

Rafał TUTAJ

MECHANIZMY KOMUNIKACYJNE W SIECIACH DeviceNet I CanOpen A KONSTRUKCJA URZĄDZEŃ TYPU Gateway

Streszczenie: Przedstawiono podstawowe własności sieci DeviceNet/ CanOpen oraz wynikające z nich zasady konstruowania urządzeń typu Gateway, konwertujące dane do/z innych przemysłowych systemów komunikacyjnych.

1. WSTĘP

Przyglądając się aktualnej ofercie systemów sieciowych nie trudno dojść do wniosku, że jest ich wiele. Po okresie, kiedy każdy producent chciał mieć swój własny protokół (a czasami i indywidualne łącze), ilość ta nieco się zmniejszyła – zaczęły krystalizować się pewne standardy, ale problem wielości pozostał. Jest on szczególnie widoczny wśród sieci poziomu urządzeń. Z jednej strony każdy dostrzega konieczność ich stosowania, co zapewnia większą niezawodność systemu, możliwość zdalnej konfiguracji i diagnostyki oraz - co najważniejsze - szybkość i stabilność działania. Z drugiej strony, końcowy użytkownik - zwłaszcza średnich i dużych instalacji - otrzymując kilka rodzajów połączeń, pomimo benefitów płynących z ich wykorzystania, ma pewien problem z przepływem danych pomiędzy nimi. Tu można postawić kolejne pytanie – czy wymiana informacji pomiędzy sieciami jest konieczna i czy nie wystarczy połączenie „obiektywne”, a więc poprzez zwykłe wejścia i wyjścia. Otóż w bardzo prostych i w pełni nie związanych systemach – tak. Ale w przypadku bardziej złożonego systemu, jak również większej ilości danych - już nie. Tym bardziej „nie” iż od kilku lat obserwujemy jest znaczny wzrost wdrożeń systemów klasy ERP, dla których jedną z podstawowych informacji jest stan procesu produkcyjnego.

Tak więc połączyć systemy trzeba. Pytanie jest jednak – jak – i jakie właściwości powinno mieć urządzenie sprzęgające? Po lekturze i wysłuchaniu referatu odpowiedź powinna być znana.

2. DEVICENET I CANOPEN – ISTOTNE WŁASNOŚCI

Obydwa systemy sieciowe zostały oparte o technologię **producent/konsument PK (producer/consumer)**. Główną ideą modelu PK jest przesyłanie danych przy użyciu mechanizmu multicast., pozwalającego na odbieranie (konsumowanie) przez wiele węzłów tych samych danych, w tym samym czasie od jednego nadawcy (producenta). Każda ramka przesyłana przez sieć jest wyposażona w pole identyfikatora, pozwalające urządzeniom odbierającym na stwierdzenie czy ta informacja jest dla nich istotna, czy też należy ją zignorować. Pozwala to na osiągnięcie takiej samej efektywności transmisji, przy niższej prędkości, a co za tym idzie również i przy niższych kosztach. Jednocześnie, zastosowanie tych samych warunków komunikacji (kabel, prędkość) pozwala na przesyłanie większej ilości danych.

Podczas normalnej pracy obydwu sieci mamy zazwyczaj do czynienia z dwoma rodzajami informacji : danymi o stanie wejść/wyjść (**I/O messages, informacje z obiektów PDO**) oraz danymi o charakterze ogólnym, konfiguracyjnym i statusowym (**Explicit messages, informacje z obiektów SDO**). Poniżej zestawiono główne cechy obydwu komunikatów:

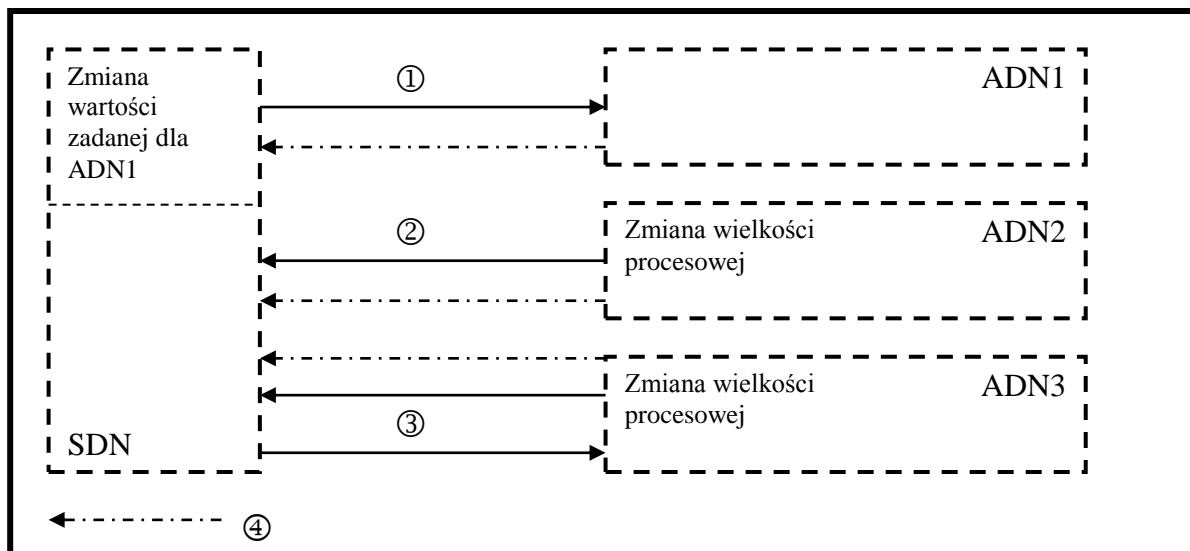
I/O message PDO	<ul style="list-style-type: none"> • Zorientowane tylko na przesyłanie danych I/O • Niewielka długość • Często zmieniają swoją zawartość (zmiennosc danych) • Posiadają duży stosunek danych do całości komunikatu • Wymagają szybkiego i częstego przesyłania (wysoki priorytet)
Explicit message SDO	<ul style="list-style-type: none"> • Zorientowane na przesyłanie informacji pomiędzy dwoma urządzeniami • Posiadają dość znaczną długość • Zawierają obok danych, informację o funkcji komunikatu, określoną ścisłym protokołem • Powstają w sytuacjach żądanie-odpowiedź (request/response) • Z uwagi na niekrytyczny rodzaj danych (konfiguracja, status, itp.) posiadają niski priorytet

Przesyłanie danych konfiguracyjnych zachodzi – jak to przedstawiono powyżej – na żądanie i wygląda podobnie w różnych systemach. Duże grupy danych są dzielone na części i przesyłane w chwilach małego obciążenia sieci. Znacznie trudniejsze do przesłania – z uwagi na swój charakter – są dane krytyczne. Ich wymiana wymaga określenia odpowiednich modeli, gwarantujących uzyskanie wysokiej efektywności.

W przypadku DeviceNet oraz CanOpen mamy trzy mechanizmy transferu danych krytycznych.

2.1. Change of State (COS), Event triggered PDO TM – 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 przesłania 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 połączenie③, w którym odebranie komunikatu jest potwierdzone.

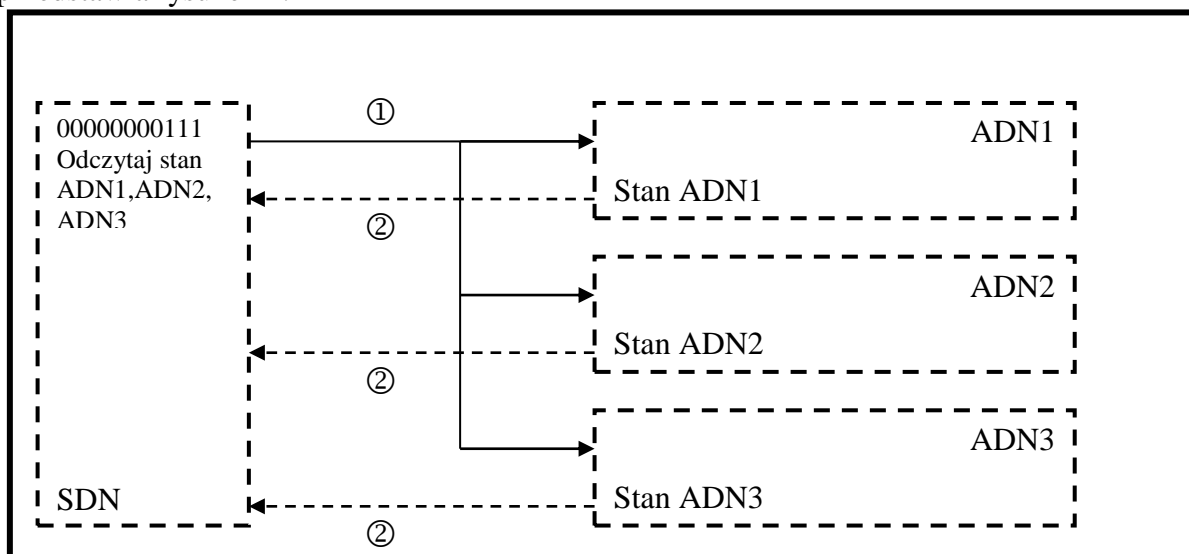


Rys.1.

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 ustawianym czasem) przesyłania informacji statusowej ④, nazywany Device Heartbeat (lub Timer Triggered PDO Mode). Transfer ten jest opcjonalny i stosowany w przypadku urządzeń, których awaria jest trudna do wykrycia dla obsługi technicznej.

2.2. Bit Strobe (BS), Sync Triggered PDO TM – 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 rysunek 2.

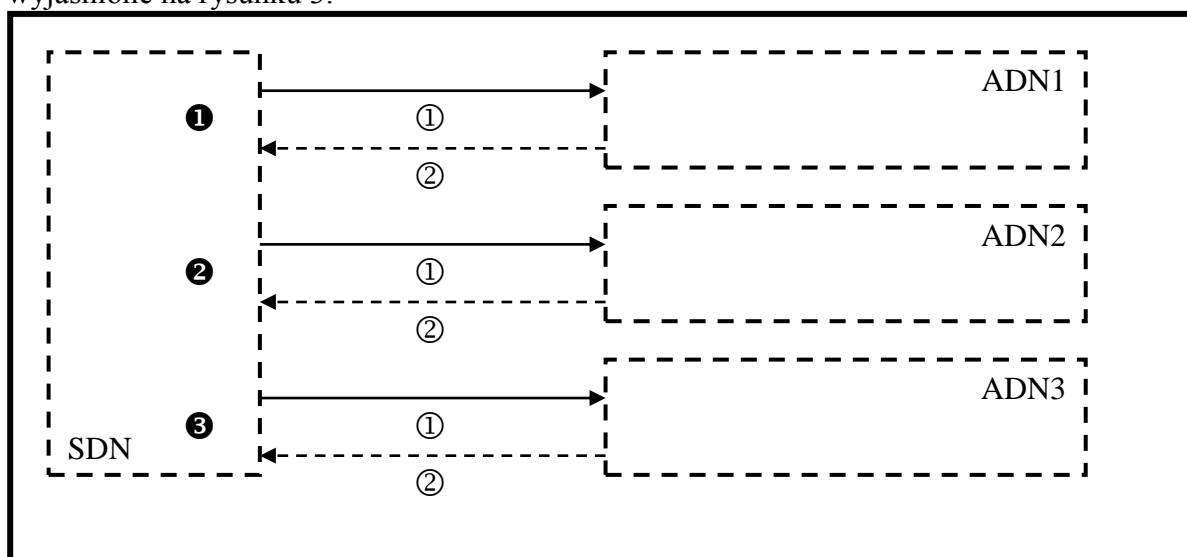


Rys.2. Idea komunikacji z wykorzystaniem strobowania bitowego

Moduł skanera wysyła w sieć komunikat ① Zgodnie z modelem Producent-Konsument dane te mogą zostać odebrane przez wszystkie urządzenia w sieci. W ramce tego komunikatu w przypadku DeviceNet, znajdują się 64 bity (8 bajtów czyli maksymalna ilość informacji w jednej ramce), z których każdy reprezentuje jeden węzeł. Jeśli pole zawiera „1”, a węzeł pracuje w trybie BS to wysyła do skanera komunikat ② zawierający aktualny stan urządzenia. Ustawienie wartości „0” jest równoznaczne z brakiem żądania odpowiedzi od urządzenia. W sieci CanOpen ramka strobuująca SYNC Message nie zawiera danych, a jej identyfikator jest sygnałem dla urządzeń o żądaniu przesłania swojego stanu. Mechanizm ten ma jak widać pewne cechy klasycznego odpytywania, jest jednak znacznie bardziej efektywny 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 fotowyłączniki, 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

2.3. Polling (PO), Remotely requested PDO TM – cykliczne odpytywanie

Komunikacja z wykorzystaniem mechanizmu PO jest realizacją standardowej (i często jedynej) idei wymiany informacji w sieciach poziomym urządzeń. Przekazywanie danych jest realizowane przez cykliczne łączenie się z każdym urządzeniem, co szczegółowo zostało wyjaśnione na rysunku 3.



Rys. 3. Ilustracja idei wymiany informacji w sieciach poziomym urządzeń

Podczas konfiguracji sieci w skanerze jest budowana lista urządzeń, według której następuje odpytywanie. W naszym przykładzie listę tworzą adaptory : ADN1 (pozycja ① na liście), ADN2 (②) oraz ADN3 (③). Rozpoczynając od pozycji ① skaner wysyła komunikat ① zawierający informacje przeznaczone dla ADN1 lub adres żądanych danych w ADN1. W odpowiedzi ② ADN1 przesyła odpowiednio potwierdzenie przyjęcia informacji lub dane wymagane przez skaner. Po zakończeniu połączenia z urządzeniem ① 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

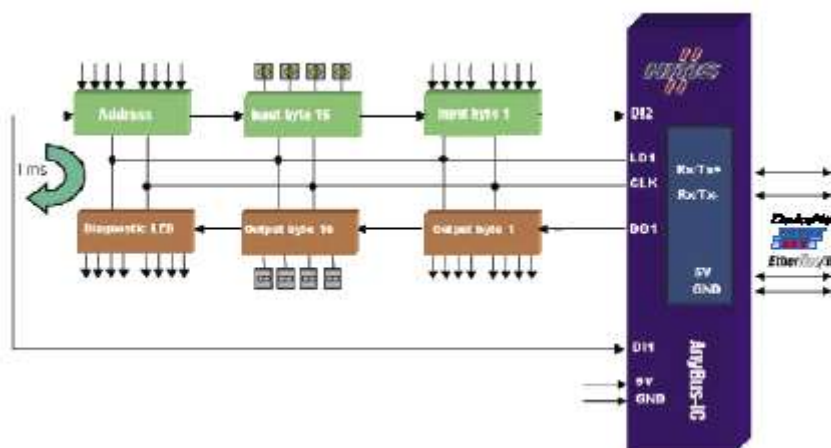
dynamiczny, zależny od rodzaju sterowanego procesu i wykorzystanych urządzeń. Mechanizm PO jest najczęściej wykorzystywany przy zbieraniu danych analogowych (temperatury, ciśnienia, przepływy itp.), których okres zmienności jest w prawdzie zróżnicowany, ale w miarę stały i łatwy do określenia.

3. URZĄDZENIA TYPU GATEWAY – KONSTRUKCJA I GŁÓWNE PROBLEMY

Konstruktorzy budujący urządzenia, których zadaniem jest sprzężenie różnych systemów sieciowych, stają zazwyczaj przed dwoma fundamentalnymi problemami:

- niezgodności medialnej, a więc różnic wynikających z zastosowania innych fizycznych połączeń oraz sterowania dostępem do łącza,
- niezgodności protokołowej, będącej efektem stosowania różnych modeli sieciowych.

Pierwszy z nich jest zazwyczaj rozwiązywany jedno- lub dwuetapowo. W przypadku sieci opartych o inny standard niż RS 485, a więc takich systemów jak Interbus-S, ASI czy Profibus (dla wysokich prędkości), istnieją dostawcy układów scalonych oferujący gotowe układy pozwalające zarówno na obsługę łącza fizycznego (a więc na zapewnieniu odpowiednich poziomów napięć, sterowania liniami itp.), jak i na realizację funkcji sterowania łączem. Jest to szczególnie istotne gdy chodzi o prędkość działania takiego układu. Warto zwrócić uwagę, że przy systemach pracujących z prędkościami powyżej 19200 baud, taki wariant, gdzie sterowanie dostępem odbywa się w sposób sprzętowy, jest wręcz standardem. Poniżej (rys.4.) pokazany jest schemat blokowy układu Anybus-IC firmy HMS.



Rys.4. Schemat blokowy układu Anybus-IC

Zastosowanie tego układu wymaga od konstruktora zbudowania własnego modelu danych (z dostępnych dla danego systemu sieciowego), zapewnienie jego obsługi i wymiany informacji z Anybus-IC. Zwalnia jednak z konieczności obsługi podstawowych zadań systemowych jak konstrukcja ramek, kontrolowanie sposobu ich wysłania i odbierania oraz obsługi samego procesu nadawania i detekcji kolizji. Te własności są szczególnie nieocenione jeśli spojrzymy na zadanie połączenia sieci DeviceNet i Profibus-DP. Z jednej strony mamy 500 kbaud i dostęp nasłuchowy CSMA z arbitrażem NBA, z drugiej zaś 12 Mbaud i dostęp tokenowo-M/S. Stosując taki układ, do rozwiązania pozostaje jedynie problem mapowania danych i konstrukcja odpowiednich obiektów widzianych w każdej sieci. Takiego komfortu nie ma w przypadku połączeń wykorzystujących jako medium standard RS-485. Tu zasadniczo dostępne są układy magistralowe (sterowniki portów), zaś cała obsługa (tworzenie ramek, koordynacja ruchu) spoczywa na konstruktorze. Realizując np. obsługę sieci Modbus,

musimy w pełni określić jakie będą zasady ruchu (stacja master, zarządzanie timeout'ami) oraz konstruować odpowiednio ramki i pilnować odstępów międzyramkowych. W przypadku połączeń wolnych jest to oczywiście możliwe do zrealizowania, ale np. przy przejściu na prędkości 38,4 lub 57,6 kbaud, znacznie efektywniej pracują sprzętowe rozwiązania.

Drugi problem – niezgodność protokołowa – wydaje się być trudniejszy do rozwiązania. Tu nasuwają się generalnie dwa sposoby podejścia:

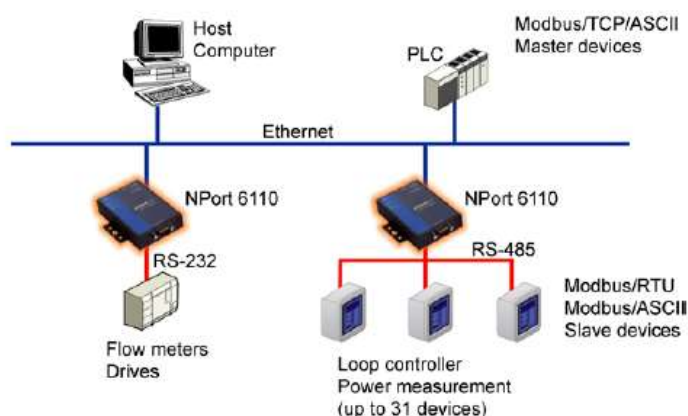
- dynamiczne przekładanie danych pomiędzy ramkami obydwu sieci,
- konstrukcja bazy danych widzianej jako zespół odpowiednich obiektów w każdej sieci.

Wariant z dynamicznym tłumaczeniem protokołów sprawdza się jedynie w przypadku systemów z polingowym modelem transmisji, z jednoznacznie określoną stacją master, odpytującą poszczególne stacje slave. W systemach tych możemy określić następujące warianty pracy :

- Master / Master,
- Master / Slave,
- Slave / Slave.

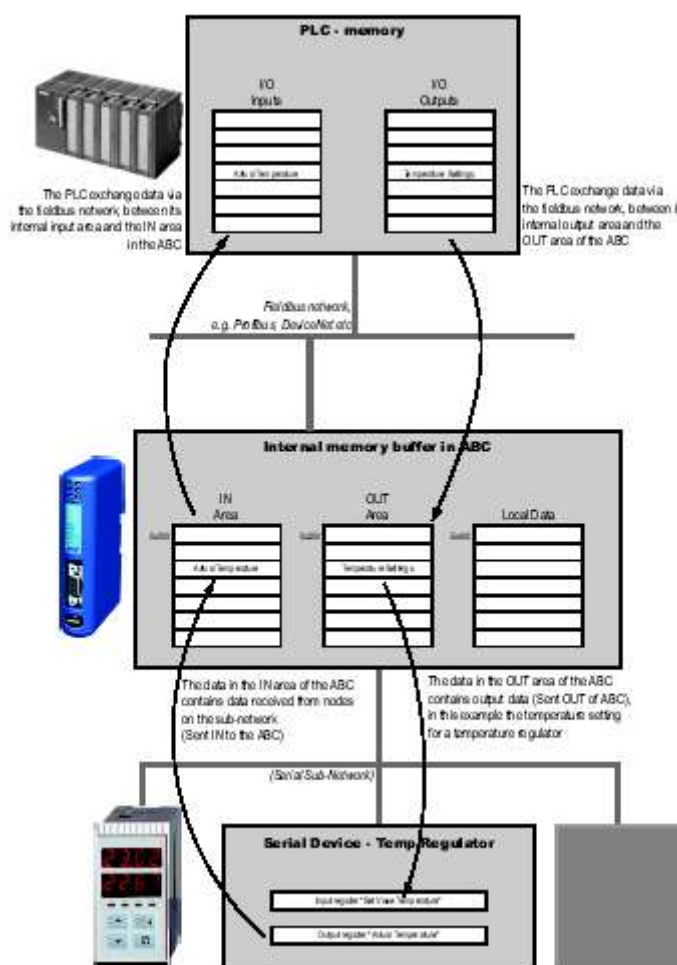
Pamiętając, iż gateway ma tylko zamieniać ramki (a właściwie ich zawartość), każdy z nich niesie pewne trudności w realizacji. Przypadek (a) wymaga korelacji odpytywania po obydwu stronach, jak również eliminuje inne stacje master, co zubaża całość aplikacji. Dodatkowo dochodzi problem składowania danych. W wariacie (c) występuje ponownie problem korelacji - znacznie głębszy. Poprawne działanie takiego gateway'a wymaga synchronizacji dwóch stacji master w niezależnych sieciach. Praktycznie jest to bardzo trudne do zrealizowania. I wreszcie tryb pracy (b) – z jednej strony master, z drugiej slave. Wariant ten jest najczęściej spotykany, gdyż strona slave jest obsługiwana przez niezależną stację master, a odebranie zapytania generuje wysłanie żądania po stronie master. Tu jednak wciąż pozostaje (podobnie jak w (a) i (c)) problem szybkiej konwersji.

W praktyce spotykamy jedynie urządzenia pracujące w klasie (b) w podobnych co do konstrukcji protokołach. Najlepszym tego przykładem jest Nport 6110 firmy Moxa, łączący systemy Modbus i Modbus TCP.



Rys.5. Ilustracja systemu Nport 6110 firmy Moxa

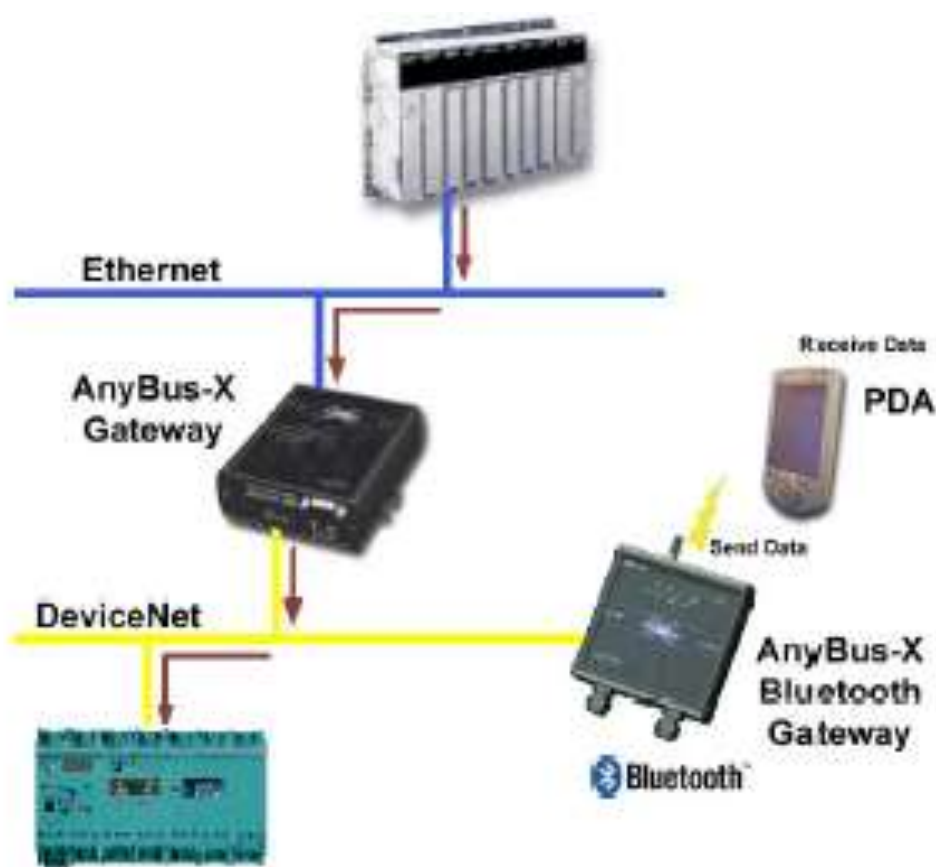
Zatem by zrealizować konwerter pozwalający na wykorzystanie wszystkich właściwości łączonych sieci – a w naszym przypadku mechanizmów COS, BS oraz podziału danych na istotne i konfiguracyjne – musimy zastosować wariant z implementacją bazy danych jako jądra gateway’ a. Rysunek 6 prezentuje ideę pomysłu.



Rys.6. Prezentacja wariantu z implementacją bazy danych

Konwerter został wyposażony (obok układów realizujących zadania warstw fizycznej i łącza) również w pewną formę bazy danych. Jest to zespół tablic, których przeznaczenie, jak i funkcjonowanie jest zależne od rodzaju sieci. Jeśli określimy jedną stronę jako stronę slave w DeviceNet, to jednocześnie definiujemy rodzaj współpracy z siecią (Change-Of-State, Polling, Bit Strobe [w praktyce tylko dla sensoryki]) oraz mamy możliwość dzielenia danych na ważne (Implicit) oraz drugoplanowe (Explicit). Z drugiej strony możemy np. zastosować port master dla magistrali ASI. Będzie on cyklicznie odpytywał całą sieć i wypełniał tablicę buforową. W przypadku zmian wartości, zadziała mechanizm COS i gateway wyśle w sieć DeviceNet odpowiednią informację. Jeśli otrzyma nowe dane od skanera, to zostaną one wpisane do tablicy buforowej i zostaną przesłane do odpowiednich stacji ASI Slave. Mechanizm ten cechuje się wysoką stabilnością i co najważniejsze niezależnością funkcjonalną obydwu procesów. Wymiana przez wspólną pamięć, pozwala zachować zarówno pełne właściwości łączonych systemów jak i nie zmniejszać ich efektywności. Warto tu zwrócić uwagę na obecność obszaru Local Data, którego zadaniem jest przechowywanie nie tylko informacji konfiguracyjnej, ale także danych statusowych i diagnostycznych. Pozwala to na osiągnięcie sprzęgu nie tylko w zakresie danych, ale i bezpieczeństwa systemu.

Rysunek 7 pokazuje przykładową realizację połączenia pomiędzy systemami Ethernet/IP oraz DeviceNet i pośrednio Bluetooth, realizowaną za pomocą konwerterów AnyBus-X firmy HMS.



Rys.7. Ilustracja przykładowej realizacji połączeń międzysystemowych

4. ZAKOŃCZENIE

Jeszcze kilka lat temu wystarczyło zainstalowanie odpowiednio dużego komputera, wyposażenie go w szereg kart sieciowych, dodanie wizualizacji i w zasadzie sprzęg był gotowy. Celem - jak widać - było jedynie doprowadzenie informacji do lokalnej stacji operatorskiej. Dzisiaj wymagania się znacznie zmieniły. Projektant musi zapewnić szereg połączeń na wszystkich trzech poziomach sieciowych spotykanych w przemyśle. I nie chodzi tu o zebranie danych w „jakimś” czasie. System ma pozwalać na pracę w czasie rzeczywistym, bez tworzenia wąskich gardeł, a jednocześnie w sposób bezpieczny. Z jednej strony mamy takie warunki, z drugiej - chcemy wykorzystywać wszystko to, co dają magistrale CANowskie, a więc model Producent-Konsument i wynikające z niego sposoby wymiany informacji, a głównie Change-Of -State i Event Triggered PDO TM. Obecnie widać - jak wynika to z niniejszego referatu - tylko jedno rozwiązanie: inteligentny gateway z wbudowaną bazą danych.

5. LITERATURA

- [1] TUTAJ R: Przemysłowe systemy komunikacyjne typu Fieldbus, PAR 4'97, s. 32-35.
- [2] TUTAJ R: DeviceNet – efektywne rozwiązanie problemu komunikacji typu Fieldbus PAR 4'98, s. 34-27.
- [3] TUTAJ R: Różne urządzenia-jedna sieć: sterowanie i pomiary za pomocą DeviceNet PAR 7-8'99, s. 39-43.
- [4] TUTAJ R: Sterownik PLC w środowisku wielosieciowym, PAR 4'00, s. 7-11.

COMMUNICATION MECHANISMS IN DEVICENET AND CANOPEN NETWORKS AND DESIGN THE GATEWAY DEVICES.

Abstract: The paper presents the basics of communication mechanisms in DeviceNet and CanOpen networks and based on them the rules of construction gateways to/from other systems.

anOpen networks and based on other systems.

Recenzent: mgr inż. Jerzy JURA