

Różne urządzenia – jedna sieć : sterowanie i pomiary za pomocą DeviceNet

Streszczenie : W referacie zaprezentowane zostały trzy mechanizmy wymiany danych w sieci DeviceNet : Change of State, Bit Strobe oraz Polling. W oparciu o ich własności przedstawiono ideę sieci łączącej wiele zróżnicowanych urządzeń, co zostało poparte przykładami i uwagami z ich praktycznego stosowania. Wskazano także na możliwość integracji urządzeń pochodzących od różnych dostawców.

Abstract : The paper present three kinds of data transfer, used in DeviceNet network : Change of State, Bit Strobe and Polling. On the base of their properties, author presents the idea of network, connecting different arts of equipment. Some examples from practical usage are also given. The possibility of integrity the devices, made by wide area of manufactures is described on the context of open network concept.

1. WSTĘP

Obserwując rozwój współczesnego przemysłu można zauważyć, że pomysł „globalnej wioski” przejawiający się m.in. masowym wykorzystywaniem Internetu, przenika coraz bardziej także do zakładów produkcyjnych. Choć oczywiście skala tej globalizacji jest znacznie mniejsza (ale znane są przykłady ogólnoświatowego dostępu do informacji procesowej) to jednak idee pozostają takie same : zebrać i udostępnić w sposób szybki i bezpieczny odpowiednią ilość danych. W tym miejscu zapewne odezwią się głosy konserwatystów, uznających, że skoro zakład nie jest aż tak wielki, to stosowanie tradycyjnego okablowania sygnałowego jest w zupełności wystarczające. Poglądom takim przeciwstawia się dzisiejsza technologia produkcji, tworząc wymagania przedstawione w [1][2], z których najważniejsze to wiarygodność danych, bezpieczeństwo transmisji, oszczędność czasu i kosztów podczas uruchamiania i normalnej pracy. Trudno jednak, z uwagi na wspomniane założenia, zastosować w aplikacjach przemysłowych tak popularny w świecie komputerowym Internet. W pracy [1] przedstawiony został przegląd współcześnie stosowanych systemów sieciowych oraz ich klasyfikacja z uwagi na funkcje i właściwości.

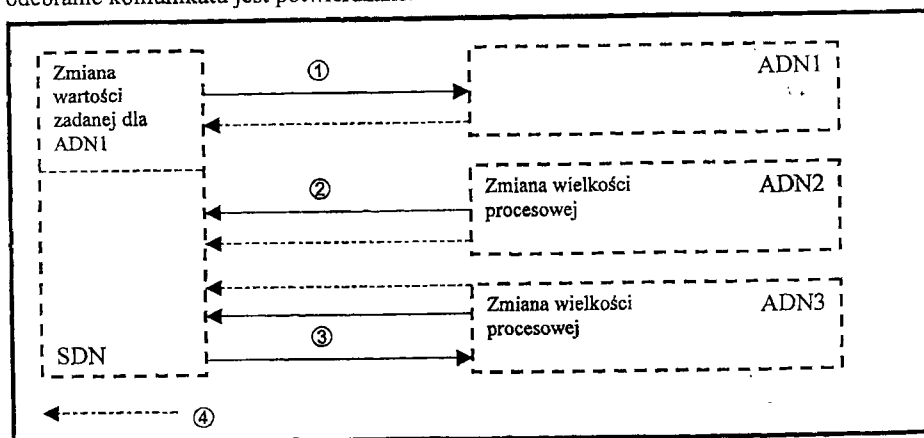
Szczególnym zainteresowaniem, zarówno projektantów jak użytkowników cieszą się sieci poziomu urządzeń, pozwalające na łączenie układów sensorycznych i wykonawczych, w których spełnienie wspomnianych wymagań jest najbardziej widoczne. Spośród dość licznej grupy systemów komunikacyjnych wyróżnić można sieć DeviceNet, przede wszystkim z uwagi na zastosowany tam interesujący model Producent-Konsument oraz sposoby wymiany informacji. Szczegóły tego modelu zostały zaprezentowane w [2], ale warto tu wspomnieć że oznacza się on znaczną redukcją czasu zajętości łącza przez poszczególne urządzenia, a co za tym idzie większą efektywnością. Podniesienie przepustowości sieci osiągnięto także dzięki rozdzieleniu danych [2] na istotne (typu I/O Message – informujące o istotnych stanach procesu) oraz pomocnicze (typu Explicit Message – stanowiące parametry konfiguracyjne). Kolejnym krokiem w celu usprawnienia sieci stało się zróżnicowanie mechanizmów wymiany danych pomiędzy skanerem (stacją master) a adapterami sieci DeviceNet (urządzeniami typu slave).

2. MECHANIZMY WYMIANY DANYCH W SIECI DEVICENET

W tradycyjnych sieciach poziomu urządzeń podstawowym i często jedynym mechanizmem przesyłania informacji jest polling czyli cykliczne odpytywanie urządzeń. W przypadku współczesnych systemów sterowania, w których w ramach jednej sieci współpracuje wiele różnorodnych urządzeń, takie podejście jest już nie wystarczające. Projektant sieci musi uwzględnić nie tylko szybkość zmian wielkości mierzonej (lub sterowania) ale także charakter i sposób pracy urządzenia. Jest to główna przyczyna, dla której konstruktorzy DeviceNet stworzyli trzy, równoprawne sposoby wymiany danych: *change of state*, *bit strobe* oraz *polling*. Na rysunkach wykorzystano następujące skróty: SDN – skaner DeviceNet (master) znajdujący się w sterowniku, ADN – adapter DeviceNet (slave).

2.1 *Change of State (COS)* – zmiana stanu

Ideą mechanizmu COS jest wykorzystywanie łącza jedynie wtedy jeśli stan urządzenia uległ zmianie. Oznacza to, że wszystkie węzły pracujące w trybie COS dokonują sprawdzenia, czy dane przeznaczone do przestania zmieniły swoją wartość i jeśli tak, to przystępują do testowania stanu sieci i następnie nadawania. Komunikat ① jest generowany w sytuacjach w których skaner modyfikuje sterowanie procesem, zaś sytuacja ② ma miejsce wtedy, gdy stan procesu uległ zmianie. Warto tu zwrócić uwagę na fakt, iż typowa transmisja COS odbywa się tylko w jedną stronę i ma charakter multicast (jeden do wielu). W przypadkach przesyłania szczególnie istotnych danych, zachodzi połączenie ③ w którym odebranie komunikatu jest potwierdzone.

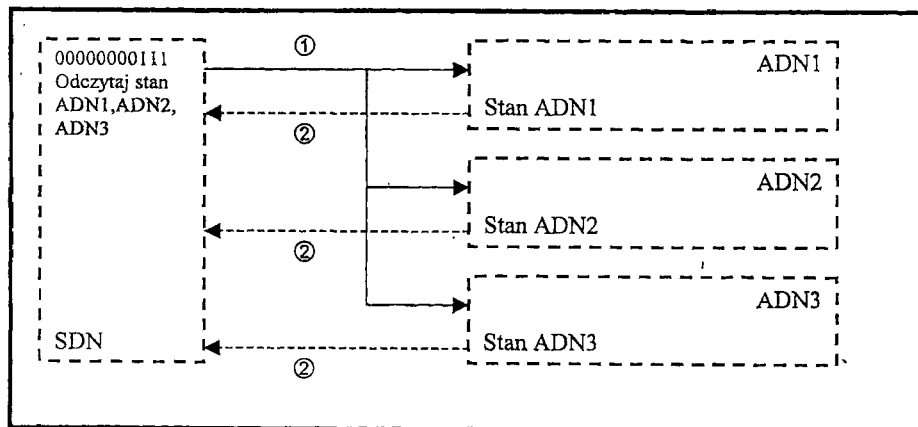


Tu jednak może powstać pytanie: jak odróżnić sytuację, w której brak kolejnych danych wynika z ich niezmienności od stanu w którym urządzenie (lub połączenie kablowe) zostało uszkodzone i jakkolwiek transmisja nie jest w ogóle możliwa. Aby skaner mógł dokonać analizy stanu węzłów typu COS, wprowadzony został mechanizm cyklicznego (z ustalaniem czasem) przesyłania informacji statusowej ④, nazywany Device Heartbeat. Transfer ten jest opcjonalny i stosowany w przypadku urządzeń, których awaria jest trudna do wykrycia dla obsługi technicznej. Typowymi przykładami węzłów COS są terminale operatorskie i układy napędowe (falowniki itp.) charakteryzujące się dużą ilością parametrów, których zmienność w czasie jest często bardzo zróżnicowana. W trybie COS pracują także interfejsy autonomicznych sterowników PLC, posiadających własny program i

synchronizujących dane procesowe ze sterownikiem nadrzędnym (w którym znajduje się moduł skanera, lub innym dołączonym przez podobny interfejs). [3][4]

2.2 Bit Strobe (BS) – strobowanie bitowe

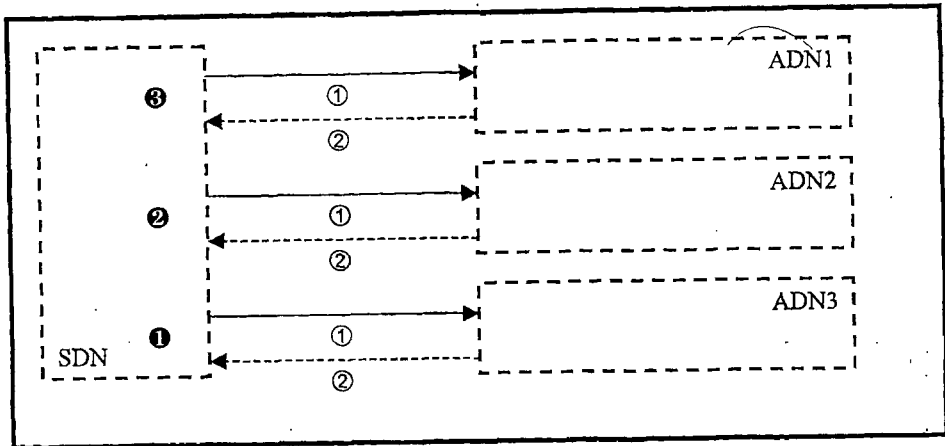
Mechanizm BS został stworzony przede wszystkim do obsługi prostej sensoryki i nieskomplikowanych układów wykonawczych. Urządzenia takie z racji swojej konstrukcji cechują się niewielkimi rozmiarami, co uniemożliwia zastosowanie w nich rozwiniętych układów procesorowych. Ideę komunikacji z wykorzystaniem strobowania bitowego przedstawia poniższy rysunek.



Moduł skanera wysłał w sieć komunikat ① Zgodnie z modelem Producent-Konsument dane te mogą zostać odebrane przez wszystkie urządzenia w sieci. W ramce tego komunikatu znajdują się 64 bity (8 bajtów czyli maksymalna ilość informacji w jednej ramce [1]), z których każdy reprezentuje jeden węzeł. Jeśli pole zawiera „1”, a węzeł pracuje w trybie BS to wysłał do skanera komunikat ② zawierający aktualny stan urządzenia. Ustawienie wartości „0” jest równoznaczne z brakiem żądania odpowiedzi od urządzenia. Mechanizm ten ma jak widać pewne cechy klasycznego odpytywania, jest jednak znacznie bardziej efektywniejszy z uwagi na charakter przesyłania ① typu broadcast. Znajduje on zastosowanie przede wszystkim przy współpracy z urządzeniami sensorycznymi typu czujniki krańcowe, obecności (pojemnościowe, indukcyjne) czy fotowłączniki, które obok swojego stanu (a więc wartości binarnej) przesyłają swój status, zamykający się zazwyczaj w kilku bitach. Typowo całość informacji zwracanej przez urządzenie w postaci ② ma długość jednego bajtu. Pozwala to na osiągnięcie dość znacznych prędkości przekazywania informacji, co jest szczególnie istotne w procesach transportowych, pakowania i drobnej, masowej produkcji. Obecnie na rynku znajduje się wiele prostych czujników, wyposażonych w interfejs DeviceNet, zaś dołączenie standardowego. dwu/trój-przewodowego nie stanowi problemu dzięki interfejsowi DeviceLink [3][4].

2.3 Polling (PO) – cykliczne odpytywanie

Komunikacja z wykorzystaniem mechanizmu PO jest realizacją standardowej (i często jedynej) idei wymiany informacji w sieciach poziomu urządzeń. Przekazywanie danych jest realizowane przez cykliczne łączenie się z każdym urządzeniem, co szczegółowo wyjaśnia przedstawiony poniżej rysunek.



Podczas konfiguracji sieci w skanerze jest budowana lista urządzeń, według której następuje odpytywanie. W naszym przykładzie listę tworzą adaptery : ADN3 (pozycja ❶ na liście), ADN2 (❷) oraz ADN1 (❸). Rozpoczynając od pozycji ❶ skaner wysyła komunikat ❶ zawierający informacje przeznaczone dla ADN3 lub adres żądanych danych w ADN3. W odpowiedzi ❷ ADN3 przesyła odpowiednio potwierdzenie przyjęcia informacji lub dane wymagane przez skaner. Po zakończeniu połączenia z urządzeniem ❶ skaner łączy się z kolejnym adapterem na liście. Proces ten jest realizowany cyklicznie, przy czym dla każdego węzła można określić indywidualny czas odpytywania. Lista w skanerze ma więc charakter dynamiczny, zależny od rodzaju sterowanego procesu i wykorzystanych urządzeń. Mechanizm PO jest najczęściej wykorzystywany przy zbieraniu danych analogowych (temperatury, ciśnienia, przepływy itp.), których okres zmienności jest wprawdzie zróżnicowany ale w miarę stały i łatwy do określenia. Zastosowanie PO do przekazywania informacji sterującej procesem ma podobne uzasadnienie. Typowymi urządzeniami pracującymi w ramach PO są moduły i bloki uniwersalnych wejść/wyjść typu Flex I/O (w wykonaniach standardowych oraz Ex) a także odporne na warunki środowiskowe ArmorBlock (praca w zakresie $-25 + 75^{\circ}\text{C}$). Z pośród każdej z nich projektant ma możliwość wybrania modułu odpowiadającego założonym funkcjom i rodzajowi obsługiwanego sygnału.

3. PROFILE URZĄDZEŃ, SIG'S – SPECIAL INTEREST GROUPS

Sieć DeviceNet, a właściwie urządzenia w niej pracujące zostały zdefiniowane [5] przy użyciu modelowania obiektowego, organizującego wszystkie parametry (nazywane atrybutami) i usługi w jedno ciało – obiekt. Obiekt jest zatem zbiorem odpowiednio dobranych usług i atrybutów. Obiekt może być przekształcany przy pomocy usług, zaś atrybuty reprezentowane przez wartości liczbowe, charakteryzują jego własności. Każdy obiekt jest opisywany przez klasy. Klasa definiuje rodzaj obiektu i określa charakterystyczne właściwości, identyczne dla wszystkich obiektów tej klasy. Dla przykładu wszyscy ludzie na Ziemi są reprezentowani przez klasę „Ludzie”, zawierającą kilka miliardów obiektów. Obiekt wewnątrz klasy jest nazywany instancją. Jest ona jego reprezentacją. Każda instancja wewnątrz jednej klasy ma taki sam zestaw atrybutów, których wartości czynią każdą instancję unikalną w ramach klasy. Kontynuując przykład ludzi, poniższa tabela pokazuje przykładową definicję każdego z nas.

Klasa	Instancja	Atrybut	Wartość atrybutu
Ludzie	Adam	Płeć	Mężczyzna
		Wiek	30
	Ewa	Płeć	Kobieta
		Wiek	25

Dzięki zastosowaniu takiej klasyfikacji cech możliwe stało określenie profili urządzeń, opisujących w sposób ścisły ich funkcje oraz zachowanie się w sieci. W każdym profilu muszą znaleźć się obiekty należące do następujących klas :

- *Connection Class* – obiekty z tej klasy są odpowiedzialne za nawiązanie i utrzymanie połączeń I/O Message oraz Explicit Message
- *DeviceNet Class* – obiekt z tej klasy opisuje zawiera atrybuty sieciowe takie jak adres, prędkość transmisji, adres skanera itp.
- *Identity Class* – obiekt z tej klasy zawiera informacje o urządzeniu (status, kod, numer seryjny, numer wersji itp.) oraz jego producencie
- *Message Router Class* – obiekt z tej klasy jest odpowiedzialny za realizację wymiany informacji w trybie Explicit Message.

Dzięki nim każde urządzenie jest w stanie poprawnie pracować w sieci, co jest rozumiane jako możliwość bezkolizyjnego dołączenia do sieci i jego identyfikacji oraz jako zdolność do wymiany danych. Występowanie i własności pozostałych klas jest związane z konkretnym profilem. Poniżej zestawione zostały aktualnie zdefiniowane profile urządzeń :

Wyłączniki krańcowe	Napędy DC	Wyspy zaworowe
Czujniki indukcyjne	Napędy Servo	Czytniki kodów paskowych
Czujniki fotoelektryczne	Napędy AC	Zabezpieczenia przeciążeniowe
Dyskretne bloki I/O	Enkodery	Rozłączniki
Analogowe bloki I/O	Rezolwery	Styczniki i startery do silników
Wagi	Stacje i terminale MMI	Soft-starty

Każdy z wymienionych profili zawiera szereg obiektów zebranych w klasie *Parameter Class*. Opisują one własności charakterystyczne dla danego typu urządzeń. Dla przykładu w przypadku falowników (napędy AC) obiekty tej klasy zawierają dane ruchowe takie jak częstotliwość zadana, czas rozpędzania, hamowania, ograniczenia prądowe itp. Nad ich poprawnym udostępnianiem w sieci, czuwają obiekty zebrane w klasie *Assembly Class*.

Z uwagi na ciągły rozwój i doskonalenie DeviceNet, w ramach każdego profilu określone zostały atrybuty obowiązkowe oraz opcjonalne. Pierwsze z nich zapewniają minimalny zestaw parametrów, pozwalający na uruchomienie i poprawne działanie całego systemu sterowania. Wykorzystanie pozostałych daje użytkownikowi niemalże nieograniczone możliwości automatycznego i zdalnego rekonfigurowania pracujących w nim urządzeń. Koncepcja minimalnego zestawu atrybutów, w połączeniu z otwartością DeviceNet spowodowała, że szereg firm, nawet niezbyt zaawansowanych technologicznie przystąpiło do tego projektu. Obecność ponad 300 certyfikowanych producentów jest dowodem na trafność takiego rozwiązania. Firmy te, działając pod egidą ODVA, tworzą grupy wytwórców zgodnie z rodzajem wytwarzanych urządzeń, nazywane SIG – Special Interest Group. Ich zadaniem jest rozwijanie technologii oraz, co najistotniejsze dla użytkownika, zapewnienie zamienności urządzeń w ramach danego profilu. Zamiennosc ta jest rozumiana (i certyfikowana) jako zdolność urządzenia, pochodzącego od jednego dostawcy, do pracy w miejsce innego, stworzonego przez drugiego producenta. Cecha ta jest gwarantowana w ramach obowiązkowego zestawu atrybutów, który dzięki pracom SIG staje się coraz bardziej pełny.

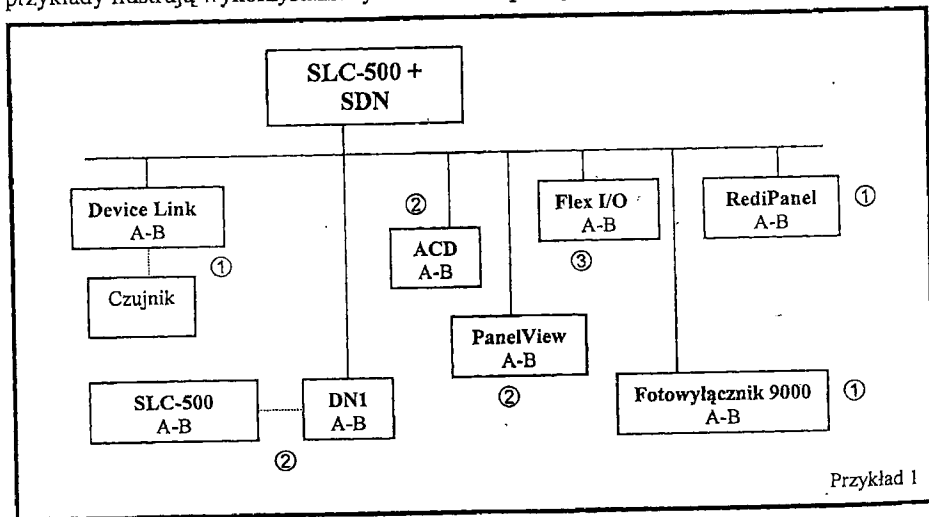
Dla użytkownika sieci DeviceNet, profile urządzeń są widziane jako pliki tekstowe, zawierające, zapisane w odpowiednim formacie, informacje o obiektach i ich atrybutach. Pliki te noszą nazwę *Electronic Data Sheet EDS* i są tworzone dla każdego urządzenia indywidualnie, określając te parametry, do których przewidywany jest dostęp poprzez sieć. Przykład fragmentu takiego pliku (dotyczącego napędu AC) został pokazany poniżej :

```
[Device]
VendCode = 1;                $ Vendor Code
ProdType = 2;                $ Product Type
ProdCode = 1;                $ Product Code
MajRev = 4;                  $ Major Rev
MinRev = 0;                  $ Minor Rev
VendName = "Allen-Bradley Company";
ProdTypeStr = "AC Drive";
ProdName = "160-SSC";
Catalog = "160-SSC SF0";

[Params]
Param1=                      $ parameter instance
0,                            $ data slot - don't care
6, "20 64 24 01 30 01",     $ path size, logical address (class, instance, attribute)
0x0074,                      $ descriptor - in hex format
2,                            $ data type (UINT)
2,                            $ data size
"Output Frequency",         $ name
"Hz",                        $ units
```

4. PRZYKŁADY INTEGRACJI RÓŻNYCH URZĄDZEŃ W PRAKTYCE

Przedstawione dotąd informacje dotyczyły zasad funkcjonowania DeviceNet jako sieci dostosowanej do obsługi wielu, różnorodnych urządzeń. Przedstawione w dalszej części przykłady ilustrują wykorzystanie tych założeń w praktyce.

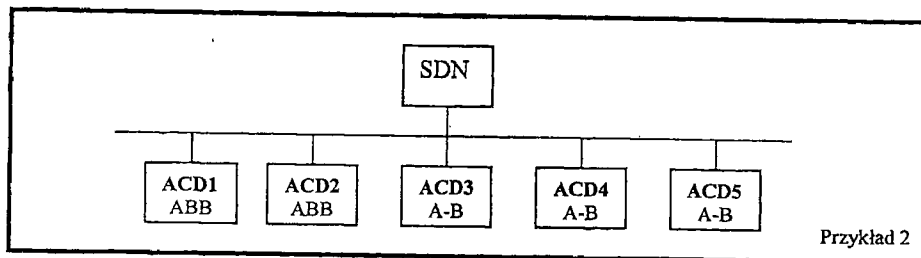


Przykład 1 zawiera przekrój najbardziej typowych urządzeń spotykanych w aplikacjach przemysłowych. Połączenie ich razem w jedną sieć, było jednym z głównych wymagań postawionych przez użytkownika. Zastosowanie DeviceNet pozwoliło na odpowiedni dobór urządzeń i przydzielenie im odpowiednich mechanizmów wymiany danych. Do wykrywania typowych stanów obecności zastosowano tradycyjne czujniki

łączone poprzez DeviceLink. Przekazywanie parametrów szczególnie istotnych (zliczanie wielkości produkcji, odrzuty, wykrywanie braków elementów składowych produktu) zostało zrealizowane poprzez czujniki posiadające zintegrowany interfejs DeviceNet (tak jak fotowłącznik 9000). Dzięki temu każde uszkodzenie, zabrudzenie lub błąd czujnika jest natychmiast sygnalizowane. Z podobnych względów wykorzystano pulpit z typowymi przyciskami RediPanel, widziany bezpośrednio w sieci, co pozwala na zamontowanie w nim głównych przełączników procesowych. Wszystkie te urządzenia (ozn ①) z uwagi na swój charakter, zostały skonfigurowane w trybie BS.

Komunikacja z operatorem oraz realizacja sterowania wymagały nieco innego podejścia. Zastosowany terminal PanelView (przykład profilu MMI) pozwala na prezentację danych procesowych oraz zadawanie nowych wartości. Nie byłoby w tym nic dziwnego (są to przecież typowe funkcje MMI), gdyby nie fakt, iż posiada on własności monitora sieciowego. Cecha ta pozwala na dostęp do atrybutów poszczególnych urządzeń bez zapręgnięcia do tego celu skanera, gdyż informacja wyświetlana pochodzi z „nasłuchu” ruchu sieciowego. Do sterowania napędami użyto grupy falowników wyposażonych w adaptory DeviceNet, co umożliwiło automatyczne ich dostrajanie przez sterownik. Warto tu w sposób szczególnie zaznaczyć obecność drugiego sterownika SLC-500, współpracującego z adapterem DN1. Konstrukcja taka pozwoliła na zachowanie jego autonomii (posiada własny procesor, wykonujący niezależnie swój program), zapewniając jednocześnie swobodę dostępu do danych sieciowych (oraz programowania poprzez sieć!). Takie rozwiązanie zostało zastosowane przy kontroli układów grzewczych, których praca nie może zostać przerwana z powodu zatrzymania skanera lub błędów w sieci. Wymienione tu urządzenia (ozn ②) z uwagi na aperiodyczny charakter przesyłanych danych skonfigurowano w trybie COS.

Pozostałe sygnały czyli mniej istotne czujniki cyfrowe, wszystkie pomiary analogowe (temperatury, ciśnienia) oraz sterowania (siłowniki, pozycjonery, elektrozwory) zostały przewidziane do obsługi poprzez bloki Flex I/O ③. Dzięki temu uzyskano możliwość dopasowania systemu do poszczególnych sygnałów (szeroki wybór modułów), co w połączeniu z pracą w trybie PO pozwoliło w sposób łatwy i czasowo-optymalny zapanować nad ich obsługą.

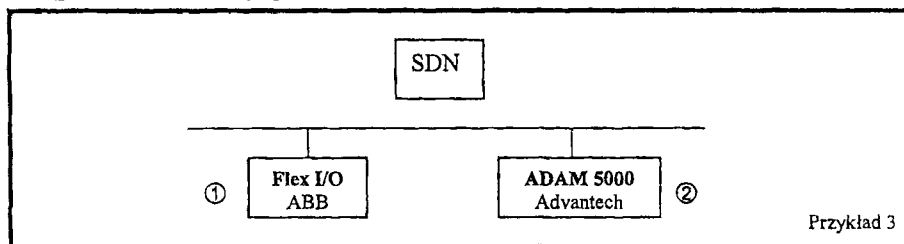


Aplikacja pokazana w przykładzie 2 prezentuje sposób realizacji przez różnych producentów założeń dotyczących profili urządzeń. W systemie tym zastosowano swa rodzaje napędów: Allen-Bradley oraz ABB. W ramach konfiguracji „obowiązkowej” wszystkie falowniki są widziane jako dwa słowa odczytywane oraz dwa słowa sterujące:

Parametry odczytywane	Parametry sterujące
Status napędu	Zadawanie kierunków, start, stop
Aktualna częstotliwość	Częstotliwość zadana

Dzięki temu można mówić o zamienności tych napędów. Niestety, z racji ograniczenia przez ABB dostępu do pozostałych informacji, wyświetlanych tylko na lokalnym terminalu, do pracy w trybie autostrojenia wykorzystano jedynie napędy Allen-Bradley. ODVA prowadząc szeroką akcję wspierania producentów poprzez SIG, stara się wpływać na zwiększanie

rozmiaru minimalnego zestawu atrybutów. Należy jednak zawsze mieć na uwadze możliwe odstępstwa od założonego profilu w części rozszerzonej.



Przykład 3

Ostatni, 3 przykład obrazuje współpracę dwóch zestawów bloków we/wy : Flex I/O oraz ADAM 5000. W obydwu przypadkach mamy możliwość dobrania rodzajów modułów do charakteru obsługiwanych sygnałów. Cóż więc różni te urządzenia? Zastosowanie ① pozwala rozróżnienie informacji procesowej i konfiguracyjnej. Pierwszy rodzaj jest przesyłany w trybie I/O Message (a więc o wysokim priorytecie), drugi zaś jako Explicit Message (o priorytecie niższym). Dzięki temu zapewniana jest deterministyczna obsługa procesu przy jednoczesnym wyeliminowaniu niepotrzebnego ruchu sieciowego. Urządzenia typu ② charakteryzują się nieco mniejszym zaawansowaniem technologicznym, co przejawia się w konieczności obsługi całej komunikacji w trybie Explicit Message. Ta własność (w powiązaniu ze znacznie niższą ceną) stawia je w roli układów pomocniczych, sterujących mniej istotnymi partiami procesu.

5. ZAKOŃCZENIE

Referat ten powstał w oparciu o doświadczenia autora zebrane podczas konfiguracji i uruchamiania sieci DeviceNet. Nie wyczerpują one całej wiedzy dotyczącej tego systemu, jak również problematyki integracji urządzeń przemysłowych. Prezentując ten referat, autor ma nadzieję pobudzić Czytelników do dyskusji zarówno poprzez E-mail, jak i spotkania osobiste podczas targów czy w biurze Elmark Automatyka.

Literatura

- [1] Rafał Tutaj : Przemysłowe Systemy Komunikacyjne, Konferencja Automation'97 Tom 2, przedruk : PAR 4'97 s 32-35
- [2] Rafał Tutaj : DeviceNet – efektywne rozwiązanie problemu komunikacji typu Fieldbus, Konferencja Automation'98 , przedruk : PAR 4'98 s 34-37
- [3] Rockwell Automation : DeviceNet System Overview, Publication DN-2.15, 1998
- [4] ODVA : Connected with DeviceNet, Product Catalog 1998
- [5] ODVA : DeviceNet Specification Release 2.0 1997