



S7-1200

Instrukcje programowania – instrukcje sterujące wykonywaniem programu i obsługujące komunikację

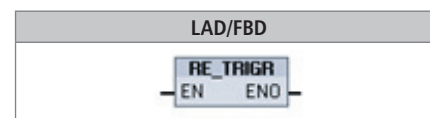


Kontynuujemy opis instrukcji programowania obsługiwanych przez sterowniki S7-1200. W tym miesiącu skupiamy się na omówieniu instrukcji sterujących przebiegiem wykonywania programu, zaczniemy także prezentować instrukcje odpowiadające za obsługę interfejsu sieciowego Ethernet, który jest standardowym wyposażeniem jednostek centralnych z serii S7-1200. Po raz kolejny widać, że możliwości języków programowania współczesnych PLC coraz bardziej zbliżają się do mechanizmów dostępnych do niedawna wyłącznie dla programistów piszących aplikacje w języku C.

Instrukcja kasowania timera nadzorującego pracę CPU

Instrukcja RE_TRIGR (*re-trigger scan time watchdog*) jest stosowana do wydłuża-

nia maksymalnego dopuszczalnego czasu zanim timer układu dozorującego poprawność wykonywania cyklu programu wygeneruje błąd.



Instrukcja RE_TRIGR służy do ponownego wyzwolenia timera nadzorującego poprawność wykonywania cyklu programu podczas pojedynczego cyklu. Dzięki temu, od czasu ostatniego wykonania funkcji RE_TRIGR dopuszczalny maksymalny czas cyklu programu wydłuża się o jeden (najdłuższy) okres cyklu programu.

Jednostka CPU w systemie S7-1200 ogranicza użycie instrukcji RE_TRIGR do cyklu programu, przykładowo do OB1 i funkcji

wywoływanych z cyklu programu. Oznacza, że jeżeli RE_TRIGR jest wywołana z dowolnego OB z listy OB cyklu programu, to timer układu dozoru jest kasowany i ENO = EN. Jeżeli RE_TRIGR jest wykonana z rozruchowego OB, OB przerwań lub OB obsługi błędów, to ENO = FALSE i timer układu dozoru nie jest kasowany.

Ustawianie maksymalnego czasu cyklu programu PLC

Użytkownik może ustawić maksymalny czas cyklu programu podczas konfiguracji urządzenia PLC w *cycle time*.

Monitor czasu cyklu	Minimalna wartość	Maksymalna wartość	Domyślna wartość
Maksymalny czas cyklu	1 ms	6000 ms	150 ms

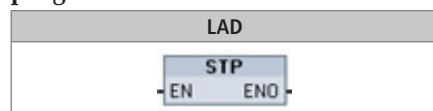
Limit czasu układu dozoru

Jeżeli przed zakończeniem cyklu programu upłynie maksymalny czas cyklu programu, to zostanie wygenerowany błąd. Jeżeli w programie użytkownika jest umieszczony blok kodu obsługi błędu OB80, to PLC wykonuje OB80, a w nim można dodać odpowiedni program reakcji na tę sytuację. Kiedy w programie nie ma OB80, wtedy pierwsze przekroczenie limitu czasu jest ignorowane.

Gdy podczas tego samego cyklu programu ponownie zostanie przekroczony maksymalny czas cyklu programu (czyli łącznie dwa maksymalnie długie cykle), to generowany jest błąd powodujący przejście PLC do trybu STOP.

W trybie STOP wykonywanie programu użytkownika jest wstrzymywane, podczas gdy komunikacja i diagnostyka systemowa PLC nadal działają.

Instrukcja zatrzymywania cyklu programu



Instrukcja STP (*stop PLC scan cycle*) wprowadza PLC w tryb STOP. Kiedy PLC jest w trybie STOP, wtedy wykonywanie programu użytkownika i uaktualnianie adresu wyjściowego Q z obrazu procesu są wstrzymane.

Wyjściowe stany bezpieczne, jakie pojawiają się podczas przechodzenia systemu w tryb STOP na wyjściach zintegrowanych oraz wyjściach płytki sygnałowej i modułów rozszerzeń (analogowych i cyfrowych), są definiowane podczas konfiguracji urządzenia PLC w zakładce *Properties*. Użytkownik może wybrać „zamrożenie” ostatniego stanu wyjść lub ustalić (analogowe i cyfrowe) stany bezpieczne. Wartością domyślną dla wyjść cyfrowych jest stan FALSE, a warto-

ścią domyślną dla wyjść analogowych jest poziom 0.

Jeżeli EN = TRUE, to PLC przejdzie do stanu STOP, wykonywanie programu zostanie zatrzymane, a stan ENO nie będzie miał znaczenia. W przeciwnym przypadku EN = ENO = 0.

Instrukcje pobierania kodu błędu

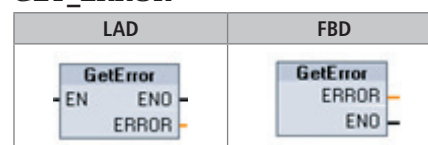
Instrukcje pobierania kodu błędu (*get error*) dostarczają informacji o błędach związanych z wykonywaniem bloków programu. Przed użyciem instrukcji GET_ERROR lub GET_ERR_ID bloki programu muszą sprawdzić atrybut *handle errors within block*: w konfiguracji właściwości bloku.

GET_ERROR sygnalizuje, że wystąpił błąd wykonania bloku programu i wypeł-

nia predefiniowaną strukturę danych błędu szczegółowymi informacjami o błędzie.

GET_ERROR_ID sygnalizuje, że wystąpił błąd wykonania bloku programu i zgłasza ID (kod identyfikacyjny) błędu.

GET_ERROR



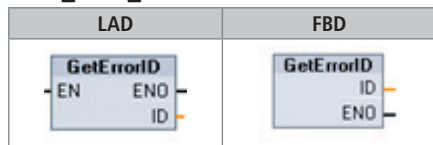
Parametr	Typ danych	Opis
ERROR	ErrorStruct	Struktura danych błędu

Element danej typu Error-Struct	Typ danych	Opis					
ERROR_ID	WORD	Identyfikator błędu					
FLAGS	BYTE	Sygnalizuje, czy błąd wystąpił podczas wywołania do innego bloku: 16#01 jeśli błąd wystąpił podczas wywołania 16#00 w przeciwnym przypadku					
REACTION	BYTE	Reakcja na błąd: 0 = zignorować; nic nie jest zapisane (błąd zapisu) 1 = zastąpić: 0) użyte jako wartość (błąd odczytu) 2 = ominąć instrukcję (błąd systemowy)					
BLOCK_TYPE	BYTE	Typ bloku, w którym wystąpił błąd: 1 = OB 2 = FC 3 = FB					
PAD_0	BYTE	Wewnętrznie wypełniany bajt dla uzyskania zgodności, wynosi 0					
CODE_BLOCK_NUMBER	UINT	Numer bloku, w którym wystąpił błąd					
ADDRESS	UDINT	Lokalizacja w wewnętrznej pamięci instrukcji, przy której wystąpił błąd					
MODE	BYTE	Wewnętrzne mapowanie dotyczące sposobu interpretowania pozostałych pól					
		Tryb	(A)	(B)	(C)	(D)	(E)
		0					
		1					Offset
		2			Area		
		3	Location	Scope		Number	
		4			Area		Offset
		5			Area	DB no.	Offset
6	PtrNo./Acc		Area	DB no.	Offset		
7	PtrNo./Acc	Slot No./Scope	Area	DB no.	Offset		
PAD_1	BYTE	Wewnętrznie wypełniany bajt dla uzyskania zgodności, nie wykorzystywany, wynosi 0					
OPERAND_NUMBER	UINT	Numer argumentu wewnętrznej instrukcji					
POINTER_NUMBER_LOCATION	UINT	(A) Położenie wskaźnika wewnętrznej instrukcji					
SLOT_NUMBER_SCOPE	UINT	(B) Położenie obszaru w wewnętrznej pamięci					
AREA	BYTE	(C) Odniesienie do obszarów pamięci podczas wykrycia błędu: L: 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE I: 16#81 Q: 16#82 M: 16#83 DB: 16#84, 85, 8A, 8B					
PAD_2	BYTE	Wewnętrznie wypełniany bajt dla uzyskania zgodności, nie wykorzystywany, wynosi 0					
DB_NUMBER	UINT	(D) DB wskazany podczas wykrycia błędu DB, w przeciwnym przypadku 0					
OFFSET	UDINT	(E) Położenie bitu wskazanego podczas wykrycia błędu (przykład: 12 = bajt 1, bit 4)					

Struktura danych parametru ERROR

Użytkownik może zmienić nazwę struktury, ale nie może zmieniać nazw występujących wewnątrz struktury.

GET_ERR_ID



Parametr	Typ danych	Opis
ID	WORD	Identyfikator błędu

Parametr ID: Wartości identyfikatora błędu jako elementu ErrorStruct ERROR_ID

ERROR_ID szesnastkowo	ERROR_ID dziesiętnie	Błąd wykonania bloku programu
2503	9475	Błąd niezainicjalizowania wskaźnika
2522	9506	Błąd odczytu – argument spoza zakresu
2523	9507	Błąd zapisu – argument spoza zakresu
2524	9508	Błąd odczytu – nieprawidłowy argument
2525	9509	Błąd zapisu – nieprawidłowy argument
2528	9512	Błąd odczytu – zgodność danych
2529	9513	Błąd zapisu – zgodność danych
2530	9520	Błąd zapisu DB
253A	9530	Globalny DB nie istnieje
253C	9532	Błędna wersja lub FC nie istnieje
253D	9533	SFC nie istnieje
253E	9534	Błędna wersja lub FB nie istnieje
253F	9535	SFB nie istnieje
2575	9589	Błąd głębokości zagnieźdzenia programu
2576	9590	Błąd alokacji danych lokalnych
2942	10562	Błąd bezpośredniego odczytu wejść
2943	10563	Błąd bezpośredniego zapisu do wyjść

Działanie

Domyślnie, PLC odpowiada na wystąpienie błędu wykonania bloku zarejestrowaniem błędu w buforze diagnostycznym i przejściem w tryb STOP. Jednakże jeżeli użytkownik umieści jedną lub więcej instrukcji GET_ERROR lub GET_ERR_ID w kodzie bloku, to tym samym blok jest skonfigurowany do obsługi błędów w ramach tego bloku. W takim przypadku PLC nie przechodzi do trybu STOP i nie rejestruje błędu w buforze diagnostycznym. Zamiast tego, informacje o błędzie są przedstawiane na wyjściu instrukcji GET_ERROR lub GET_ERR_ID.

ID. Użytkownik może uzyskać szczegółowe informacje o błędzie za pomocą instrukcji GET_ERROR lub odczytać identyfikator błędu za pomocą instrukcji GET_ERR_ID. Zwykle pierwszy błąd jest najważniejszy – kolejne błędy są tylko konsekwencją tego pierwszego.

Pierwsze wykonanie instrukcji GET_ERROR lub GET_ERR_ID w ramach bloku, zwraca pierwszy wykryty błąd jaki powstał podczas wykonywania bloku. Ten błąd mógł wystąpić gdziekolwiek między startem bloku i wykonaniem dowolnej z instrukcji GET_ERROR albo GET_ERR_ID. Kolejne wykonania instrukcji GET_ERROR lub GET_ERR_ID zwracają pierwszy błąd od czasu poprzedniego wykonania instrukcji GET_ERROR lub GET_ERR_ID. Historia błędów nie jest zachowywana i wykonanie dowolnej z tych instrukcji uzbraja system PLC do wyłapywania następnego błędu.

Dana typu ErrorStruct wykorzystywana przez instrukcję GET_ERROR może być dodana w edytorze bloku danych i edytorach bloku interfejsu, tak że program użytkownika może mieć dostęp do zawartych w niej informacji. W celu dodania tej struktury należy z rozwijanej listy wybrać ErrorStruct. Stosując unikalne nazwy, użytkownik może stworzyć wiele ErrorStruct. Nazwy elementów wewnętrznych ErrorStruct nie mogą być zmieniane.

Błędy wskazywane przez ENO

Jeżeli EN = TRUE i zostaje wykonana instrukcja GET_ERROR lub GET_ERR_ID, to:

- ENO = TRUE sygnalizuje, że wystąpił błąd wykonania bloku i informacje o błędzie są dostępne.
- ENO = FALSE sygnalizuje, że nie wystąpił żaden błąd wykonania bloku.

Użytkownik może powiązać program reakcji na błąd z ENO, które jest aktywowane po wystawieniu błędu. Jeżeli błąd istnieje, to parametry wyjściowe przechowują informacje o błędzie i program użytkownika ma do nich dostęp.

Instrukcje GET_ERROR i GET_ERR_ID mogą być wykorzystane do przesłania informacji o błędzie z aktualnie wykonywanego bloku (zwanego blokiem) do bloku wywołującego. W celu uzyskania ostatecznego statusu wykonania bloku wywołującego, instrukcje należy umieścić w ostatnim obwodzie bloku programu wywołującego.

Instrukcje obsługujące komunikację Ethernet Otwarcie komunikacji ethernetowej

Opis TSEND_C

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 sobie funkcje TCON, TDISCON i TSEND.

Funkcja TSEND_C

W celu ustanowienia połączenia należy wykonać TSEND_C z CONT = 1.

Po pomyślnym ustanowieniu połączenia, TSEND_C ustawia parametr DONE na jeden cykl.

W celu zakończenia połączenia komunikacyjnego należy wykonać TSEND_C z CONT = 0. Komunikacja zostanie natychmiast przerwana. Ma to również wpływ na stację odbiorczą. Połączenie zostanie tam zakończone i dane z bufora odbiorczego mogą zostać utracone.

W celu wysłania danych ustanowionym kanałem połączeniowym należy wykonać TSEND_C z narastającym zboczem na REQ. Po pomyślnym wykonaniu operacji wysłania danych, TSEND_C ustawia parametr DONE na jeden cykl.

W celu ustanowienia połączenia i wysłania danych należy wykonać TSEND_C z CONT = 1 i REQ = 1. Po pomyślnym wykonaniu operacji wysłania danych, TSEND_C ustawia parametr DONE na jeden cykl.

Opis TRCV_C

TRCV_C ustala ze stacją partnerską 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 sobie funkcje TCON, TDISCON i TRCV.

Funkcja TRCV_C

Ustanowienie połączenia: należy wykonać TRCV_C z parametrem CONT = 1.

Odbiór danych: należy wykonać TRCV_C z parametrem EN_R = 1. Dane można odbierać w sposób ciągły gdy EN_R = 1 i CONT = 1.

Zakończenie połączenia: należy wykonać TRCV_C z CONT = 0. Komunikacja zostanie natychmiast przerwana i dane mogą zostać utracone.

Tryby odbiorcze

TRCV_C obsługuje te same tryby odbiorcze co instrukcja TRCV. W następującej tabeli znajdują się informacje jak dane są wprowadzane do obszaru odbiorczego.

Wariant protokołu	Wprowadzanie danych do obszaru odbiorczego	Parametr „connection_type”	Wartość parametru LEN
TCP	Tryb „ad hoc”	B#16#11	0

TCP	Odbiór danych o określonej długości	B#16#11	<> 0
ISO on TCP	Kontrolowane przez protokół	B#16#12	0 (rekomendowana) lub <> 0

Tryb TCP *ad hoc*

Tryb *ad hoc* istnieje tylko w wariantcie protokołu TCP. Użytkownik może ustawić tryb *ad hoc* przypisując parametrowi LEN wartość 0. Obszar odbiorczy jest identyczny z obszarem uformowanym przez DATA. Maksymalnie są odbierane 1472 bajty.

TCP – odbiór danych o określonej długości

Użytkownik może ustawić tryb odbioru danych o określonej długości przypisując parametrowi LEN wartość inną niż 0. Obszar odbiorczy jest definiowany przez parametry LEN i DATA.

ISO on TCP – przepływ danych kontrolowany protokołem

W wariantcie protokołu *ISO on TCP*, przesyłane dane są kontrolowane przez protokół. Obszar odbiorczy jest definiowany przez parametry LEN i DATA.

Uwaga

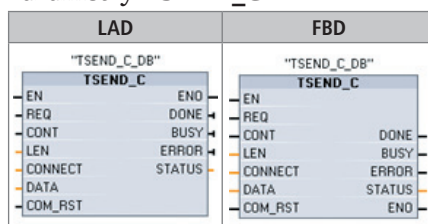
Ze względu na asynchroniczne przetwarzanie instrukcji TSEND_C, użytkownik musi utrzymywać w obszarze nadawczym spójne dane aż do chwili gdy parametr DONE lub parametr ERROR przyjmie wartość TRUE. W przypadku TSEND_C, wartość TRUE parametru DONE oznacza, że dane zostały pomyślnie wysłane. Nie oznacza to natomiast, że połączona stacja partnerska CPU odczytała bufor odbiorczy. Ze względu na asynchroniczne przetwarzanie TRCV_C, dane w obszarze odbiorczym są spójne tylko wtedy, kiedy DONE = 1.

W poniższej tabeli przedstawiono związki między parametrami BUSY, DONE i ERROR.

BUSY	DONE	ERROR	Opis
TRUE	nieistotny	nieistotny	Zadanie jest w toku wykonywania
FALSE	TRUE	FALSE	Zadanie zostało pomyślnie zakończone
FALSE	FALSE	TRUE	Zadanie zostało zakończone z błędem. Przyczynę błędu można odczytać z parametru STATUS

FALSE	FALSE	FALSE	Nowe zadanie nie zostało przydzielone
-------	-------	-------	---------------------------------------

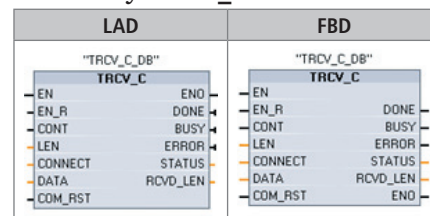
Parametry TSEND_C



Parametr	Typ parametru	Typ danych	Opis
REQ	INPUT	BOOL	Parametr sterujący REQ rozpoczyna wysyłanie zadania w trakcie połączenia opisanego w CONNECT w chwili wystąpienia narastającego zbocza.
CONT	INPUT	BOOL	Parametr sterujący CONT: 0: rozłącz 1: ustanów i utrzymuj połączenie
LEN	INPUT	INT	Maksymalna liczba bajtów do wysłania w zadaniu. Por. zależność między CPU i Protocol Variant oraz Transferable Data Length.
CONNECT	IN_OUT	ANY	Wskaźnik do opisu połączenia.
DATA	IN_OUT	ANY	Obszar nadawczy; zawiera adres i długość danych do nadania.
COM_RST	IN_OUT	BOOL	Parametr COM_RST: 1: Całkowity restart bloku funkcji, istniejące połączenie zostaje zakończone.
DONE	OUTPUT	BOOL	Parametr DONE statusu: 0: Zadanie jeszcze się nie rozpoczęło lub ciągle jest w toku. 1: Zadanie wykonane bez błędu.
BUSY	OUTPUT	BOOL	Parametr BUSY statusu: 0: Zadanie wykonane. 1: Zadanie jeszcze nie wykonane. Nowe zadanie nie może zostać rozpoczęte.
ERROR	OUTPUT	BOOL	Parametr ERROR statusu: 1: Podczas przetwarzania wystąpił błąd. Szczegółowe informacje o typie błędu zawiera STATUS.

STATUS	OUTPUT	WORD	Parametr STATUS statusu: Informacje o błędzie.
--------	--------	------	--

Parametry TRCV_C

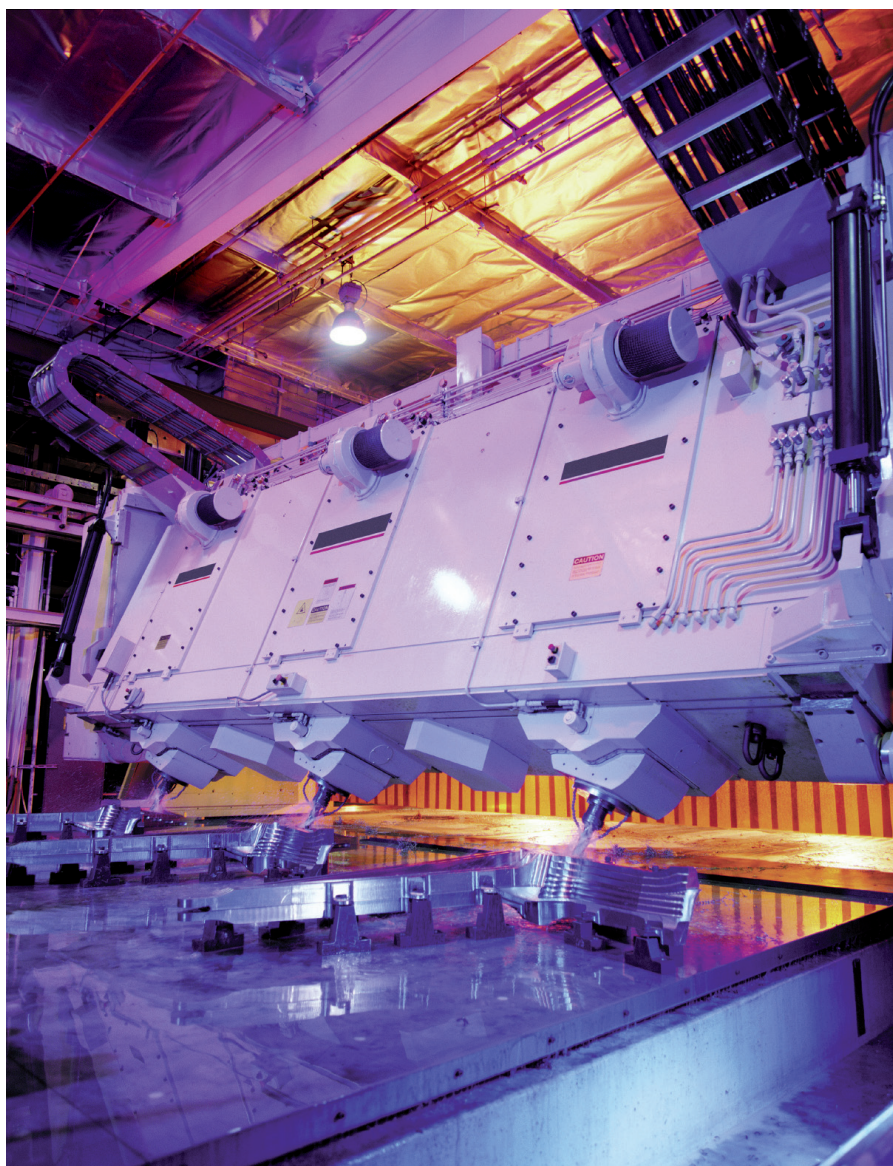


Parametr	Typ parametru	Typ danych	Opis
EN_R	IN	BOOL	Parametr sterujący uaktywniany do odbioru: Kiedy EN_R = 1, wtedy TRCV_C jest gotowa do odbioru. Zadanie odbioru jest wykonywane.
CONT	IN	BOOL	Parametr sterujący CONT: 0: rozłącz 1: ustanów i utrzymuj połączenie
LEN	IN	INT	Długość obszaru odbiorczego w bajtach. W celu poznania znaczenia LEN = 0 lub LEN <> 0 por. wyżej (tryby odbiorcze). W celu poznania wartości zakresów, por. zależność między CPU i Protocol Variant (connection_type) oraz Transferable Data Length.
CONNECT	IN_OUT	ANY	Wskaźnik do opisu połączenia.
DATA	IN_OUT	ANY	Obszar odbiorczy zawiera adres początkowy i maksymalną długość danych odbieranych.
COM_RST	IN_OUT	BOOL	Parametr COM_RST: 1: Całkowity restart bloku funkcji, istniejące połączenie zostaje zakończone.
DONE	OUT	BOOL	Parametr DONE statusu: 0: Zadanie jeszcze się nie rozpoczęło lub ciągle jest w toku. 1: Zadanie wykonane bez błędu.
BUSY	OUT	BOOL	Parametr BUSY statusu: 0: Zadanie wykonane. 1: Zadanie jeszcze nie wykonane. Nowe zadanie nie może zostać rozpoczęte.

ERROR	OUT	BOOL	Parametr ERROR statusu: 1: Podczas przetwarzania wystąpił błąd. Szczegółowe informacje o typie błędu zawiera STATUS.
STATUS	OUT	WORD	Parametr STATUS statusu: Informacje o błędzie.
RCVD_C	OUT	INT	Rzeczywista ilość odebranych danych wyrażona w bajtach.

Parametry ERROR i STATUS

ER-ROR	STATUS (W#16#...)	Opis
0	0000	Zadanie wykonane bez błędu
0	7000	Żadne zadanie nie jest aktualnie wykonywane
0	7001	Start wykonywania zadania, ustanowienie połączenia, oczekiwanie na połączenie partnera
0	7002	Rozpoczęcie otrzymywania danych
0	7003	Połączenie jest zakończone
0	7004	Połączenie ustanowione i monitorowane, żadne zadanie nie jest wykonywane
1	8085	Parametr LEN ma wartość 0 lub większą od największej dopuszczalnej wartości
1	8086	Parametr ID wykroczył poza dozwolony zakres
1	8087	Osiągnięto maksymalną liczbę połączeń; nie jest możliwe żadne dodatkowe połączenie
1	8088	Parametr LEN ma wartość większą niż obszar pamięci wyspecyfikowany w DATA; obszar pamięci odbiorczej jest za mały
1	8089	Parametr CONNECT nie wskazuje na blok danych
1	8091	Przekroczona głębokość zagnieżdżenia
1	809A	Parametr CONNECT wskazuje na pole, które nie odpowiada długości w opisie połączenia
1	809B	ID urządzenia lokalnego w opisie połączenia jest niezgodne z CPU
1	80A1	Błąd komunikacji: Wyspecyfikowane połączenie nie zostało jeszcze ustanowione Wyspecyfikowane połączenie jest aktualnie zakończone; transmisja tym kanałem połączeniowym jest niemożliwa Interfejs jest aktualnie reinicjalizowany
1	80A3	Wykonywana jest próba zakończenia nieistniejącego połączenia



1	80A7	Błąd komunikacji: wywołano TDISCON zanim został zakończony TCON (TDISCON musi najpierw całkowicie zakończyć połączenie wskazywane przez ID)
1	80B2	Parametr CONNECT wskazuje blok danych wygenerowany za pomocą słowa kluczowego UNLINKED
1	80B3	Niespójne parametry: Błąd w opisie połączenia Lokalny port (parametr local_tsap_id) występuje już w opisie innego połączenia ID w opisie połączenia różni się od ID wyspecyfikowanego jako parametr
1	80B4	Podczas używania wariantu protokołu ISO on TCP (connection_type = B#16#12), dla ustanowionego pasywnego połączenia (active_est = FALSE), został pogwałcony jeden lub oba następujące warunki: „local_tsap_id_len >= B#16#02” i/lub „local_tsap_id[1] = B#16#E0”

1	80C3	Wszystkie zasoby połączenia są w użyciu
1	80C4	Prześciowy błąd komunikacji: Połączenie nie może być aktualnie ustanowione Interfejs odbiera nowe parametry To skonfigurowane połączenie jest aktualnie usuwane przez TDISCON
1	8722	Parametr CONNECT: Nieprawidłowy obszar źródłowy: obszar nie istnieje w DB
1	873A	Parametr CONNECT: Niemożliwy dostęp do opisu połączenia (np. nie dostępny DB)
1	877F	Parametr CONNECT: Błąd wewnętrzny, taki jak niepoprawna referencja ANY

Tomasz Starak