ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1997
Seria: INFORMATYKA z. 33 Nr kol. 1377

Wojciech MIELCZAREK, Ryszard PAWŁOWSKI
Politechnika Śląska, Instytut Informatyki

HIERARCHICZNA SIEĆ POLOWA STEROWANA ROZKAZAMI JĘZYKA SCPI*

Streszczenie. W publikacji przedstawiono koncepcję hierarchicznej sieci polowej sterowanej rozkazami języka SCPI. Sieć potraktowano jako rozproszone urządzenie pomiarowe wyposażone w hierarchiczny dekoder rozkazów SCPI składający się z dwóch części. Za przepływ komunikatów w sieci odpowiada część rozproszona dekodera zaimplementowana w węzłach sieci, a za realizację poleceń część skupiona znajdująca się w urządzeniu końcowym, dla którego przeznaczony jest rozkaz.

HIERARCHICAL FIELDBUS CONTROLLED BY SCPI COMMANDS

Summary. In the paper an idea of hierarchical fieldbus controled by SCPI commands is presented. The network was treated as an distributed programmable instrument with hierarchical SCPI commands decoder consisting of two parts. The distributed part of SCPI decoder (implemented in the network nodes) is responsible for flowing messages through the network. The second part of SCPI decoder is implemented in the terminal device, to which a command is addressed, and simply executes a command.

1. Wprowadzenie

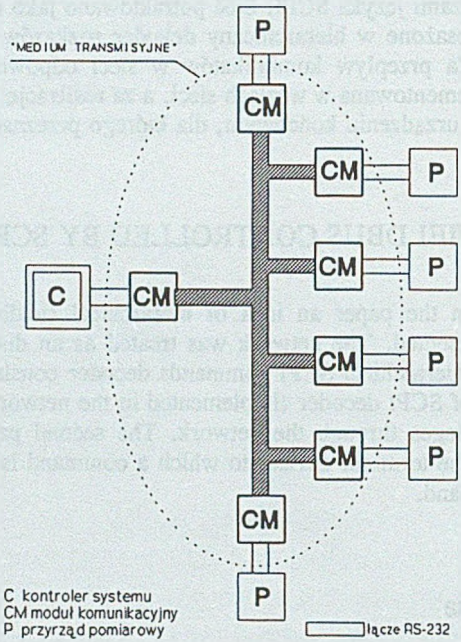
Niniejsza publikacja przedstawia koncepcję hierarchicznej, wielopoziomowej sieci polowej o następujących cechach:

- deterministyczny, sterowany centralnie dostęp do łącza pod kontrolą protokołu MODBUS,
- konfiguracja magistralowa oparta na łączu w standardzie RS-485,

*) Praca finansowana przez KBN w ramach projektu badawczego 302/T11/97/12.

- sterowanie operacjami w sieci za pośrednictwem rozkazów w języku SCPI (Standards Commands for Programable Instruments),
- możliwość bezpośredniego podłączenia do sieci urządzeń wyposażonych w interfejs szeregowy RS-232C lub urządzeń wyposażonych w łącze RS-422A za pośrednictwem konwertera RS232C<->RS422A.

Projektowana sieć stanowi medium transmisyjne przeznaczone dla rozproszonego systemu pomiarowo-kontrolnego (przemysłowego lub laboratoryjnego) opartego na urządzeniach wyposażonych w łącze RS-232, w którym przesyłane są niewielkie bloki danych, a czasy odpowiedzi nie są krytyczne. Takie medium może być tanią alternatywą dla istniejących sieci polowych, a podstawowe jego zastosowanie obejmuje małe zakłady produkcyjne, które nie stać na duże inwestycje związane z siecią sterowników przemysłowych. Na rys.1 przedstawiono projektowaną sieć polową jako medium transmisyjne dla urządzeń wyposażonych w łącze RS-232.



Rys. 1. Projektowana sieć jako medium transmisyjne dla urządzeń wyposażonych w RS-232

Fig. 1. The fieldbus under project as an transmission medium for the RS-232 devices

2. Charakterystyka sieci polowych i geneza projektu

Szeregowe łącza komunikacyjne stały się kluczowe dla rozwoju rozproszonych systemów pomiarowych i kontrolnych mających podstawowe znaczenie w automatyzacji procesów wytwórczych oraz w wielu innych zastosowaniach (medycynie, transporcie, budownictwie, badaniach naukowych). Szczególnego znaczenia nabrały przy tym sieci polowe bezpośrednio łączące urządzenia pomiarowo-kontrolne (dalej będziemy również używać określenia "urządzenia polowe") wykorzystywane do akwizycji danych i sterowania konkretnym, wydzielonym procesem [3]. Wiele funkcji, tradycyjnie wykonywanych na wyższych poziomach hierarchicznego systemu sterowania, zostało obecnie przeniesionych niżej do urządzeń polowych. Pozwala to na ograniczenie zadań wyższych poziomów systemu jedynie do funkcji nadzorczych, ale jednocześnie oznacza konieczność wyposażenia systemów polowych w dużą "inteligencję".

Sieciom polowym stawia się wysokie wymagania. Przede wszystkim ich operacje muszą być ograniczone czasowo, co wynika z wymagań sterowania w czasie rzeczywistym oraz monitoringu. Protokół komunikacyjny sieci polowej musi zapewnić transakcje cykliczne oraz aperiodyczne (sterowane zdarzeniami) z możliwością przypisania priorytetu poszczególnym zdarzeniom. Dołączanie i usuwanie urządzeń nie może zmieniać parametrów krytycznych, a ewentualne uszkodzenie urządzenia nie może zakłócać pracy sieci. Sieć polowa musi zapewniać ponadto wysoką jakość transferu informacji, a jakiekolwiek błędy muszą być wykrywane i raportowane [1,3].

Szerokie zapotrzebowanie na sieci polowe spowodowało powstanie wielu rozwiązań firmowych. Niektóre z nich, wspierane przez krajowe komitety normalizacyjne, stały się powszechnie uznanymi standardami krajowymi (np. sieć Profibus w Niemczech lub FIP we Francji). Nie zakończono pracy nad długo oczekiwanym standardem IEC FIELDBUS (IEC DIS 1158).

W tabeli 1 zebrano kilka najważniejszych sieci zaprojektowanych do zastosowań polowych i uzupełniono ten zestaw o projektowaną sieć hierarchiczną. Z porównania przedstawionych powyżej sieci polowych wynika duża różnorodność rozwiązań i to pod każdym względem (różne sposoby kodowania bitów, metody dostępu do łącza, parametry i modele komunikacyjne). Wszystkie jednak spełniają wymagania stawiane sieciom polowym, a o ich rozpowszechnieniu zadecydowała dostępność scalonych kontrolerów sieciowych, które pozwoliły na znaczące obniżenie kosztów urządzeń polowych. Mimo to sieci polowe nie są tanie i wydaje się potrzebne poszukiwanie rozwiązań pozwalających na dalsze ograniczenie kosztów.

Wśród projektantów sieci polowych istnieje tendencja do zbudowania sieci uniwersalnej, zdolnej do spełnienia najbardziej surowych i różnorodnych wymagań. Taka sieć zawsze

będzie droga. Stale istnieje jednak wiele zastosowań, w których wystarczy sieć prostsza, oparta na łatwo dostępnych, uniwersalnych i tanich układach mikrokontrolerów (np. układy serii MCS-51). Taka sieć też musi wiernie przenosić informację, być odporna na błędy i uszkodzenia urządzeń, też musi dopuszczać do wymiany urządzeń w trakcie pracy i zezwalać na realizację transakcji periodycznych i aperiodycznych. W przypadku sterowania lub monitorowania wolnozmiennego procesu przemysłowego czas transmisji danych w sieci przestaje być parametrem krytycznym. Dla takich zastosowań można obniżyć szybkość transmisji danych i wykorzystać prosty protokół komunikacyjny, możliwy do implementacji w systemach mikroprocesorowych. Spowoduje to zmniejszenie kosztów sieci polowej.

Z takimi właśnie problemami zetknęliśmy się w naszej praktyce zawodowej. Projektując na przykład sieć polową do monitorowania produkcji i sterowania pracą agregatów na wydziale włóknin w Zakładzie Lentex w Lublińcu rozwiązania firmowe sieci polowych okazały się za drogie i "za dobre". Stosunkowo nieduża liczba urządzeń polowych, niewielki rozmiar bloków informacyjnych przesyłanych w systemie, dopuszczalny duży czas reakcji na zmianę mierzonych wielkości, niewielka częstotliwość próbkowania spowodowały, że zupełnie wystarczający okazał się prosty system oparty na łączu RS-485 i protokole MODBUS, wykorzystujący transmisję znakową o szybkości transmisji 9600 bit/s. Przy okazji wystąpił dodatkowy problem. Do pomiarów różnych wielkości fizycznych na agregatach były już zainstalowane urządzenia wyposażone w interfejs RS-232 i trzeba było projektowaną sieć polową przystosować do tych urządzeń, a nie urządzenia do sieci polowej.

Z podobną potrzebą adaptacji sieci do istniejących urządzeń pomiarowych zetknęliśmy się podczas projektowania rozproszonego systemu pomiarowego opartego na łączu szeregowym, przeznaczonego do pomiarów laboratoryjnych i testowania wyrobów przemysłowych. W tym przypadku chodziło o połączenie przyrządów firmy Hewlett Packard wyposażonych w interfejs RS-232 w rozproszony system oparty na łączu RS-485. Przyrządy pomiarowe były zgodne z normą IEEE 488.2, a ich programowanie odbywało się w języku SCPI.

Poznanie języka SCPI, a w szczególności jego hierarchicznej struktury podsunęło pomysł wykorzystania go nie tylko w charakterze języka programowania przyrządów pomiarowych, ale także do sterowania przesyłem danych w sieci hierarchicznej. Analizując przykłady sieci polowych łatwo zauważyć ich wspólną cechę - sieci te są zazwyczaj jednopoziomowe. Fakt ten wynika z ich przeznaczenia, są to bowiem sieci mające łączyć urządzenia pomiarowo-sterujące jednej linii technologicznej (jednego procesu). W wielu wypadkach jednak celowe wydaje się wyodrębnienie pewnych fragmentów procesu i ich niezależna obsługa, z czym

Tablica 1

Porównanie standardów sieci polowych

Sieć	Kodowanie bitu	Reguła dostępu do łącza	Warstwy ISO-OSI	Kontroler scalony	Standard
Arcnet	NRZ	Token passing	1,2	SMC	TA
ARINC 629	Manchester	CSMA	1,2	MCE	TA(SAE)
CAN	NRZ	CSMA	1,2	Intel, Phillips, Motorola	P
CEbus	NRZ	CSMA	1,2,3,7	?	N(EIA)
Echelon LON	Manchester (diff.)	CSMA	1,2,3,4,5 6,7	Neuron chip	P
FIP	Manchester	Master/slave	1,2,7	VLSI Tech.	N
IEEE 1118 (Bitbus)	NRZI	HDLC	1,2,7	Intel	N
IEEE P1073	NRZI	Master/slave	1,2,7	-	N
IEC Fieldbus	Manchester	Token passing i Master/slave	1,2,7	Ship Star	I
MIL-STD-1553B	Manchester	Master/slave	1,2	wielu producentów	NATO
PROFIBUS	Manchester	Token passing i Master/slave	1,2,7	ISP	N
Projektowana sieć	**NRZ**	**Master/slave**	**1,2,3,4,5 6,7**	**MCS-51**	**??**

Standardy:

N narodowy I międzynarodowy

TA handlowy P firmowy

wiąże się podział sieci na moduły (podsieci pomiarowo-kontrolne). Zazwyczaj wykorzystuje się do tego celu sieci polowe na niższym poziomie połączone siecią lokalną (LAN) na poziomie wyższym. Wynika to z faktu, że protokoły sieci polowych zazwyczaj nie przewidują wymiany informacji z urządzeniami z innej sieci (brak w nich zaimplementowanych protokołów routingu, warstwy 3 i 4 modelu OSI-ISO). Rozwiązanie z różnymi protokołami na poszczególnych poziomach hierarchii sprawia jednak problemy komunikacyjne (skomplikowana konwersja protokołów). Wyjątkami są pod tym względem sieci Bitbus i Echelon.

Wymienione doświadczenia przekonały nas o potrzebie zaprojektowania prostej, hierarchicznej sieci polowej opartej na łączu RS-485, sterowanej jednolicie na wszystkich poziomach hierarchii oraz pozwalającej na podłączenie urządzeń wyposażonych w interfejs RS-232. Sieć taka powinna wykorzystywać możliwie nieskomplikowany protokół, a jej moduły komunikacyjne należy zbudować na tanich i dostępnych (a więc uniwersalnych) układach scalonych. Projektowana sieć powinna ponadto łączyć w sobie najlepsze cechy sieci przemysłowych i laboratoryjnych. Z punktu widzenia operatora systemu powinna się prezentować jako rozproszone, wielofunkcyjne urządzenie pomiarowe.

Oznacza to, że konieczne jest zaprojektowanie sprzętu oraz oprogramowania sieci polowej, która dla podłączanych do niej urządzeń stanowi "medium transmisyjne", do którego dostęp odbywa się poprzez łącze RS-232. Każde wejście wyposażone jest w sterownik komunikacyjny, który:

- umożliwia przypisanie urządzeniu adresu,
- realizuje wymianę komunikatów pod kontrolą wybranego protokółu komunikacyjnego,
- dokonuje konwersji sygnałów ze standardu RS-232 na standard elektryczny łącza sieci i odwrotnie.

Zaproponowane przez nas rozwiązanie ujednolica sposób komunikacji, zarówno pomiędzy urządzeniami obsługującymi dany proces (pracującymi na pojedynczej sieci), jak i pomiędzy urządzeniami z różnych sieci (pomiędzy różnymi poziomami hierarchii). Proponowany protokół dopuszcza dowolną liczbę poziomów hierarchii (w praktycznych rozwiązaniach wystarczą 2 lub 3 poziomy), z pełną autonomią pracy każdej podsieci. W każdej podsieci moduł master może realizować swój własny program sterujący, a jednocześnie jego praca może być asynchronicznie wstrzymywana na czas realizacji polecenia od układu mastera na poziomie wyższym.

3. Założenia do projektu hierarchicznej sieci polowej

Dla projektowanej sieci polowej przyjęto następujące założenia:

1. Sieć ma hierarchiczną strukturę wielopoziomową reprezentowaną przez drzewo sieci.

2. W każdej gałęzi (podsieci) dowolnego poziomu wyróżnia się jedną stację nadrzędną, dla której przyjęto nazwę master, oraz stacje podrzędne (wszystkie pozostałe) nazywane slave. Stacja master zarządza dostępem (dostęp deterministyczny) do sterowanej przez siebie podsieci realizując tzw. transakcję. Transakcja jest cyklem złożonym z dwóch operacji:
- stacja master adresuje wybrane urządzenie i przesyła do niego polecenie,
- urządzenie zaadresowane przez mastera wykonuje polecenie i przesyła do stacji master odpowiedź potwierdzającą odebranie polecenia i jego wykonanie (lub niewykonanie),

Dopuszcza się wysyłanie tzw. rozgłoszenia (broadcast) jako polecenia skierowanego do wszystkich stacji slave. Na polecenie rozgłoszeniowe stacje slave nie wysyłają odpowiedzi. Z określonych powyżej operacji stacji master wynika podrzędna rola stacji slave jako urządzenia, które:
- nie może samodzielnie (nie zapytane) wysłać komunikatu zdalnego,
- odpowiedzi kieruje zawsze do stacji master, co oznacza, że ewentualny dialog pomiędzy stacjami podrzędnymi może odbywać się tylko przez stację master i pod jej kontrolą. Na rys.2 przedstawiono hierarchiczną sieć polową z modułami master i slave.

3. Przejście pomiędzy dwoma kolejnymi poziomami sieci odbywa się za pośrednictwem pary urządzeń: slave podłączony do podsieci na wyższym poziomie (tzw. slave pośredniczący) i współpracujący z nim master sterujący pracą podsieci na niższym poziomie.

Oba wymienione urządzenia mogą znajdować się w bezpośrednim sąsiedztwie lub mogą być urządzeniami odległymi. Łącze komunikacyjne pomiędzy stacją slave wyższego poziomu a skojarzoną z nią stacją master niższego poziomu musi zapewnić transmisję na odległości do 1200 m. Na rys. 2 pary "slave pośredniczący-master na niższy poziomie" reprezentują S2-M2, S4-M3 i S6-M4.

4. Gałęzie sieci mogą pracować autonomicznie, co oznacza, że znajdujący się w gałęzi master może wykonywać własny cykl zbierania danych ze stacji slave znajdujących się w jego gałęzi. Dane te mogą być udostępniane masterowi na wyższym poziomie za pośrednictwem zmiennych stanu.

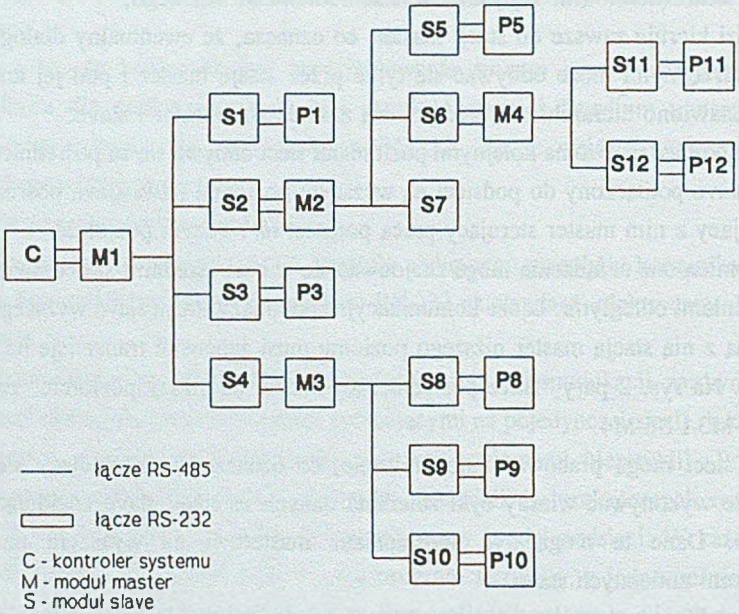
5. Polecenia skierowane z mastera na wyższym poziomie do mastera na niższym poziomie mają wyższy piorytet od operacji mastera znajdującego się na poziomie niższym. Oznacza, to że po odebraniu polecenia z wyższego poziomu, master na niższym poziomie musi przerwać aktualnie wykonywany program i zrealizować operację zleconą mu przez mastera znajdującego się na poziomie wyższym. W tym sensie będziemy mówić o masterze

nadrzędnym (stacja na poziomie wyższym) i masterze podrzędnym (stacja na poziomie niższym).

6. Każdy moduł wyposażony jest w dwa multipleksowane łącza szeregowe: jedno łącze od strony sieci (w standardzie RS-485) i drugie dla bezpośredniej komunikacji z urządzeniem pomiarowym lub innym modułem sieciowym (w standardzie RS-232).

7. Moduły master i slave oparte są na mikrokontrolerze rodziny MCS-51 i pozostają identyczne od strony sprzętowej. Jednorodność elementów sieci ograniczy koszt projektowanego medium.

8. Sterowanie pracą systemu odbywa się za pośrednictwem rozkazów w języku SCPI. W przypadku modułów komunikacyjnych (para modułów slave pośredniczący i master podrzędny tworząca węzeł sieci) rozkazy języka SCPI służą do adresowania węzła sieci. W przypadku modułów slave obsługujących bezpośrednio przyrządy pomiarowo-kontrolne (slave końcowy) rozkazy SCPI określają właściwą operację, którą ma wykonać przyrząd pomiarowy.



Rys. 2. Hierarchiczna sieć polowa z modułami master i slave
Fig. 2. Hierarchical fieldbus with master and slave modules

4. Protokół komunikacyjny

Protokół komunikacyjny musi przede wszystkim realizować przyjętą zasadę dostępu do łącza, a więc dostęp deterministyczny oparty na transakcji kontrolowanej przez stację master. Odpowiedni wydał nam się protokół MODBUS [5] ze względu na następujące cechy:

- prosta reguła dostępu oparta na zasadzie "master-slave",
- potwierdzanie wykonania rozkazów zdalnych i sygnalizacja błędów,
- zabezpieczenie przesyłanych komunikatów przed błędami,
- wykorzystanie asynchronicznej transmisji znakowej zgodnej z RS-232.

W protokole MODBUS polecenia i odpowiedzi przesyłane są w postaci bloków (ramek) o ściśle określonym formacie:

(znacznik początku ramki)(adres slave)(kod rozkazu)(pole danych)
(suma kontrolna)(znacznik końca ramki)

W projektowanej sieci rozkazy SCPI przesyłane będą jako argumenty rozkazów protokołu MODBUS w postaci ciągu znaków ASCII. Każdy master podrzędny musi sprawdzić, dla kogo przeznaczony jest dany rozkaz i przesłać go do właściwego adresata (stacji) w zarządzanej przez siebie gałęzi sieci. W praktyce oznacza to przesłanie "nowego" rozkazu SCPI, który jest fragmentem rozkazu odebranego od mastera nadrzędnego po odrzuceniu części początkowej określającej adres modułu slave. Adresatem rozkazu może być stacja slave współpracująca z urządzeniem pomiarowym mającym wykonać rozkaz lub stacja slave współpracująca z masterem podrzędnym znajdującym się na kolejnym poziomie hierarchii. W tym drugim przypadku rozkaz przekazywany jest przez stację slave do związanego z nią mastera, gdzie następuje analogiczny proces analizy i przekazania rozkazu do właściwego adresata.

Analizę i przekazywanie polecenia w sieci hierarchicznej wyjaśnimy za pośrednictwem przykładowej konfiguracji przedstawionej na rys.3a. W sieci tej występują trzy poziomy hierarchiczne. Na pierwszym poziomie układem mastera jest M1, natomiast S1, S2 i S3 są układami typu slave. S3 jest układem pośredniczącym umożliwiającym współpracę z podsiecią sterowaną przez master M2. W sieci tej układami typu slave są S4 i S5. S5, podobnie jak S3, jest układem pośredniczącym do podsieci sterowanej przez master M3. W jego podsieci pracują układy typu slave S6 i S7. Zakładając, że rozkaz *DEV<n>* adresuje urządzenie <n> na kolejnym poziomie hierarchii, a rozkaz *COMM<n>* jest rozkazem realizującym akcję <n> w zaadresowanym urządzeniu, prześledzimy sposób analizy i wykonania dwóch przykładowych rozkazów. W chwili początkowej każdy układ w sieci oczekuje na rozkaz z "poziomu wyższego" np: M1 z sieci A, S3 z sieci B i M3 z sieci E.

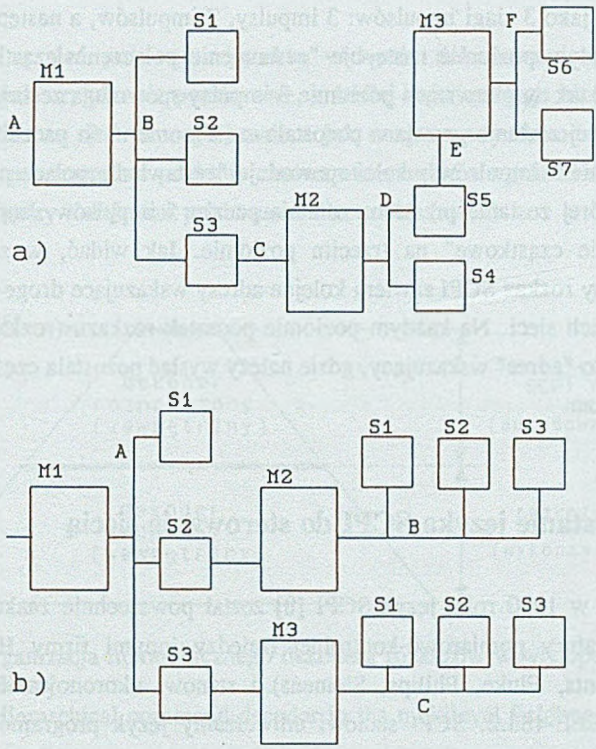
"DEV1:COMM1"

- układ M1 otrzymuje tekst rozkazu "DEV1:COMM1";
- analiza treści wskazuje, że otrzymany rozkaz jest rozkazem adresowym "DEV<n>";
- z rozkazu wyodrębniany jest adres na niższym poziomie "DEV1" i treść rozkazu do przekazania "COMM1";
- układ M1 przełącza się na sieć B;
- do układu o adresie 1 na sieci B przekazywany jest rozkaz "COMM1";
- układ M1 czeka na odpowiedź z sieci B;
- rozkaz odbiera układ S1;
- analiza treści wskazuje, że jest to rozkaz wykonawczy "COMM1";
- układ S1 wykonuje rozkaz;
- do oczekującego układu M1 zostaje odesłana odpowiedź;
- układ M1 po odebraniu odpowiedzi przełącza się na sieć A i odsyła odpowiedź, którą otrzymał od układu S1.

"DEV3:DEV5:DEV7:COMM2"

- układ M1 otrzymuje tekst rozkazu "DEV3:DEV5:DEV7:COMM2";
- otrzymany rozkaz jest rozkazem adresowym, następuje więc jego rozbicie na adres na niższym poziomie "DEV3" i treść rozkazu do przekazania "DEV5:DEV7:COMM2";
- układ M1 przełącza się na sieć B;
- do układu o adresie 3 przekazywany jest rozkaz "DEV5:DEV7:COMM2";
- układ M1 czeka na odpowiedź z sieci B;
- rozkaz odbiera układ S3;
- otrzymany rozkaz jest rozkazem adresowym, ale ponieważ układ S3 jest układem slave, otrzymany rozkaz nie będzie analizowany, tylko przekazywany bez modyfikacji do układu mastera na niższym poziomie;
- układ S3 przełącza się na sieć C, przekazuje rozkaz "DEV5:DEV7:COMM2" i czeka na odpowiedź;
- rozkaz otrzymuje układ M2 i podobnie jak M1 dekoduje go jako rozkaz adresowy, a następnie rozbija na część adresową "DEV5" i właściwy rozkaz dla niższego poziomu "DEV7:COMM2";
- po przełączeniu się na sieć D układ M2 przesyła rozkaz "DEV7:COMM2" do urządzenia o adresie 5 i czeka na odpowiedź;
- rozkaz odbiera S5 i zachowuje się identycznie jak S3, tj. zmienia sieć na E, przekazuje nie zmieniony rozkaz "DEV7:COMM2" do układu M3, a następnie czeka na odpowiedź;
- układ M3, podobnie jak M1 i M2, analizuje otrzymany rozkaz i po stwierdzeniu, że jest to rozkaz adresowy, rozbija go na część adresową "DEV7" i właściwy rozkaz dla niższego poziomu "COMM2";

- po przełączeniu się na sieć F układ M3 przesyła rozkaz "COMM2" do urządzenia o adresie 7 i czeka na odpowiedź;
- rozkaz "COMM2" zostaje odebrany przez układ S7, zdekodowany i wykonany;
- układ S7 po wykonaniu rozkazu odsyła odpowiedź, która zostaje odebrana przez M3;
- w kilku następnych krokach odpowiedź ta jest przekazywana kolejno przez układy M3->S5->M2->S3->M1. Po przekazaniu odpowiedzi każdy z układów oczekuje na kolejny rozkaz.



Rys. 3. Przykładowe konfiguracje sieci polowej do analizy przepływu komunikatów
Fig. 3. Two examples of the fieldbus configuration for messages flow analysis

Każda podsieć ma niezależne (na poziomie protokołu MODBUS) adresowanie, zatem w systemie na różnych podsieciach mogą pracować układy o tym samym adresie. Przykładowo w systemie przedstawionym na rys.3b wszystkie układy S1 mają adres 1. Są jednak widoczne jako odrębne urządzenia:

"DEV1:COMM1" polecenie skierowane do układu S1 na sieci A;

"DEV2:DEV1:COMM1" polecenie skierowane do S1 na sieci B;
"DEV3:DEV1:COMM1" polecenie skierowane do S1 na sieci C.

Układy typu master, oprócz wykonywania rozkazów adresowych, mogą realizować także rozkazy zwykłe. Przykładem niech będzie rozkaz "COMM2" skierowany do układu M3 na rys.3b: "DEV3:COMM3".

Przedstawiony powyżej uproszczony algorytm działania sieci hierarchicznej można przyrównać do działania telefonicznych central wybierakowych. W centralach tych wybrany przez abonenta numer jest przekazywany jako ciąg pogrupowanych impulsów: np. numer 375 będzie przekazany jako 3 ciągi impulsów: 3 impulsy, 7 impulsów, a następnie 5 impulsów. W centrali na każdym poziomie następuje "zestawienie połączenia cząstkowego" według liczby impulsów. I tak na pierwszym poziomie 3 impulsy spowodują zestawienie połączenia z "linią 3", po której zostanie przesłana pozostała część numeru, tj. paczki 7 i 5 impulsów. Na drugim poziomie 7 impulsów z kolei spowoduje "zestawienie połączenia cząstkowego" z "linią 7", po której zostanie przesłana ostatnia paczka 5 impulsów. Impulsy te zestawią kolejne "połączenie cząstkowe" na trzecim poziomie. Jak widać, w zaproponowanym systemie przesyłany rozkaz SCPI zawiera kolejne adresy wskazujące drogę jego przesyłu na kolejnych poziomach sieci. Na każdym poziomie początek rozkazu (rozkaz "DEV<n>") jest traktowany jako "adres" wskazujący, gdzie należy wysłać pozostałą część (za pierwszym znakiem ':') rozkazu.

5. Wykorzystanie języka SCPI do sterowania siecią

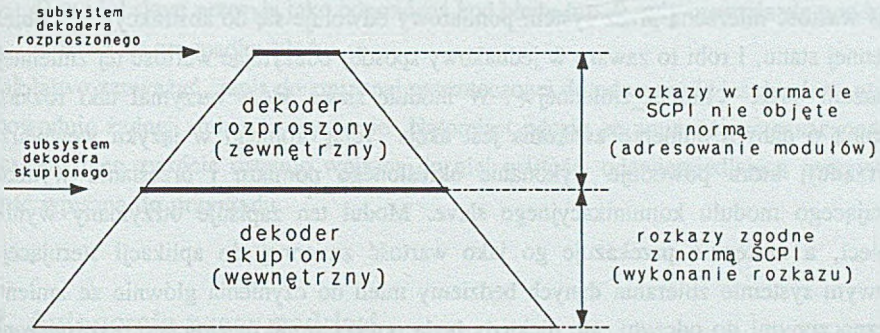
Wprowadzony w 1990 roku język SCPI [8] został powszechnie zaakceptowany przez producentów aparatury pomiarowo-kontrolnej (między innymi firmy Hewlett Packard, National Instruments, Fluke, Philips, Siemens) i stanowi ukoronowanie programowego rozwoju normy IEEE-488.2. SCPI stanowi uniwersalny język programowania aparatury pomiarowo-kontrolnej i definitywnie kończy okres, w którym, wobec braku powszechnie przyjętego standardu, występowało wiele różnych języków programowania przyrządów pomiarowych. Przed ogłoszeniem normy SCPI powszechna była sytuacja, że te same funkcjonalnie przyrządy pochodzące od różnych producentów programowało się w różny sposób i za pomocą różnych rozkazów. Język SCPI umożliwia jednolite i wygodne sterowanie systemem ATE (Automatic Test Equipment), w którym występuje kontroler jako urządzenie sterujące pracą systemu oraz urządzenia wykonujące polecenia kontrolera (przyrządy pomiarowo-kontrolne). Wybór języka SCPI dla programowania operacji projektowanej sieci polowej podyktowany jest hierarchiczną, wielopoziomową strukturą rozkazów, która dokładnie odpowiada strukturze fizycznej i logicznej naszej sieci.

Typowy rozkaz SCPI składa się z podrozkazów oddzielonych dwukropkami i pola argumentów:

":DEV1:DEV2:COMM1:COMM2 <arguments>"

Dwukropek reprezentuje przejście pomiędzy dwoma kolejnymi poziomami hierarchii w dekoderze rozkazów SCPI. Lewy skrajny dwukropek informuje, że znajdujące się za nim polecenie jest rozkazem na najwyższym poziomie hierarchii (tzw. komenda subsystemu). Dekoder SCPI zawsze rozpoczyna analizę rozkazu wychodząc od komendy subsystemu, a potem przechodzi poprzez kolejne poziomy aż do polecenia znajdującego się na poziomie najniższym.

Projektowaną sieć polową, z punktu widzenia dekodowania poleceń, będziemy traktować tak jak urządzenie pomiarowe, a ściślej mówiąc, jego hierachiczny dekoder. Nasza hierarchiczna sieć pracuje bowiem jak rozproszony dekoder rozkazów języka SCPI, którego poszczególne poziomy można utożsamiać z węzłami sieci przenoszącymi polecenia pomiędzy kolejnymi poziomami hierarchii (rys.4).



Rys. 4. Organizacja hierarchicznego dekodera rozkazów w wielopoziomowej sieci polowej

Fig. 4. Hierarchical command decoder in the multilevel fieldbus under project

Sterowanie pracą dekodera rozproszonego odbywa się za pośrednictwem przypisania adresów urządzeniom komunikacyjnym systemu. Każdy rozkaz SCPI musi zaczynać się od części określającej adres urządzenia (rozkazy typu "DEV<n>", gdzie n jest adresem modułu podrzędnego), dla którego dany rozkaz jest przeznaczony. Jest to adres złożony, zawierający adresy kolejnych modułów komunikacyjnych, przez które przechodzi komunikat zanim dotrze do urządzenia końcowego. Ostatnim adresem w tym ciągu będzie adres modułu slave końcowy. Za częścią adresową rozkazu następuje właściwe polecenie określające operację, którą powinien wykonać przyrząd pomiarowo-kontrolny (rozkazy typu "COMM<n>", gdzie n określa numer rozkazu).

Można tu mówić o dekoderze zewnętrznym (rozproszonym) oraz o dekoderze wewnętrznym (skupionym). Dekoder zewnętrzny zaimplementowany jest w modułach komunikacyjnych i sterowany adresami (rozkazy "DEV<n>"). Dekoder wewnętrzny zaimplementowany jest w modułach typu "slave końcowy" i sterowany rozkazami COMM<n>.

6. Zmienne stanu

Proponowany system rozproszonych przyrządów pomiarowych wymaga ujednolicenia sposobów dostępu do mierzonych wartości w poszczególnych przyrządach pomiarowych. Rozwiązaniem tego problemu może być mechanizm *zmiennej stanu*. Zmienna stanu jest obiektem, z którym związana jest wartość oraz pewna procedura. Mechanizm ten zaimplementowany w modułach typu slave pozwala "zasłonić" przed aplikacją rzeczywisty przyrząd pomiarowy z jego językiem komend. Oprogramowanie aplikacyjne chcąc odczytać jakąś wartość mierzoną przez system pomiarowy odwołuje się do abstrakcyjnego obiektu - zmiennej stanu, i robi to zawsze w jednakowy sposób, odczytując wartość tej zmiennej np. rozkazem VAR <numer_zmiennej>. W module slave, który otrzymał taki rozkaz, ze zmienną o danym numerze związana jest akcja (ciąg komend w języku wewnętrznym przyrządu), która powoduje wykonanie określonego pomiaru i przesłanie wyniku do czekającego modułu komunikacyjnego slave. Moduł ten zapisuje otrzymany wynik do pamięci, a następnie przekazuje go jako wartość zmiennej do aplikacji sterującej. W typowym systemie zbierania danych będziemy mieli do czynienia głównie ze zmiennymi przeznaczonymi do odczytu, ale niekiedy będą potrzebne także zmienne przeznaczone do zapisu. Jako przykład można tutaj podać zmienne stanu reprezentujące zakres pomiarowy przyrządu. Podobnie jak przy zmiennych do odczytu, aplikacja sterująca odwołuje się do abstrakcyjnego obiektu wpisujac tam wartość (np. zakres pomiarowy), natomiast moduł slave, w którym ta zmienna jest zaimplementowana, realizuje akcję związaną z tą zmienną wysyłając odpowiedni ciąg komend w języku wewnętrznym przyrządu. Jak łatwo zauważyć, oba typy zmiennych charakteryzują się ujednoliconym, zgodnym z normą SCPI sposobem dostępu do wartości od strony aplikacji:

- odczyt zmiennej: VAR? 5
- zapis zmiennej: VAR 6, 3.21

Inna jest jednak reakcja modułu slave na wymienione rozkazy:

- odczyt zmiennej: VAR? 5

a) moduł slave określa numer zmiennej (5);

b) realizowana jest akcja związana z tą zmienną: do przyrządu pomiarowego wysyłany jest ciąg rozkazów w języku wewnętrznym tego przyrządu, który powinien wyzwolić pomiar określonej wartości, a następnie z przyrządu odczytywana jest wartość danej zmiennej;

c) dane otrzymane z przyrządu pomiarowego (zazwyczaj jako ciąg znaków ASCII) zostaną wpisane do pola z wartością zmiennej;

d) ten sam ciąg znaków przekazywany jest jako wartość zmiennej w ramce odpowiedzi;

- zapis zmiennej: VAR 6, 3.21

a) moduł slave określa numer zmiennej (6);

b) ciąg znaków przekazany przez aplikację jako wartość (3.21) jest wpisywany do pola z wartością zmiennej;

c) moduł slave realizuje akcję związaną z tą zmienną (6): do przyrządu wysyłany jest ciąg rozkazów w języku wewnętrznym przyrządu oraz wartość zmiennej (3.21), co w naszym przykładzie powoduje ustawienie zakresu pomiarowego;

d) moduł slave przesyła jako odpowiedź kod błędu (np. 0, gdy operacja się powiodła, <1,n> w przypadku błędu).

Jak łatwo zauważyć, zapis do zmiennej przeznaczonej do odczytu nie jest błędem, ale też nie powoduje żadnej zmiany w systemie. Natomiast odczyt ze zmiennej przeznaczonej do zapisu powinien zwrócić ostatnio wpisaną do niej wartość, odzwierciedlającą rzeczywistą wartość wpisaną do przyrządu.

7. Autonomia pracy podsieci

Przyjęty algorytm działania sieci zakłada, że wszystkie elementy sieci są bierne, a więc oczekują na rozkaz pochodzący od układu nadrzędnego, dekodują go i wykonują (moduły slave) albo (w przypadku modułów master) po przełączeniu się na sieć niższego poziomu (podsieć) przekazują do wykonania jednemu z modułów slave na tej podsieci. W drugim przypadku układ oczekuje na odpowiedź od slave'a, a następnie przełącza się z powrotem na sieć wyższego poziomu. Wynik wykonania rozkazu (odpowiedź) jest odsyłany do układu nadrzędnego. Jak widać, po wydaniu polecenia przez układ nadrzędny (w stosunku do całej sieci) kolejne układy "tworzą ścieżkę", przez którą jest transmitowany rozkaz, a następnie w odwrotnym kierunku wraca odpowiedź. W systemie tym zakładamy, że układy są zawsze gotowe do przyjęcia rozkazu pochodzącego od układu na poziomie wyższym (np. każdy slave zawsze jest gotowy do przyjęcia rozkazu od swojego mastera, a także każdy master jest

gotowy na przyjęcie rozkazu od slave'a pośredniczącego pomiędzy nim a siecią wyższego poziomu).

W wielu systemach zbierania danych i sterowania obiektami przemysłowymi celowe wydaje się wyodrębnienie niezależnych podsieci do obsługi pewnych fragmentów obiektu przemysłowego. W przedstawionym projekcie takie wyodrębnienie jest możliwe, ponieważ założyliśmy zhierarchizowaną, wielopoziomową strukturę sieci. Autonomia pracy podsieci umożliwi rozproszoną realizację zbierania danych i sterowania.

W proponowanym rozwiązaniu elementem sterujacym taką wydzieloną podsiecią jest moduł master. Zawarty w module master program zbierania danych z "podległych" modułów typu slave może działać cyklicznie (transakcje periodyczne) lub na zasadzie sterowania zdarzeniami (transakcje aperiodyczne, np. polecenia z modułu master na wyższym poziomie). Moduł master może następnie te dane udostępniać swoim modułom nadrzędnym wykorzystując przedstawiony wcześniej mechanizm zmiennych stanu. Należy w tym miejscu wspomnieć o pewnym ubocznym efekcie, który ogromnie uprości tworzenie aplikacji sterujacych.

W systemach zbierania danych możemy się spotkać z przyrządami realizującymi pomiar w różnym czasie. Różnice te mogą wynikać z budowy przyrządu lub z algorytmu pomiaru danej wartości. Przykładowo pomiar napięcia jest realizowany stosunkowo szybko, natomiast pomiar częstotliwości wymaga zdecydowanie dłuższego czasu. Z drugiej strony chcielibyśmy, aby system pomiarowy "ukrywał" przed nami te różnice. Jak łatwo zauważyć, opisany w poprzednim punkcie mechanizm zmiennych stanu częściowo to umożliwia. W przypadku autonomicznej pracy wydzielonych podsieci pojawia się dodatkowa możliwość. Moduł master, jako układ sterujący podsiecią, realizuje cykl pomiarowy i znając czasy realizacji pomiaru poszczególnych zmiennych jest w stanie się do tego dostosować. Można to zrealizować poprzez proste rozszerzenie możliwych trybów dostępu do zmiennej w modułach slave:

- odczyt zmiennej (natychmiastowy z pamięci modułu slave) VAR? 5
- wyzwolenie akcji związanej ze zmienną VAR 5
- zapis wartości zmiennej VAR 6, 3.21

Rozbudowując opisane mechanizmy o możliwość udostępniania wartości zmiennych także przez moduły master otrzymujemy system, w którym każda z podsieci ma możliwość autonomicznej pracy zgodnie ze swoim programem zapisanym w module master. Moduł ten cyklicznie przepytuje układy slave na swojej podsieci przechowując odczytane wartości zmiennych stanu w swojej pamięci. Z jednej strony, na podstawie tych zmiennych może on sterować lokalnymi układami typu slave (odpowiednio szybko reagować na zdarzenia lokalne), a z drugiej, modułom nadrzędnym może udostępniać szereg zmiennych stanu, które mają istotne znaczenie w systemie nie tylko lokalnym. Dodatkowo, jak łatwo zauważyć,

zmienne te dostępne są w sposób ciągły z zastrzeżeniem, że ich wartość jest aktualna z dokładnością do okresu pomiarowego dla danej zmiennej wynikającego z programu akwizycji danych na lokalnej podsieci danego modułu master. Opisany mechanizm umożliwiający odczyt wartości zmiennych pośrednio z wykorzystaniem pamięci modułu master pozwala nadal odwoływać się bezpośrednio do modułów slave i ich zmiennych.

8. Podsumowanie

Przedstawiona hierarchiczna sieć polowa odpowiada funkcjonalnie rozwiązaniom standardowym. Oryginalne rozwiązania zaproponowane w projekcie obejmują:

- potraktowanie sieci jako przyrządu z dekoderem SCPI i wykorzystanie języka SCPI do sterowania przepływem komunikatów pomiędzy poziomami sieci (adresacja modułów komunikacyjnych),
- zapewnienie ścisłego związku pomiędzy hierarchiczną architekturą sieci a strukturą języka sterującego przepływem komunikatów w sieci,
- wykorzystanie w charakterze układów wykonawczych (dołączanych do sieci) dowolnych urządzeń wyposażonych w łącze RS-232.

Za charakterystyczne cechy projektowanej sieci można natomiast uznać:

- możliwość autonomicznej pracy podsieci,
- ujednolicenie dostępu do danych (zmienne stanu),
- jednorodne rozwiązania sprzętowe.

Ważne jest też, że technologia sieci nie jest skomplikowana. Wszystkie elementy można wykonać lub nabyć w kraju (złącza, moduły i przewody) uzyskując rozwiązanie znacznie tańsze od standardowych sieci polowych. Aktualnie prowadzone są prace nad modelem przedstawionej sieci polowej, których wyniki zamierzamy opublikować po uruchomieniu i przetestowaniu sieci.

LITERATURA

[1] Chatha A.: The foundation for field control systems. Control Engineering, May 1994.

[2] Johnson D.: Bus Systems Characteristics. Control Engineering, Dec. 1995.

[3] Jordan J.R.: Serial Networked Field Instrumentation. John Wiley & Sons, 1995.

[4] Lasher R.J.: Fielbus advancement and their implications. Control Engineering, July 1994.

[5] Mielczarek W.: Szeregowe interfejsy cyfrowe. Helion, Gliwice 1993.

[6] Pierson L.: Broader Fieldbus standards will improve system functionality. Control Engineering, Nov. 1994.

[7] Pimentel J.R.: Communication architectures for fieldbus network. Control Engineering, Oct. 1989.

[8] IEEE Std. 488.2-1987 "IEEE Standard Codes, Formats, Protocols, and Common Commands" i jej rozszerzenie SCPI (Standard Commands for Programmable Instruments) version 1993.

Recenzent: Doc dr hab. Adam Mrózek

Wpłynęło do Redakcji 19 marca 1997 r.

Abstract

The hierarchical fildbus presented in the paper performs the same functions as standard fieldbuses. The unconventional ideas proposed in the project are as follows:

1. The fieldbus is taken as a distributed programmable instrument with SCPI (Standard Commands for Programmable Instruments) decoder. The SCPI commands were applied to control meassages flow through the network nodes connecting levels in the hierarchical structure. SCPI commands usage to control messages flow seems to be a good choice because of close relationship between hierarchical architecture of the fielbus and hierarchical structure of SCPI language.

2. The terminal devices connected to the network can be any with RS-232 interface.

The characteristic features of the hierarchical fieldbus are:

- autonomus work of any subnet,
- uniformed access to any variable,
- uniformed communication hardware in network nodes and network terminals.

We belive also that the technology of our network wil not be complicated. All of the network hardware can be made in Poland and the whole solution should be much cheaper than any standard fieldbus.