



S7-1200

Przechowywanie danych, obszary pamięci i ich adresowanie



W artykule przedstawiamy wybrane możliwości sterowników z serii S7-1200 – doskonale wyposażonych następców niezwykle popularnych w Polsce sterowników S7-200.

Jednostka centralna sterowników S7-1200 dysponuje kilkoma sposobami przechowywania danych podczas wykonywania programu użytkownika (**tab. 1**):

- lokalizacje pamięci: CPU dzieli pamięć na różne specjalizowane obszary – wejścia (I), wyjścia (Q), pamięci bitowej (M), bloków danych (DB) i pamięci lokalnej lub chwilowej (L). Program użytkownika ma dostęp (dla odczytu i zapisu) do danych przechowywanych w tych obszarach.
- bloki danych (DB): DB mogą być wykorzystywane w programie użytkownika do przechowywania danych. Dane prze-

chowywane w DB nie są wymazywane, gdy blok danych zostaje zamknięty, lub gdy blok kodu skojarzony z tym blokiem danych zostanie zakończony. Są dwie kategorie DB: globalne DB: przechowują dane, które mogą być wykorzystywane przez inne wszystkie inne bloki; blok danych *instance* DB: przechowują dane dla określonych FB i ich struktura jest zgodna z parametrami używanymi przez FB.

- pamięć chwilowa: za każdym razem gdy wywołany jest blok kodu, system operacyjny CPU alokuje chwilową (lokalną) pamięć (L), która jest wykorzystywana podczas wykonywania tego bloku. Po

zakończeniu wykonywania tego bloku kodu, CPU realokuje pamięć lokalną na potrzeby innych bloków.

- odwołania, takie jak as 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”).

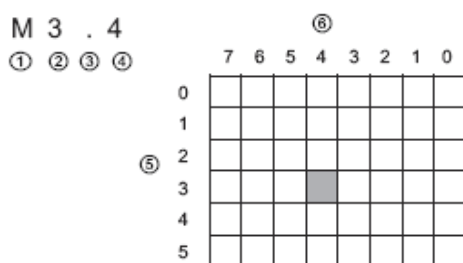
Różne obszary pamięci mają swoje unikalne adresy. Program użytkownika wykorzystuje te adresy w celu uzyskania dostępu do informacji przechowywanych w tych miejscach pamięci.

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. Przykład dostępu do bitu (zwanego również adresowaniem *byte.bit*) jest pokazany poniżej. W tym przykładzie po identyfikatorze

Tab. 1.

Obszar pamięci	Opis	Wymuszony	Trwały
I obraz procesu – wejście I_:P (fizyczne wejście)	Skopiowany na początku cyklu programu stan wejść fizycznych	Tak	Nie
	Bezpośredni odczyt wejściowych punktów fizycznych CPU, SB, SM	Nie	Nie
Q obraz procesu – wyjście Q_:P (fizyczne wyjście)	Stan skopiowany na początku cyklu programu do wyjść fizycznych	Tak	Nie
	Bezpośredni zapis do wyjściowych punktów fizycznych CPU, SB, SM	Nie	Nie
M pamięć bitowa	Pamięć sterująca i danych	Nie	Tak
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

obszaru pamięci i adresie bajtu (I = wejście i 3 = bajt 3) podany jest, oddzielony kropką („.”), adres bitu (bit 4).



(1) Identyfikator obszaru pamięci	(4) położenie bitu w bajcie (bit 4 z 8)
(2) Adres bajtu: bajt 3 (trzeci bajt)	(5) Bajty obszaru pamięci
(3) Kropka rozdzielająca adres bajtu od numeru bitu	(6) Bity wybranego bajtu

Użytkownik może uzyskać dostęp do danych zawartych w większości obszarów pamięci (I, Q, M, DB i L) jako bajtów, słów lub podwójnych słów stosując format „adresowania bajtowego”. W celu uzyskania dostępu do bajtu, słowa lub podwójnego słowa w pamięci, należy podać adres w podobny sposób, jaki stosuje się do adresowania bitów. Ten adres zawiera identyfikator obszaru, oznaczenie rozmiaru danych i adres bajtu początkowego bajtu, słowa lub podwójnego słowa. Rozmiar oznacza się jako bajt (B), słowo (W) lub podwójne słowo (DW) (przykładowo: IB0, MW20, QD8).

W celu uzyskania bezpośredniego dostępu do fizycznych wejść lub fizycznych wyjść, do adresu lub tagu należy dołączyć „:P” (na przykład: I0.3:P, Q1.7:P lub „Stop:P”).

Dostęp do danych w obszarach pamięci CPU

Pakiet narzędziowy TIA umożliwia programowanie symboliczne. Typowo, w tagach PLC, bloku danych lub blokach OB, FC lub FB są tworzone tagi. Te tagi zawierają nazwę, typ danych, przesunięcie i komentarz. Ponadto w bloku danych można wyspecyfikować wartość początkową. Te tagi można wykorzystywać podczas programowania,

podając nazwę tagu jako parametr instrukcji. Opcjonalnie, jako parametr można również podać argument bezwzględny (pamięć, obszar, rozmiar i przesunięcie). W przykładach podanych w kolejnych częściach przedstawiono w jaki sposób podawać argument bezwzględny. Na początku argumentu bezwzględnego, program edytora automatycznie dostawia znak %. W programie edytora można przełączać aktualny widok na jeden z trzech: symboliczny, bezwzględny i symboliczny lub bezwzględny.

I (obszar wejściowy obrazu procesu): CPU próbuje stan punktów wejściowych peryferii (fizycznych) tuż przed wykonaniem cyklicznego OB w każdym cyklu programu. Użytkownik ma dostęp do bitów, bajtów, słów i podwójnych słów należących do obszaru wejściowego obrazu procesu. Dopuszczalny jest zarówno zapis jak i odczyt danych, ale zwykle dane z obszaru wejściowego obrazu procesu są tylko odczytywane.

bit	I[adres bajtu]. [adres bitu]	I0.1
bajt, słowo lub podwójne słowo	I[rozmiar][adres startowego bajtu]	IB4, IW5 lub ID12

Dołączając do adresu „:P” można bezpośrednio odczytywać cyfrowe i analogowe wejścia CPU, SB lub SM. Różnica w dostępie przy wykorzystaniu adresowania I_:P zamiast I polega na tym, że dane są pobierane bezpośrednio z odczytywanych punktów, a nie z obszaru wejściowego obrazu procesu. Ponieważ dane są odczytywane bezpośrednio ze swojego źródła, a nie z kopii utworzonej podczas ostatniego uaktualnienia obszaru wejściowego obrazu procesu, więc dostęp poprzez I_:P jest nazywany „bezpośrednim odczytem”.

Ponieważ stan fizycznych punktów wejściowych jest ustawiany bezpośrednio z urządzeń zainstalowanych na obiekcie i połączonych z tymi punktami, więc zapis do tych punktów jest zabroniony. Inaczej mówiąc dostęp poprzez I_:P może być tylko odczytem, w przeciwieństwie do dostępu poprzez I, który obejmuje zarówno odczyt, jak i zapis.

Dostęp poprzez I_:P jest również ograniczony do takiej liczby wejść, która jest obsługiwana przez pojedynczą CPU, SB lub SM zaokrąglonej w górę do najbliższego pełnego bajtu. Na przykład, jeżeli wejścia 2 DI/2 DQ SB są tak skonfigurowane, że ich adresowanie rozpoczyna się od I4.0, to te punkty wejściowe są dostępne jako I4.0:P i I4.1:P lub jako IB4:P. Dostęp do I4.2:P ÷ I4.7:P nie jest odrzucany, ale nie ma żadnego sensu ponieważ te punkty nie są używane. Dostęp do IW4:P i ID4:P jest zabroniony ponieważ jest przekroczony bajt przesunięcia powiązany z tym SB.

Dostęp poprzez I_:P nie wpływa na wartości pamiętane w obszarze wejściowym obrazu procesu.

bit	I[adres bajtu]. [adres bitu]:P	I0.1:P
bajt, słowo lub podwójne słowo	I[rozmiar][adres startowego bajtu]:P	IB4:P, IW5:P lub ID12:P

Q (obszar wyjściowy obrazu procesu): CPU kopiuje wartości pamiętane w obszarze wyjściowym obrazu procesu do fizycznych punktów wyjściowych. Użytkownik ma dostęp do bitów, bajtów, słów i podwójnych słów należących do obszaru wyjściowego obrazu procesu. Dla obszaru wyjściowego obrazu procesu dopuszczalny jest zarówno zapis jak i odczyt danych.

bit	Q[adres bajtu]. [adres bitu]	Q1.1
bajt, słowo lub podwójne słowo	Q[rozmiar][adres startowego bajtu]	QB5, QW10, QD40

Dołączając do adresu „:P” można bezpośrednio zapisywać dane do cyfrowych i analogowych wyjść CPU, SB lub SM. Różnica w dostępie przy wykorzystaniu adresowania Q_:P zamiast Q polega na tym, że dane, oprócz wpisania do obszaru wyjściowego obrazu procesu są również przesyłane bezpośrednio do adresowanych punktów (są zapisywane do dwóch miejsc). Ponieważ dane są przesyłane bezpośrednio do punktów docelowych, które nie muszą czekać na kolejne

uaktualnienie obszaru wyjściowego obrazu procesu, więc dostęp poprzez Q:P jest nazywany „bezpośrednim zapisem”.

Ponieważ wyjściowe punkty fizyczne bezpośrednio sterują urządzeniami zainstalowanymi na obiekcie, które są do tych punktów podłączone, więc odczyt tych punktów jest zabroniony. Inaczej mówiąc dostęp poprzez Q:P może być tylko zapisem, w przeciwnieństwie do dostępu poprzez Q, który obejmuje zarówno odczyt, jak i zapis.

Dostęp poprzez I:P jest również ograniczony do takiej liczby wyjść, która jest obsługiwana przez pojedynczą CPU, SB lub SM zaokrąglonej w górę do najbliższego pełnego bajtu. Na przykład, jeżeli wyjścia 2 DI/2 DQ SB są tak skonfigurowane, że ich adresowanie rozpoczyna się od Q4.0, to te punkty wyjściowe są dostępne jako Q4.0:P i Q4.1:P lub jako QB4:P. Dostęp do Q4.2:P ÷ Q4.7:P nie jest odrzucany, ale nie ma żadnego sensu ponieważ te punkty nie są używane. Dostęp do QW4:P i QD4:P jest zabroniony ponieważ jest przekroczony bajt przesunięcia powiązany z tym SB.

Dostęp poprzez Q:P wpływa zarówno na stan wyjść fizycznych, jak i na wartości pamiętane w obszarze wyjściowym obrazu procesu.

bit	Q[adres bajtu]. [adres bitu]:P	Q1.1:P
bajt, słowo lub podwójne słowo	Q[rozmiar][adres startowego bajtu]:P	QB5:P, QW10:P lub QD40:P

M (obszar pamięci bitowej): Obszaru pamięci bitowej M używa się do sterowania zarówno przekaźników, jak i danych do przechowywania pośredniego statusu operacji lub innych informacji sterujących. Użytkownik ma dostęp do bitów, bajtów, słów i podwójnych słów należących do obszaru pamięci bitowej. Dla pamięci M dopuszczalny jest zarówno zapis jak i odczyt danych.

bit	M[adres bajtu].[adres bitu]	M26.7
bajt, słowo lub podwójne słowo	M[rozmiar][adres startowego bajtu]	MB20, MW30, MD50

Temp (pamięć chwilowa): CPU zapewnia pamięć chwilową (lokalną) dla każdej z trzech grup priorytetów OB: 16 kB dla rozruchu i cyklu programu, włączając w to FB i FC; 4 kB dla zdarzeń przerwań standardowych, włączając w to FB i FC i 4 kB dla zdarzeń przerwań błędów, włączając w to FB i FC.

Pamięć Temp jest podobna do pamięci M z jednym zasadniczym wyjątkiem: pamięć M ma charakter globalny, a pamięć Temp jest pamięcią lokalną:

Pamięć M: Dowolny OB, FC lub FB ma dostęp do danych w pamięci M, co oznacza, że dane są dostępne globalnie dla wszystkich elementów programu użytkownika.

Pamięć Temp: Dostęp do danych w pamięci Temp mają tylko te OB, FC lub FB, które stworzyły lub zadeklarowały lokalizację pamięci Temp. Lokalizacje pamięci Temp pozostają lokalne i nie są współdzielone przez różne bloki kodu, nawet jeśli jeden blok kodu wywołuje inny blok kodu. Na przykład: jeśli OB wywołuje FC, to FC nie ma dostępu do pamięci chwilowej należącej do wywołującego OB.

CPU alokuje pamięć chwilową wtedy, kiedy jest ona potrzebna. CPU alokuje pamięć chwilową dla bloku kodu w chwili, gdy blok kodu jest uruchamiany (dla OB) lub jest wywoływany (dla FC lub FB). Alokacja pamięci chwilowej dla bloku kodu może dotyczyć tej samej lokalizacji pamięci Temp, która była poprzednio używana przez inne OB, FC lub FB. CPU nie inicjalizuje pamięci chwilowej w momencie alokacji i w związku z tym mogą się w niej znajdować jakieś wartości.

Dostęp do pamięci chwilowej jest możliwy wyłącznie za pomocą adresowania symbolicznego.

DB (blok danych): Pamięć DB stosuje się do pamiętania różnych typów danych, włączając w to pośredni status operacji lub inne parametry sterujące dla FB i struktury danych wymagane przez wiele instrukcji, takich jak timery i liczniki. Użytkownik może określić, czy blok danych będzie umożliwiał odczyt/zapis, czy też będzie tylko do odczytu. Użytkownik ma dostęp do bitów, bajtów, słów i podwójnych słów należących do pamięci bloku danych. Dla bloków danych typu czytaj/zapisz dopuszczalny jest zarówno zapis jak i odczyt danych. Dla bloków danych typu czytaj dozwolony jest tylko odczyt danych.

bit	DB[numer bloku danych]. DBX[adres bajtu].[adres bitu]	DB1.DBX2.3
bajt, słowo lub podwójne słowo	DB[numer bloku danych]. DB [rozmiar][adres startowego bajtu]	DB1.DBB4, DB10.DBW2, DB20.DBD8

Adresowanie I/O w CPU i modułów I/O

Użytkownik może zmienić domyślne adresowanie wybierając na ekranie konfiguracyjnym pole adresu i wpisując tam nowe liczby. Wejściom i wyjściom cyfrowym przypisuje się pełne bajty, niezależnie od tego, czy moduł ma wszystkie punkty, czy nie. Wejścia i wyjścia analogowe tworzą grupy po dwa punkty (4 bajty). W podanym przykładzie, użytkownik może zmienić adres DI16 z 8...9 na 2...3. Program asystuje użytkownikowi i zmienia zakres adresów, które mają niewłaściwy rozmiar lub wchodzą w konflikt z innymi adresami.

Andrzej Gawryluk

Opracowano na podstawie dokumentacji firmy Siemens.



Kiedy na ekranie konfiguracyjnym są dodawane CPU i moduły I/O, to automatycznie są alokowane adresy I oraz Q.
Wejściom CPU odpowiadają bity adresowane od I0.0 do I0.7 i od I1.0 do I1.5 (łącznie 14 punktów). Wyjściom CPU odpowiadają bity adresowane od Q0.0 do Q0.7 i od Q1.0 do Q1.1 (łącznie 10 punktów). Wejściom analogowym CPU odpowiadają słowa o adresach IW64 i IW66 (2 punkty analogowe, łącznie 4 bajty). Wejścia DI16 są adresowane od I8.0 do I9.7.
AI4 / AO2 – wejścia to IW112, IW114, IW116, IW118, a wyjścia to QW112 i QW114
DI8 / DO8 – zakres wejść jest od I16.0 do I17.7, a wyjść od Q16.0 do Q17.7.
Na rysunku przedstawiono przykładowy CPU 1214C z dwoma SM.

Module	Slot	Address	Quad.	Type
	103			
	102			
PS485_1	101			CM 1241 (PS485)
PLC_1	1			CPU 1214C DC/DC/DC
DI14/DO10	1.1	0...1	0...1	DI14/DO10
AIZ	1.2	64...67		AIZ
AO1 x 12Bit_1	1.3		80...81	AO1 signal board
HSC_1	1.16			High speed counters (H)
HSC_2	1.17			High speed counters (H)
HSC_3	1.18			High speed counters (H)
HSC_4	1.19			High speed counters (H)
HSC_5	1.20			High speed counters (H)
HSC_6	1.21			High speed counters (H)
Pulse_1	1.32			Pulse generator (PTOP)
Pulse_2	1.33			Pulse generator (PTOP)
PROFINET int.	1.327			PROFINET interface
DB x 24VDC_1	2			SM 1221 DB x 24VDC
AI4 x 12bit_1	3	112...1		SM 1231 AI4