>Actuator_H State</p> <p>Actuator_L Error</p> <p>Feedback ErrorBits</p> <p>Feedback_PER </p> <p>Reset</p> | <pre> "PID_3Step_1"(SetpInt:= _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_, SetpIntLimitH=> _bool_out_, SetpIntLimitL=> _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 PIDT1 z mechanizmem zabezpieczenia przed „nawijaniem” (anti-windup) 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.

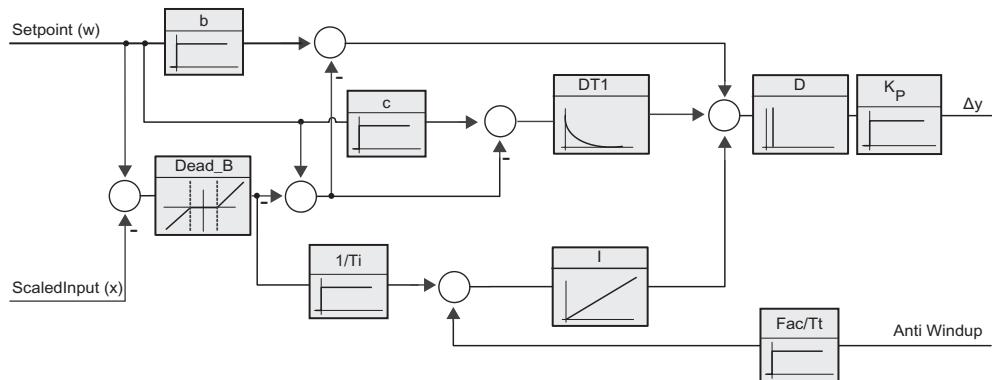
² W przykładzie SCL „PID_3Step_1” jest nazwą instancji DB.

Tabela 8.7. Typy danych dla określonych parametrów

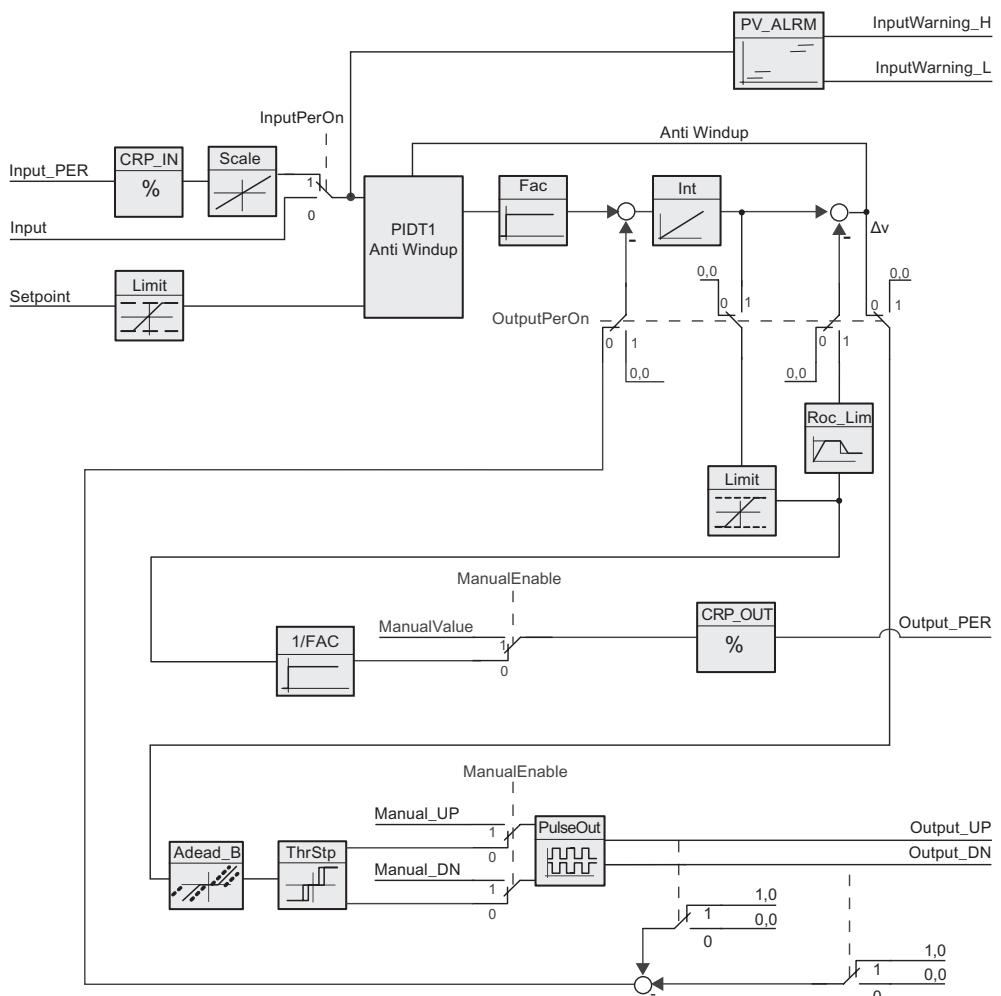
| Parametr i jego typ | | Typ danych | Opis |
|---------------------|----|------------|---|
| Setpoint | IN | Real | Wartość zadana dla regulatora PID pracującego w trycie automatycznym. Wartość domyślna: 0.0. |
| Input | IN | Real | Wartość bieżąca procesu. Wartość domyślna: 0.0 Należy także ustawić Config.InputPEROn = FALSE. |
| Input_PER | IN | Word | Wartość analogowa procesu (opcjonalnie). Wartość domyślna: W#16#0. Należy także ustawić Config.InputPEROn = TRUE. |
| ManualEnable | IN | Bool | Włącza lub wyłącza manualny tryb pracy. Wartość domyślna: FALSE <ul style="list-style-type: none"> Przy zmianie zbocza z FALSE na TRUE, Regulator przełącza się na tryb ręczny - State = 4; niezmienione Retain.Mode. Przy zmianie zbocza z TRUE na FALSE, regulator przełącza się na ostatnio używany tryb pracy; State = Retain.Mode. |

| Parametr i jego typ | Typ danych | Opis |
|---------------------|------------|--|
| ManualUP | IN | Bool W trybie ręcznym każde zbocze narastające powoduje rozwarcie zaworu o 5% całego zakresu ruchu lub na minimalny okres czasu potrzebny do uruchomienia silnika. Wartość ManualUP jest szacowana tylko jeśli nie jest wykorzystywana instrukcja Output_PER i nie ma sygnału zwrotnego z czujnika położenia. Wartość domyślna: FALSE <ul style="list-style-type: none"> Jeśli Output_PER = FALSE, ręczne wejście włącza Output_UP na czas odpowiadający czasowi 5% ruchu urządzenia. Jeśli Config.ActuatorEndStopOn = TRUE, Output_UP nie załącza się nawet jeśli Actuator_H = TRUE. |
| ManualDN | IN | Bool W trybie ręcznym każde zbocze narastające powoduje rozwarcie zaworu o 5% całego zakresu ruchu lub na minimalny okres czasu potrzebny do uruchomienia silnika. Wartość ManualDN jest szacowana tylko jeśli nie jest wykorzystywana instrukcja Output_PER i nie ma sygnału zwrotnego z czujnika położenia. Wartość domyślna: FALSE <ul style="list-style-type: none"> Jeśli Output_PER = FALSE, ręczne wejście włącza Output_DN na czas odpowiadający czasowi 5% ruchu urządzenia. Jeśli Config.ActuatorEndStopOn = TRUE, Output_DN nie załącza się nawet jeśli Actuator_L = TRUE. |
| ManualValue | IN | Real Wartość bieżąca procesu dla ręcznego trybu pracy. Wartość domyślna: 0.0 W trybie ręcznym użytkownik określa bezwzględną pozycję zaworu. ManualValue jest obliczany tylko jeśli użytkownik używa OutputPer, lub jeśli dostępne jest sprzężenie zwrotne. Wartość domyślna: 0.0 |
| Feedback | IN | Real Sprzężenie zwrotne od położenia zaworu. Wartość domyślna: 0.0 Aby użyć Feedback należy ustawić Config.FeedbackPerOn = FALSE. |
| Feedback_PER | IN | Word Analogowe sprzężenie zwrotne od pozycji zaworu. Wartość domyślna: W#16#0 Aby użyć Feedback_PER, należy ustawić Config.FeedbackPerOn = TRUE. Feedback_PER jest skalowane za pomocą następujących parametrów: <ul style="list-style-type: none"> Config.FeedbackScaling.LowerPointIn Config.FeedbackScaling.UpperPointIn Config.FeedbackScaling.LowerPointOut Config.FeedbackScaling.UpperPointOut |
| Actuator_H | IN | Bool Jeśli Actuator_H = TRUE, zawór jest w górnej pozycji krańcowej i nie może kontynuować ruchu w tym kierunku. Wartość domyślna: FALSE |
| Actuator_L | IN | Bool Jeśli Actuator_L = TRUE, zawór jest w dolnej pozycji krańcowej i nie może kontynuować ruchu w tym kierunku. Wartość domyślna: FALSE |
| Reset | IN | Bool Restartuje regulator. Wartość domyślna: FALSE Przejście z FALSE do TRUE powoduje: <ul style="list-style-type: none"> Nieaktywny tryb pracy Wartość wejściowa= 0 Pośrednie wartości systemowe zostają skasowane (PIDParameter zostanie zachowany) Przejście z TRUE do FALSE powoduje powrót do ostatniego stanu aktywnego. |

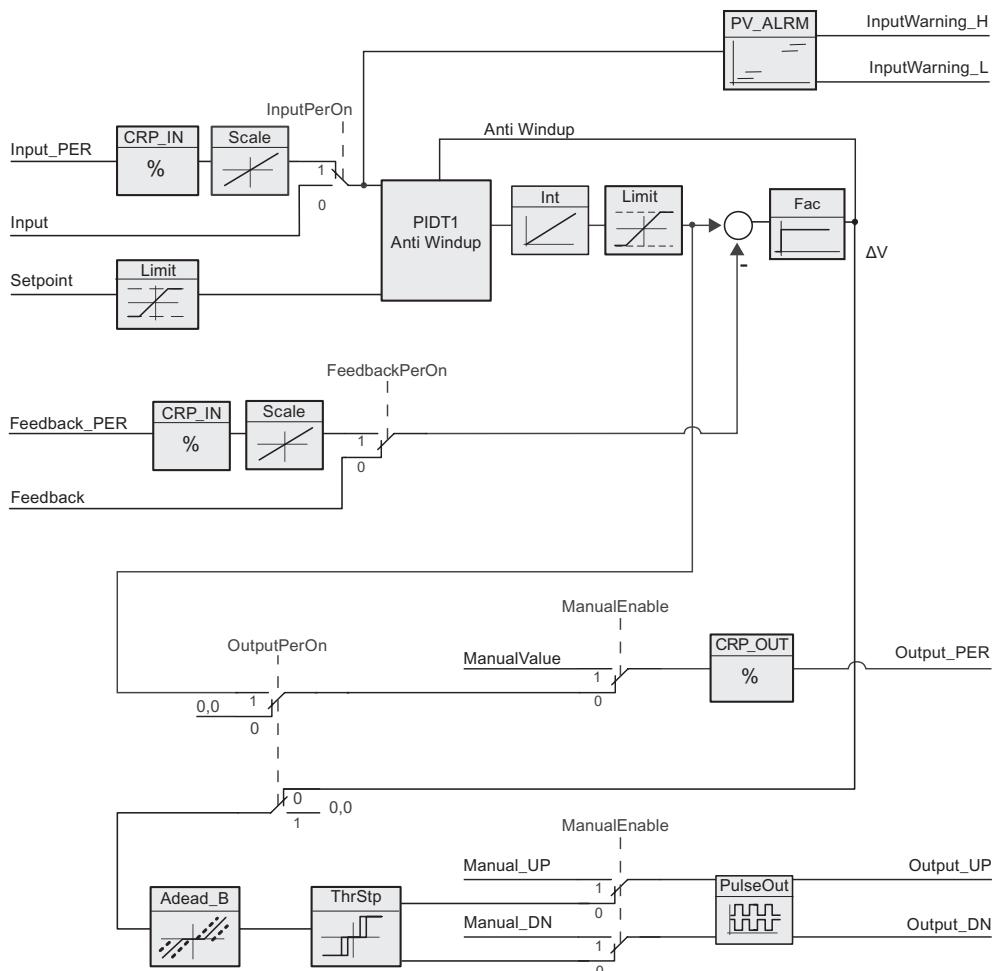
| Parametr i jego typ | | Typ danych | Opis |
|---------------------|-----|------------|---|
| ScaledInput | OUT | Real | Przeskalowane wartości procesu |
| ScaledFeedback | OUT | Real | Przeskalowane pozycje zaworu |
| Output_PER | OUT | Word | Analogowa wartość wyjściowa. Jeśli Config.OutputPerOn = TRUE obliczana jest Output_PER. |
| Output_UP | OUT | Bool | Cyfrowa wartość wyjściowa otwarcia zaworu. Wartość domyślna: FALSE. Jeśli Config.OutputPerOn = FALSE to parametr Output_UP jest obliczany. |
| Output_DN | OUT | Bool | Cyfrowa wartość wyjściowa zamknięcia zaworu. Wartość domyślna: FALSE Jeśli Config.OutputPerOn = FALSE to parametr Output_DN jest obliczany. |
| SetpointLimitH | OUT | Bool | Górna granica wartości zadanej. Wartość domyślna: FALSE Jeśli SetpointLimit_H = TRUE to bezwzględna wartość górnego limitu wartości zadanej została osiągnięta. W CPU wartość zadana jest ograniczona przez aktualnie skonfigurowaną bieżącą wartość limitu górnego. |
| SetpointLimitL | OUT | Bool | Dolna granica wartości zadanej. Wartość domyślna: FALSE Jeśli SetpointLimit_L = TRUE to bezwzględna wartość dolnego limitu wartości zadanej została osiągnięta. W CPU wartość zadana jest ograniczona przez aktualnie skonfigurowaną bieżącą wartość limitu dolnego. |
| InputWarningH | OUT | Bool | Jeśli InputWarningH = TRUE, to wartość wejściowa osiągnęła lub przekroczyła wartość górnego ostrzeżenia. Wartość domyślna: FALSE. |
| InputWarningL | OUT | Bool | Jeśli InputWarningL = TRUE, to wartość wejściowa osiągnęła lub przekroczyła wartość dolnego ostrzeżenia. Wartość domyślna: FALSE. |
| State | OUT | Int | Bieżący tryb pracy regulatora PID. Wartość domyślna: 0 Do zmiany trybu należy użyć sRet.i_Mode: <ul style="list-style-type: none">• State = 0: Inactive (nieaktywny).• State = 1: Pretuning (dostrajanie podczas rozruchu).• State = 2: Manual fine tuning (ręczne strojenie).• State = 3: Automatic mode (tryb automatyczny).• State = 4: Manual mode (tryb ręczny).• State = 5: Substitute output value approach (zastąpienie wartości wyjściowej przybliżoną).• State = 6: Transition time measurement (pomiar czasu przejścia).• State = 7: Substitute output value approach with error monitoring (zastąpienie wartości wyjściowej przybliżoną z monitorowaniem błędów).• State = 8: Error monitoring (monitorowanie błędów). |
| Error | OUT | Bool | Jeśli Error = TRUE to przynajmniej jeden komunikat o błędzie jest w toku. Wartość domyślna: FALSE. |
| ErrorBits | OUT | DWord | Parametry ErrorBits instrukcji PID_3STEP zawierają kod komunikatu wykrytego błędu (strona 167). Wartość domyślna: DW#16#0000. |



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.

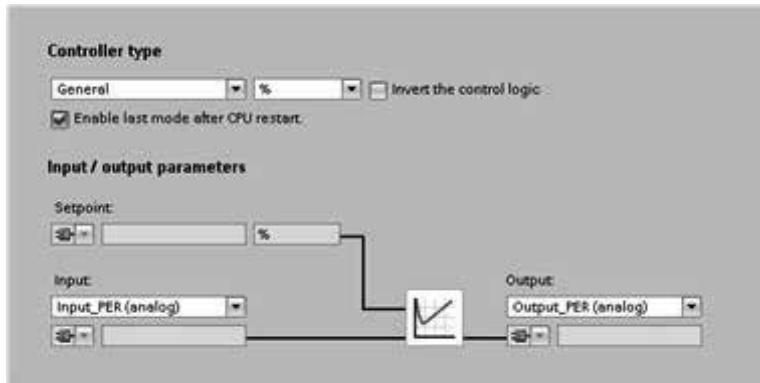
Tabela 8.8. Parametry ErrorBits instrukcji PID_3STEP

| Wartość ErrorBit (DW#16#...) | Opis |
|---------------------------------|--|
| 0000 | Brak błędu. |
| 0001 | Wartość parametru „Input” jest poza granicami wartości procesowych: Input> Config.InputUpperLimit lub Input<Config.InputLowerLimit Jeśli ActivateRecoverMode = TRUE i ErrorBehaviour = 1, siłownik przesuwa się do zastępczej wartości wyjściowej. Jeśli ActivateRecoverMode = TRUE i ErrorBehaviour = 0, siłownik zatrzymuje się w aktualnym położeniu. Jeśli ActivateRecoverMode = FALSE, siłownik zatrzymuje się w aktualnym położeniu. PID_3STEP V1.1: Można przesunąć siłownika w trybie ręcznym. PID_3STEP V1.0: Tryb ręczny nie jest możliwy w tym stanie. Nie można uruchomić silownika ponownie, dopóki błąd nie zostanie usunięty. |
| 0002 | Nieprawidłowa wartość parametru „Input_PER”. Należy sprawdzić czy błąd nie wystąpił na wejściu analogowym. Jeżeli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE, wskaźnik błędu jest kasowany, a PID_3STEP przełącza się w tryb automatyczny. |
| 0004 | Błąd wykryty podczas precyzyjnego dobierania wartości Oscillation - brak możliwości utrzymania wartości procesowej. |
| 0008 | Błąd wykryty na etapie strojenia wstępne. Wartość procesowa zbyt bliska wartości zadanej. Start strojenia precyzyjnego. |
| 0010 | Wartość zadana nie może się zmieniać się podczas strojenia regulatora. |
| 0020 | Strojenie wstępne nie może być zrealizowane w trybie automatycznym ani podczas strojenia precyzyjnego. |
| 0040 | Błąd wykryty na etapie strojenia precyzyjnego. Wartość zadana jest zbyt bliska dopuszczalnym wartościom skrajnym. |
| 0080 | Nieprawidłowa konfiguracja zakresów wartości wyjściowej. Wymagane sprawdzenie czy wartości wyjściowe skonfigurowano prawidłowo i czy są zgodne z wymaganymi kierunkami regulacji. |
| 0100 | Błąd wykryty podczas strojenia regulatora wymusił nieprawidłowe wartości parametrów. |
| 0200 | Nieprawidłowa wartość parametru „Input”: format numeryczny jest nieprawidłowy. Jeżeli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE, wskaźnik błędu jest kasowany, a PID_3STEP przełącza się w tryb automatyczny. |
| 0400 | Błąd obliczania wartości wyjściowej. Wymagane sprawdzenie parametrów PID. |

| Wartość ErrorBit (DW#16#...) | Opis |
|---------------------------------|--|
| 0800 | Błąd czasu próbkowania: instrukcja PID_3STEP nie została wywołana w czasie cyklicznego wywołania przerwania OB. Jeżeli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE, wskaźnik błędu jest kasowany, a PID_3STEP przełącza się w tryb automatyczny. |
| 1000 | Nieprawidłowa wartość parametru „Setpoint”: format numeryczny jest nieprawidłowy. Jeżeli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE, wskaźnik błędu jest kasowany, a PID_3STEP przełącza się w tryb automatyczny. |
| 2000 | Nieprawidłowa wartość parametru „Feedback_PER”: format numeryczny jest nieprawidłowy. Konieczna weryfikacja czy błąd wystąpił na wejściu analogowym. Siłownik nie może przesunąć się do zastępczej wartości wyjściowej i nie może zmienić aktualnego położenia. Tryb ręczny nie jest możliwy w tym stanie. Przesunięcie siłownika z tego stanu jest możliwe po zablokowaniu sprzężenia zwrotnego z czujnika położenia (Config. FeedbackOn = FALSE). Jeżeli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE, wskaźnik błędu jest kasowany, a PID_3STEP przełącza się w tryb automatyczny. |
| 4000 | Nieprawidłowa wartość parametru „Feedback „: format numeryczny jest nieprawidłowy. Siłownik nie może przesunąć się do zastępczej wartości wyjściowej i nie może zmienić aktualnego położenia. Tryb ręczny nie jest możliwy w tym stanie. Przesunięcie siłownika z tego stanu jest możliwe po zablokowaniu sprzężenia zwrotnego z czujnika położenia (Config. FeedbackOn = FALSE). Jeżeli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE, wskaźnik błędu jest kasowany, a PID_3STEP przełącza się w tryb automatyczny. |
| 8000 | Błąd sprzężenia zwrotnego cyfrowego czujnika położenia. Actuator_H = TRUE i ActuatorL = TRUE. Siłownik nie może przesunąć się do zastępczej wartości wyjściowej i nie może zmienić aktualnego położenia. Tryb ręczny nie jest możliwy w tym stanie. Przesunięcie siłownika z tego stanu jest możliwe po zablokowaniu sprzężenia zwrotnego z czujnika położenia (Config. FeedbackOn = FALSE). Jeżeli tryb automatyczny był aktywny przed wystąpieniem błędu, ActivateRecoverMode = TRUE, wskaźnik błędu jest kasowany, a PID_3STEP przełącza się w tryb automatyczny. |

8.6. Konfigurowanie regulatora PID

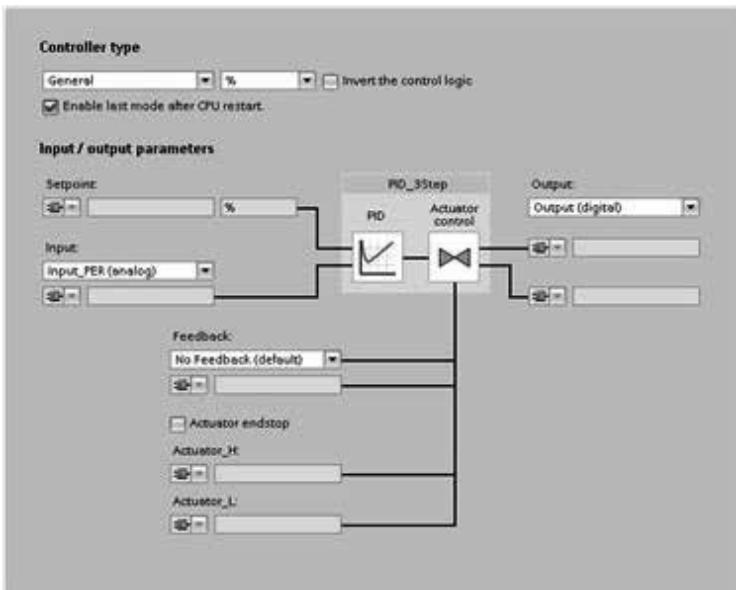
Parametry obiektu technologicznego określają pracę regulatora PID. Użyj ikony do otwarcia edytora konfiguracji.



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 inżynierską |
| | Odwrócenie 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 zwiększone jeśli wartość wejściowa jest mniejsza od zadanej. Jeśli jest zaznaczone, wyjście pętli PID jest zwiększone 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 przeskalać 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 jednostkę inżynierską |
| | Odwrócenie 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 cyfrowych wyjść (Output_UP i Output_DN) lub analogowych wyjść (Output_PER) dla wartości wyjściowych. |
| Process value | Feedback (sprzężenie zwrotne) | Wybór typu stanu urządzenia zwracanego do pętli PID <ul style="list-style-type: none"> Brak sprzężenia (domyślny) Sprzężenie zwrotne Feedback_PER |
| | | 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 przeskalaować analogowe wartości procesu (<i>input value</i>). |

| Ustawienia | | Opis |
|------------|--------------------------------------|--|
| Actuator | Motor transition time | Ustawia czas od otwarcia do zamknięcia zaworu (należy odszukać tą wartość w danych technicznych zaworu lub w opisie bezpośrednio na zaworze). |
| | Minimum ON time | Ustawia najmniejsze przemieszczenie zaworu (należy odszukać tą wartość w danych technicznych zaworu lub w opisie bezpośrednio na zaworze). |
| | Minimum OFF time | Ustawia najmniejszą wartość czasu przestoju zaworu (należy odszukać tą wartość w danych technicznych zaworu lub w opisie bezpośrednio na zaworze). |
| | Error behavior | 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 otworzenie 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 (Low oraz High)</i> definiują analogowe sprzężenie zwrotne od pozycji zaworu. <i>FeedbackPER High</i> musi być większe niż <i>FeedbackPER Low</i>. |

¹ Scale Position Feedback jest edytowalne tylko jeśli użytkownik uaktywnił w ustawieniach podstawowych opcję *Feedback*.

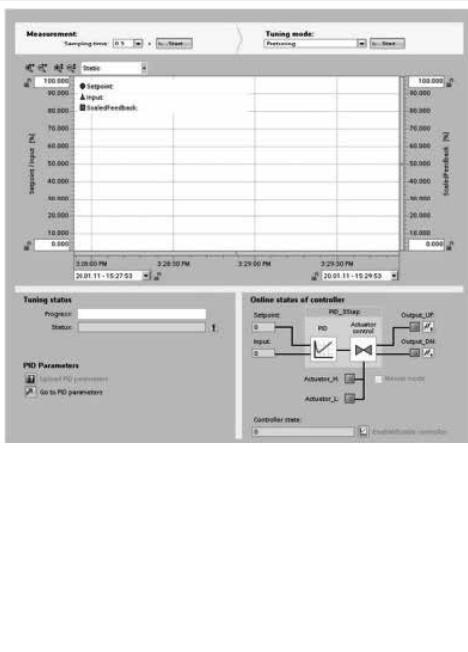
8.7. Uruchomienie regulatora PID

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. Przykład okna konfiguracyjnego (PID_3Step)

| | |
|--|--|
|  | <ul style="list-style-type: none"> Measurement: wyświetla wartość zadaną, wartość procesu (wartość wejściową) oraz trend wartości wyjściowej w czasie rzeczywistym. Należy wprowadzić czas próbkowania i naciśnij przycisk Start. Tuning mode: aby nastroić pętlę PID należy wybrać <i>Pretuning</i> lub <i>Fine tuning</i> (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. <p>Po zakończeniu procesu strojenia, użytkownik może zachować te parametry klikając przycisk <i>Upload PID parameters</i> w sekcji <i>PID Parameters</i> edytora uruchomienia.</p> <p>Jeżeli wystąpi błąd podczas strojenia, to wartość wyjściowa z regulatora wyniesie 0. Regulator przejdzie w tryb „nieaktywny”. Ten stan wywoła błąd.</p> |
|--|--|

Wbudowany serwer WWW ułatwiający komunikację z Internetem

Wbudowany w sterownik serwer WWW pozwala na dostęp do danych 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. Zestaw stron Web pozwala na użycie wielu funkcji:

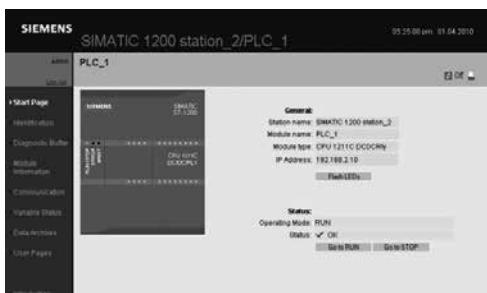
- Zamiana trybu pracy sterownika (STOP oraz RUN).
- Nadzorowanie i modyfikowanie stanu zmiennych PLC.
- Podgląd i ściąganie wszelkich dzienników zdarzeń gromadzonych przez CPU.
- Podgląd bufora diagnostycznego CPU.
- Możliwa jest aktualizacja firmware'u w CPU.

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 może użyć każdej przeglądarki internetowej, która wspiera HTTP w wersji 1.1.

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 połączonego CPU oraz listę ogólnych informacji o CPU.

Jeśli użytkownik jest zalogowany na koncie typu „admin”, może zmieniać tryb pracy CPU (STOP i RUN), a także zapalać diody LED.

The screenshot shows the SIMATIC Manager interface for monitoring variable status. It displays a table with columns: Address, Display Format, Monitor Value, Modify Value, and Modify. The table includes entries for various tags such as Q0_1, I0_1, Converge_00440, Motor_On, Start_Motor, Data_Block_1, Data_Block_2, Tag_1, and Ano_Constant.

Strona **Variable Status** (Zmienne stanu) pozwala na obserwowanie lub zmianę każdego I/O lub danych pamięci w CPU. 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.

The screenshot shows the SIMATIC Manager interface for the diagnostic buffer. It lists several entries with details like number, time, date, and event description. For example, entry 1 shows a transition from STOP to RUN state at 05.25.57 17:00 pm on 01.04.2019.

Strona **Diagnostic Buffer** (bufor diagnostyczny) wyświetla bufor diagnostyczny. 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.

The screenshot shows the SIMATIC Manager interface for data logs. It lists log entries with columns for Date, Time, Recent Entries, Download, Download & Clear, and Delete. The entries show dates from 30.09.2018 to 30.10.2018.

Strona **Data Logs** pozwala na wgląd lub ściągnięcie określonej liczby wpisów dziennika zdarzeń. Serwer Web ściąga dzienniki zdarzeń na komputer użytkownika. Pliki dzienników mają standardowy format .csv. Więcej informacji znajduje się w sekcji poświęconej dziennikom zdarzeń (strona 103).

The screenshot shows the SIMATIC Manager interface for updating firmware. It has a section for selecting a firmware update file and a note about clicking the Load button to transfer the file to the CPU.

Strona **Upgrade Firmware** umożliwia wygodną wymianę firmware'u w CPU.

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 listwie.



Ostrzeżenie

Nieautoryzowany dostęp do CPU lub zmiana wartości zmiennych w PLC na nieprawidłowe mogą spowodować przerwanie prawidłowej pracy sterownika, wywołać zagrożenie życia i/lub uszkodzenia lub zniszczenia urządzeń i/lub mienia.

Ponieważ użytkownicy „admin” wbudowanego serwera Web mogą wpływać na tryb i parametry pracy sterownika, a także zmieniać firmware, firma Siemens zaleca stosowanie następujących mechanizmów zmniejszających ryzyko nieprawidłowej pracy sterownika:

Komunikacja ze sterownikiem powinna odbywać się wyłącznie za pomocą protokołu HTTPS.

Hasło chroniące dostęp do sterownika (strona 75) powinno być silne co znaczy, że powinno składać się z kombinacji co najmniej ośmiu liter, cyfr i znaków specjalnych, nie tworzących popularnych i łatwych do skojarzenia słów lub danych osobistych. Hasło powinno być utajnione i zmieniane tak często jak to możliwe.

Zalecana jest implementacja w programie mechanizmów weryfikujących poprawność wartości zmiennych, co zmniejszy ryzyko działania aplikacji w przypadku zmiany wartości zmiennych przez użytkowników mających do strony Web CPU.

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 albo do stron Web definiowanych przez użytkownika. 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 obsługa JavaScript lub "ciasteczek" jest wyłączona w przeglądarce sieciowej, to należy ją ponownie włączyć. 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 definiowanej

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 połączeniem się z serwerem Web za pomocą `https://` ukaże się powiadomienie o weryfikacji zabezpieczeń.

Liczba połączeń

Wbudowany w CPU serwer Web może jednocześnie obsługiwać do 30 połączeń HTTP. Połączenia te mogą być wykorzystywane na różne sposoby, w zależności m.in. od rodzaju przeglądarki wykorzystanej do nawiązania połączenia oraz liczby formatów danych (CSS, HTML, obrazy itp.) wykorzystywanych do generacji treści stron. Niektóre z połączeń są aktywne podczas przeglądania stron, inne są automatycznie zrywane po pobraniu treści wyświetlanej strony.

Jeżeli – przykładowo – do przeglądania strony Web generowanej przez CPU jest używana przeglądarka Firefox 8, która obsługuje maks. do 6 stałych połączeń jednocześnie, możliwe jest przeglądanie stron Web CPU jednocześnie za pomocą pięciu takich przeglądarki.

Należy pamiętać, że prędkość generacji stron Web przez CPU zmniejsza się ze wzrostem liczby aktywnych połączeń.

9.2.1. Ograniczenia związane z wyłączeniem obsługi JavaScript

Wyłączenie JavaScript ogranicza niektóre funkcje

Standardowe strony Web są implementowane przy użyciu języka HTML, skryptów JavaScript oraz „ciasteczek”. Strony Web nie będą działały poprawnie dopóki nie zostanie załączona obsługa JavaScript i „ciasteczek”. Jeśli użytkownik nie może uruchomić obsługi JavaScript, to kontrolowane przez nie funkcje mogą zostać nieuruchomione.

Tabela 9.1. Nieprawidłowe działanie stron Web po wyłączeniu obsługi JavaScript

| Standardowe strony Web | Efekt |
|------------------------|--|
| Ogólne | <ul style="list-style-type: none">Dane na stronie nie są odświeżane dynamicznie. Użytkownik musi ręcznie odświeżać stronę poprzez naciśnięcie ikony odświeżenia.Nie istnieje możliwość zalogowania się jako „admin“. |
| Informacje o module | <ul style="list-style-type: none">Nie można filtrować danych.Nie można sortować pól. |
| Diagnostyka | <ul style="list-style-type: none">Wyswietlanie szczegółów zdarzeń: Użytkownik po wyłączeniu obsługi JavaScript musi nacisnąć hiperlink pola zdarzenia bufora diagnostycznego, aby mieć wgląd w dane zdarzenia wyświetlane na dole sekcji.Zmiana zakresu widocznych wpisów bufora diagnostycznego: Użytkownik po wyłączeniu obsługi JavaScript używa listy rozwijalnej znajdującej się na górze okna, aby wybrać zakres wyświetlanych wpisów bufora diagnostycznego. Użytkownik musi wówczas kliknąć w link „Go“, aby odświeżyć stronę bufora diagnostycznego z zakresem wybranym z listy rozwijalnej. |

| Standardowe strony Web | Efekt |
|------------------------|---|
| Zmienna | <ul style="list-style-type: none"> Po wpisaniu każdej zmiennej użytkownik musi ręcznie powiększyć wiersz <i>New variable</i> aby dodać nową zmienną. Wybranie formatu wyświetlanego nie powoduje automatyczne zmiany wartości danych wyświetlanych w określonym formacie. Należy wtedy kliknąć przycisk <i>Monitor value</i>, aby odświeżyć wyświetlanie w nowym formacie. |
| Dzienniki zdarzeń | <ul style="list-style-type: none"> Użytkownik nie może nacisnąć na nazwie pliku pod <i>Recent entries</i> (Ostatnie wpisy), aby zapisać lub otworzyć dziennik zdarzeń. Można jednak użyć ikony Download, która ma taką samą funkcjonalność. Strona Data Logs (Dziennik zdarzeń) nie odświeża się. Przyciski „+” oraz „–” służące zwiększeniu lub zmniejszeniu liczby zdarzeń pozostają nieaktywne. Użytkownik może opuścić stronę Data Log, a następnie wejść na nią ponownie, aby dostać ostatnie 25 wpisów. |

9.2.2. Funkcje ograniczone poprzez wyłączenie obsługi „ciasteczek”

Jeśli przeglądarka użytkownika ma wyłączoną obsługę „ciasteczek”, użytkownik nie może zalogować się pod nazwą „admin”.

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.

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 komplikowany 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 kompliuje wszystko ze stron HTML do obsługiwanych przez siebie bloków danych. Te bloki danych składają się z jednego sterującego bloku danych, który kieruje wyświetlanie 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 wyświetlały 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 załadunku 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.

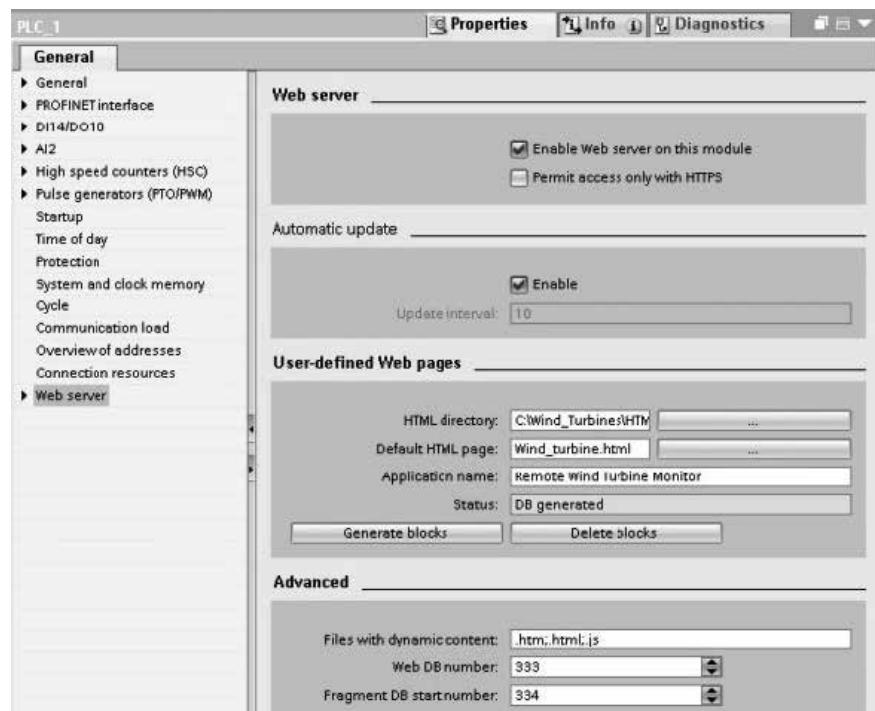
Użytkownik może sprawdzić rozmiar obszaru pamięci i dostępną, wolną pamięć za pomocą narzędzia Online and Diagnostic. Można także zobaczyć właściwości każdego pojedynczego bloku, który został przez STEP 7 wygenerowany ze zdefiniowanej przez użytkownika strony Web i określić jego rozmiar.

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:

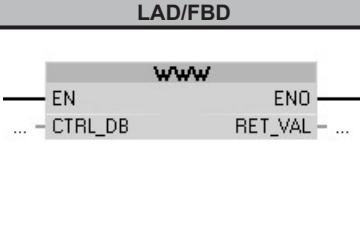
- Lokacja domyślnej strony startowej w celu utworzenia bloku DB dla stron Web zdefiniowanych przez użytkownika.
- Nazwę aplikacji (opcjonalnie). Nazwa aplikacji jest używana dla późniejszej organizacji lub grupowania stron Web. Jeśli stworzono nazwę aplikacji, adres URL powinien pojawić się w poniższej wersji:
`http://www.xx.yy.zz/awp/<nazwa aplikacji>/<nazwa strony>.html.`
- Rozszerzenia nazwy pliku w celu sprawdzenia obecności komend AWP. Domyslnie 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

Instrukcje WWW pozwalają stronom zdefiniowanym przez użytkownika na dostępność z poziomu standardowych stron Web. Program użytkownika wykonuje instrukcję WWW tylko raz, np. w rozruchowym OB. Strony zdefiniowane przez użytkownika mogą być jednak dostępne także w specjalnych okolicznościach. Program użytkownika może je wywoływać zgodnie z wymaganiami aplikacji.

Tabela 9.2. 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.

Łatwe sterowanie napędami

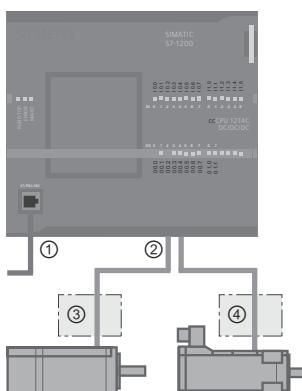
CPU posiada funkcje sterowania napędami, zarówno silników krokowych jak i serwonapędów z interfejsem impulsowym. Funkcje sterowania napędami sterują i nadzorują pracę napędów.

- 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 osiąmi i rozpoczęcia wykonania zadań dla napędu.

Uwaga

Zmiany wprowadzone w konfiguracji sterownika napędów i wprowadzone do pamięci CPU znajdującego się w trybie RUN nie zostaną uwzględnione do chwili przełączenia CPU z trybu STOP do 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.

Płytką 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 obsługiwać sterowanie impulsowe i sterowanie kierunkiem dla dwóch silników. Wbudowane wyjścia przekaźnikowe nie mogą zostać użyte w roli wyjść impulsowych w sterowaniu silnikami.

Uwaga

Wyjścia impulsowe nie mogą być używane przez inne instrukcje w programie użytkownika

Po skonfigurowaniu wyjść w CPU lub na płytce sygnałowej jako wyjścia impulsowe (w celu użycia instrukcji sterowania PWM lub innych podstawowych instrukcji sterowania ruchem), odpowiadające im adresy (Q0.0 do Q0.3, Q4.0 do Q4.3) są usuwane z pamięci Q i nie mogą być użyte do innych celów w programie użytkownika. Jeśli program użytkownika zapisuje wartości do wyjścia używanego jako generator impulsów, CPU nie przepisuje tych wartości do wyjść fizycznych.

Tabela 10.1. Największa możliwa liczba sterowanych napędów

| Typ CPU | | Bez zainstalowanej płytki SB | Z płytka SB (2 x wyjście DC) | Z płytka SB (4 x wyjście DC) |
|-----------|-----------|------------------------------|---------------------------------|---------------------------------|
| CPU 1211C | DC/DC/DC | 2 | 2 | 2 |
| | AC/DC/RLY | 0 | 1 | 2 |
| | DC/DC/RLY | 0 | 1 | 2 |
| CPU 1212C | DC/DC/DC | 2 | 2 | 2 |
| | AC/DC/RLY | 0 | 1 | 2 |
| | DC/DC/RLY | 0 | 1 | 2 |
| CPU 1214C | DC/DC/DC | 2 | 2 | 2 |
| | AC/DC/RLY | 0 | 1 | 2 |
| | DC/DC/RLY | 0 | 1 | 2 |
| CPU 1215C | DC/DC/DC | 4 | 4 | 4 |
| | AC/DC/RLY | 0 | 1 | 2 |
| | DC/DC/RLY | 0 | 1 | 2 |

Tabela 10.2. Zakres częstotliwości dla wyjść impulsowych

| Wyjście impulsowe | Częstotliwość |
|----------------------|---|
| Zintegrowane z CPU | 2 PTO: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ 2 PTO: $2 \text{ Hz} \leq f \leq 20 \text{ kHz}$ |
| Standardowa SB | $2 \text{ Hz} \leq f \leq 20 \text{ kHz}$ |
| Szybkie (200 kHz) SB | Instrukcje MC V2: $2 \text{ Hz} \leq f \leq 200 \text{ kHz}$ Instrukcje MC V1: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ ¹ |

¹ Instrukcje MC V1 obsługują maksymalną częstotliwość równą 100 kHz.

Uwaga

Maksymalna częstotliwość impulsowa dla generatorów wyjścia impulsowego wynosi 100 kHz (cyfrowe wyjścia CPU), 20 kHz (dla standardowej płytke sygnałowej SB) i 200 kHz (dla cyfrowych wyjść szybkiej SB) lub 100 kHz w przypadku użycia instrukcji MC V1.

Konfiguracja generatora impulsów

1. Dodaj obiekt mechaniczny:

- W drzewie projektu należy rozwinąć węzeł *Technological 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 wymagany sygnał wyjściowy. Zapisz dwa wyjścia przypisane do impulsów i kierunku.

Uwaga

Jeżeli PTO nie został wcześniej skonfigurowany we właściwościach CPU (CPU Properties), domyślnie są wykorzystywane lokalne wyjścia CPU.

Jeżeli w aplikacji jest wykorzystywany moduł sygnałowy należy wcisnąć przycisk „Device configuration”, co spowoduje przejście do opcji CPU. W opcji „Parametr assignment” należy skonfigurować „Pulse options” do obsługi modułu sygnałowego. Dla PTO na module sygnałowym dostępne są dwa wyjścia: „Pulse_1” i „Pulse_3”.

- 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

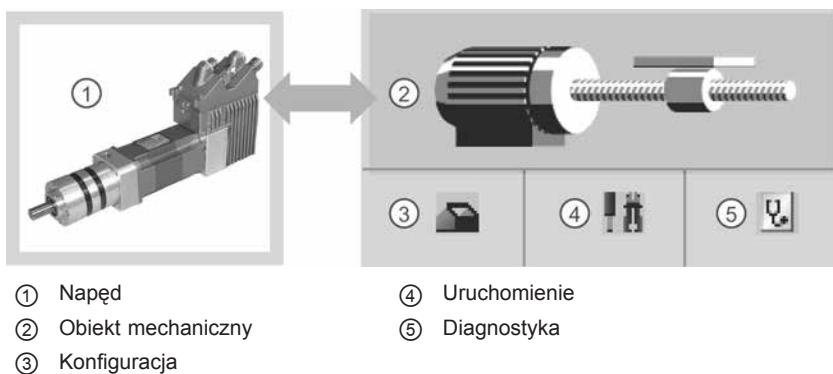
Konfiguracja generatora impulsów do współpracy z modelem sygnałowym: należy wybrać właściwości „Pulse generators (PTO/PWM)” i uruchomić generator impulsów. Dwa takie generatory są dostępne w każdym CPU S7-1200: V1.0, V2.0, V2.1 i V2.2. Wersja V3.0 CPU jest wyposażona w cztery generatory impulsów. W tym samym obszarze konfiguracji „Pulse options” należy wybrać pracę generatora impulsów jako „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 wykonywanie 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.

10.1. 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

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. Ograniczenie występuje w CPU S7-1200 w wersjach: V1.0, V2.0, V2.1 oraz V2.2. W wersji V3.0 ograniczenie nie występuje.

Użytkownik nie może monitorować bieżącej wartości (np. w ID 1000) kiedy generowane są脉sy.

Tabela 10.3. Narzędzia STEP 7 do sterowania ruchem

| Narzędzie | Opis |
|------------------------------|--|
| Configuration – 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. |
| Commissioning – uruchamianie | Sprawdza działanie osi bez konieczności tworzenia programu użytkownika. Podczas rozpoczęcia pracy narzędzia wyświetlny jest panel sterujący. Oferuje on następujące instrukcje: <ul style="list-style-type: none"> Załączanie i wyłączanie osi. Ruch osi w trybie jog – ręczne przesuwanie. Pozycja osi w układzie wzajemnym i bezwzajemnym. Oś bazowa. Potwierdzanie błędów. Prędkość i przyspieszenie/opóźnienie mogą być określone przez komendy napędu. Panel sterujący wyświetla także bieżący stan osi. |
| Diagnostics – diagnostyka | Monitoruje bieżący stan i informacje o błędzie dla osi i napędu. |



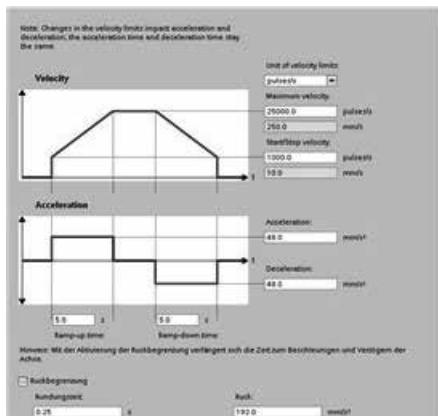
Po stworzeniu obiektu technologicznego, użytkownik konfiguruje osie poprzez definiowanie parametrów, takich jak PTO i definiowanie interfejsu napędu. Użytkownik musi skonfigurować także pozostałe parametry osi, takie jak graniczne położenie, dynamikę i bazowanie.

Uwaga

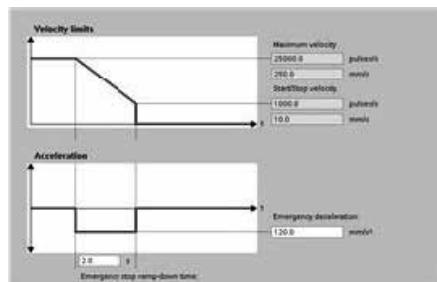
Użytkownik może przeprowadzić adaptację wartości parametrów wejściowych do nowych wymiarów systemu z poziomu programu.



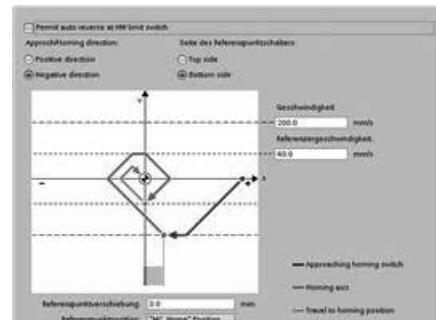
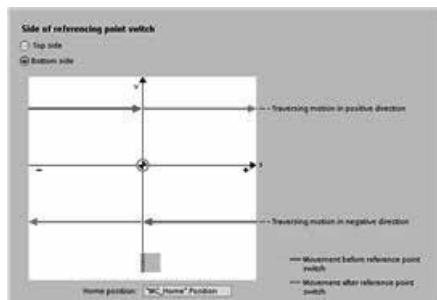
Konfiguracja właściwości sygnałów z napędu i nadzorowania pozycji (sprzętowe i programowe krańcówki).



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.

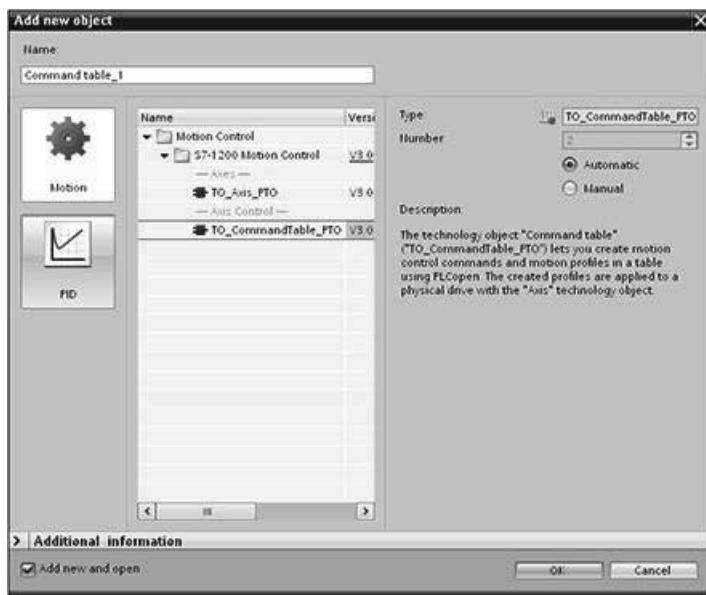
Panel sterowania wyświetla bieżący stan osi. Użytkownik poza załączeniem i wyłączeniem osi może także sprawdzać pozycjonowanie osi (zarówno wartość bezwzględną jak i względną) oraz określić jej prędkość, przyspieszenie lub opóźnienie. Panel sterowania pozwala także na potwierdzenie błędów.

10.2. Konfiguracja TO_CommandTable_PTO

Konfigurowanie instrukcji CommandTable jest możliwe za pomocą obiektów technologicznych (*Technological objects*).

Dodawanie obiektów technologicznych

1. W drzewie projektu rozwiń węzeł *Technological Objects* i wybierz *Add new object*.
2. Wybierz ikonę *CommandTable* i kliknij *OK* żeby otworzyć edytor konfiguracji obiektu CommandTable.



Tworzenie sekwencji aplikacji

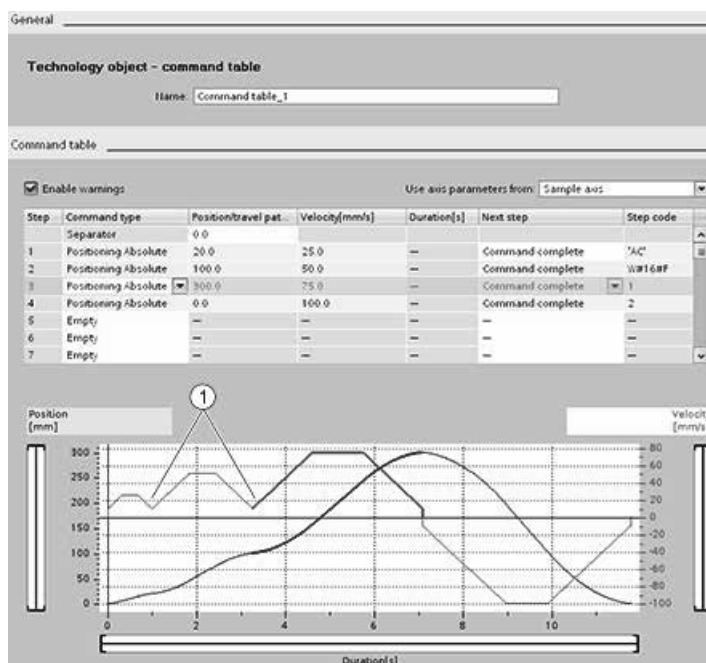
Użytkownik może w oknie *CommandTable* utworzyć pożądaną sekwencję ruchów i zweryfikować ją na wykresie za pomocą diagramu trendów.

Typy poleceń użytych w projektowanym procesie mogą zostać przez użytkownika dowolnie wybrane. Maksymalnie można zadać 32 kroki, które są odtwarzane sekwencyjnie, pozwalając w łatwy sposób zaprojektować złożone profile ruchu.

Tabela 10.4. Typy poleceń MC_CommandTable

| Typ polecenia | Opis |
|-----------------------------|--|
| <i>Empty</i> | Polecenie ignorowane podczas realizacji sekwencji zapisanej w tabeli ruchów (<i>command table</i>). |
| <i>Halt</i> | Zatrzymanie napędu. Uwaga: stosowane wyłącznie po poleceniu <i>Velocity setpoint</i> . |
| <i>Positioning Relative</i> | Przesunięcie bazujące na odległości. Polecenie powoduje przesunięcie na zadaną odległość zadaną prędkością. |
| <i>Positioning Absolute</i> | Przesunięcie bazujące na położeniu. Polecenie powoduje przesunięcie do zadanego położenia zadaną prędkością. |
| <i>Velocity setpoint</i> | Ustalenie prędkości przesunięcia. |
| <i>Wait</i> | Wstrzymanie ruchu na zadany czas. Polecenie nie zatrzymuje silników. |
| <i>Separator</i> | Polecenie pozwalające zapisać więcej niż jeden profil ruchu w pojedynczej tabeli. |

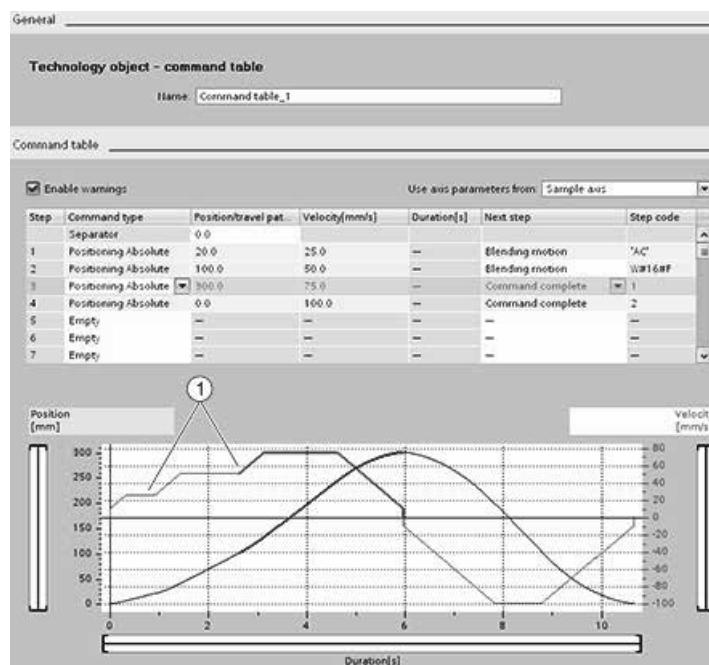
Na poniższym rysunku *Command complete* spełnia rolę przejścia do kolejnego kroku. Ten typ przejścia umożliwia silnikowi zmniejszenie prędkości obrotowej pod koniec sekwencji i jej zwiększenie na początku kolejnego kroku.



- ① Zmniejszenie prędkości pomiędzy kolejnymi krokami do wartości zadanej dla start/stop.

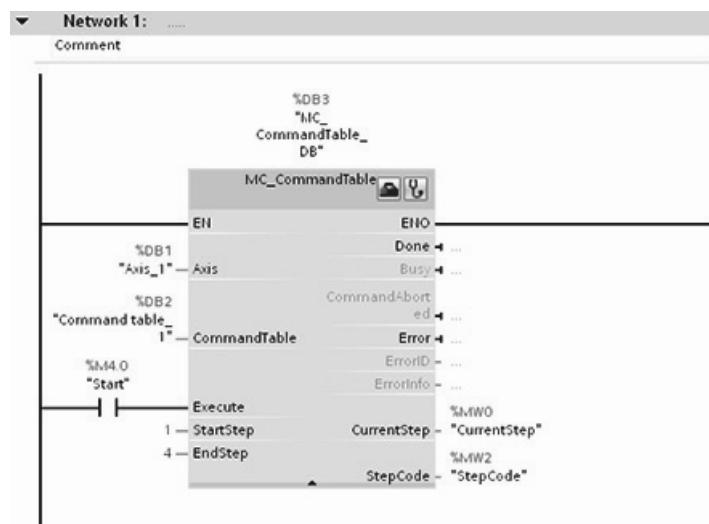
Na poniższym rysunku *Blending motor* wartość jest wykorzystywana do przejścia do następnego kroku. Ten typ przejścia zapewnia płynne przejścia pomiędzy krokami, co jest uzyskiwane dzięki utrzymywaniu prędkości silnika przed kolejnym krokiem. Plynne przejścia pomiędzy krokami zmniejszają czas wykonania całej sekwencji. Sekwencja ruchów w prezentowanym przykładzie jest wykonywana

w czasie ok. 6 sekund, bez wykorzystania wspomnianego mechanizmu czas wykonania sekwencji przekracza 7 sekund.



- ① Napęd utrzymuje prędkość pomiędzy kolejnymi krokami, co zapewnia oszczędność czasu i zmniejszenie zużycia maszyn.

Na rysunku poniżej pokazano tabelę poleceń (*command table*) zarządzaną za pomocą polecenia MC_CommandTable:



10.3. Instrukcja MC_Power

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 10.5. Instrukcja MC_Power

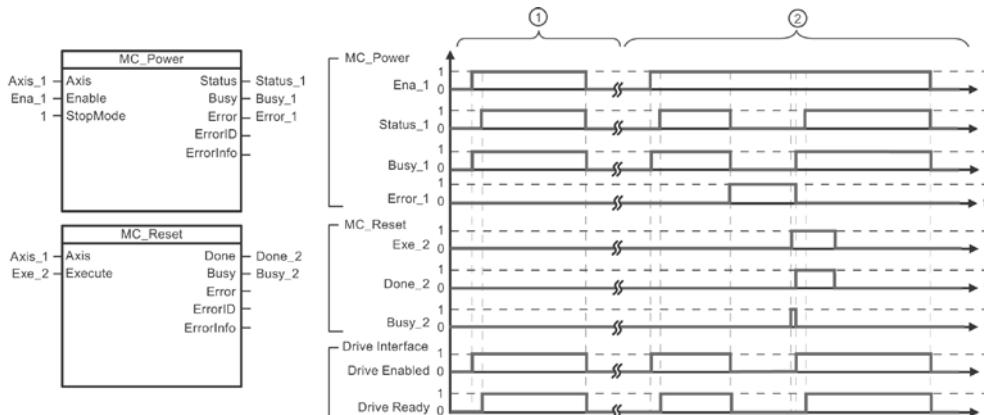
| LAD/FBD | SCL | Opis |
|---|--|--|
| <pre>"MC_Power_DB" MC_Power EN ENDO Axis Status Enable Busy StopMode Error Status=>_bool_out_ Busy=>_bool_out_ Error=>_bool_out_ ErrorID=>_word_out_ ErrorInfo=>_word_out_;</pre> | <pre>"MC_Power_DB"(Axis:=_multi_fb_in_, Enable:=_bool_in_, StopMode:=_int_in_, Status=>_bool_out_, 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 wywołalby 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 nazwa „MC_Power_DB” odnosi się do instancji DB.

Tabela 10.6. 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 z ograniczeniem szarpnięć: jeżeli blokada osi jest aktywna, hamulce zmniejszają prędkość jej ruchu do zadanego poziomu. Jeżeli ogranicznik szarpnięć jest aktywny, podczas hamowania jest uwzględniona jego konfiguracja. Oś jest blokowana od razu po zatrzymaniu. |
| 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 | FALSE: MC_Power nie jest aktywne. TRUE: MC Power jest aktywne. |
| Error | OUT | Bool | <p>FALSE: Brak błędu.</p> <p>TRUE: Wystąpił błąd w instrukcji sterowania napędem („MC_Power”) lub w związanym 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 zasygnalizowaniu 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 zwierdzony 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. Doprzedzić 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.

10.4. Instrukcja MC_Reset

Tabela 10.7. Instrukcja MC_Reset

| LAD/FBD | SCL | Opis |
|---|---|--|
| <pre>"MC_Reset_DB" MC_Reset [EN] [Axis] [Execute] [ENO] [Done] [Busy] [Error] [ErrorCode] [ErrorInfo]</pre> | <pre>"MC_Reset_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Restart:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, ErrorCode=>_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>W wersji V3.0 i nowszych polecenie Restart umożliwia załadowanie nowej konfiguracji napędu do pamięci roboczej modułu CPU znajdującego się w trybie RUN.</p> |

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL nazwa „MC_Reset_DB” odnosi się do instancji DB.

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 10.8. Parametry instrukcji MC_Reset

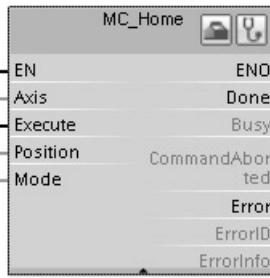
| Parametr i jego typ | Typ danych | Opis |
|---------------------|------------|--|
| Axis – oś | IN | TO_Axis_1 Obiekt mechaniczny |
| Execute – wykonanie | IN | Bool Rozpoczęcie zadania po wystąpieniu zbocza narastającego. |
| Restart | IN | Bool TRUE = ładowanie konfiguracji ruchu osi do pamięci roboczej. Polecenie może być wykonane tylko wtedy, gdy os robocza jest zablokowana. FALSE = Oczekiwanie na wykryte błędy. |
| 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. Instrukcja MC_Home

Tabela 10.9. Instrukcja MC_Home

| LAD/FBD | SCL | Opis |
|--|---|---|
|  <p>"MC_Home_DB"</p> <pre> MC_Home +-----+ EN ENO Axis Done Execute Busy Position CommandAborted Mode Error ErrorID ErrorInfo </pre> | <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. Instrukcja MC_Home może być użyta tylko w wypadku, gdy osi została wcześniej włączona.</p> |

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

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 10.10. Parametry instrukcji MC_Home

| Parametr i jego typ | Typ danych | Opis |
|---------------------|------------|---|
| Axis – osi | 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: $-1.0e^{12} \leq \text{Pozycje} \leq 1.0e^{12}$ |

| Parametr i jego typ | | Typ danych | Opis |
|-------------------------------------|-----|------------|--|
| Mode – tryb | IN | Int | <p>Tryb bazowania:</p> <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 ustawia się na TRUE i zmienna obiektu technologicznego <Nazwa osi>. StatusBits.HomingDone także będzie miała wartość TRUE, to proces bazowania został zakończony.

Tabela 10.11. 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 |

Bazowanie odnosi się do dopasowania osi współrzędnych do rzeczywistej (fizycznej) pozycji napędu (jeśli napęd jest w danej chwili w pozycji x, osie zostaną dopasowane do bycia w pozycji x). Dla osi sterowanych pozycyjnie, wpisy i wyświetlenia pozycji odnoszą się dokładnie do tych współrzędnych osi.

Uwaga

Zgodność pomiędzy współrzędnymi osiowymi i stanem rzeczywistym jest niezwykle ważna. Niezbędne jest uzyskanie pewności o tym, że bezwzględna pozycja docelowa osi jest osiągnięta także przez napęd.

Instrukcja MC_Home inicjuje bazowanie osi.

Istnieją 4 różne funkcje bazowania. Pierwsze dwie pozwalają użytkownikowi na ustawienie bieżącej pozycji osi, a dwie kolejne ustalają pozycję osi z uwzględnieniem czujnika odniesienia bazowania (*Home Reference Sensor*).

- Tryb 0 – Bezwzględne odniesienie bezpośrednie (*Direct Referencing Absolute*): Podczas pracy w tym trybie osie wskazują swoje dokładne położenie. Zmienna obsługująca pozycję ma ustawianą wartość zgodnie z wejściem *Position* instrukcji bazowania. Można użyć tej własności do kalibracji i ustawień. Pozycja osi jest ustawiana niezależnie od krańcówki punktu odniesienia. Przesuw nie jest przerwany. Wartość parametru *Position* instrukcji MC_Home ustawiany jest jako punkt odniesienia osi (*reference point of the axis*). Aby przypisać punkt odniesienia do dokładnej pozycji mechanicznej, oś musi zostać w tej pozycji nieruchomo podczas wykonywania instrukcji bazowania.
- Tryb 1 – Względne odniesienie bezpośrednie (*Direct Referencing Relative*): Podczas pracy w tym trybie używana jest wewnętrzna zmienna pozycji, do

której dodana jest wartość wejścia *Position* instrukcji bazowania. Jest to zwykle wykorzystywane dla przesunięcia maszynowego (offsetu).

Pozycja osi jest ustawiana niezależnie od krańcówki punktu odniesienia. Przesuw nie jest przerywany. Następujące wyrażenie określa pozycję osi po bazowaniu:

Nowa pozycja osi = bieżąca pozycja osi + wartość parametru *Position* z instrukcji MC_Home.

- Tryb 2 – Pasywne odniesienie (*Passive Referencing*): Kiedy oś jest w ruchu i przechodzi przez krańcówkę punktu odniesienia, to bieżąca pozycja jest ustawiana jako pozycja bazowania. Ta funkcja jest pomocna w przypadku wystąpienia normalnego zużycia maszyn i luzów przekładni. W konsekwencji pozwala ominąć konieczność ręcznej kompensacji zużycia. Wejście *Position* instrukcji bazowania, tak jak poprzednio, jest dodawane do pozycji wskazanej przez krańcówkę punktu odniesienia, co ułatwia stworzenie offsetu pozycji bazowania.

Podczas pasywnego bazowania, instrukcja MC_Home nie przeprowadza żadnego ruchu bazującego. Przesuw wymagany dla tego kroku musi zostać zaimplementowany przez użytkownika za pomocą innych instrukcji sterowania ruchem. Gdy krańcówka punktu odniesienia zostanie wykryta, oś jest naprowadzana zgodnie z konfiguracją. Przesuw nie jest zatrzymywany podczas pasywnego bazowania.

- Tryb 3 – Aktywne odniesienie (*Active Referencing*): tryb ten jest najdokładniejszą metodą bazowania osi. Wstępny kierunek i prędkość ruchu zostają skonfigurowane w rozszerzonych parametrach konfiguracji technologicznego obiektu (*Technology Object Configuration Extended Parameters-Homing*). Zależy to od konfiguracji maszyny. Istnieje także możliwość określenia, czy sygnałem krańcówki punktu odniesienia jest zbocze narastające, czy też opadające. Praktycznie wszystkie czujniki mają różny aktywny obszar działania i jeżeli pozycja *Steady State On* została użyta jako sygnał bazowania, to mógłby powstać związek właśnie z różnym obszarem działania czujników. Przez użycie zbocza opadającego lub narastającego sygnału można uzyskać znacznie bardziej precyzyjną pozycję bazy. Tak jak w pozostałych trybach bazowania, wartość parametru *Position* instrukcji bazowania dodawana jest do pozycji odniesionej sprzętowo.

W trybie aktywnego bazowania, instrukcja MC_Home przeprowadza wymagane dojście do punktu referencyjnego. Gdy krańcówka punktu odniesienia zostanie wykryta, oś jest bazowana zgodnie z konfiguracją. Przesuw nie jest zatrzymywany.

Tryby 0 i 1 nie wymagają w ogóle, aby oś się poruszała. Zwykle służą one kalibracji i parametryzacji. Tryby 2 i 3 wymagają poruszania się osi i miniecia czujnika, który skonfigurowany jest jako krańcówka punktu odniesienia w mechaniczny. Punkt odniesienia może być umieszczony w obszarze roboczym osi lub poza normalnym obszarem roboczym, ale w zakresie ruchu.

10.6. Instrukcja MC_Halt

Tabela 10.12. Instrukcja MC_Halt

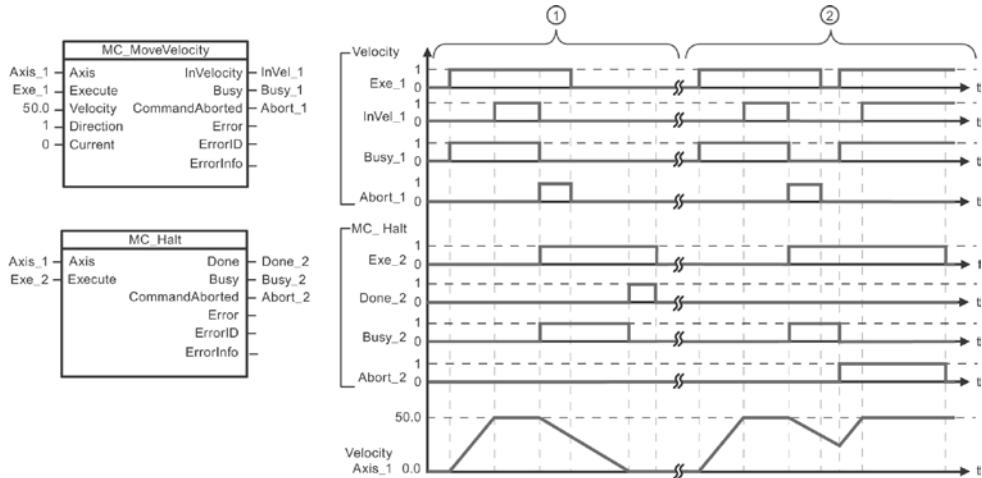
| LAD/FBD | SCL | Opis |
|---|---|--|
|  <p>"MC_Halt_DB"</p> <pre> MC_Halt [EN] [Axis] [Execute] [ENO] [Done] [Busy] [CommandAborted] [Error] [ErrorID] [ErrorInfo] </pre> | <pre> "MC_Halt_DB"(Axis:=multi_fb_in_, Execute:=bool_in_, Done=>bool_out_, Busy=>bool_out_, CommandAborted=>bool_out_, Error=>bool_out_, 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 oś została wcześniej załączona.</p> |

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL nazwa „MC_Halt_DB” odnosi się do instancji DB.

Tabela 10.13. Parametry instrukcji MC_Halt

| Parametr i jego typ | Typ danych | Opis |
|---------------------|------------|---|
| Axis – oś | IN | Obiekt mechaniczny. |
| Execute – wykonanie | IN | Rozpoczęcie zadania po wystąpieniu zbocza narastającego. |
| Done – wykonane | OUT | TRUE = Osiągnięto prędkość 0. |
| Busy – zajęte | OUT | TRUE = Zadanie jest wykonywane. |
| CommandAborted | OUT | TRUE = Wykonanie zadania zostało przerwane przez inne zadanie. |
| Error – błąd | OUT | TRUE = Wystąpił błąd podczas wykonywania zadania. Przyczyna wystąpienia błędu może zostać znaleziona w parametrach „ErrorID” i „ErrorInfo”. |
| ErrorID | OUT | ID błędu dla parametru „Error”. |
| ErrorInfo | OUT | Informacje o ID błędu dla parametru „ErrorID”. |



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

10.7. Instrukcja MC_MoveAbsolute

Tabela 10.14. Instrukcja MC_MoveAbsolute

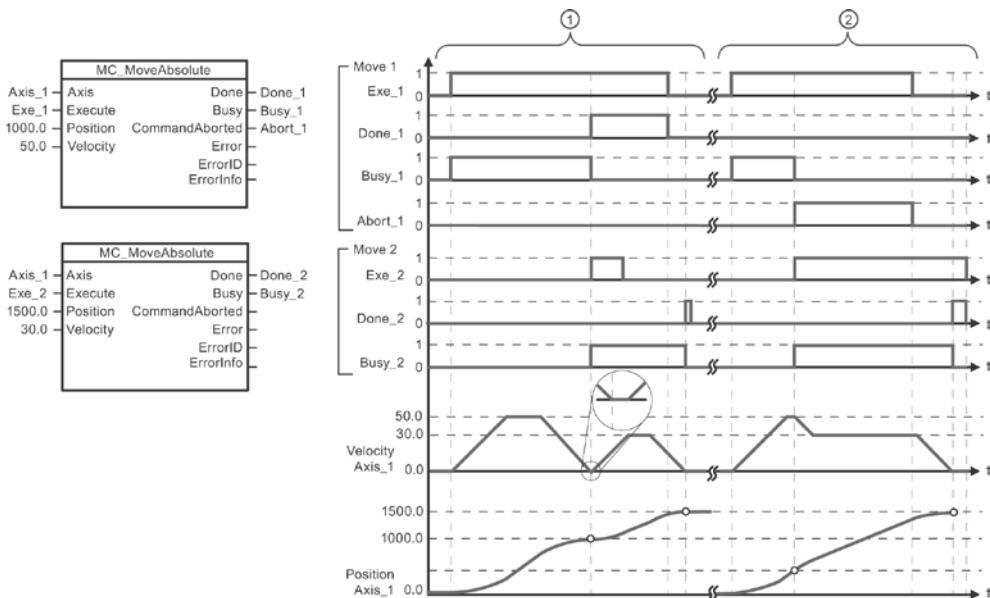
| LAD/FBD | SCL | Opis |
|---|---|---|
| <pre>"MC_MoveAbsolute_DB" MC_MoveAbsolute(Axis:= _multi_fb_in_, Execute:= _bool_in_, Position:= _real_in_, Velocity:= _real_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre> | <pre>"MC_MoveAbsolute_DB" MC_MoveAbsolute(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 rozpoczęta 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 nazwa „MC_MoveAbsolute_DB” odnosi się do instancji DB.

Tabela 10.15. Parametry instrukcji MC_MoveAbsolute

| 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 domyślana: FALSE). |
| Position – pozycja | IN | Real Bezwzględna pozycja docelowa (Wartość domyślana: 0.0). Graniczne wartości: $-1,0e^{12} \leq \text{Pozycja} \leq 1,0e^{12}$. |
| Velocity – prędkość | IN | Real Prędkość osi (Wartość domyślana: 10.0). Prędkość nie zawsze jest osiągana ze względu na dobrane wartości przyspieszeń i opóźnień oraz drogi dojścia do zadanej pozycji. Graniczne wartości: Prędkość Start/Stop \leq Prędkość \leq maksymalna Prędkość. |
| 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 domyślana: 0000). |
| ErrorInfo | OUT | Word Informacje o ID błędu dla parametru „ErrorID” (Wartość domyślana: 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 przerwane 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:

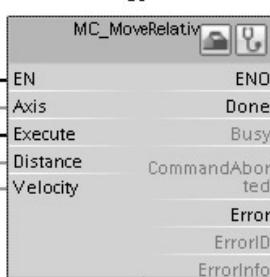
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

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

10.8. Instrukcja MC_MoveRelative

Tabela 10.16. Instrukcja MC_MoveRelative

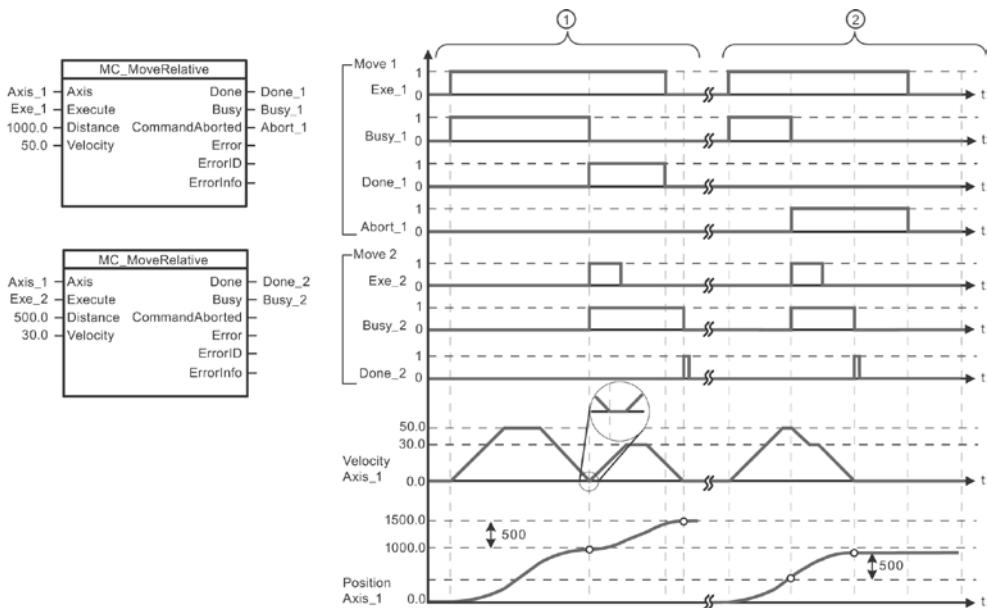
| LAD/FBD | SCL | Opis |
|--|--|---|
|  <p>"MC_MoveRelative_DB" MC_MoveRelative EN Axis Execute Distance Velocity ENO Done Busy CommandAborted Error ErrorID ErrorInfo</p> | <pre>,,MC_MoveRelative_DB"(Axis:= _multi_fb_in_, Execute:= _bool_in_, Distance:= _real_in_, Velocity:= _real_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre> | <p>Instrukcja MC_MoveRelative rozpoczęta ruch pozycjonujący względem pozycji startowej.</p> <p>Instrukcja MC_MoveRelative może być użyta tylko w wypadku, gdy oś została wcześniej załączona oraz naprowadzona.</p> |

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL „MC_MoveRelative_DB” jest nazwą instancji DB.

Tabela 10.17. 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 domyślana: FALSE). |
| Position - pozycja | IN | Real | Długość przesunięcia dla operacji pozycjonowania (Wartość domyślana: 0.0) Graniczne wartości: $-1,0e^{12} \leq \text{Pozycja} \leq 1,0e^{12}$. |
| Velocity - prędkość | IN | Real | Prędkość osi (Wartość domyślana: 10.0) Ta prędkość nie, zawsze jest osiągana ze względu na dobrane wartości przyspieszeń i opóźnień oraz drogi dojścia do zadanej pozycji. Graniczne wartości: Prędkość Start/stop \leq Prędkość \leq maksymalna prędkość |
| 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 domyślana: 0000) |
| ErrorInfo | OUT | Word | Informacje o ID błędu dla parametru „ErrorID” (Wartość domyślana: 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 przerwane 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

10.9. Instrukcja MC_MoveVelocity

Tabela 10.18. 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> | <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> |

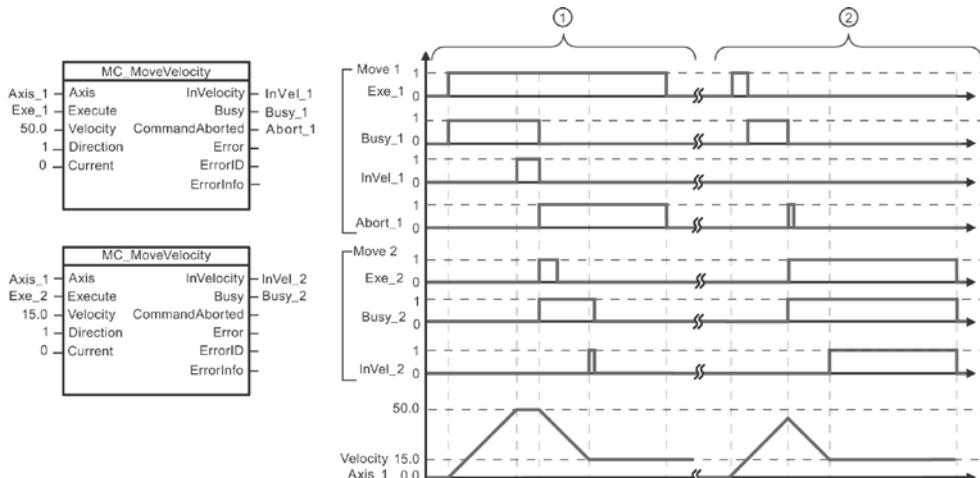
¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL nazwa „MC_MoveVelocity_DB” odnosi się do instancji DB.

Tabela 10.19. 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). |
| 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. |

| Parametr i jego typ | Typ danych | Opis |
|---------------------|------------|--|
| 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)

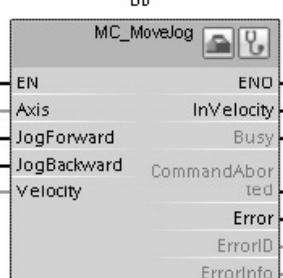
Zadanie MC_MoveVelocity z ustawionym parametrem „Velocity” = 0,0 (tak jak zadanie MC_Halt) przerwa aktywne zadanie sterowania napędem i hamuje os z nastawnionym opóźnieniem. Po zatrzymaniu osi, parametr wyjściowy „InVelocity” zostaje ustawiony na TRUE na przynajmniej jeden cykl programu.

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 podtrzymywane.

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.

10.10. Instrukcja MC_MoveJog

Tabela 10.20. 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 osią ze stałą określona prędkością w trybie Jog. Instrukcja ta jest szczególnie przydatna podczas testowania i rozruchu. Instrukcja MC_MoveJog może być użyta tylko w wypadku, gdy os została wcześniej załączona.</p> |

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

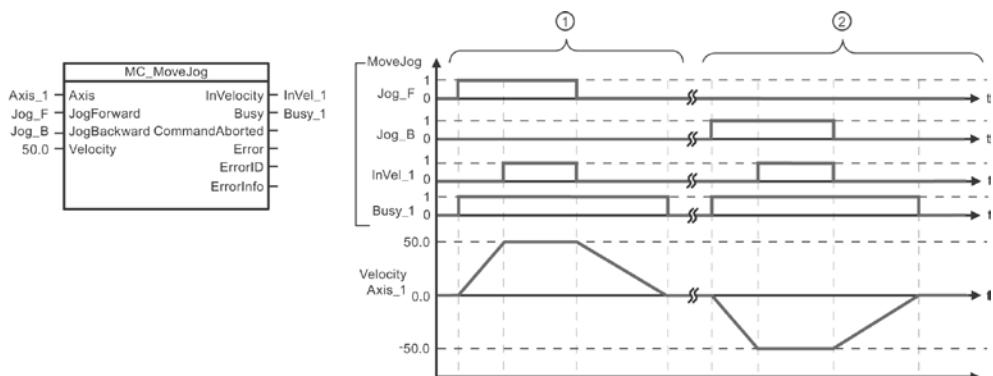
² W przykładzie SCL nazwa „MC_MoveJog_DB” odnosi się do instancji DB.

Tabela 10.21. 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, os 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, os porusza się w kierunku ujemnym z prędkością określoną w parametrze „Velocity”. Znak wartości parametru „Velocity” jest ignorowany (wartość domyślna: False). |

| Parametr i jego typ | | Typ danych | Opis |
|---------------------|-----|------------|---|
| Velocity | IN | Real | Ustawiona wartość prędkości dla trybu jog (Domyślna wartość: 10.0). Graniczne wartości: Prędkość Start/stop \leq Prędkość \leq 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 domyślana: 0000) |
| ErrorInfo | OUT | Word | Informacje o ID błędu dla parametru „ErrorID” (Wartość domyślana: 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.

- ② 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:

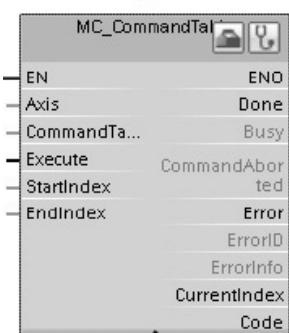
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

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

10.11. Instrukcja MC_CommandTable

Tabela 10.22. 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.

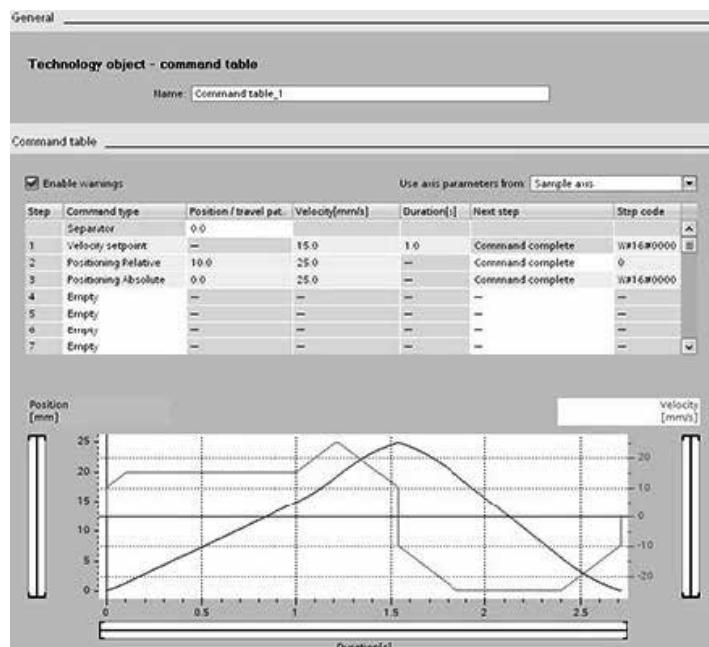
² W przykładzie SCL nazwa „MC_CommandTable_DB” odnosi się do instancji DB.

Tabela 10.23. Parametry instrukcji MC_CommandTable

| Parametr i jego typ | Typ danych | Wartość początkowa | Opis |
|---------------------|------------|--------------------|---|
| Axis | IN | TO_Axis_1 | – Obiekt 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. |

Sekwencję ruchów można utworzyć za okna konfiguracji pomocą tabeli „Command Table”, jej weryfikację umożliwia podgląd graficzny wyświetlony w oknie terndów.



Użytkownik może wybrać rodzaje poleceń wykorzystywanych w programowaniu sekwencji ruchów. Można zdefiniować do 32 zadań. Polecenia są wykonywane sekwencyjnie.

Tab. 10.24. Rodzaje poleceń MC_CommandTable

| Rodzaj polecenia | Opis |
|-----------------------------|---|
| <i>Empty</i> | Spełnia rolę instrukcji pustej, jest ignorowana podczas wykonywania sekwencji zapisanej w tabeli polecień. |
| <i>Halt</i> | Zatrzymanie osi. Uwaga: polecenie można użyć wyłącznie po poleceniu <i>Velocity setpoint</i> . |
| <i>Positioning Relative</i> | Pozycjonuje oś na określona odległość. Polecenie przesuwa oś roboczą do zadanego położenia z zadaną prędkością. |
| <i>Positioning Absolute</i> | Pozycjonuje oś do określonego położenia. Oś jest przesuwana z zadaną prędkością do określonego położenia. |
| <i>Velocity setpoint</i> | Przesuwa oś z określona prędkością. |
| <i>Wait</i> | Wstrzymanie ruchu na określony czas. Polecenie to nie wstrzymuje ruchów wykonywanych w danej chwili. |
| <i>Separator</i> | Dodaje linię separującą powyżej linii bieżącej. Polecenie pozwala zapisać więcej niż jedną sekwencję ruchów w pojedynczej tabeli. |

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
- Bieżące zadanie sterowania z załączeniem pierwszych komend *Positioning Relative*, *Positioning Absolute*, *Velocity set point* lub *Halt*.

10.12. Instrukcja MC_ChangeDynamic

Tabela 10.25. MC_ChangeDynamic

| LAD/FBD | SCL | Opis |
|--|--|---|
|  <pre>"MC_ChangeDynamic_DB" . "MC_ChangeDynamic"(Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre> | <pre>"MC_ChangeDynamic_DB" . "MC_ChangeDynamic"(Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre> | <p>Zmienia dynamiczne ustawienia sterowania ruchem osi:</p> <ul style="list-style-type: none"> • Zmiana czasu przyspieszenia (czasu <i>ramp-up</i>). • Zmiana czasu hamowania (czasu <i>ramp-down</i>). • Zmiana czasu hamowania w trybie zatrzymania alarmowego (czasu <i>emergency ramp-down</i>). • Zmiana czasu niwelacji szarpięć. |

¹ STEP 7 po wstawieniu instrukcji automatycznie tworzy blok DB.

² W przykładzie SCL nazwa „MC_ChangeDynamic_DB” odnosi się do instancji DB.

Tabela 10.26. Parametry instrukcji MC_ChangeDynamic

| Parametr i jego typ | Typ danej | Opis |
|---------------------|-----------|---|
| Axis | IN | TO_Axis_1 Oś obiektu technologicznego |
| Execute | IN | Bool Inicjacja wykonania polecenia za pomocą zbocza narastającego. Wartość domyślana: FALSE |
| ChangeRampUp | IN | Bool TRUE = zmiana czasu przyspieszania podanego jak parametr „RampUpTime”. Wartość domyślana: FALSE |
| RampUpTime | IN | Real Czas przyspieszania (w sekundach) od zatrzymania do maksymalnej zadanej prędkości bez niwelacji szarpięć. Wartość domyślana: 5.00. Zmiana wartości wpływa na znacznik <Axis name>. Config.DynamicDefaults.Acceleration. Skutek wprowadzonych zmian jest sygnalizowany w opisie znacznika. |
| ChangeRampDown | IN | Bool TRUE = zmiana czasu hamowania podanego jak parametr „RampDownTime”. Wartość domyślana: FALSE |
| RampDownTime | IN | Real Czas hamowania (w sekundach) bez niwelacji szarpięć. Wartość domyślana: 5.00. Zmiana wartości wpływa na znacznik <Axis name>. Config.DynamicDefaults.Acceleration. Skutek wprowadzonych zmian jest sygnalizowany w opisie znacznika. |

| Parametr i jego typ | | Typ danej | Opis |
|---------------------|-----|-----------|--|
| ChangeEmergency | IN | Bool | TRUE = zmiana czasu hamowania w trybie awaryjnym podanego jako parametr „EmergencyRampTime”. Wartość domyślna: FALSE |
| EmergencyRampTime | IN | Real | Czas hamowania (w sekundach) do zatrzymania od zadanej prędkości maksymalnej bez niwelacji szarpnięć. Wartość domyślana: 2.00. Zmiana wartości wpływa na znacznik <Axis name>. Config.DynamicDefaults.Acceleration. Skutek wprowadzonych zmian jest sygnalizowany w opisie znacznika. |
| ChangeJerkTime | IN | Bool | TRUE = zmiana czasu niwelacji szarpień podanego jako parametr „JerkTime”. Wartość domyślna: FALSE |
| JerkTime | IN | Real | Czas niwelacji szarpień (w sekundach) wykorzystywany podczas przyspieszania i hamowania. Wartość domyślana: 0.25. Zmiana wartości wpływa na znacznik <Axis name>. Config.DynamicDefaults.Acceleration. Skutek wprowadzonych zmian jest sygnalizowany w opisie znacznika. |
| Done | OUT | Bool | TRUE = nowa wartość została zapisana w pamięci sterownika w bloku nastaw technologicznych. Opis znacznika jest widoczny, gdy wprowadzone zmiany zostaną uwzględnione. Wartość domyślna: FALSE |
| Error | OUT | Bool | TRUE = błąd podczas wykonywania polecenia. Przyczyna błędów jest identyfikowana przez parametry „ErrorID” oraz „ErrorInfo”. Wartość domyślna: FALSE |
| ErrorID | OUT | Bool | Identyfikator błędu. Wartość domyślna: 16#0000 |
| ErrorInfo | IN | Word | Informacja 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

Wykonywanie polecenia MC_ChangeDynamic nie może być przerwane przez inne polecenia sterowania napędem.

Polecenie MC_ChangeDynamic nie przerywa wykonywania innych poleceń sterowania napędami.

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 mieścią się w zakresie ograniczeń konfiguracji obiektu mechanicznego.

Łatwe w użyciu narzędzia online

11

11.1. Tryb online i połączenie z CPU

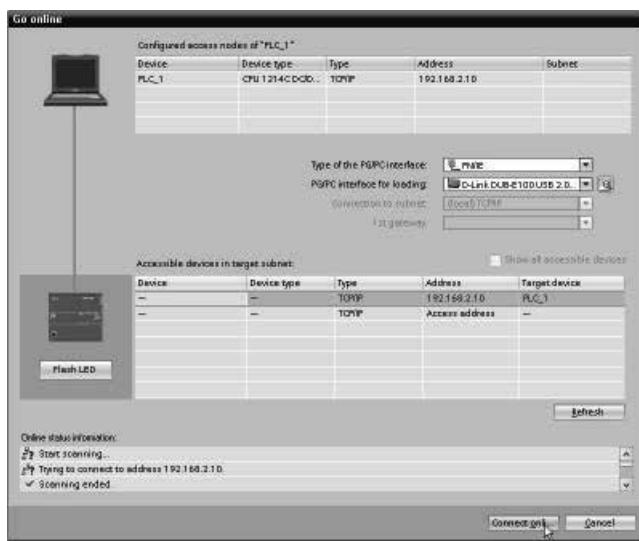
Połączenie online pomiędzy urządzeniem programującym i CPU jest wymagane do załadowania programów i danych projektów inżynierskich.

Połączenie online pozwala na następujące operacje:

- Testowanie programów użytkownika.
- Wyświetlanie i zmianę trybu pracy CPU (strona 215).
- Wyświetlanie i ustawienie daty i czasu dnia dla CPU (strona 224).
- Wyświetlanie informacji o modułach.
- Porównanie i synchronizacja bloków programów online i offline (strona 223).
- Zgrywanie i wgrywanie bloków programu.
- Wyświetlenie danych diagnostycznych i bufora diagnostycznego (strona 224).
- Użycie tablicy monitorującej (*Watch table*, strona 217) do sprawdzenia programu użytkownika przez sprawdzenie i modyfikację wartości.
- Użycie tabeli wymuszeń do ustawiania wartości w CPU (strona 218).

Podłączenie do określonego CPU, zanajdującego się w rozbudowanej sieci, następuje po wybraniu z drzewa nawigacji projektu (*Project Navigation*) odpowiedniego CPU i kliknięciu *Go online* w widoku projektu.

 Go online



Jeśli CPU jest podłączane 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*.

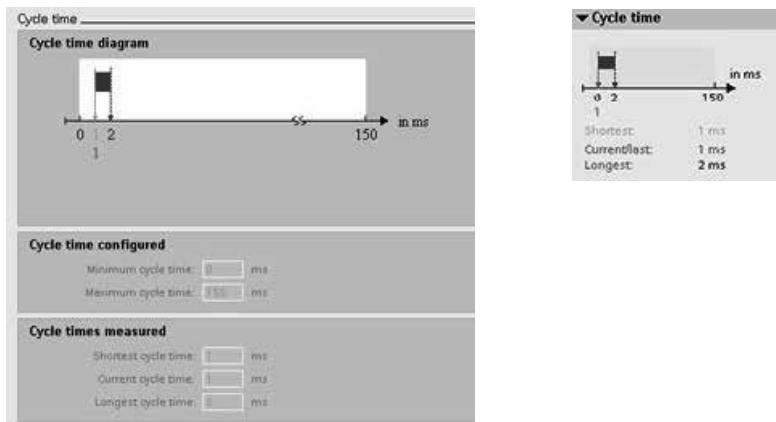
11.2. Interakcja z połączonym CPU

Widok *online* i diagnostyczny zawiera panel operatorski do zmiany trybu pracy CPU. Karta *Online tools* wyświetla panel operatorski, który pokazuje tryb pracy CPU i pozwala na zmianę jego typu pracy. Należy użyć przycisku na panelu operatorskim do zmiany trybu pracy CPU (RUN lub STOP). Ma on także przycisk MRES, który kasuje pamięć.

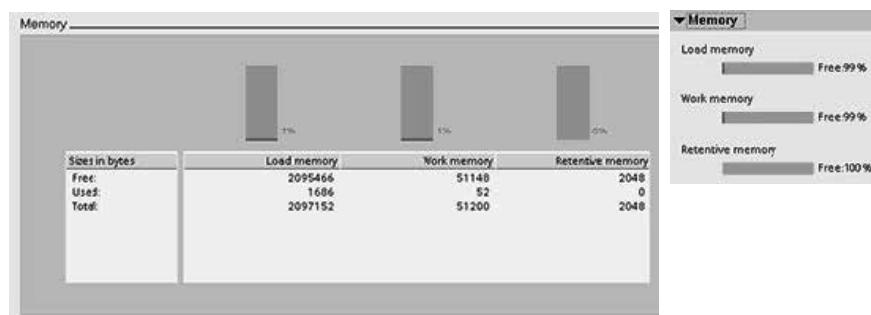


Kolor wskaźnika RUN/STOP pokazuje tryb pracy CPU: żółty wskazuje tryb STOP, zielony wskazuje tryb RUN.

Należy być podłączonym do CPU, aby używać panelu operatorskiego. Po wybraniu CPU z konfiguracji urządzenia lub wyświetleniu bloku kodu z połączonym CPU, użytkownik może wyświetlić panel operatorski z karty zadań *Online tools*.



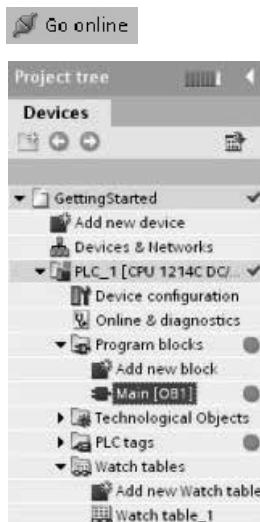
Można monitorować czas cyklu dla połączonego CPU.



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 tą samą konfigurację i program użytkownika.

Tablica zmiennych pokazuje listę tagów/zmiennych. Mogą być one pokazywane także przez tablice monitorującą, także jako bezpośrednie adresy.

| GettingStarted_1 > PLC_1 > Watch tables > Watch table_1 | | | | | |
|---|-------|---------|----------------|---------------|--------------|
| | Name | Address | Display format | Monitor value | Modify value |
| 1 | "On" | %I0.0 | Bool | | |
| 2 | "Off" | %I0.1 | Bool | | |
| 3 | "Run" | %Q0.0 | Bool | | |

Należy nacisnąć w pasku narzędzi przycisk *Monitor all*, aby nadzorować wykonanie programu użytkownika i wyświetlić wartości zmiennych.

| GettingStarted_1 > PLC_1 > Watch tables > Watch table_1 | | | | | |
|---|-------|---------|----------------|---|--------------|
| | Name | Address | Display format | Monitor value | Modify value |
| 1 | "On" | %I0.0 | Bool | <input checked="" type="checkbox"/> FALSE | |
| 2 | "Off" | %I0.1 | Bool | <input checked="" type="checkbox"/> FALSE | |
| 3 | "Run" | %Q0.0 | Bool | <input checked="" type="checkbox"/> FALSE | |

Pole *Monitor value* pokazuje wartość każdej ze zmiennych.

11.4. Wyświetlanie stanu programu użytkownika

Użytkownik może nadzorować stan zmiennych w edytorech programu LAD i FBD. Należy użyć paska edytora aby wyświetlić edytor programu LAD. Pasek edytora pozwala na zmianę widoku pomiędzy otwartymi edytorami bez konieczności zamknięcia 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 punkty danych podczas wykonywania przez CPU programu użytkownika. Punkty danych 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 faktyczne 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).

| | i | Name | Address | Display format | Monitor value | Force value | F |
|---|---|---------|---------|----------------|---|-------------|---|
| 1 | | "On":P | %I0.0:P | Bool | | TRUE | <input checked="" type="checkbox"/>  |
| 2 | | "Off":P | %I0.1:P | Bool | | | <input type="checkbox"/>  |
| 3 | | "Run":P | %Q0.1:P | Bool |  | | <input type="checkbox"/>  |

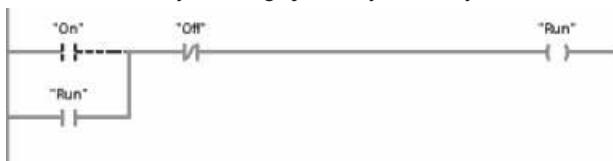
Wartość wejścia lub wyjścia, które ma zostać wymuszona, musi być wpisana w polu *Force value*. Użytkownik może użyć wtedy komórki sprawdzającej w kolumnie *Force*, aby włączyć wymuszenie wejścia lub wyjścia.



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 wymuszanie wartości na wejściach i wyjściach poprzez wyspecyfikowanie adresu wejścia lub wyjścia (I:_P lub Q:_P) w tabeli wymuszeń i zastosowanie 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.

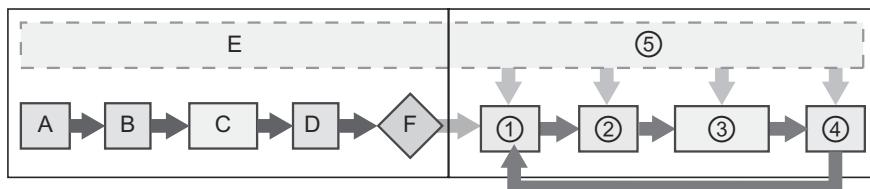
Jeśli program użytkownika jest wykonywany z karty pamięci zabezpieczonej przed zapisem, użytkownik nie może inicjować lub zmieniać wymuszenia I/O z poziomu tablicy monitorującej, ponieważ nie może nadpisywać wartości w chronionym

przed zapisem programie użytkownika. Każda próba wymuszenia wartości chronionej przed zapisem będzie skutkowała 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. Jeśli adresy cyfrowych I/O są przypisane do tych urządzeń, ich wartości nie mogą być modyfikowane przez funkcję *Force* z tabeli monitorującej.



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 przerwań w kolejce nie jest zakłócone.
- E Odblokowywanie zapisu do wyjść nie jest zakłócone.

RUN

- ① Podczas zapisywania obszaru pamięci Q do fizycznych wyjść, CPU uaktualnia stan wyjść wartościami wymuszonimi.
- 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 przerwań 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 wybraną 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 przejęciu 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.

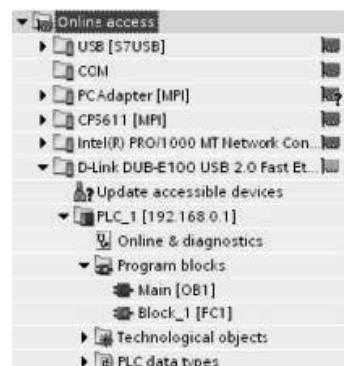


Żeby załadować program z CPU pracującego online do projektu należy wykonać następujące czynności:

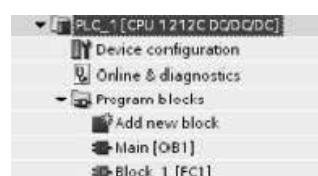
1. Kliknąć folder *Program blocks* w projekcie *offline*.
2. Kliknąć przycisk *Go online*.
3. Kliknąć przycisk *Upload*.
4. Zatwierdzić wybór w oknie dialogowym *Upload*.

Należy z okna nawigatora projektu rozszerzyć węzeł *Online access*, aby wybrać blok programu w połączonym CPU:

1. Należy rozwinąć węzeł sieci i kliknąć dwukrotnie w *Update accessible devices*.
2. Następnie należy rozwinąć węzeł CPU.
3. Należy przeciągnąć folder *Program blocks* z połączonym online CPU i upuścić go w folderze *Program blocks* projektu offline.
4. W oknie dialogowym *Upload preview* należy wybrać boks *Continue*, a następnie kliknąć przycisk *Upload from device*.



Kiedy nastąpi zakończenie zgrywania, wszystkie bloki programu, bloki technologiczne oraz zmienne będą wyświetlane w obszarze *offline*.



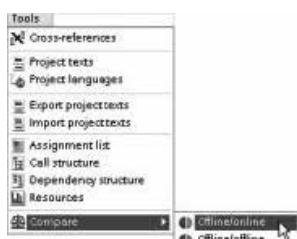
Uwaga

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 i synchronizacja CPU będących *online* i *offline*

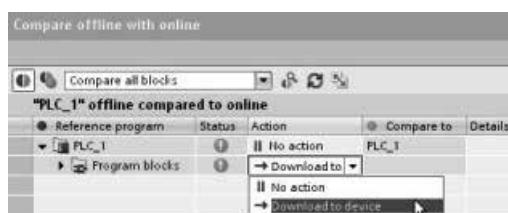
Istnieje możliwość porównania bloków kodu znajdujących się w połączonym CPU z blokiem kodu w programie użytkownika. Jeśli blok kodu z programu użytkownika nie zgadza się z kodem w połączonym CPU, użytkownik może użyć edytora *Compare*, który pozwala na synchronizację projektu z połączonym CPU poprzez zgranie kodu ze sterownika do programu lub wgranie programu do CPU.

Synchronizacja jest następnym sposobem na zgranie projektu z połączonego CPU.

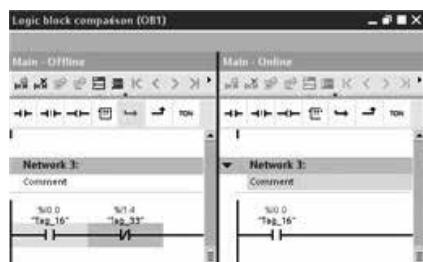


W projekcie należy wybrać CPU.

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).



Naciśnięcie ikony *Action* pozwala na wybór akcji: zgranie programu, wgranie programu lub brak akcji.



Kliknięcie przycisku *Synchronize* powoduje zgranie programu z CPU.

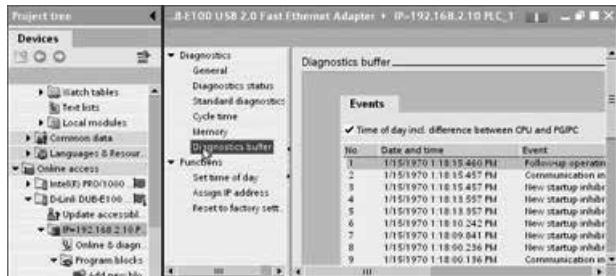
Kliknięcie przycisku *detailed comparison* (szczegółowe porównanie) pokazuje bloki kodu obok siebie.

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 połączonego CPU, użytkownik może wyświetlić lub zmienić adres IP. Istnieje także możliwość wyświetlania lub ustawienia parametrów czasu i daty dla połączonego 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. Ustawienia fabryczne

W każdej chwili użytkownik może powrócić do fabrycznych ustawień sterownika po warunkiem, że:

- Karta pamięci nie jest zainstalowana w sterowniku.
- CPU ma połączenie *online*.
- CPU znajduje się w trybie STOP.

Uwaga

Jeżeli CPU jest w trybie RUN i zostanie zainicjowane przywracanie ustawień fabrycznych, przełączenie w tryb STOP jest możliwe dopiero po potwierdzeniu ostrzeżenia.

Procedura

Przywrócenie ustawień fabrycznych wymaga wykonania następujących kroków:

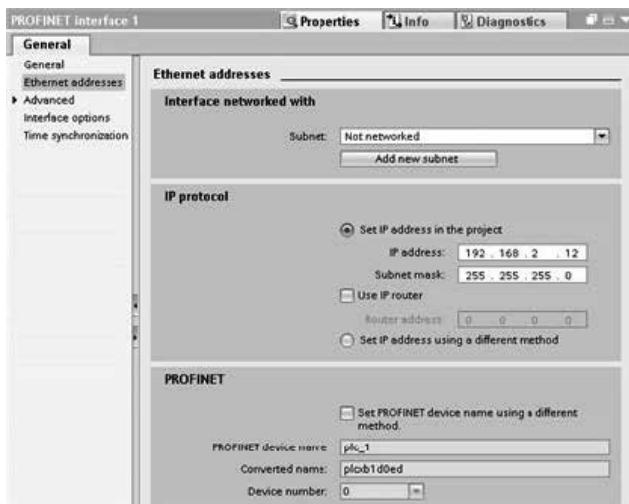
1. Otwarcie w CPU widoku *Online and Diagnostic*.
2. Wybranie opcji *Reset to factory settings* w folderze *Functions*.
3. Zaznaczenie opcji *Keep IP address*, jeżeli adres IP ma pozostać niezmieniony. Jeżeli konieczna jest zmiana adresu IP należy zaznaczyć opcję *Reset IP address*.
4. Przyciśnięcie przycisku *Reset*.
5. Potwierdzenie ostrzeżenia za pomocą *OK*.

Skutki

- Pamięć robocza, pamięć load oraz operatory są zerowane.
- Parametrom są przypisane wartości domysльne.
- Bufor diagnostyczny jest zerowany.
- Czas dzienny jest zerowany.
- Adres IP zmienia się lub nie, w zależności od opcji wybranej podczas zerowania. Adres MAC jest stały, nie można go zmodyfikować.

11.13. Załadowanie adresu IP do połączonego online CPU

Aby przypisać adres IP należy przeprowadzić następujące działania:



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.



Po ściągnięciu konfiguracji urządzenia wraz z adresem IP, użytkownik może zobaczyć adres IP w folderze *Online access*.

- Skonfigurowanie dla CPU adresu IP (strona 74).
- Zapisanie i załadowanie konfiguracji do CPU.

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 prze PLC.

11.14. 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.

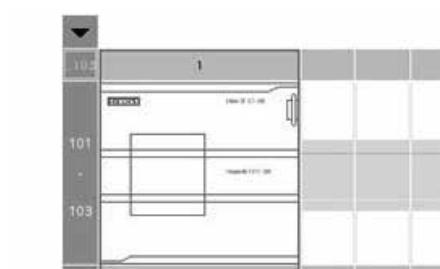


W konfiguracji urządzenia (widok projektu lub widok portalu) należy dodać nowe urządzenie, ale zaznaczając *unspecified CPU* zamiast wyboru określonego urządzenia. STEP 7 utworzy obiekt nieznanego CPU.



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.



The device is not specified.
→ Please use the [hardware catalog](#) to specify the CPU,
→ or [detect](#) the configuration of the connected device.

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*.

11.15. Ładowanie programu do pamięci sterownika w trybie pracy RUN

CPU sterownika jest przystosowane do wymiany programu sterującego w trakcie pracy sterownika w trybie RUN („Download in RUN mode”). Ta funkcja umożliwia wprowadzanie niewielkich modyfikacji do programu przy minimalnym zakłócaniu pracy sterownika, pozwala także na całkowitą wymianę programu sterującego, co wiąże się jednak z ryzykiem zakłócenia pracy aplikacji.



OSTRZEŻENIE

Wymiana fragmentów programu sterującego pracą CPU znajdującego się w trybie RUN powoduje natychmiastową zmianę sposobu jego pracy. Wymiana programu w trybie RUN może spowodować nieprzewidziane zachowanie sterownika, w konsekwencji którego może wystąpić zagrożenie życia i/lub ryzyko powstania uszkodzeń osprzętu.

Wymianę oprogramowania sterownika w trybie RUN może przeprowadzać wyłącznie autoryzowany personel, przygotowany do realizacji tego typu zadań.

Funkcja „Download in RUN mode” umożliwia modyfikację i wymianę programu znajdującego się w pamięci sterownika bez konieczności przełączania go w tryb STOP, co pozwala na:

- Wprowadzanie niewielkich zmian w przebiegu procesu bez konieczności jego zatrzymywania (na przykład wymiana parametrów procesu).
- Szybkie debugowanie pracy programu (na przykład poprzez zmianę stanu wyjść sterownika bez oczekiwania na wystąpienie zadanych warunków zmiany).

Użytkownik może wykonać następujące modyfikacje w blokach i znacznikach, następnie zapisać ich nowe wartości w pamięci sterownika:

- Tworzyć, modyfikować i kasować funkcje (FC), bloki funkcyjne (FB oraz tabele znaczników).
- Tworzyć i kasować bloki danych DB, przy czym nie jest możliwe modyfikowanie struktur DB, dopuszczalne są natomiast wartości początkowe DB. W trybie RUN nie jest możliwa modyfikacja DB webserwera.
- Modyfikacja bloków organizacyjnych OB, nie jest możliwe ich dodawanie i kasowanie.

Jednocześnie można zapisywać maksymalnie do 10 bloków. W przypadku konieczności zapisania więcej niż 10 bloków konieczne jest przełączenie sterownika w tryb STOP.

W przypadku modyfikowania w trybie RUN prawdziwego procesu, przed rozpoczęciem wymiany programu trzeba uwzględnić potencjalne niebezpieczeństwa dla sterowanych maszyn i urządzeń, a także ich obsługi.

Uwaga

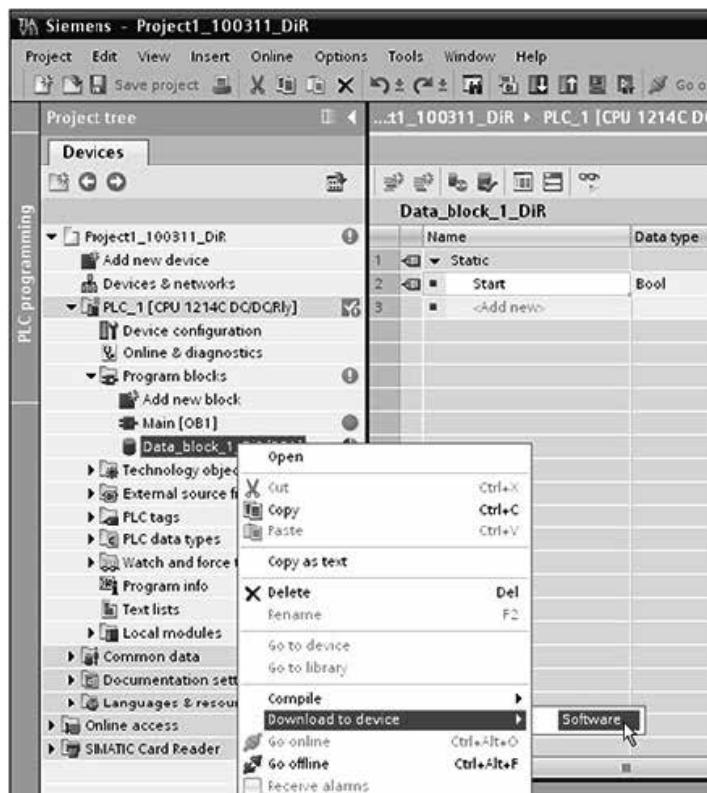
Jeżeli sterownik jest w trybie RUN i zakończono modyfikowanie programu, STEP 7 zawsze najpierw podejmie próbę jego załadowania do pamięci sterownika w trybie RUN. Jeżeli użytkownik nie chce tego musi przełączyć sterownik w tryb STOP.

Jeżeli modyfikacje programu uniemożliwiają jego załadowanie do pamięci sterownika będącego w trybie RUN, STEP 7 poinformuje użytkownika, że musi on samodzielnie przełączyć go w tryb STOP.

11.15.1. Modyfikacje programu w trybie RUN

Żeby zmodyfikować program w trybie RUN sterownika w pierwszym kroku należy sprawdzić czy sterownik obsługuje tryb „Download In RUN mode” oraz czy CPU jest w trybie RUN:

1. Żeby załadować program do pamięci sterownika znajdującego się w trybie RUN należy wybrać jedną z metod:
 - użyć polecenia *Download to device* z menu *Online*,
 - użyć przycisku *Download to device* z paska narzędziowego,
 - kliknąć prawym przyciskiem myszy *Program blocks* w drzewie projektu i wybrać polecenie *Download to device -> Software*.



- Jeżeli program skompilował się bez błędów STEP 7 ładuje go do pamięci CPU.
- STEP7 proponuje opcje: ładowanie programu do pamięci lub zaniechanie dalszych kroków.
- Jeżeli zostanie wybrana opcja „Load”, STEP 7 załaduje program do pamięci CPU.

Dane techniczne

A.1. Dane techniczne ogólne

Zgodność z normami

System automatyki S7-1200 jest zgodny z podanymi niżej normami i procedurami testowymi. Kryteria testowania systemu automatyki S7-1200 są oparte o te normy i procedury testowe.

Nie wszystkie modele sterowników z serii S7-1200 mają obecnie wszystkie certyfikaty, a statusy ich certyfikacji mogą zmienić się bez uprzedzenia. Sprawdzenie aktualnego statusu certyfikatów sterowników stosowanych w aplikacjach należy do obowiązków użytkownika. W doborze odpowiedniego sterownika pomocą służby lokalne biuro firmy Siemens oraz dystrybutorzy, którzy udzielają także aktualnych informacji o statusach certyfikacji poszczególnych modeli sterowników.

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 emisjyjności 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 WYJĄTEK: sprawdź w specyfikacji technicznej maksymalną dopuszczalną liczbę wejść lub wyjść włączonych jednocześnie. Niektóre modele wymagają zmniejszenia ich liczby przy temperaturze otoczenia Ta = 60°C.

Ostrzeżenie

Zastąpienie komponentów innymi może spowodować niezgodność z zaleceniami Class 1, Division 2 oraz Zone 2. Naprawy mogą być wykonywane wyłącznie przez autoryzowane centra serwisowe firmy Siemens.

Zatwierdzenie ATEX



Zatwierdzenie ATEX dotyczy wyłącznie modeli DC, nie dotyczy zaś modeli AC oraz Relay.

EN 60079-0:2006: Środowisko wybuchowe – wymagania ogólne,

EN 60079-15:2005: Aparatura elektryczna do pracy w środowisku narażonym na zagrożenie wybuchem.

Typ zabezpieczenia „nA”

II 3 G Ex nA II T3

WAŻNY WYJĄTEK: sprawdź w specyfikacji technicznej maksymalną dopuszczalną liczbę wejść lub wyjść włączonych jednocześnie. Niektóre modele wymagają zmniejszenia ich liczby przy temperaturze otoczenia $T_a = 60^{\circ}\text{C}$.

Moduły muszą być instalowane w odpowiednich obudowach o stopniu zabezpieczenia co najmniej IP54 zgodnie z normą EN 60529 oraz uwzględniających warunki środowiskowe miejsca pracy urządzeń.

Jeżeli nominalna temperatura pracy przekracza 70°C w miejscu wprowadzenia kabla lub 80°C w miejscu rozgałęziania przewodów, to dane temperaturowe wybranych kabli muszą być zgodne z faktycznie zmierzoną temperaturą.

Należy podjąć środki zabezpieczające, by zakłócenia o charakterze przejściowym nie przekroczyły nominalnych wartości napięć o więcej niż 40%.

Zatwierdzenie C-Tick



System automatyki S7-1200 spełnia wymagania norm AS/NZS 2064 (Class A).



Korea Certification

System automatyki S7-1200 spełnia wymagania certyfikatu Korean Certification (KC Mark). Obejmuje on urządzenia zdefiniowane w Class A Equipment, przeznaczone do zastosowań przemysłowych, których stosowanie w aplikacjach domowych nie jest zalecane. Dopuszczenia 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 ostatnio zatwierdzeń sporzązonej 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)

Środowiska przemysłowe

System automatyki S7-1200 jest zaprojektowany do zastosowania w środowiskach przemysłowych.

Tabela A.1. Środowiska przemysłowe

| Zakres zastosowania | Wymagania dotyczące emisjności 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脉冲 przejściowe | 2 kV, 5 kHz z obwodem sprzągającym do zasilania AC i DC systemu. 2 kV, 5 kHz z zaciskami sprzągającymi do I/O. |
| EN 61000-4-5 Odporność na udary | 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 (sygnały I/O, systemy zasilania DC) wymagane jest zewnętrzne zabezpieczenie. |
| 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. |

Tabela A.3. Emisjyność przewodzona i promieniowana wg EN 61000-6-4

| Kompatybilność elektromagnetyczna – emisjyność przewodzona i promieniowana wg EN 61000-6-4 | | |
|--|------------------------|--|
| Emisjyność przewodzona EN 55011, Class A, Group 11 | Od 0,15 MHz do 0,5 MHz | <79 dB (μ V) quasi-pik; <66 dB (μ V) średnio |
| | Od 0,5 MHz do 5 MHz | <73 dB (μ V) quasi-pik; <60 dB (μ V) średnio |
| | Od 5 MHz do 30 MHz | <73 dB (μ V) quasi-pik; <60 dB (μ V) średnio |
| Emisjyność promieniowana EN 55011, Class A, Group 1 | Od 30 MHz do 230 MHz | <40dB (μ V/m) quasi-pik; pomiar w odległości 10 m |
| | Od 230 MHz do 1 GHz | <47dB (μ V/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 | Od -40°C do +70°C |
| EN60068-2-1, Test Ab, zimno | |
| 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% Chyba, że wyspecyfikowano inaczej |
| Ciśnienie atmosferyczne | Od 1080 do 795 hPa (odpowiadające wysokości od 1000 do 2000 m) |
| Stężenie zanieczyszczeń | S _{O₂} : <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 uderów w każdym z 3 kierunków |
| EN60068-2-6 vibracje 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 |

Uwaga

Systemy, które muszą być włączane w temperaturze od -20°C do 0°C program użytkownika powinien zapewniać 10-sekundowe opóźnienie zadziałania obciążanych wyjść od chwili włączenia.

Tabela A.6. Test izolacji wysokim napięciem

| Test izolacji wysokim napięciem | |
|---|---|
| Obwody o napięciu znamionowym 24/5 V | 520 VAC (test typu granic izolacji optycznej) |
| Obwody 115/230 V do uziemienia | 1500 VAC |
| Obwody 115/230 V do obwodów 115/230 V | 1500 VAC |
| Obwody 115/230 V do obwodów 24/5 V | 1500 VAC (3000 VAC/4242 VDC test typu) |
| Interfejs Ethernet w odniesieniu do obwodów 24V/5V oraz masy zasilania ¹ | 1500 (tylko test typu) |

¹ Izolacja interfejsu sieciowego jest zaprojektowana w taki sposób, żeby zminimalizować niebezpieczeństwo podczas krótkotrwałych zwarć linii do napięć o wartościach zagrażających życiu. Nie jest to jednoznaczne ze spełnieniem wymogów bezpieczeństwa stawianych klasycznym izolacjom stosowanym w energetyce.

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

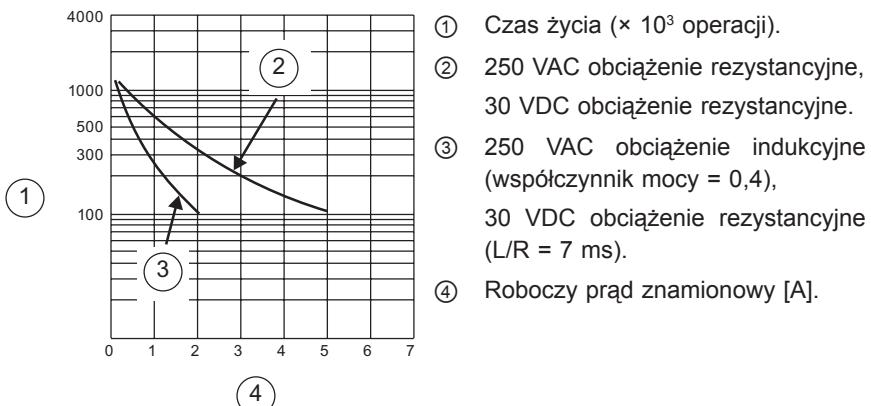
| Napięcie znamionowe | Tolerancja |
|---------------------|--|
| 24 VDC | Od 20,4 VDC do 28,8 VDC Od 22,0 VDC do 28,8 VDC dla temperatury z przedziału -20°C...0°C |
| 120/230 VAC | Od 85 VAC do 264 VAC, od 47 do 63 Hz |

Uwaga

Kiedy przełącznik mechaniczny łączy 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.

Żywotność przekaźników elektrycznych

Typowe dane dla przekaźników, dostarczane przez ich producentów, 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.



A.2. Specyfikacja CPU

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku S7-1200 System Manual lub na internetowej stronie wsparcia technicznego (<http://www.siemens.com/automation/support-request>).

Tabela A.8. Ogólna specyfikacja

| Ogólna specyfikacja | | CPU 1211C | CPU 1212C | CPU 1214C | CPU 1215C |
|---|--|--|--|--|--|
| Wymiary W × H × D | | 90 × 100 × 75 | 90 × 100 × 75 | 110 × 100 × 75 | 130 × 100 × 70 |
| Masa | <ul style="list-style-type: none"> • AC/DC/Przekaźnik • DC/DC/ Przekaźnik • 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 |
| Pobór mocy | <ul style="list-style-type: none"> • AC/DC/ Przekaźnik • DC/DC/ Przekaźnik • 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 |
| Wydajność prądowa (5 VDC) magistrala CM lub SM | 750 mA maks. | 1000 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. | |
| Pobór prądu przez wejścia cyfrowe (24 VDC) | 4 mA/ wykorzystane wejście | 4 mA/ wykorzystane wejście | 4 mA/ wykorzystane wejście | 4 mA/ wykorzystane wejście | |

Tabela A.9. Charakterystyka CPU

| Charakterystyka CPU | CPU 1211C | CPU 1212C | CPU 1214C | CPU 1215C |
|--|--|--|--|--|
| Pamięć użytkownika | | | | |
| • Pamięć robocza | • 30 kB | • 50 kB | • 75 kB | • 100 kB |
| • Pamięć ładowania | • 1 MB | • 1 MB | • 4 MB | • 4 MB |
| • Pamięć trwała | • 10 kB | • 10 kB | • 10 kB | • 10 kB |
| Wbudowane cyfrowe I/O (strona 247). | 6 wejść 4 wyjścia | 8 wejść 6 wyjścia | 14 wejść 10 wyjścia | 14 wejść 10 wyjścia |
| Wbudowane analogowe I/O (strona 255). | 2 wejścia | 2 wejścia | 2 wejścia | 2 wejścia |
| Rozmiar obrazu procesu: | | | | |
| • Wejście | • 1024 bajtów | • 1024 bajtów | • 1024 bajtów | • 1024 bajtów |
| • Wyjście | • 1024 bajtów | • 1024 bajtów | • 1024 bajtów | • 1024 bajtów |
| Pamięć bitowa (M) | 4096 B | 4096 B | 8192 B | 8192 B |
| Rozszerzające moduły sygnałowe (SM) | brak | 2 SM maks. | 8 SM maks. | 8 SM maks. |
| Rozszerzająca płytka sygnałowa (SB) | 1 maks. | 1 maks. | 1 maks. | 1 maks. |
| Rozszerzające moduły komunikacyjne (CM) | 3 maks. | 3 maks. | 3 maks. | 3 maks. |
| Szybkie liczniki | | | | |
| • jednofazowe (częstotliwości zegara) | Wbudowane 3 linie I/O, 5 z SB • 3 przy 100 kHz SB: 2 at 30 kHz | Wbudowane 4 linie I/O, 6 z SB • 3 przy 100 kHz, 1 przy 30 kHz SB: 2 at 30 kHz | 6 łącznie • 3 przy 100 kHz, 3 przy 30 kHz | 6 łącznie • 3 przy 100 kHz, 3 przy 30 kHz |
| • kwadraturowe (częstotliwości zegara) | • 3 przy 80 kHz SB: 2 at 20 kHz | • 3 przy 80 kHz, 1 przy 20 kHz SB: 2 at 20 kHz | • 3 przy 80 kHz i 3 przy 20 kHz | • 3 przy 80 kHz, 3 przy 20 kHz |
| Wyjścia impulsowe ¹ | 4 | 4 | 4 | 4 |
| Wejścia rejestrujące impulsy | 6 | 8 | 14 | 14 |
| Przerwania od opóźnienia/cykliczne | 4 łącznie z rozdzielcością 1 ms | 4 łącznie z rozdzielcością 1 ms | 4 łącznie z rozdzielcością 1 ms | 4 łącznie z rozdzielcością 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 |
| Zegar czasu rzeczywistego | | | | |
| • Dokładność | • +/- 60 sekund/miesiąc | • +/- 60 sekund/miesiąc | • +/- 60 sekund/miesiąc | • +/- 60 sekund/miesiąc |
| • Czas podtrzymywania (bezobsługowy Super-kondensator) | • 20 dni typ./12 dni min. przy 40°C | • 20 dni typ./12 dni min. przy 40°C | • 20 dni typ./12 dni min. przy 40°C | • 20 dni typ./12 dni min. przy 40°C |
| Szybkość wykonywania | | | | |
| • operacji boolowskich | • 0,08 µs/instrukcję | • 0,08 µs/instrukcję | • 0,08 µs/instrukcję | • 0,08 µs/instrukcję |
| • operacji Move Word | • 12 µs/instrukcję | • 12 µs/instrukcję | • 12 µs/instrukcję | • 12 µs/instrukcję |
| • operacji Real Math | • 18 µs/instrukcję | • 18 µs/instrukcję | • 18 µs/instrukcję | • 18 µs/instrukcję |

¹ Dla CPU z wyjściami przekaźnikowymi użytkownik musi zainstalować cyfrową płytę sygnałową (SB), aby móc używać wyjść impulsowych.

Tabela A.10. Komunikacja

| Dane techniczne | CPU 1211C, CPU 1212C, CPU 1214C | CPU 1215C |
|----------------------------------|--|--|
| Komunikacja | 1 port Ethernet <ul style="list-style-type: none"> • Szybkość przesyłu danych • Izolacja (sygnału zewnętrznego od logiki PLC) • Typ kabla | 2 port Ethernet <ul style="list-style-type: none"> • 10/100 Mb/s • Izolacja transformatorowa, 1500 VDC • CAT5e ekranowany |
| Urządzenia | • 3 HMI ¹ <ul style="list-style-type: none"> • 1 PG | |
| Połączenia Ethernet ² | 8 (aktywne lub pasywne) | |
| Połączenia CPU-CPU (GET/PUT) | • 8 (klient) <ul style="list-style-type: none"> • 3 (serwer) | |

¹ Każde CPU posiada trzy przyłącza HMI do obsługi HMI (do 3 urządzeń HMI (do 2 paneli SIMATIC Comfort). Łączna ilość interfejsów HMI jest zależna od typu paneli HMI użytych w aplikacji. Przykładowo, można przyłączyć do trzech paneli SIMATIC Basic do CPU lub do dwóch paneli SIMATIC Comfort i dodatkowo jeden panel SIMATIC Basic.

² Otwieranie połączenia komunikacyjnego użytkownika (aktywne lub pasywne): TSEND_C, TRCV_C, TCON, TDICON, TSEND oraz TRCV.

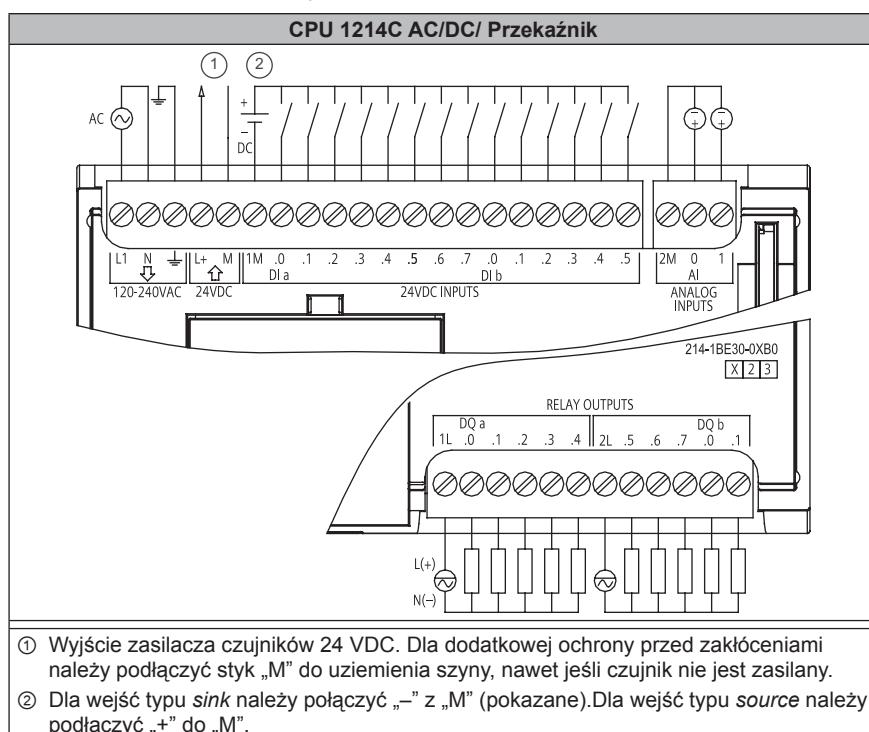
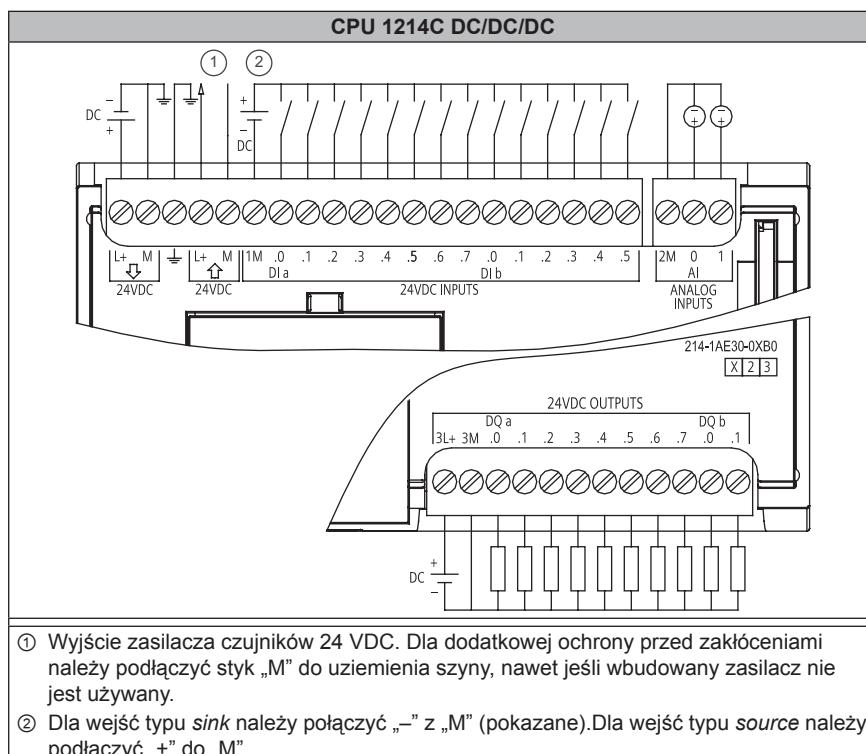
Tabela A.11. Schemat podłączenia CPU1214C AC/DC/Przekaźnik

Tabela A.12. Schemat podłączenia CPU1214C DC/DC/DC



A.3. Moduły cyfrowych I/O

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku S7-1200 System Manual lub na internetowej stronie wsparcia technicznego (<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.13. 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 | | 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 |
| 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: source) | 4 wyjścia (półprzewodnik – MOSFET) |

Tabela A.14. Moduł wejść/wyjść cyfrowych (DI/DQ) SM 1223

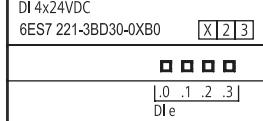
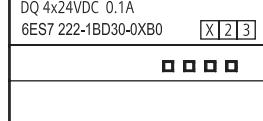
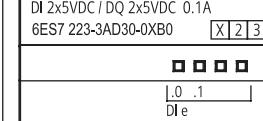
| Ogólne | | SB 1223 DI/DQ (200 kHz) | SB 1223 2 DI/2 DQ |
|-------------------|---------------|--|---|
| Nr zamówieniowy | | <ul style="list-style-type: none"> • 24 VDC: 6ES7 223-3BD30-0XB0 • 5 VDC: 6ES7 223-3AD30-0XB0 | 6ES7 223-0BD30-0XB0 |
| Wymiary W × H × D | | 38 × 62 × 21 (mm) | 38 × 62 × 21 (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: source) 2 wyjścia (półprzewodnik – MOSFET) | 2 wejścia (IEC Type 1 sink) 2 wyjścia (półprzewodnik – MOSFET) |

Uwaga

Szybkie (200 kHz) SB używa wejścia typu *source*. Standardowa płytka SB (20 kHz) używa wejścia typu *sink*. Więcej informacji znajduje się w specyfikacji cyfrowych wejść i wyjść (strona 247).

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 „+”. 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.15. 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>Detailed description: This section shows the internal connection of the SB 1221 module. It has four digital input terminals (I.0, I.1, I.2, I.3) each connected to a 24VDC power source via a diode. The terminals are labeled DI e.</p> | <p>SB 1222 DC 200 KHz DQ 4x24VDC 0.1A 6ES7 222-1BD30-0XB0 [X 2 3]</p>  <p>Detailed description: This section shows the internal connection of the SB 1222 module. It has four digital output terminals (I.0, I.1, I.2, I.3) each connected to a 24VDC power source via a diode. The terminals are labeled DQ e.</p> | <p>SB 1223 DC/DC 200 KHz DI 2x5VDC / DQ 2x5VDC 0.1A 6ES7 223-3AD30-0XB0 [X 2 3]</p>  <p>Detailed description: This section shows the internal connection of the SB 1223 module. It has two digital input terminals (I.0, I.1) and two digital output terminals (I.0, I.1) each connected to a 5VDC power source via a diode. The terminals are labeled DI e and DQ e.</p> |
| <p>① Obsługują tylko wejścia typu <i>source</i>.</p> | <p>① Dla wyjść <i>source</i> należy połączyć <i>Load</i> do „–” (na rysunku). Dla wyjść <i>sink</i> należy połączyć <i>Load</i> do „+”.</p> | <p>① Obsługują tylko wejścia typu <i>source</i>. ② Dla wyjść <i>source</i> należy połączyć <i>Load</i> do „–” (na rysunku). Dla wyjść <i>sink</i> należy połączyć <i>Load</i> do „+”.</p> |

Uwaga

Szybkie (200 kHz) SB (SB 1221 i SB 1223) używa wejść tylko typu *sink*. Standardowa płytka 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 „+”. 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

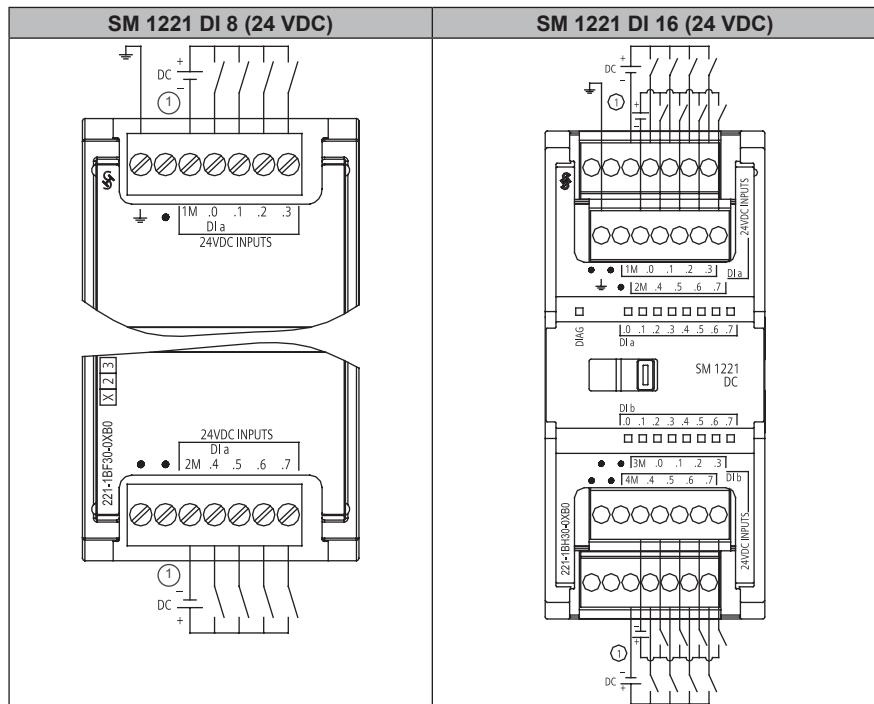
występuje przepływ prądu). Jeżeli moduł jest podłączony bez programu użytkownika, domyślnym napięciem dla tego modułu jest 0 V, co oznacza, że obciążenie typu *sink* będzie w stanie ON.

A.3.2. Moduł SM 1221 – cyfrowe wejścia (DI)

Tabela A.16. SM 1221 – cyfrowe wejścia (DI)

| Dane techniczne | SM 1221 DI 8 (24 VDC) | SM 1221 DI 16 (24 VDC) |
|---|--|-------------------------------------|
| Nr zamówieniowy | 6ES7 221-1BF30-0XB0 | 6ES7 221-1BH30-0XB0 |
| Liczba wejść (DI) (Specyfikacja na stronie 236). | 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 24 VDC 4 mA/wykorzystane wejście | 130 mA 4 mA/wykorzystane wejście |

Tabela A.17. Schemat podłączenia dla modułów cyfrowych wejść (DI) dla SM 1221



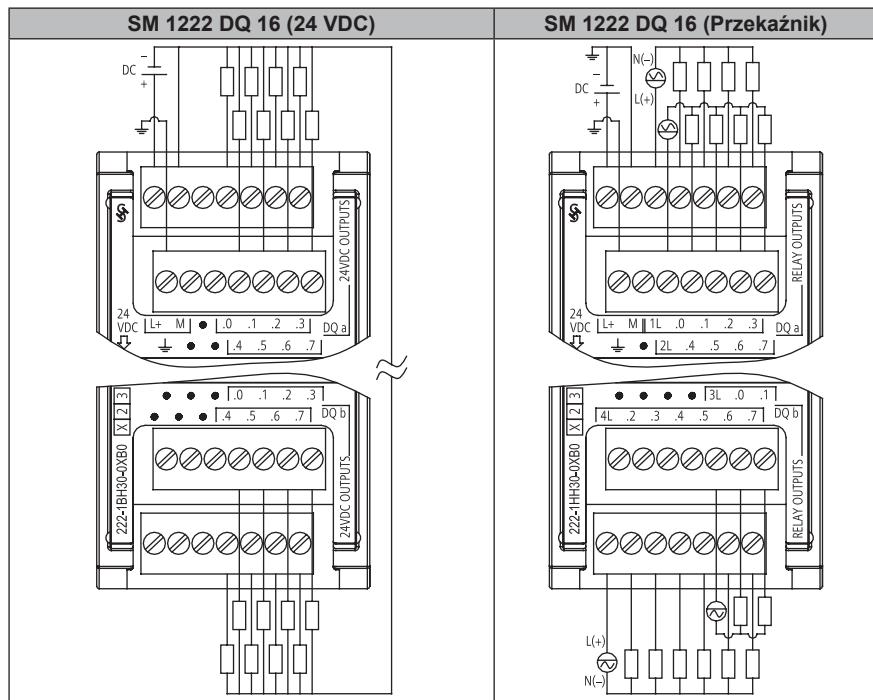
- ① Dla wejść typu *sink* należy podłączyć „–” do „M” (na rysunku). Dla wejść typu *source* należy podłączyć „+” do „M”.

A.3.3. Moduł SM 1222 – cyfrowe wyjścia (DQ)

Tabela A.18. SM 1222 – cyfrowe wyjścia (DQ)

| Dane techniczne | | SM 1222 DQ (przekaźnik) | SM1222 DQ (24VDC) |
|--|---------------|--|---|
| Nr zamówieniowy | | <ul style="list-style-type: none"> • DQ 8: 6ES7 222-1HF30-0XB0 • DQ 8 zamiennik: 6ES7 222-1XF30-0XB0 • DQ 16: 6ES7 222-1HH30-0XB0 | <ul style="list-style-type: none"> • DQ 8: 6ES7 222-1BF30-0XB0 • DQ 16: 6ES7 222-1BH30-0XB0 |
| Liczba wyjść (DQ) (Specyfikacja na stronie 236) | | <ul style="list-style-type: none"> • (DQ 8) lub 16 (DQ 16) • 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 zamiennik: 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 zamiennik: 5 W • DQ 16: 8,5 W | <ul style="list-style-type: none"> • DQ 8: 1,5 W • DQ 16: 2,5 W |
| Pobór prądu | magistrala SM | <ul style="list-style-type: none"> • DQ 8: 120 mA • DQ 8 zamiennik: 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 oraz DQ 16: 11 mA aktywną cewkę przekaźnika • DQ 8 zamiennik: 16,7 mA aktywną cewkę przekaźnika | <ul style="list-style-type: none"> • DQ 8: -- • DQ 16: -- |

Tabela A.19. Schemat podłączenia dla modułów cyfrowych wyjść (DQ) dla SM 1222

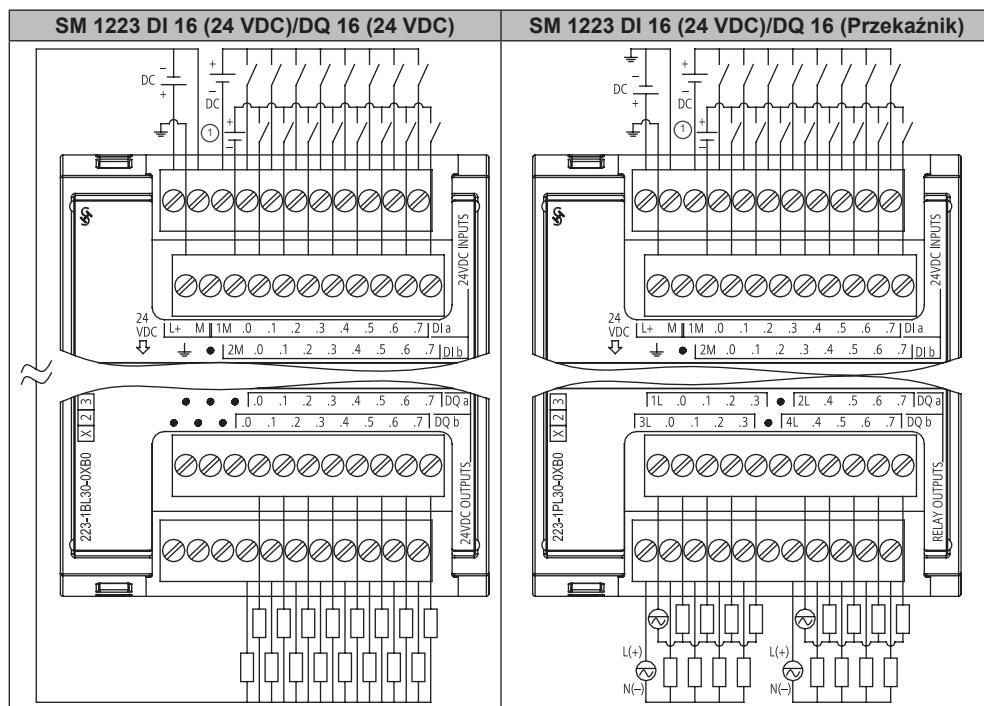


A.3.4. Moduł SM 1223 cyfrowe wejścia/wyjścia VDC (DI/DQ)

Tabela A.20. SM 1223 – cyfrowe wejścia wyjścia (DI/DQ)

| Dane techniczne | SM 1223 DI (24 VDC)/DQ (Przekaźnik) | SM 1223 DI (24 VDC)/DQ (24 VDC) |
|---|--|--|
| Nr zamówieniowy | DI 8/DQ 8: 6ES7 223-1PH30-0XB0 DI 16/DQ 16: 6ES7 223-1PL30-0XB0 | DI 8/DQ 8: 6ES7 223-1BH30-0XB0 DI 8/DQ 8: 6ES7 223-1BL30-0XB0 |
| Liczba wejść/wyjść (DI/DQ) (Specyfikacja na stronie 236). | <ul style="list-style-type: none"> Wejście: 8 lub 16 (24 VDC) Wyjście: 8 lub 16 (przekaźnik) | <ul style="list-style-type: none"> Wejście: 8 lub 16 (24 VDC) Wyjście: 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 |
| 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 |
| | 24 VDC | 4 mA/wykorzystane wejście 11 mA/wykorzystaną cewkę przekaźnika |
| | | 4 mA/wykorzystane wejście |

Tabela A.21. Schemat podłączenia dla modułów cyfrowych wejśc/wyjśc (DI/DQ) dla SM 1223



① 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

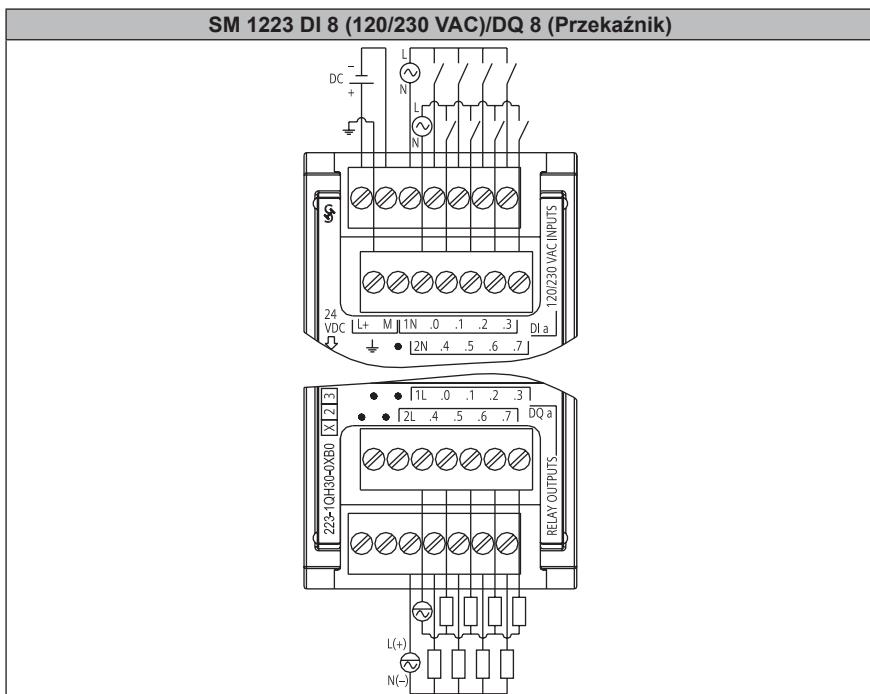
Tabela A.22. Moduł 1223 – cyfrowe wejścia VAC/wyjścia (DI/DQ)

| Dane techniczne | | SM 1223 DI (120/230 VAC)/DQ (Przekaźnik) |
|----------------------------|---------------|---|
| Nr zamówieniowy | | DI 8/DQ 8: 6ES7 223-1QH30-0XB0 |
| Liczba wejść/wyjść (DI/DQ) | | Wejścia: 8 (120/230 VAC) Specyfikacja dla wejść 120/230 VAC: Strona 238. Wyjścia: 8 (przekaźnik) Specyfikacja dla cyfrowych wyjść: Strona 238. |
| 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 × 120/230 VAC, DQ 8 × Przekaźnik (6ES7 223-1QH30-0XB0) jest zatwierdzony do użytku w Class 1, Division 2, Gas Group A, B, C, D, Temperature Class T4 Ta = 40°C.

Tabela A.23. Schemat podłączenia dla modułów SM 1223 DI 8 (120/230 VAC)/DQ (Przekaźnik)



A.4. Specyfikacja cyfrowych wejść i wyjść

A.4.1. Cyfrowe wyjścia 24 VDC (DI)

Tabela A.24. 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 |

| Dane techniczne | CPU, SM oraz SB | Szybkie SB (200 kHz) |
|---|--|--|
| 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 |
| Sygnal 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 |
| Sygnal 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.25. 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) |
| 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.26. 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 | <table border="1" style="width: 100%;"> <tr> <td style="padding: 2px;">Nieekranowany</td> <td style="padding: 2px;">300 metrów</td> </tr> <tr> <td style="padding: 2px;">Ekranowany</td> <td style="padding: 2px;">500 metrów</td> </tr> </table> | Nieekranowany | 300 metrów | Ekranowany | 500 metrów |
| 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.27. Specyfikacja cyfrowych wyjść (DQ)

| Dane techniczne | Przekaźnik (CPU oraz SM) | 24V DC (CPU, SM, oraz SB) | 200 KHZ 24V DC (SB) |
|--------------------------------------|-------------------------------|---------------------------------|---|
| Typ | przekaźnik, 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 |
| Sygnał logiczny 1 przy maks. prądzie | N/A | 20 VDC min. | L+ minus 1,5 V ¹ L+ minus 0,7 V ² |

A. Dane techniczne

| Dane techniczne | Przekaźnik (CPU oraz SM) | 24V DC (CPU, SM, oraz SB) | 200 KHZ 24V DC (SB) |
|---|---|--|--|
| 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 Changeover 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 | Przekaźnik (CPU oraz SM) | 24V DC (CPU, SM, oraz SB) | 200 KHZ 24V DC (SB) |
|---|---|--|---|
| Prąd wspólny | CPU: SM przekaźnik: • SM 1222: 10 A (DQ 8 oraz DQ 16) • SM 1223 DI 8/ DQ 8 przekaźnik: 10 A • SM 1223 DI 16/DQ 16 przekaźnik: 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 przekaźnika | 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) | Przekaźnik: 10 mln cykli załącz/wyłącz | N/A | N/A |
| Trwałość styków przy nominalnym obciążeniu | Przekaźnik: 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 przekaźnikowe użytkownik musi zainstalować płytę 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 wejścia.

A.5. Moduły analogowych I/O

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku S7-1200 System Manual lub na internetowej stronie wsparcia technicznego (<http://www.siemens.com/automation/>).

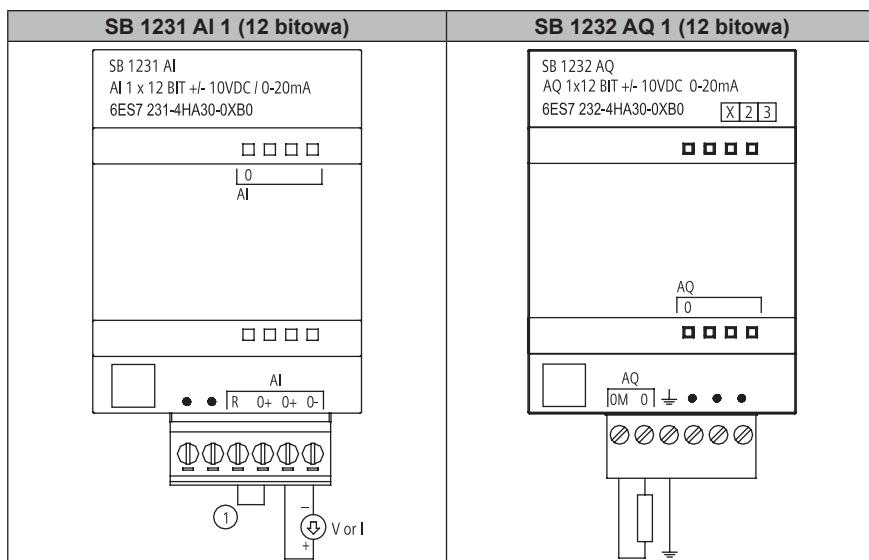
A.5.1. Płytki sygnałowe analogowych wejść (AI)- SB 1231 i analogowych wyjść (AQ) – SB 1232

Tabela A.28. Ogólna specyfikacja

| Dane techniczne | SB 1231 AI 1 (12 bitowa) ¹ | SB 1232 AQ 1 (12bitowa) |
|-----------------------------|---------------------------------------|-------------------------|
| 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.29. 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.30. SM 1231 – analogowe wejścia (AI)

| Dane techniczne | SM 1231 AI 4 (13-bitowy) | SM 1231 AI 8 (13-bitowy) | SM 1231 AI 4 (16-bitowy) |
|-----------------------------|---|---|---------------------------------------|
| Nr zamówieniowy (MLFB) | 6ES7 231-4HD30-0XB0 | 6ES7 231-4HF30-0XB0 | 6ES7 231-5ND30-0XB0 |
| 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 (wejście 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.31. SM 1232 – analogowe wyjścia (AQ)

| Dane techniczne | SM 1232 AQ 2 (14bitowy) | SM 1232 AQ 4 (14bitowy) |
|-----------------------------|-------------------------|-------------------------|
| Nr zamówieniowy (MLFB) | 6ES7 232-4HB30-0XB0 | 6ES7 232-4HD30-0XB0 |
| 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.32. SM 1234 – analogowe wejścia/wyjścia (AI/AQ)

| Dane techniczne | SM 1234 AI 4 (13 bitowy)/AQ 2 (14 bitowy) |
|-----------------------------|---|
| Nr zamówieniowy (MLFB) | 6ES7 234-4HE30-0XB0 |
| 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) |

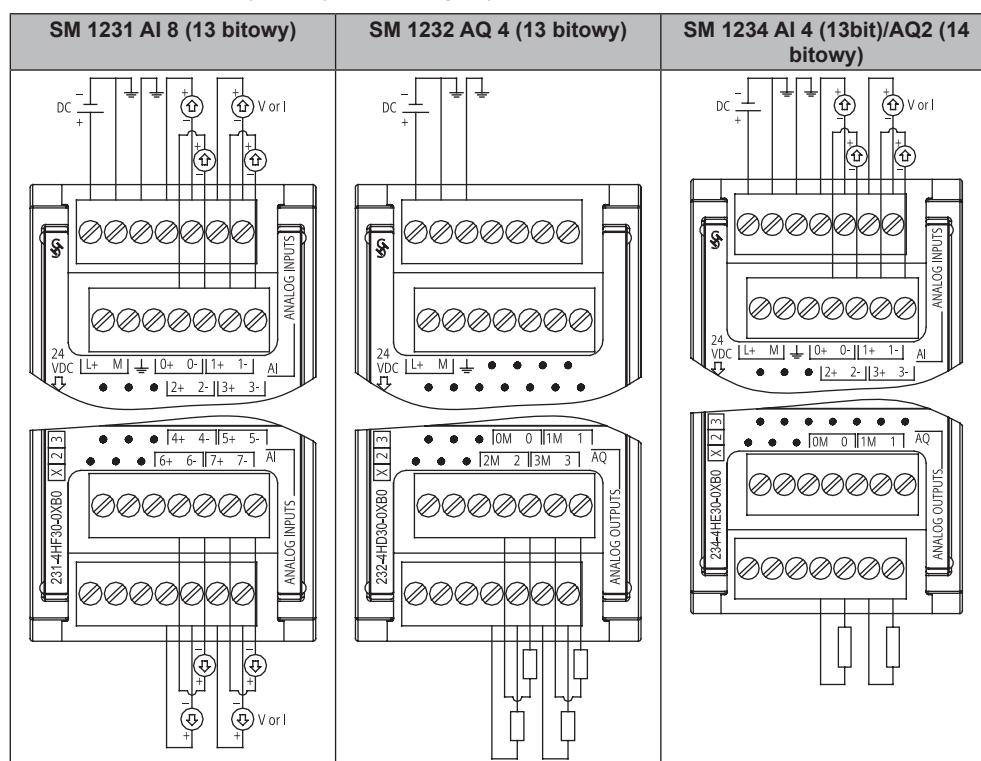
A.5.5. Schematy podłączeń dla SM 1231 (AI), SM 1232 (AQ) oraz SM 1234 (AI/AQ)

Uwaga

Nie używane wejścia analogowe powinny zostać dodane.

Kiedy wejścia analogowe są skonfigurowane w trybie prądowym, prąd nie popłynie dopóki do modułu nie zostanie dołączone zewnętrzne źródło zasilania.

Tabela A.33. Schematy podłączeń analogowych SM



A.6. Moduł baterii BB1297

BB1297 Battery Board

Tabela A.34. Specyfikacja

| Parametr | BB1297 Battery |
|-----------------------------------|--|
| Numer zamówienia | 6ES7 297-1AX30-0XA0 |
| Wymiary szer. x wys. x głęb. (mm) | 38 x 62 x 21 |
| Waga | |
| Czas podtrzymania pracy zegara | do 1 roku |
| Typ baterii | Typowa, wymiana bez użycia specjalnych narzędzi |
| LED w CPU | Sygnalizuje konieczność wymiany baterii |
| Program użytkownika | Aplikacja/system może oszacować stan naładowania baterii |

Moduł BB1297 jest używany w aplikacjach wymagających podtrzymywania pracy zegara czasu rzeczywistego po wyłączeniu zasilania CPU przez czas dłuższy niż 1 miesiąc. Właściwości modułu BB1297:

- Podtrzymuje pracę zegara czasu rzeczywistego po wyłączeniu zasilania CPU. Zegary w sterownikach S7-1200 wyposażonych w moduł BB1297 pracują prawidłowo przez czas do jednego roku bez konieczności zasilania zewnętrznego.
- Tylko jeden moduł BB1297 lub SB może być użyty w danej chwili.
- Dołączanie/odłączanie modułu SB1297 podczas pracy sterownika nie jest dopuszczalne. Moduł BB1297 można montować lub demontować tylko przy wyłączonym zasilaniu CPU. Jeżeli zasilanie CPU jest odłączone, a moduł BB1297 zostanie zdemontowany na przykład w celu wymiany baterii, pracę zegara czasu rzeczywistego podtrzyma superkondensator wbudowany w CPU.
- LED „Maint” w CPU będzie sygnalizował miganiem konieczność wymiany baterii na nową.
- Aplikacja użytkownika może monitorować stan naładowania baterii i informować użytkownika o konieczności jej wymiany za pomocą komunikatu wyświetlanego 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.35. 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, ±2,5, lub 0 do 20 mA | ±10 V, ±5 V, ±2,5 V, lub 0 do 20 mA |
| Rozdzielcość | 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 | ±0,3%/±0,6% pełnego zakresu | ±0,1%/±0,2% pełnego zakresu |

| Dane techniczne | CPU | SB | SM |
|--|--|--|--|
| Zakres przerzutu górnego i dolnego (słowo danych) (Przypis 1) | Napięcie: Od 27,649 do 32,511 Prąd: N/A | Napięcie: Od 32,511 do 27,649/ Od -27,649 do -32,512 Prąd: Od 32,511 do 27,649/ od 0 do -4864 | Napięcie: Od 32,511 do 27,649/od -27,649 do -32,512 Prąd: Od 32,511 do 27,649/od 0 do -4864 |
| Przepelnienie górne i dolne (słowo danych) (Przypis 1) | Napięcie: Od 32,512 do 32,767 Prąd: N/A | Napięcie: Od 32,767 do 32,512/ od -32,513 do -32,768 Prąd: Od 32,767 do 32,512/ od -4865 do -32,768 | Napięcie: Od 32,767 do 32,512/od -32,513 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 | Różnicowe: $\geq 100\text{ k}\Omega$ | Różnicowe: 220 kΩ (napięcie), 250 Ω (prąd) Tryb sumacyjny: 55 kΩ (napięcie), 55 kΩ (prąd) | Różnicowe: 9 MΩ (napięcie), 250 Ω (prąd) Tryb sumacyjny: 4,5 MΩ (napięcie), 4,5 MΩ (prąd) |
| 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 | Przepelnienie górne i dolne | Przepelnienie górne i dolne | Przepelnienie górne i dolne (Przypis 3) Za niskie napięcie 24 VDC |

Przypis 1: Zobacz zakresy pomiarowe wejść analogowych dla napięcia i prądu (strona 257), aby określić zakres przerzutu górnego i dolnego oraz zakres przepelnienia górnego i dolnego.

Przypis 2: Zobacz czasy odpowiedzi skokowej (strona 258) aby określić wartości wygładzenia i tłumienia zakłóceń.

Przypis 3: SM 1231 AI 4 (13-bitowy): Jeśli napięcie większe niż +30 VDC lub mniejsze niż -15 VDC zostanie przyłożone do wejścia, to wartość wynikowa nie będzie znana, a odpowiadające jej przepelnienie górne lub dolne może nie być aktywne.

A.7.2. Napięciowe i prądowe zakresy pomiarowe dla analogowych wejścia (AI)

Tabela A.36. Napięciowa reprezentacja wejścia analogowego

| System | | Zakres pomiarowy napięcia | | | | | |
|-------------|-----------------|---------------------------|----------|----------|----------|-------------------------------------|--------------------------|
| Dziesiętnie | Heksadecymalnie | ±10 V | ±5 V | ±2,5 V | 1,25 V | 0 do 10 V | |
| 32767 | 7FFF | 11,851 V | 5,926 V | 2,963 V | 1,481 V | Przepelenie górne | 11,851 V |
| 32512 | 7F00 | | | | | | Przepelenie górne |
| 32511 | 7EFF | 11,759 V | 5,879 V | 2,940 V | 1,470 V | Zakres przerzutu od góry | 11,759 V |
| 27649 | 6C01 | | | | | | Zakres przerzutu od góry |
| 27648 | 6C00 | 10 V | 5 V | 2,5 V | 1,250 V | Zakres nominalny | 10 V |
| 20736 | 5100 | 7,5 V | 3,75 V | 1,875 V | 0,938 V | | 7,5 V |
| 1 | 1 | 361,7 µV | 180,8 µV | 90,4 µV | 45,2 µV | | 361,7 µV |
| 0 | 0 | 0 V | 0 V | 0 V | 0 V | | 0 V |
| -1 | FFFF | | | | | Ujemne wartości nie są przetwarzane | |
| -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 | | |
| -27649 | 93FF | | | | | | |
| -32512 | 8100 | -11,759 V | -5,879 V | -2,940 V | -1,470 V | Zakres przerzutu od dołu | |
| -32513 | 80FF | | | | | | |
| -32768 | 8000 | -11,851 V | -5,926 V | -2,963 V | -1,481 V | Przepelenie dolne | |

Tabela A.37. Prądowa reprezentacja wejścia analogowego

| System | | Zakres pomiarowy prądu | | |
|-------------|-----------------|------------------------|------------------|--------------------------|
| Dziesiętnie | Heksadecymalnie | Od 0 do 20 mA | Od 4 mA do 20 mA | |
| 32767 | 7FFF | 23,70 mA | 22,96 mA | Przepelenie górne |
| 32512 | 7F00 | | | |
| 32511 | 7EFF | 23,52 mA | 22,81 mA | Zakres przerzutu od góry |
| 27649 | 6C01 | | | |
| 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 | | | Przepelenie dolne |
| -32768 | 8000 | | | |

A.7.3. Odpowiedź skokowa analogowych wejść (AI)

Poniższa tabela pokazuje czasy odpowiedzi skokowej wejścia analogowego dla CPU, SB oraz SM.

Tabela A.38. 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 |
| 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) | • 0,625 | • 4,17 | • 5 | • 25 |
| | • (8 kanałów) | • 1,25 | • 4,17 | • 5 | • 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.39. 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.40. 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.41. Specyfikacja analogowych wyjść (AQ)

| Dane techniczne | SB | SM |
|---|---|--|
| Typ | Napięcie lub prąd | Napięcie lub prąd |
| Zakres | ±10 V lub od 0 do 20 mA | ±10 V lub od 0 do 20 mA |
| Rozdzielcość | 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 |
| Czas ustalania (do 95% nowej wartości) | Napięcie: 300 µS (R), 750 µS (1 uF) Prąd: 600 µS (1 mH), 2 ms (10 mH) | Napięcie: 300 µS (R), 750 µS (1 uF) Prąd: 600 µS (1 mH), 2 ms (10 mH) |
| Impedancja obciążenia | Napięcie: ≥ 1000 Ω Prąd: ≤ 600 Ω | Napięcie: ≥ 1000 Ω Prąd: ≤ 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"> • Przepelnienie górne i dolne • Zwarcie do uziemienia (tylko tryb napięciowy) • Przerwa przewodu (tylko tryb prądowy) | <ul style="list-style-type: none"> • Przepelnienie 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 259) dla określenia pełnego zakresu. | |

A.7.6. Napięciowe i prądowe zakresy pomiarowe dla analogowych wyjść (AQ)

Tabela A.42. Napięciowa reprezentacja wyjścia analogowego

| System | | Zakres pomiarowy napięcia | |
|-------------|-----------------|---------------------------|--------------------------|
| Dziesiętnie | Heksadecymalnie | ± 10 V | |
| 32767 | 7FFF | Przypis 1 | Przepelnienie górne |
| 32512 | 7F00 | Przypis 1 | |
| 32511 | 7EFF | 11,76 V | Zakres przerzutu od góry |
| 27649 | 6C01 | | |
| 27648 | 6C00 | 10 V | |
| 20736 | 5100 | 7,5 V | |
| 1 | 1 | 361,7 µ V | |
| 0 | 0 | 0 V | Zakres nominalny |
| -1 | FFFF | -361,7 µ V | |
| -20736 | AF00 | -7,5 V | |
| -27648 | 9400 | -10 V | |

| System | | Zakres pomiarowy napięcia | |
|-------------|-----------------|---------------------------|--------------------------|
| Dziesiętnie | Heksadecymalnie | $\pm 10 \text{ V}$ | |
| -27649 | 93FF | | Zakres przerzutu od dołu |
| -32512 | 8100 | -11,76 V | |
| -32513 | 80FF | Przypis 1 | |
| -32768 | 8000 | Przypis 1 | Przepełnienie dolne |

¹ W przypadku wystąpienia przepełnienia górnego lub dolnego, wyjścia analogowe będą zachowywały się zgodnie z właściwościami konfiguracji urządzenia ustawionymi dla modułu analogowych sygnałów. Parametr „Reaction to CPU STOP” może przyjmować dwie wartości: Use substitute value (Użyj zastępczej wartości) lub Keep last value (Zachowaj ostatnią wartość).

Tabela A.43. 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, wyjścia analogowe będą zachowywały się zgodnie z właściwościami konfiguracji urządzenia ustawionymi dla modułu analogowych sygnałów. Parametr „Reaction to CPU STOP” może przyjmować dwie wartości: Use substitute value (użyj zastępczej wartości) lub 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 zmierzonych (np. 25,3 stopni będzie wyrażone jako 253 (dziesiętnie)).

Moduły RTD (SB 1231 RTD oraz SM 1231 RTD) mierzą wartość rezystancji podłączanej 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 zmierzonych (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 wstępna 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 płytka sygnałowych TC i RTD.

Tabela A.44. Ogólna specyfikacja

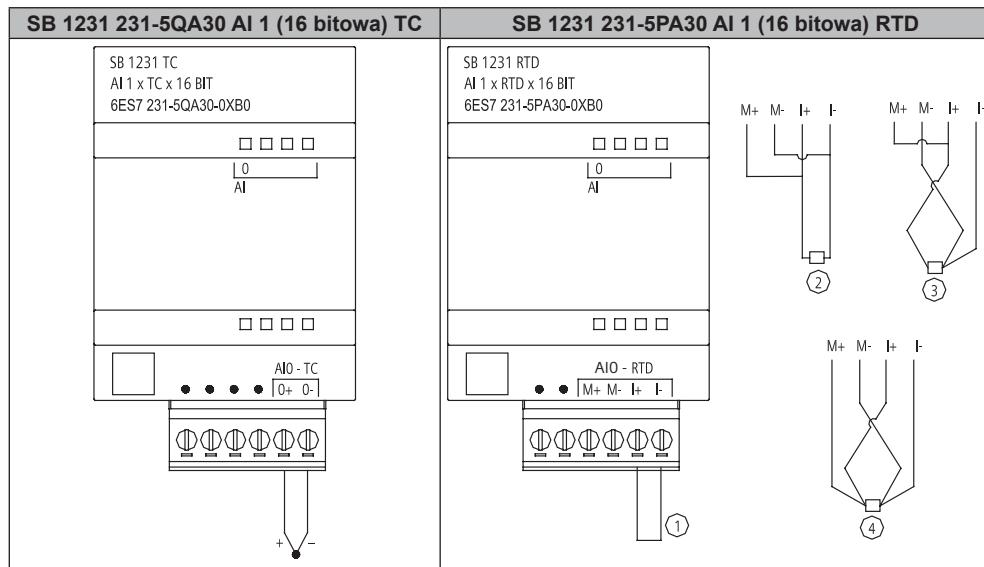
| Dane techniczne | SB 1231 AI 1 (16 bitowa) TC | SB 1231 AI 1 (16 bitowa) RTD |
|-----------------------------|--|--|
| Nr zamówieniowy | 6ES7 231-5QA30-0XB0 | 6ES7 231-5PA30-0XB0 |
| Wymiary W × H × D (mm) | 38 × 62 × 21 mm | 38 × 62 × 21 mm |
| Masa | 35 gramów | 35 gramów |
| Pobór mocy | 0,5 W | 0,7 W |
| Pobór prądu (magistrala SM) | 5 mA | 5 mA |
| Pobór prądu (24 VDC) | 20 mA | 25 mA |
| Liczba wejść (strona 258) | 1 | 1 |
| Typ | Nieuziemiona TC/mV | Odniesione do modułu RTD/Ω |
| Diagnostyka | <ul style="list-style-type: none"> Przepelnienie górne i dolne^{1,2} Przerwanie obwodu³ | <ul style="list-style-type: none"> Przepelnienie górne i dolne^{1,2} Przerwanie obwodu³ |

¹ Informacja o alarmie diagnostycznym z powodu przepelenienia górnego lub dolnego będzie zgłoszona w wartościach danych analogowych nawet jeśli alarma zostały wyłączone podczas konfiguracji modułu.

² RTD: Wykrywanie przepelenienia 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.45. 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.46. Ogólna specyfikacja

| Dane techniczne | SM 1231 AI 4 × RTD × 16 bit | SM 1231 AI 8 × RTD × 16 bit |
|--------------------------------------|--|--|
| Nr zamówieniowy | 6ES7 231-5PD30-0XB0 | 6ES7 231-5PF30-0XB0 |
| 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 258) | 4 | 8 |
| Typ | Odniesione do modułu RTD/Ω | Odniesione do modułu RTD/Ω |
| Diagnostyka | <ul style="list-style-type: none"> Przepelnienie górne i dolne^{1,2} Za niskie napięcie 24 VDC² Przerwanie obwodu (tylko tryb prądowy)⁴ | <ul style="list-style-type: none"> Przepelnienie górne i dolne^{1,2} Za niskie napięcie 24 VDC² Przerwanie obwodu (tylko tryb prądowy)⁴ |

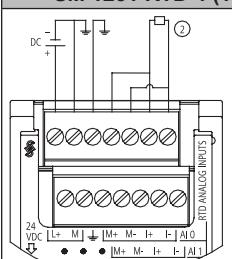
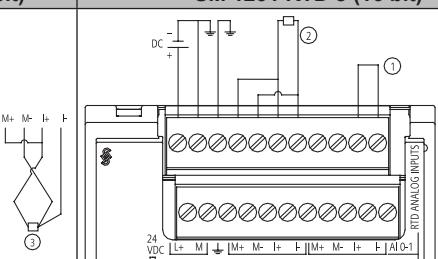
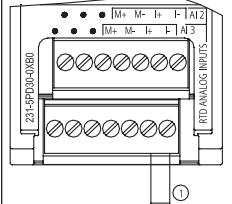
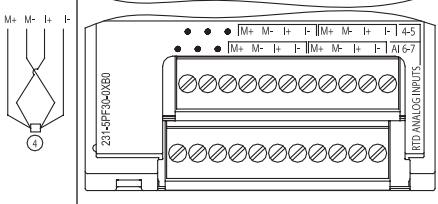
¹ 20,4 do 28,8 VDC (Class 2, ograniczone zasilanie lub zasilanie czujnika z CPU).

² Informacja o alarmie diagnostycznym z powodu przepelnienia górnego lub dolnego będzie zgłoszona w wartościach danych analogowych nawet jeśli alarma zostały wyłączone podczas konfiguracji modułu.

³ Wykrywanie przepelnienia 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.47. Schemat podłączeń modułu rozszerzeń RTD

| SM 1231 RTD 4 (16 bit) | SM 1231 RTD 8 (16 bit) | Odniesienie |
|---|---|--|
|  |  | <ul style="list-style-type: none"> ① Pętla zwrotna nieużywanego wejścia RTD ② 2-przewodowa RTD ③ 3-przewodowa RTD ④ 4-przewodowa RTD |
|  |  | |

A.8.3. Specyfikacja modułu SM 1231 TC

Tabela A.48. Ogólna specyfikacja

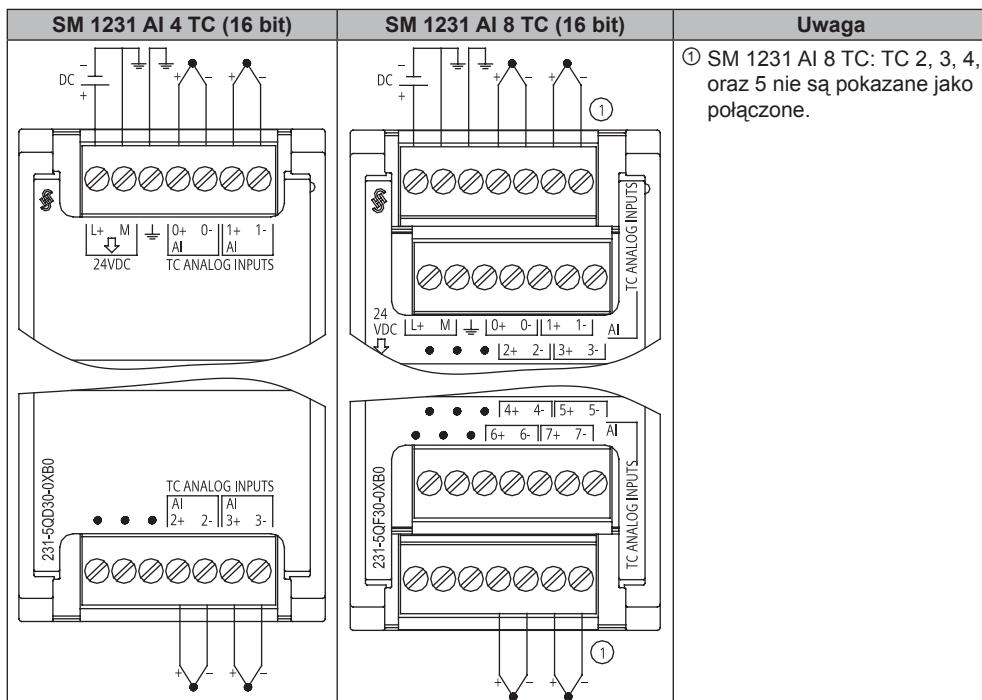
| Dane techniczne | SM 1231 AI4 × 16 bit TC | SM 1231 AI8 × 16 bit TC |
|--------------------------------------|---|---|
| Nr zamówieniowy | 6ES7 231-5QD30-0XB0 | 6ES7 231-5QF30-0XB0 |
| Wymiary W × H × D (mm) | 45 × 100 × 75 | 45 × 100 × 75 |
| Masa | 180 gramów | xxx gramów |
| Pobór mocy | 1,5 W | 1,5 W |
| Pobór prądu (magistrala SM) | 80 mA | 80 mA |
| Pobór prądu ¹ (24 VDC) | 40 mA | 40 mA |
| Liczba wejść (strona 258) | 4 | 8 |
| Typ | Nieuziemiona TC/mV | Nieuziemiona TC/mV |
| Diagnostyka | <ul style="list-style-type: none"> • Przepelenienie gorne i dolne² • Za niskie napięcie 24 VDC² • Przerwanie obwodu (tylko tryb pradowy)³ | <ul style="list-style-type: none"> • Przepelenienie gorne i dolne² • Za niskie napięcie 24 VDC² • Przerwanie obwodu (tylko tryb pradowy)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 przepelenienia 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.49. Schemat podłączeń modułu rozszerzeń TC



A.8.4. Specyfikacja analogowego wejścia dla RTD oraz TC (SM i SB)

Tabela A.50. 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 | | Zobacz tabele typów RTD/TC: <ul style="list-style-type: none"> • Zakres nominalny (słowo danych) • Zakres przerzutu od góry i od dołu (słowo danych) • Przepełnienie górne i dolne (słowo danych) |
| Rozdzielcość | 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 MΩ |
| Izolacja | Sygnalu zewnętrznego od logiki | 500 VAC |
| | Pola 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 267) • TC (strona 266) |
| Powtarzalność | | ±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 268) • TC (strona 266) |
| Błąd styku odniesienia | | <ul style="list-style-type: none"> • RTD: Nie dotyczy • TC: ±1,5°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.51. 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 zgłasiane są jako -32 768.

² Wartości z termopar powyżej maksymalnego górnego zakresu zgłasiane 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-kanałowego SM TC: Dokładność wyników może spaść w przypadku obecności w pobliżu czujników nadajników częstotliwości radiowych (częstotliwości 970–990 MHz).

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.52. Wybór filtra termopary i czasy odświeżania

| Tłumiona częstotliwość [Hz] | Czas całkowania [ms] | Czas odświeżania [s] | | |
|-----------------------------|----------------------|----------------------|---------------|---------------|
| | | 1-kanałowa SB | 4-kanałowy SM | 8-kanałowy 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 rozdzielcość i dokładność modułu, jeśli tłumiona jest częstotliwość 400 Hz, należy użyć czasu całkowania wynoszącego 10 ms. Taka wartość pozwoli również tłumić szумy o częstotliwości 100 Hz oraz 200 Hz.

A.8.7. Tabela wyboru typu czujnika RTD

Tabela A.53. 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 przy normalnym zakresie od -20°C do 60°C |
|---|------------|--------------------------------|------------------------|------------------------|---------------------------------|--|---|
| Pt 0,003850 ITS90 DIN EN 60751 | 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 100 | -243,0°C | -200,0°C | 850,0°C | 1000,0°C | ±0,5°C | ±1,0°C |
| Pt 0,003916 | Pt 200 | -243,0°C | -200,0°C | 850,0°C | 1000,0°C | ±0,5°C | ±1,0°C |
| Pt 0,003920 | 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 |
| Cu 0,004260 | Cu 10 | -60,0°C | -50,0°C | 200,0°C | 240,0°C | ±1,0°C | ±2,0°C |
| | Cu 50 | -60,0°C | -50,0°C | 200,0°C | 240,0°C | ±0,6°C | ±1,2°C |
| | Cu 100 | | | | | | |
| Cu 0,004280 | Cu 10 | -240,0°C | -200,0°C | 200,0°C | 240,0°C | ±1,0°C | ±2,0°C |
| | Cu 50 | -240,0°C | -200,0°C | 200,0°C | 240,0°C | ±0,7°C | ±1,4°C |
| | Cu 100 | | | | | | |

¹ Wartości z RTD poniżej minimalnego dolnego zakresu zgłasiane są jako -32 768.

² Wartości z RTD powyżej maksymalnego górnego zakresu zgłasiane są jako 32 767.

Tabela A.54. Rezystancje

| Zakres | Min. dolny zakres | Nominalny dolny zakres | Nominalny górny zakres | Maks. górny zakres ¹ | Dokładność przy normalnym zakresie przy 25°C | Dokładność przy normalnym zakresie od -20°C do 60°C |
|--------|-------------------|------------------------|------------------------|---------------------------------|--|---|
| 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 z RTD powyżej maksymalnego górnego zakresu zgłasiane są jako 32 767.

A.8.8. Wybór filtra RTD i czasy odświeżania

Tabela A.55. 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 rozdzielcość i dokładność modułu, jeśli tłumiona jest częstotliwość 400 Hz, należy użyć czasu całkowania wynoszącego 10 ms. Taka wartość pozwoli również tłumić szумy o częstotliwości 100 Hz oraz 200 Hz.

A.9. Interfejsy komunikacyjne

Pełna lista modułów dostępnych dla S7-1200 znajduje się w podręczniku S7-1200 System Manual lub na internetowej stronie wsparcia technicznego (www.siemens.pl/simatic).

A.9.1. PROFIBUS master/slave

A.9.1.1. Moduł CM 1242-5 PROFIBUS slave

Tabela A.56. 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 zwarciem) dla zakończenia magistrali.

Interfejs PROFIBUS

Tabela A.57. Wyprowadzenia z gniazda D-sub

| | Pin | Opis | Pin | Opis |
|---|-----|--|---------|---------------------------|
| | | | Pin | |
| 6 | 1 | – Nieużywane – | 6 | P5V2: zasilanie +5 V |
| 7 | 2 | – Nieużywane – | 7 | – Nieużywane – |
| 8 | 3 | RxD/TxD-P: Linia danych B | 8 | RxD/TxD-N: Linia danych A |
| 9 | 4 | RTS | 9 | – Nieużywane – |
| | 5 | M5V2: Potencjał odniesienia danych (uziemienie DGND) | Obudowa | Masa obudowy |

A.9.1.2. Moduł CM 1243-5 PROFIBUS master**Tabela A.58.** 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 | <ul style="list-style-type: none"> 24 V • 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) |
| 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 zwarciem) dla zakończenia magistrali.

Interfejs PROFIBUS

Tabela A.59. Wyprowadzenia z gniazda D-sub

| | Pin | Opis | Pin | Opis |
|--|-----|-------------------------------------|---------|---|
| | 1 | – Nieużywane – | 6 | VP: Zasilanie +5 V tylko dla oporników zamykających magistralę nie można stosować do zasilania zewnętrznych odbiorników |
| | 2 | – Nieużywane – | 7 | – Nieużywane – |
| | 3 | RxD/TxD-P: Linia danych B | 8 | RxD/TxD-N: Linia danych A |
| | 4 | CNTR-P: RTS | 9 | – Nieużywane – |
| | 5 | DGND: masa dla sygnałów z danymi VP | Obudowa | Masa obudowy |

Przewód PROFIBUS

Uwaga

Uziemienie ekranu przewodu PROFIBUS

Ekran przewodu PROFIBUS musi być uziemiony.

W tym celu należy ściągnąć izolację z końca przewodu i podłączyć ekran do uziemienia.

A.9.2. Procesor komunikacyjny GPRS

Uwaga

Procesory komunikacyjne (CP) nie mają dopuszczenia morskiego.

Następujące moduły nie mają 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.60.** 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 |
| GRPS | 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 | 24 V |
| <ul style="list-style-type: none"> • minimum • maksimum | <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) |
| 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 promień 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 | |
| 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°C | |
| Łatopalność | UL 94 V2 | |
| Materiały obudowy | ABS Polylac PA-765, jasnoszary (RAL 7035) | |
| Wymiary (W × L × H) w mm | 70,5 × 146,5 × 20,5 | |
| Masa | 130 g | |

A.9.3. TeleService (TS)

Następujące podręczniki zawierają specyfikacje techniczną dla urządzenia TS Adapter IE Basic oraz modułarnego urządzenia TS Adapter:

- Industrial Software Engineering Tools Modular TS Adapter
- Industrial Software Engineering Tools TS Adapter IE Basic

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.61. Specyfikacja ogólna

| Dane techniczne | CB 1241 RS485 |
|-------------------|---------------------|
| Nr zamówieniowy | 6ES7 241-1CH30-1XB0 |
| Wymiary W × H × D | 38 × 62 × 21 |
| Masa | 40 gramów |

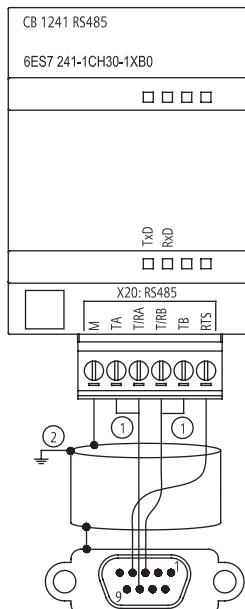
Tabela A.62. Urządzenia nadawcze i odbiorcze

| Dane techniczne | CB 1241 RS485 |
|--|--|
| Typ | RS485 (2-przewodowy półduplex) |
| Zakres napięcia w trybie wspólnym | od -7 V do +12 V, 1 sekunda, 3 VRMS ciągłe |
| 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 127 Ω , połączenie do RS485 Pin 3 Krótki Pin TA do Pin T/RA, efektywna impedancja zakończenia wynosi 127 Ω , 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 | |
| Sygnal RS485 do uziemienia szyny | 500 VAC, 1 minuta |
| Sygnal RS485 do wspólnej logiki CPU | |
| 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.63. 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 (zakańcza tylko końcowe urządzenie 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 sekcji „Obciążanie i terminowanie złącza sieci RS485” w publikacji „Podręcznik systemowy S7-1200”.

Tabela A.64. Rozmieszczenie sygnałów w złączu interfejsu RS485 w module CB 1241 RS485 (6ES7 241-1CH30-1XB0)

| Pin | Złącze DB9 | X20 |
|-------|------------------------------|----------|
| 1 | RS485/Masa sygnału cyfrowego | – |
| 2 | RS485/nieużywany | – |
| 3 | RS485/TxD+ | 3 – T/RB |
| 4 | RS485/RTS | 1 – RTS |
| 5 | RS485/Masa sygnału cyfrowego | – |
| 6 | RS485/5 V (zasilanie) | – |
| 7 | RS485/nieużywany | – |
| 8 | RS485/TxD– | 4 – T/RA |
| 9 | RS485/nieużywany | – |
| Shell | | 7 – M |

A.9.4.2. Moduły CM 1241 RS485/RS232

Tabela A.65. Specyfikacja ogólna

| Dane techniczne | CM 1241 RS232 |
|---------------------|------------------|
| Wymiary (W × H × D) | 30 × 100 × 75 mm |
| Masa | 150 gramów |
| Pobór mocy | 1,1 W |
| Z +5 VDC | 220 mA |

Tabela A.66. Urządzenia nadawcze i odbiorcze

| Dane techniczne | | Opis |
|---------------------------------------|---|---|
| Typ | | RS232 (pełen duplex). |
| Urządzenie nadawcze (RS485) | Zakres napięcia w trybie wspólnym | od -7 V do +12 V, 1 sekunda, 3 VRMS ciągłe |
| | 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 | 10 kΩ do +5 V na B, PROFIBUS Pin 3 10 kΩ do GND na A, PROFIBUS Pin 8 |
| Urządzenie nadawcze (RS232) | Napięcie wyjścia urządzenia nadawczego | +/- 5 V min. przy $R_L = 3 \text{ k}\Omega$ |
| | Napięcie transmisji | +/- 15 VDC maks. |
| Odbiornik | Impedancja wejścia odbiornika | RS232: 3 kΩ min. |
| | Odbiornik: próg/czułość | RS232: 0,8 V min. stan niski, 2,4 maks. stan wysoki, typowa histereza 0,5 V |
| | Napięcie wejścia odbiornika (Tylko RS232) | +/- 30 VDC maks. |
| Izolacja | Sygnal RS485 do uziemienia szyny Sygnal RS485 do wspólnej logiki CPU | 500 VAC, 1 minuta |
| Długość przewodu, ekranowany | | RS232: 10 m |
| 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 stop | | 1 (domyślnie), 2 |
| Sterowanie strumieniem danych (RS232) | | Nie obsługuje |
| Czas oczekiwania | | od 0 do 65 535 ms |

Tabela A.67. Wtyczka RS232 dla CM 1241 i okablowanie

| 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 | | 2 | 7 RTS | Żądanie wysłania: Wyjście |
| 3 TxD | Dane wysyłane do DCE: Wyjście | | 3 | 8 CTS | Gotowość do wysłania: Wejście |
| 4 DTR | Gotowość terminala danych: Wyjście | | 4 | 9 RI | Sygnal dzwonka (nieużywany) |
| 5 GND | Masa sygnałowa | | 5 | SHELL | Masa obudowy |

A.9.4.3. Moduły CM 1241 RS422/485

Tabela A.68. Specyfikacja ogólna

| Dane techniczne | CM 1241 RS422/485 |
|---------------------|---------------------|
| Numer zamówienia | 6ES7 241-1CH31-0XB0 |
| Wymiary (W × H × D) | 30 × 100 × 75 mm |
| Masa | 155 g |

Tabela A.69. Urządzenia nadawcze i odbiorcze

| Dane techniczne | Opis |
|--|---|
| Typ | RS422 lub RS485, 9-stykowe złącze żeńskie DB9 |
| Zakres napięcia w trybie wspólnym | Od -7 V do +12 V, 1 sekunda, 3 VRMS ciągłe |
| 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 kΩ do +5 V na B, PROFIBUS Pin 3 Od 10 kΩ do GND na A, PROFIBUS Pin 8 Opcjonalnie dostępny generator polaryzacji spoczynkowej (bias) lub bez generatora polaryzacji spoczynkowej. We wszystkich przypadkach konieczne są zewnętrzne terminatory. Dodatkowe informacje są dostępne w odpowiednich rozdziałach podręcznika systemowego S7-1200. |
| Impedancja wejścia odbiornika | +/- 5 V min. z uwzględnieniem terminatora |
| Odbiornik: próg/czułość | +/- 0,2 V maks. typowa histereza |
| Izolacja | |
| Sygnal RS485 do uziemienia szyny | 500 VAC, 1 minuta |
| Sygnal RS485 do wspólnej logiki CPU | |
| Długość przewodu, ekranowany | 1000 m maks. (w zależności od szybkości transmisji) |
| 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 |

| Dane techniczne | Opis |
|---------------------------------------|---|
| Parzystość | Bez parzystości (domyślnie), parzysty, nieparzysty, Mark (bit parzystości ustawiony zawsze na 1), Space (bit parzystości ustawiony zawsze na 0) |
| Liczba bitów stop | 1 (domyślnie), 2 |
| Sterowanie strumieniem danych (RS232) | W trybie RS422 obsługiwany protokół XON/XOFF |
| Czas oczekiwania | od 0 do 65 535 ms |

Tabela A.70. Zasilanie

| Dane techniczne | CB 1241 RS422/485 |
|-----------------|-------------------|
| Pobór mocy | 1,2 W |
| Od +5 VDC | 240 mA |

Tabela A.71. Wtyczka RS485 lub RS422 (żeńska)

| 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 | Stosowany w trybie RS422 Nie wykorzystywany w trybie RS385: wyjście | | 5 | Niepodłączona | |
| 3 TxD+ | Sygnal B (RxD/TxD+): Wejście/Wyjście | | 4 | | |
| 4 RTS ² | Żądanie wysłania (poziom TTL): Wyjście | | 3 | 8 TXD-: Sygnal A (RxD/TxD-): Wejście/Wyjście | |
| 5 GND | Masa logiki lub sygnałów komunikacyjnych | | 2 | Connected for RS422 Not used for RS485: Output | |
| | | | 1 | | |
| | | | | SHELL | Masa obudowy |
| | | | | | |

¹ Styki 2 i 9 są wykorzystywane jako wyjścia wyłącznie w trybie RS422.

² Sygnał RTS ma poziomy TTL i może być wykorzystywany do sterowania innego urządzenia z interfejsem komunikacyjnym pracującym w trybie half-duplex. Sygnał jest aktywny podczas nadawania i nieaktywny w każdym innym przypadku.

A.10. Urządzenia towarzyszące

A.10.1. Zasilacz PM1207

PM 1207 jest modułem zasilającym dla S7-1200. Ma on następujące funkcje:

- wejście 120/230 VAC, wyjście VDC/2,5A
- Nr zamówieniowy 6ESP 332-1SH71

Więcej informacji o tym produkcie oraz jego dokumentacja znajduje się na stronie wsparcia technicznego (www.siemens.pl/simatic).

A.10.2. Kompaktowy switch CSM 1277

CSM 1277 jest kompaktowym modułem 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 dołączenia się w przemysłową sieć.
- 3-biegunkowe 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 stronie wsparcia technicznego (www.siemens.pl/simatic).