

Wojciech MIELCZAREK, Ryszard PAWŁOWSKI  
Politechnika Śląska, Instytut Informatyki

## PODŁĄCZENIE SIECI POŁOWEJ OPARTEJ NA PROTOKOLE ADVANTECH DO SIECI HIERARCHICZNEJ STEROWANEJ ROZKAZAMI JĘZYKA SCPI

**Streszczenie.** W publikacji przedstawiono trzy możliwości podłączenia sieci polowej opartej na protokole Advantech i łącza RS-485 do sieci hierarchicznej opartej na protokole Modbus w warstwie łącza danych i rozkazach języka SCPI w warstwach sieciowej i aplikacji:

- podłączenie, w którym surowe komunikaty sieci polowej przenoszone są bezpośrednio na polu argumentów ramek Modbus,
- wykorzystanie mechanizmu tunelowania protokołów,
- wykorzystanie mechanizmu zmiennych stanu hierarchicznej sieci polowej.

## CONNECTING A FIELBUS BASED ON THE ADVANTECH PROTOCOL TO THE HIERARCHICAL NETWORK CONTROLLED BY SCPI COMMANDS

**Summary.** Three ways of connecting a fieldbus based on the Advantech protocol and RS-485 to the hierarchical network controlled by the SCPI commands (Network and Application Layers) and the Modbus protocol (Data Link Layer) are presented in the paper:

- the connection, in which unconverted messages are transferred directly by a Modbus frame data field,
- the connection using the protocol tunneling mechanism,
- the connection using the hierarchical network state variables mechanism.

## 1. Wprowadzenie

Hierarchiczna sieć polowa sterowana rozkazami języka SCPI [1] została opracowana w Instytucie Informatyki Politechniki Śląskiej w wyniku realizacji projektu badawczego KBN 302/T11/97/12. Najważniejsze cechy wymienionej sieci są następujące:

- hierarchiczna, wielopoziomowa struktura,
- topologia magistralowa oparta na łączu RS-485,
- routing komunikatów przez węzły sieci sterowany rozkazami języka SCPI (Standard Commands for Programmable Instruments) [2] przesyłanymi w ramach protokołu Modbus ASCII [7],
- autonomia pracy podsieci na poszczególnych poziomach hierarchii,
- możliwość dostępu do dowolnego urządzenia w sieci z kontrolera centralnego znajdującego się na najwyższym poziomie hierarchii,
- możliwość podłączenia urządzeń wyposażonych w interfejsy RS-232, RS-423 lub RS-422, jak również całych sieci opartych na łączu RS-485 i dowolnym protokole znakowym.

Niniejsza publikacja przedstawia trzy możliwości podłączenia sieci polowej opartej na protokole Advantech [3] i łączu RS-485 do sieci hierarchicznej sterowanej rozkazami języka SCPI:

- podłączenie, w którym surowe komunikaty sieci polowej przesyłane są bezpośrednio na polu argumentów ramek Modbus,
- wykorzystanie mechanizmu tunelowania protokołów,
- wykorzystanie mechanizmu zmiennych stanu hierarchicznej sieci polowej.

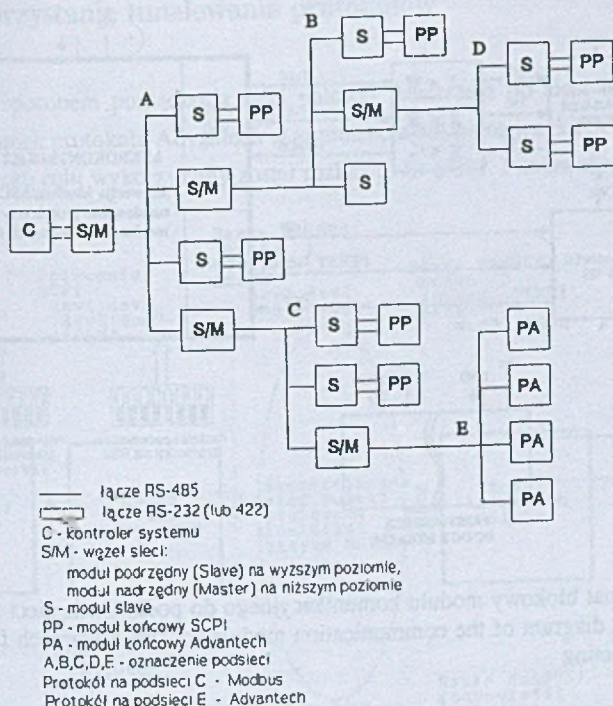
Należy podkreślić, że chociaż problem podłączenia sieci polowej do sieci hierarchicznej prezentowany jest na konkretnym przykładzie, to omówione tu rozwiązania są uniwersalne i z niewielkimi tylko zmianami mogą być zastosowane do podłączenia dowolnej sieci polowej wykorzystującej łącze RS-485 i protokół znakowy.

Na rys.1 przedstawiono strukturę sieci hierarchicznej z podłączoną siecią polową Advantech.

## 2. Moduł komunikacyjny pośredniczący w podłączeniu sieci Advantech

Podłączenie sieci Advantech do sieci hierarchicznej wykonano za pośrednictwem modułu komunikacyjnego, który na sieci hierarchicznej jest stacją slave, a na sieci Advantech stacją master (dalej moduł ten nazywamy Advantech Slave). Moduł wyposażony jest w dwa





Rys. 1. Struktura sieci hierarchicznej sterowanej rozkazami SCPI

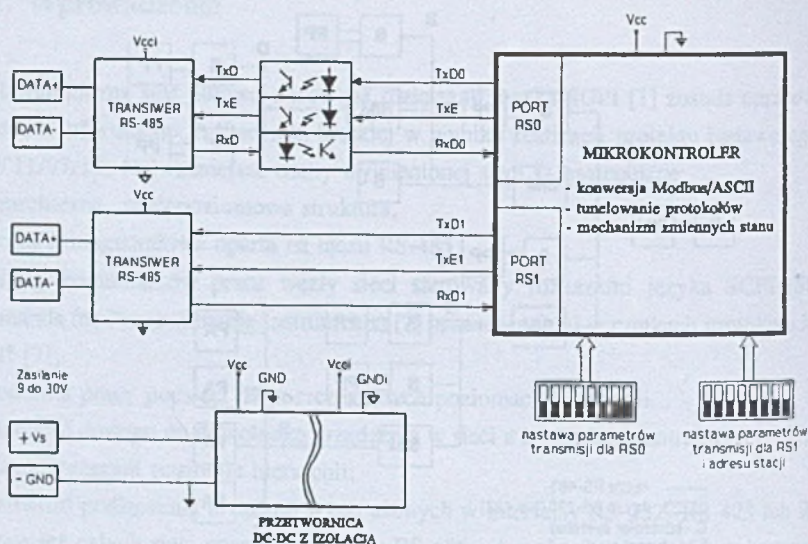
Fig. 1. The hierarchical network controlled by the SCPI commands structure

separowane galwanicznie porty RS-485: RS1 dla podłączenia sieci hierarchicznej oraz RS0 do podłączenia sieci Advantech. Schemat blokowy modułu przedstawiono na rys.2. Wszystkie trzy przedstawione dalej sposoby podłączenia sieci polowej korzystają z identycznych od strony sprzętowej modułów pośredniczących, różniących się tylko oprogramowaniem.

### 3. Podłączenie surowe

Nazwa "podłączenie surowe" określa przypadek, w którym oprogramowanie modułu pośredniczącego ogranicza się do:

- wydzielenia rozkazu Advantech z pola argumentów polecenia Modbus sieci hierarchicznej i wprowadzenia go do sieci Advantech,
- odebrania odpowiedzi z sieci Advantech i wprowadzenia jej na pole danych odpowiedzi Modbus.



Rys. 2. Schemat blokowy modułu komunikacyjnego do podłączenia sieci Advantech  
 Fig. 2. Block diagram of the communication module for the Advantech fieldbus connecting

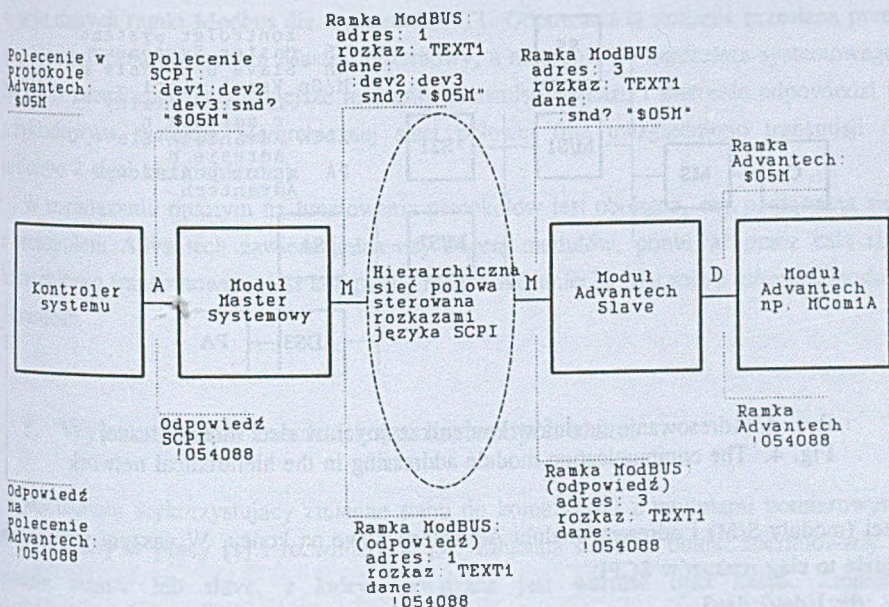
Przy wprowadzaniu rozkazu do sieci Advantech moduł dodaje znak CR (0x0D), który jest znacznikiem końca ramki protokołu Advantech, a przy odbiorze odpowiedzi znak ten usuwa. Odbiór odpowiedzi ograniczony jest czasowo przez timeout, który domyślnie ustawiony jest na 1 s. Wartość timeoutu można zmienić odpowiednim rozkazem protokołu Modbus. W przypadku podłączenia surowego moduł pośredniczący jest w zasadzie konwerterem protokołu Modbus na ASCII [6], gdzie przez ASCII rozumiemy tutaj ciąg znaków ASCII wprowadzany do sieci polowej lub odbierany z sieci polowej. W pewnym sensie stanowi on również przedłużenie portu szeregowego kontrolera systemu, który za jego pośrednictwem steruje podłączaną siecią.

Przedstawiony sposób podłączenia podsieci angażuje kontroler sieci hierarchicznej do bezpośredniego sterowania siecią polową i dlatego przewidziano go przede wszystkim do celów testowania i oceny sprawności połączeń w sieci.



#### 4. Wykorzystanie tunelowania protokołów

Kolejnym sposobem podłączenia sieci polowej Advantech do sieci hierarchicznej jest tunelowanie ramek protokołu Advantech w argumencie polecenia języka SCPI przedstawione na rys.3. Do tego celu wykorzystany został rozkaz SND SCPI. Ponieważ wszystkie rozkazy

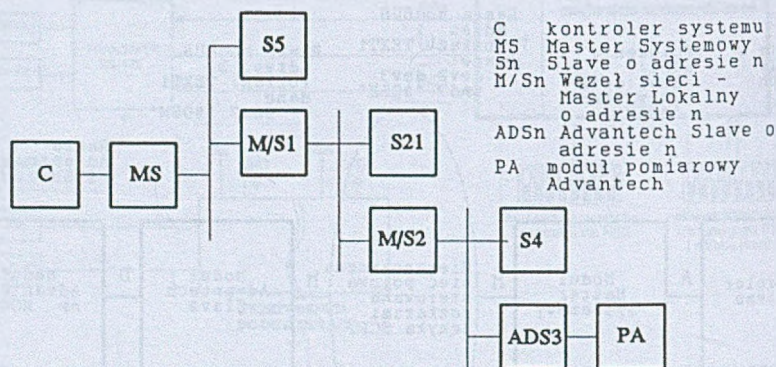


Rys. 3. Tunelowanie ramek protokołu Advantech w argumencie polecenia SCPI COM  
Fig. 3. Advantech frames tunneling within the SCPI command argument

protokołu Advantech (z wyjątkiem rzadko wykorzystywanego rozkazu globalnego wyzwolenia pomiaru "####") zwrótnie przekazują odpowiedź, aby wymusić transmisję odpowiedzi, należy zastosować formę zapytania (query command) w rozkazie SND: `":snd?"`. Ramki protokołu Advantech są ramkami znakowymi, w których wykorzystuje się znaki ASCII: dużych liter, cyfr dziesiętnych i kilka znaków specjalnych ('%', '\$', '#', '@', '!', '>' i '?'). Treść takiej ramki można przesłać jako parametr typu napis (quoted string) polecenia SCPI. Jedyną dodatkową operacją na argumencie polecenia SND jest konieczność "doklejenia" znaku CR na końcu ramki protokołu Advantech przez oprogramowanie modułu pośredniczącego (Advantech Slave). W przypadku ramek odpowiedzi znak ten należy usunąć przed odesłaniem odpowiedzi SCPI. Przesłanie przykładowego rozkazu Advantech przez hierarchiczną sieć



połową ilustrują rys.3 i 4. Rozkaz Advantech (na przykład: "\$05M" - polecenie do modułu o adresie 05, aby podał swój typ) przekazany zostaje kontrolerowi systemu. Kontroler jako dodatkową informację musi posiadać ciąg adresów dla protokołu Modbus, które składają się na "ścieżkę transmisyjną" wewnątrz hierarchicznej sieci polowej, pozwalającą na dotarcie do modułu Advantech Slave. Przykładową sieć hierarchiczną, przez którą będzie transmitowany rozkaz, przedstawiono na rys.4. Ścieżkę tworzą kolejne rozkazy DEV z adresami węzłów



Rys. 4. Adresowanie modułów komunikacyjnych w sieci hierarchicznej  
Fig. 4. The communication module addressing in the hierarchical network

sieci (moduły S/M) i adresem modułu Advantech Slave na końcu. W naszym przykładzie będzie to ciąg rozkazów SCPI:

```
:dev1:dev2:dev3
```

Do podanego ciągu dołączony zostanie w module kontrolera systemowego rozkaz SND oraz w charakterze jego argumentu rozkaz Advantech. W wyniku połączenia tych elementów powstaje rozkaz SCPI, który zostanie przekazany modułowi Master Systemowy przez łącze RS-232, przy wykorzystaniu protokołu znakowego ASCII [4]:

```
:dev1:dev2:dev3:snd? "$05M"
```

Rozkaz ten zostaje przekazany do modułu Mastera Systemowego i zgodnie z zasadami opisanymi w pracy [1] zostaje wprowadzony do sieci. Do modułu o adresie 1 na pierwszym poziomie hierarchii zostaje wysłana ramka w protokole Modbus z rozkazem TEXT1 (polecenie przenoszące rozkaz SCPI typu query) i argumentem w postaci polecenia SCPI:

```
:dev2:dev3:snd? "$05M"
```

Kolejne transmisje doprowadzają na 3 poziom hierarchii sieci, gdzie moduł M/S2 przesyła do stacji o adresie 3 ramkę Modbus z rozkazem TEXT1 i argumentem w postaci tekstu polecenia SCPI:

```
:snd? "$05M"
```

Stacja o adresie 3 jest modulem typu Advantech Slave. Po otrzymaniu ramki dekodowany jest rozkaz Modbus, a następnie polecenie SCPI. Realizując polecenie SND oprogramowanie modułu wyodrębnia argument ("\$05M"), usuwa z niego znaki cudzysłowu (""), a następnie, po kontroli poprawności syntaktycznej ramki (dla protokołu Advantech), wysyła ją na sieć podrzędną (sieć z protokołem Advantech). Realizując transakcję Advantech należy odebrać odpowiedź od modułu slave (na rys.4 modul PA), a następnie otrzymaną odpowiedź odesłać w polu danych ramki Modbus dla rozkazu TEXT1. Odpowiedź ta zostanie przesłana przez sieć hierarchiczną do modułu Master Systemowy, a następnie do kontrolera systemowego. Na rys.3 zaznaczono najważniejsze momenty transmisji rozkazu i zwrotnie odpowiedzi w przykładowym systemie hierarchicznej sieci polowej (nie uwzględniono transmisji na poziomie 2 sieci hierarchicznej).

W rozwiązaniu opartym na tunelowaniu protokołów jest obojętne, czy podłączana sieć z protokołem Advantech zawiera jeden czy więcej modułów, ponieważ przez całą sieć hierarchiczną transmitowany jest kompletny rozkaz Advantech, włącznie z adresem modułu Advantech.

## 5. Wykorzystanie mechanizmu zmiennych stanu

Mechanizm wykorzystujący zmienne stanu do komunikacji z modułami pomiarowymi został opisany w pracy [1] i rozwinęty w [5]. Zmienna stanu to obiekt zdefiniowany w module master lub slave, z którym związana jest wartość oraz akcja. Zmienne identyfikowane są przez numery. Dostęp do zmiennych odbywa się przy wykorzystaniu rozkazu SCPI VAR. Dla każdej zmiennej dostępne są trzy operacje:

- wyzwolenie pomiaru zmiennej

:var5

- zapis wartości do zmiennej

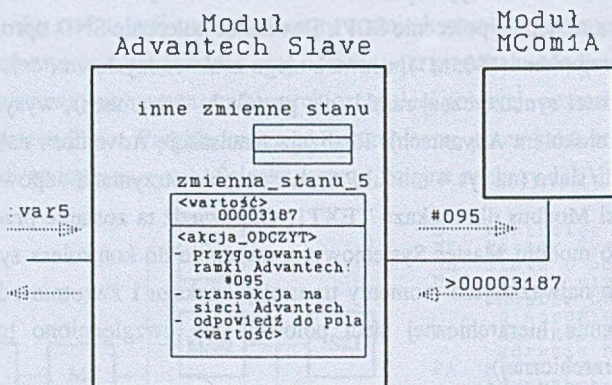
:var5 129

- odczyt wartości zmiennej

:var5?

Rozważmy realizację tych trzech operacji dostępu do zmiennej na przykładach. Moduł Advantech Slave otrzymuje ramkę w protokole Modbus (adresowaną do siebie) z rozkazem TEXT1 i argumentem w postaci rozkazu SCPI VAR. Po zdekodowaniu rozkazu Modbus i polecenia SCPI wyodrębniony zostaje numer zmiennej. Następnie sprawdzana jest obecność znaku zapytania ('?'), co oznacza polecenie odczytu wartości zmiennej lub obecność argumentu i związana z jego wystąpieniem operację zapisu wartości do zmiennej.





Rys. 5. Wyzwolenie pomiaru zmiennej

Fig. 5. Triggering a state variable measurement

Dla operacji wyzwolenia pomiaru (rys.5), po otrzymaniu polecenia:

```
:var5
```

wyszukiwany jest w bazie danych ze zmiennymi opis zmiennej (w naszym przykładzie zmiennej numer 5). Z opisu akcji ODCZYT wynika, że należy przygotować ramkę z rozkazem Advantech (#095 - odczyt wyniku pomiaru z kanału 5 w module o adresie 09) i zrealizować transakcję Advantech na sieci podrzędnej. Po zakończeniu transakcji odpowiedź modułu (>00003187) należy wpisać do pola z wartością zmiennej (wcześniej należy usunąć wszelkie znaki charakterystyczne dla protokołu Advantech). W naszym przykładzie nową wartością zmiennej numer 5 będzie 00003187. Ponieważ polecenie wyzwolenia pomiaru jest poleceniem typu command języka SCPI, nie jest odsyłana żadna ramka odpowiedzi SCPI. Oczywiście odsyłana jest ramka potwierdzenia pozytywnego w protokole Modbus.

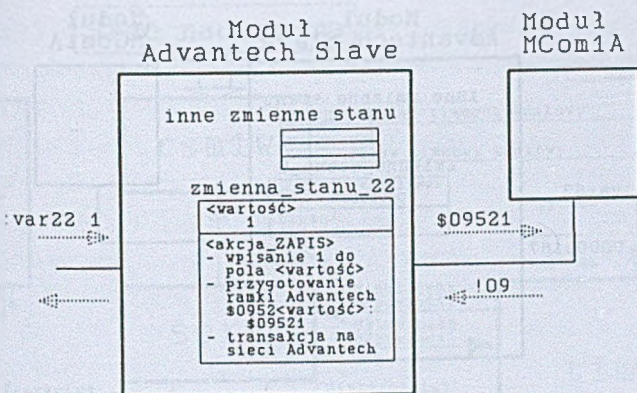
Operację zapisu wartości do zmiennej przedstawiono na rys.6. Moduł Advantech Slave po otrzymaniu polecenia SCPI:

```
:var22 1
```

i wyodrębnieniu numeru zmiennej (22) oraz argumentu (1) wyszukuje w bazie danych ze zmiennymi opis zmiennej numer 22. Po znalezieniu wyzwalana jest akcja ZAPIS związana z tą zmienną. W pierwszym kroku należy argument polecenia VAR wpisać do pola WARTOŚĆ zmiennej o numerze 22. Następnie przygotowana zostaje ramka polecenia Advantech. W naszym przykładzie ramka z rozkazem obejmuje zarówno część stałą (\$0952), jak i część zmienną, zależną od wartości zmiennej numer 22. W wyniku złożenia tych dwóch elementów powstaje rozkaz:

```
$09521
```





Rys. 6. Zapis wartości do zmiennej  
Fig. 6. A state variable write operation

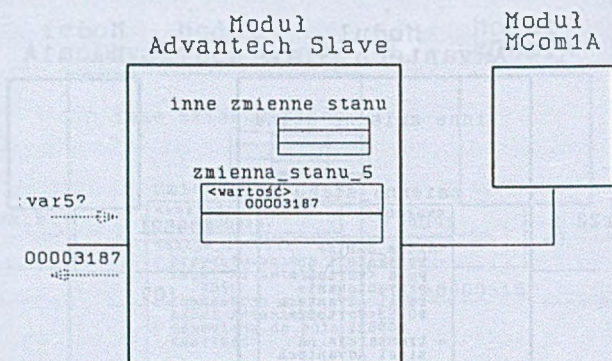
Następnie zrealizowana zostaje transakcja, w wyniku której oprogramowanie modułu Advantech Slave otrzymuje od stacji slave na sieci Advantech potwierdzenie pozytywne wykonanej operacji (ramka !09). To pozwala na odesłanie pozytywnego potwierdzenia w protokole MODBUS do stacji nadrzędnej. Podobnie jak w przypadku wyzwolenia pomiaru zmiennej nie jest odsyłana odpowiedź SCPI.

Ostatnią operacją jest odczyt (rys.7) wartości zmiennej realizowany rozkazem `:var5?`

W tym wypadku moduł Advantech Slave nie musi kontaktować się z modulem slave na sieci Advantech, bowiem wartość zmiennej może odczytać ze swojej pamięci (pole WARTOŚĆ określonej zmiennej). Po znalezieniu określonej zmiennej w bazie danych zostaje odesłana jej wartość jako odpowiedź SCPI:

003187

Problemem, który pojawia się w przypadku realizacji dostępu do modułów na sieci Advantech za pomocą mechanizmu zmiennych stanu, jest sposób powiązania określonej zmiennej (o danym numerze) z modulem na sieci Advantech i konkretnym rozkazem dla tego modułu. Można tutaj rozważać powiązanie ściśle statyczne (określone w konfiguracji modułu Advantech Slave), jak i powiązanie dynamiczne. To drugie może się wiązać z koniecznością detekcji adresów i rodzajów modułów na sieci Advantech przez moduł Advantech Slave i dostosowaniem jego konfiguracji do zastanej konfiguracji sieci podrzędnej.



Rys. 7. Odczyt wartości zmiennej

Fig. 7. A state variable read operation

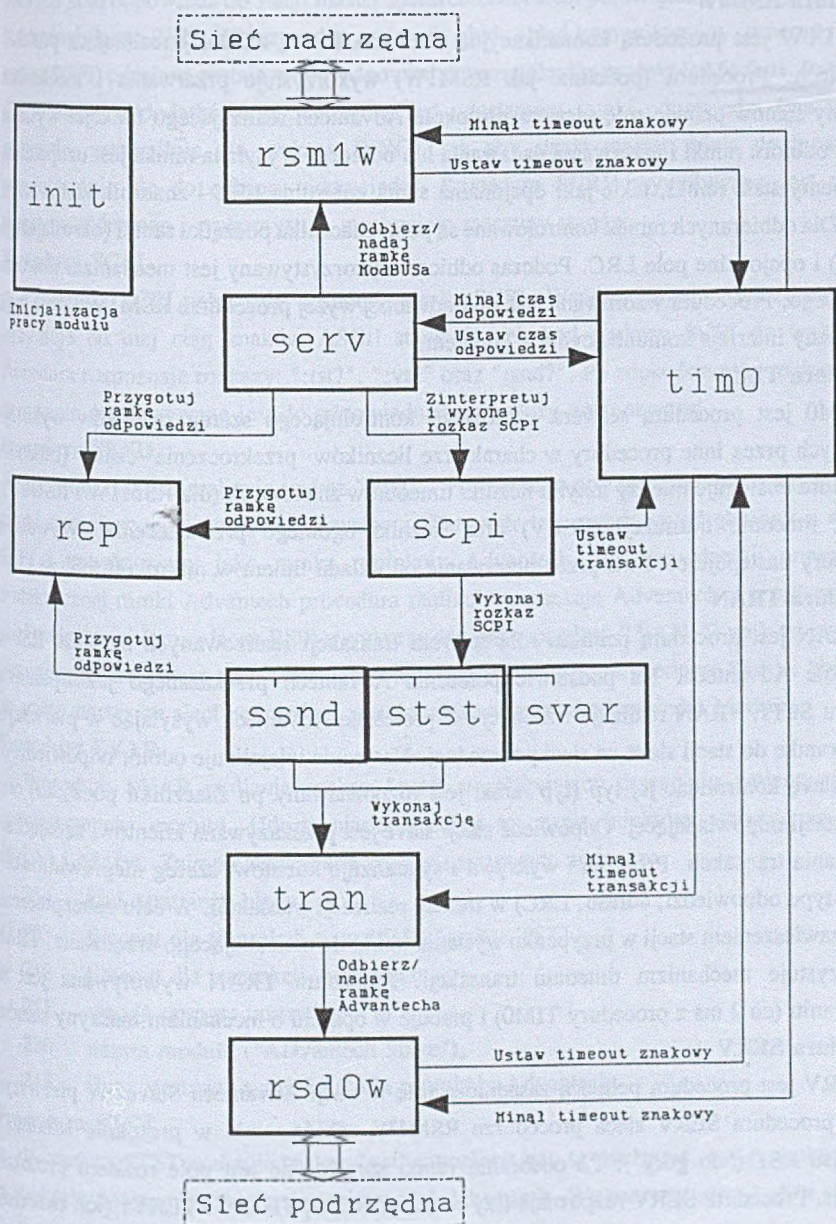
## 6. Oprogramowanie stacji Advantech Slave w wersji wykorzystującej tunelowanie

Tworząc oprogramowanie modułu Advantech Slave w wersji wykorzystującej tunelowanie przyjęto koncepcję podziału oprogramowania na niezależne moduły komunikujące się ze sobą na zasadzie klient-serwer. Na rys.8 przedstawiono podstawowe procedury tworzące oprogramowanie modułu Advantech Slave.

### Procedura RSM1W

RSM1W jest procedurą komunikacyjną dla łącza RS1 ("do góry") realizującą protokół Modbus ASCII. Procedura ta wykorzystuje przerwanie od układu RS1 i pracuje w oparciu o mechanizm maszyny stanów. Pełni ona rolę serwera protokołu Modbus oferując funkcje: wysłania ramki, odbioru ramki oraz natychmiastowego przerwania odbioru lub nadawania. Funkcja wysłania ramki uzupełnia wysłaną informację o stałe elementy ramki Modbus, takie jak: znacznik początku ramki (':'), sumę kontrolną LRC i znaki kończące ramkę (CR i LF). Funkcja odbioru ramki z kolei analizuje pole adresowe pozwalając selektywnie odbierać ramki przeznaczone dla określonego modułu slave (odbierane są także ramki w trybie broadcast). Dla odbieranych ramek kontrolowane są poszczególne znaki (ich poprawność i liczba) oraz suma kontrolna LRC. Podczas odbioru wykorzystywany jest także mechanizm timeoutu znakowego. Ostatnia funkcja pozwala przerwać transakcję (nadawanie lub odbiór) w dowolnym momencie.





Rys. 8. Oprogramowanie modułu Advantech Slave w wersji wykorzystującej tunelowanie  
 Fig. 8. The Advantech Slave software - tunneling version



### Procedura RSD0W

RSD0W jest procedurą komunikacyjną dla łącza RS0 ("na dół") realizującą protokół Advantech. Procedura (podobnie jak RSM1W) wykorzystuje przerwania i mechanizm maszyny stanów pełniąc rolę serwera protokołu Advantech realizującego funkcje: wysłania ramki, odbioru ramki i przerwania nadawania lub odbioru. Wysyłana ramka jest uzupełniana o elementy stałe ramki, takie jak: opcjonalna suma kontrolna LRC i znacznik końca ramki (CR). Dla odbieranych ramek kontrolowane są pola: znacznika początku ramki (określającego jej typ) i opcjonalne pole LRC. Podczas odbioru wykorzystywany jest mechanizm timeoutu znakowego. Procedura wzorowana jest na omówionej wyżej procedurze RSM1W - obie mają identyczny interfejs komunikacyjny z klientem.

### Procedura TIM0

TIM0 jest procedurą serwera czasowego kontrolującego szereg liczników wykorzystywanych przez inne procedury w charakterze liczników przekroczenia czasu (timeout). Procedura obsługuje między innymi liczniki timeoutów znakowych (dla RSM1W i RSD0W), licznik timeoutu transakcji (SERV) oraz liczniki ogólnego przeznaczenia. Wywołanie procedury następuje co 1 ms przez przerwanie od układu timera w mikrokontrolerze.

### Procedura TRAN

TRAN jest procedurą pełniącą rolę serwera transakcji realizowanych na łączu RS0 w protokole Advantech. Na podstawie polecenia Advantech przekazanego jako parametr rozkazu SCPI, TRAN realizuje transakcję w protokole Advantech wysyłając w pierwszym kroku ramkę do stacji slave na sieci podrzędnej. Następnie inicjalizuje odbiór odpowiedzi od stacji slave kontrolując jej typ (typ ramki jest rozpoznawany po znaczniku początku) oraz adres stacji odpowiadającej. Odpowiedź stacji slave jest przekazywana klientowi żądającemu wykonania transakcji. Procedura wykrywa i sygnalizuje klientowi szereg nieprawidłowości (błędy typu odpowiedzi, adresu, LRC) w trakcie realizacji transakcji. W celu zabezpieczenia przed zawieszeniem stacji w przypadku wysłania ramki do nieistniejącego urządzenia, TRAN wykorzystuje mechanizm timeoutu transakcji. Procedura TRAN wywoływana jest na przerwaniu (co 2 ms z procedury TIM0) i pracuje w oparciu o mechanizm maszyny stanów.

### Procedura SERV

SERV jest procedurą pełniącą zasadniczą rolę w stacji Advantech Slave. W pierwszym kroku procedura SERV zleca procedurze RSM1W odbiór ramki w protokole Modbus z interfejsu RS1 ("do góry"). Po odebraniu ramki sprawdzane jest pole rozkazu protokołu Modbus. Procedura SERV rozpoznaje trzy rozkazy: PING, TEXT0 i TEXT1 (ich znaczenie opisane zostało w [5]). Dla rozkazu PING przygotowana zostaje ramka odpowiedzi z zawartością "ADS" (ADvantech Slave). Dla rozkazów TEXT0 i TEXT1 treścią ramki jest tekst ASCII rozkazu w języku SCPI. Procedura SERV przekazuje treść ramki do analizy i wykonania procedurze SCPI. Ramka odpowiedzi przekazana przez procedurę SCPI zostaje



odesłana jako odpowiedź do stacji master hierarchicznej sieci polowej znajdującej się na sieci nadrzędnej (łącze RS1). W przypadku wykrycia błędu (błąd komunikacyjny, timeout, błędny rozkaz SCPI) odesłana zostaje ramka odpowiedzi szczególnej w protokole Modbus. Procedura SERV realizuje dodatkowo opóźnienie przed odesłaniem ramki odpowiedzi (ma to duże znaczenie, szczególnie dla rozkazu PING), tak aby stacja master miała dość czasu na przygotowanie się do odbioru odpowiedzi. Procedura SERV wywoływana jest w pętli programu głównego i wykorzystuje mechanizm maszyny stanów.

#### Procedura SCPI

Procedura SCPI pełni rolę dekodera języka SCPI. Współpracując z procedurą SERV otrzymuje od niej ciąg znaków ASCII stanowiących kod rozkazu SCPI do wykonania. Procedura rozpoznaje rozkazy: ":tst?", ":var" oraz ":snd?". Po zdekodowaniu rozkazu SCPI sterowanie przekazywane jest do odpowiedniej procedury wykonawczej.

#### Procedura SSND

Procedura SSND realizuje rozkaz ":snd?". Polecenie to posiada jeden parametr w postaci tekstu ograniczonego znakami cudzysłowu. Parametr ten zostaje wyodrębniony z rozkazu SCPI i potraktowany jako ramka protokołu Advantech. Po sprawdzeniu poprawności syntaktycznej ramki Advantech procedura realizuje transakcję Advantech z urządzeniem na sieci podrzędnej (przez łącze RS0) z wykorzystaniem procedury TRAN. Wynik tej transakcji jest następnie kopiowany do ramki odpowiedzi i przekazywany procedurze SERV. Na koniec do stacji master na sieci nadrzędnej zostaje odesłana ramka odpowiedzi Modbus.

#### Procedura SVAR

Procedura SVAR realizuje rozkaz ":var" umożliwiający dostęp do zmiennych stanu (wewnętrznych) modułu. Udostępniane operacje to: zapis (niektóre zmienne są READ ONLY) i odczyt. Zaimplementowane zostały następujące zmienne:

- 254 kod ostatniego błędu,
- 253 timeout dla transakcji typu PING (rozkaż TST),
- 252 timeout dla transakcji Advantech,
- 251 wersja oprogramowania,
- 250 nazwa modułu ("Advantech Slave"),
- 245 flaga sterująca sumą LRC dla protokołu Advantech.

#### Procedura STST

Procedura STST realizuje rozkaż ":tst" umożliwiający sprawdzenie, czy pod określonym w rozkazie adresem znajduje się jakiś moduł Advantech. Wykorzystywany jest w tym celu rozkaż Saa2 protokołu Advantech.

#### Procedura REP

REP jest procedurą pomocniczą, która przygotowuje ramki odpowiedzi na potrzeby innych procedur (SERV, SCPI, SVAR, STST i SSND).

## 7. Podsumowanie

W publikacji przedstawiono trzy możliwości podłączenia sieci polowej opartej na protokole Advantech i łączy RS-485 do sieci hierarchicznej (rys.1) opartej na protokole Modbus w warstwie łączy i rozkazach języka SCPI w warstwach sieciowej i aplikacji:

- podłączenie, w którym surowe komunikaty sieci polowej przenoszone są bezpośrednio na polu argumentów ramek Modbus,
- wykorzystanie mechanizmu tunelowania protokołów (rys.3),
- wykorzystanie mechanizmu zmiennych stanu hierarchicznej sieci polowej (rys.5,6,7).

Rozwiązanie pierwsze jest najprostsze, ponieważ moduł pośredniczący w podłączeniu pracuje tylko jako konwerter protokołu Modbus na ASCII, gdzie ASCII reprezentuje tutaj ciąg znaków wprowadzany do lub odbierany z sieci polowej.

Rozwiązanie oparte na tunelowaniu ramek protokołu Advantech w rozkazie SND języka SCPI, który z kolei przenoszony jest jako argument polecenia Modbus, wymaga więcej przetwarzania, ale bardziej odpowiada specyfice sieci hierarchicznej jako obiektu sterowanego rozkazami języka SCPI.

W obu wymienionych przypadkach dostęp do jednostek pracujących w podłączanej sieci polowej wymaga znajomości protokołu obowiązującego w tej sieci przez kontroler sieci hierarchicznej.

Trzeci sposób, najbardziej złożony, dzięki wykorzystaniu mechanizmu zmiennych stanu umożliwia dostęp do jednostek podłączanej sieci za pośrednictwem rozkazów SCPI zapisu, odczytu oraz wyzwolenia pomiaru zmiennej. Kontroler sieci hierarchicznej nie musi znać protokołu podłączanej sieci, ponieważ zmienna stanu "ukrywa" przed nim całą specyfikę sterowania operacjami na podłączanej sieci. W tym przypadku oczywiście bardziej skomplikowane staje się oprogramowanie modułu pośredniczącego, który staje się konwerterem prostych rozkazów języka SCPI na polecenia protokołu podłączanej sieci.

## LITERATURA

1. Mielczarek W., Pawłowski R.: Hierarchiczna sieć polowa sterowana rozkazami języka SCPI. ZN Pol. Śl. s. Informatyka z. 33, Gliwice 1997.
2. Mielczarek W.: Urządzenia pomiarowe i systemy kompatybilne ze standardem SCPI. Helion, Gliwice 1999.
3. Mielczarek W., Pawłowski R.: System liczników impulsów oparty na łączy RS-485 i protokole Advantech. ZN Pol. Śl. s. Informatyka z. 32, Gliwice 1997.



4. Pawlowski R.: Stacja master hierarchicznej sieci polowej sterowanej rozkazami języka SCPI. ZN Pol. Śl. s. Informatyka z. 34, Gliwice 1998.
5. Pawlowski R.: Autonomia pracy podsieci w hierarchicznej sieci polowej sterowanej rozkazami języka SCPI. ZN Pol. Śl. s. Informatyka z. 36, Gliwice 1999.
6. Mielczarek W.: Stacja slave hierarchicznej sieci polowej sterowanej rozkazami języka SCPI. ZN Pol. Śl. s. Informatyka z. 33, Gliwice 1997.
7. Modicon: Modicon Modbus Protocol Reference Guide. PI-MBUS-300 Rev.D. Modicon Inc. 1992.

Recenzent: Dr inż. Włodzimierz Borof

Wpłynęło do Redakcji 20 kwietnia 2000 r.

#### Abstract

Three ways of connecting a fieldbus based on the Advantech protocol and RS-485 to the hierarchical network (fig.1) controlled by the SCPI commands (Network and Application Layers) and the Modbus protocol (Data Link Layer) were discussed in the paper:

- the connection, in which unconverted fieldbus messages are transferred directly by a Modbus frame data field,
- the connection using the protocol tunneling mechanism (fig.3),
- the connection using the hierarchical network state variables mechanism (fig.5,6,7).

The first solution is rather simple, because the communication module, that connects a fieldbus to the hierarchical network works only as an Modbus to ASCII converter, where ASCII represents a set of characters entered to or outputed from a fieldbus.

The solution based on Advantech frames tunneling within SCPI SND command, that in turn is transfered as an argument within Modbus frame, requires more conversion, but better corresponds to the hierarchical network, as an object controlled by the SCPI commands.

In these both cases access to the units in the connected fieldbus is possible only if the system controller knows the fieldbus protocol. The third solution, most complex, enables access to units in a fieldbus using the following SCPI operations: WRITE, READ and TRIGGER a state variable. The System Controller needn't to know the protocol in a fieldbus connected to the hierarchical network, because a state variable "hides" all details responsible for a fieldbus controlling. In this case, evidently, more complex becomes the software of the communication module, that in fact acts as a SCPI to "a fieldbus protocol" converter.