ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 34 Nr kol.1381

Andrzej KWIECIEŃ, Zdzisław MRÓWKA, Piotr GAJ
Politechnika Śląska, Instytut Informatyki

O PEWNEJ IMPLEMENTACJI INTERFEJSU SIECI TYPU FIP

Streszczenie. Artykuł przedstawia zagadnienia związane z nowymi interfejsami sieci FIP przeznaczonymi do pracy w urządzeniach obiektowych. Główny nacisk położony jest na wykorzystanie koprocesora MicroFIP i na zagadnienie zdalnej konfiguracji tego typu abonentów. W artykule zawarte są odniesienia do rzeczywistych rozwiązań.

ABOUT SOME IMPLEMENTATION OF FIP NETWORK INTERFACE

Summary. This article presents the basic description of new FIP type network interface dedicated to work at remote device. Main topic is using MicroFIP coprocessor and issue of remote configuration subscribers type like this. The real treatment links are including in this article.

1. Wstęp

Niniejsza praca traktuje o praktycznej implementacji protokołu sieci FIP (Factory Instrumentation Protocol) dla specjalnych zastosowań. Nie przedstawiono w niej dokładnego opisu protokołu, gdyż można go znaleźć między innymi w [7], [8], [9], [10]. Ograniczono się jedynie do przypomnienia podstawowych informacji związanych z modelem tej sieci.

2. Model i założenia protokołu FIP

Jako sieć terenowa (polowa) sieć FIP zdecydowanie ułatwia budowę rozproszonych systemów automatycznego sterowania przez zamianę konwencjonalnych połączeń typu „punkt-

punkt" pomiędzy zadajnikami sygnału i urządzeniami wykonawczymi a jednostką centralną, na przykład sterownika swobodnie programowalnego, dzięki stosowaniu połączeń typu „multipoint". Jest to rozwiązanie tańsze i prostsze w realizacji przy zachowaniu zalet związanych z takimi parametrami, jak czas odpowiedzi lub bezpieczeństwo poprawnej transmisji. Inną zaletą tej sieci jest ogromna łatwość w jej modyfikacji, rozbudowie i obsłudze. Może ona zapewnić przyłączenie albo połączenie w sieć elementów wykonawczych i zadajników sygnałów bez względu na ich położenie na obiekcie. Dzięki temu uzyskuje się możliwość użycia urządzeń jako bezpośrednich układów typu wejście / wyjście bez pośrednictwa sterowników przemysłowych i traktowania ich jako urządzeń zdalnych. Kolejną zaletą sieci FIP jest możliwość tworzenia systemów otwartych. Sieć ta jest rozwiązaniem dającym standardowy dostęp wszystkim producentom i wewnętrznym użytkownikom do rozległego systemu informatycznego.

Sieć FIP jest siecią terenową (field network) realizującą niższe warstwy procesów sterowania i produkcji „dyskretnej" [1]. W obszarze realizacji tych warstw rozróżnia się różne rodzaje wymian informacji:

- informacje (dane) pochodzące z procesu tworzone przez układy wejścia (inicjatory),
- rozkazy tworzone przez system, zwracane do procesu i realizowane przez układy wyjścia (układy wykonawcze),
- informacje wymieniane przez różnych abonentów sieci i przez różne systemy sterowania w kontekście zastosowanej aparatury, takiej jak:
  - przemysłowe sterowniki swobodnie programowalne,
  - klasyczne bądź „inteligentne" inicjatory i układy wykonawcze,
- informacje niezbędne dla monitorowania, sterowania i pomiarów.

Pierwsze trzy typy wymian dotyczą komunikacji poziomej w sieci, gdzie następuje podział informacji od inicjatorów do układów wykonawczych lub sterowników przemysłowych. Te rodzaje wymian mają miejsce również w trakcie aktualizacji rozproszonej bazy danych. Natomiast ostatni rodzaj wymian można określić jako „pionowy" ruch danych, służący monitorowaniu i sterowaniu procesem. Te wymiany są mniej krytyczne z punktu widzenia wymagań czasowych. Sieć FIP stara się jednocześnie zaspokoić wymagania wynikające z „pionowej" i „poziomej" wymiany danych ze szczególnym uwzględnieniem periodycznej wymiany obwarowanej czasem krytycznym i aperiodycznej wymiany danych z wyższymi warstwami oprogramowania.

System komunikacyjny sieci FIP jest podzielony na trzy warstwy, które odpowiadają modelowi OSI/ISO. Warstwami tymi są:

- warstwa fizyczna (physical layer) (1),
- warstwa łącza (data link layer) (2),
- warstwa aplikacji (aplication layer) (7).

Obowiązującym modelem tej sieci jest model „Producent – Dystrybutor – Konsument" (PDK).

„Producent" jednostki danych jest odpowiedzialny, na poziomie aplikacji, za „produkowanie" danych, które mogą być generowane periodycznie lub nie, synchroniczne lub nie, z innymi aplikacjami lub „produkcjami" danych. „Dystrybutor" danych jest natomiast odpowiedzialny za transfer danych od „Producenta" do wszystkich „Konsumentów".

„Konsumenci" danych są aplikacjami, które, będąc wykonywane, żądają danych. Używanie danych znów może być periodyczne albo nie, synchroniczne albo nie z innymi aplikacjami. Dystrybucja danych powinna gwarantować czasową charakterystykę „produkcji" danych uwzględniającą ich „konsumpcję" i dostępność zgodnie z wymaganiami „konsumentów". Model PDK stawia, w odróżnieniu od modelu „klient-serwer", na uprzywilejowanej pozycji „Konsumentów". Poza tym, funkcja dystrybucji, statycznie skonfigurowana i zaprojektowana, może być modyfikowana, spełniając czasowe wymagania „konsumentów". Sieć o modelu PDK doskonale nadaje się do zaadaptowania dla systemów realizujących wymiany poziome, jak również może być zadowalająca dla realizacji niektórych wymian pionowych.

3. Charakterystyka koprocesorów sieci FIP

Jedną z cech charakterystycznych sieci typu FIP jest możliwość podłączenia do niej abonentów stanowiących specjalizowane urządzenia nie będące swobodnie programowalnymi sterownikami. Interfejs sieciowy takiego urządzenia (jak np. termometru, pompy, zaworu itp.) nie musi obsługiwać wszystkich funkcji protokołu (np. arbitrażu magistrali). Poza tym istotne jest, aby jego koszt był minimalny. Pojawiła się zatem konieczność stworzenia kilku rodzajów koprocesorów sieciowych dedykowanych dla konkretnego typu abonentów.

W związku z faktem, iż prace związane ze sprzętem i oprogramowaniem prowadzone są w oparciu o bazę procesorów firmy CEGELEC, dlatego dalsze rozważania dotyczyć będą wyłącznie układów FULLFIP2 i MicroFIP [14].

Zadaniem wymienionych kontrolerów jest obsługa protokołu sieci FIP. W odniesieniu do siedmiowarstwowego modelu ISO/OSI sieć FIP posiada trzy warstwy tegoż modelu: warstwę fizyczną, łącza oraz warstwę aplikacji. Zarówno FULLFIP2, jak i MicroFIP realizują funkcje obsługujące poszczególne warstwy protokołu, jednak każdy z nich w różnym stopniu, w zależności od przeznaczenia.

Układ FULLFIP2 posiada najwięcej zaimplementowanych funkcji protokołu FIP; z tego względu że jest to układ mikroprogramowalny, w dużym stopniu zależą one od jego systemu operacyjnego, którym jest FIPCODE. Przez zmianę tegoż oprogramowania możemy zmieniać zachowanie koprocesora.

Z kolei koprocesor MicroFIP jest układem konfigurowalnym. Jego możliwości jako koprocesora sieciowego ściśle zależą od trybu pracy, w jaki został wprowadzony. Z kolei możliwości wyboru sprowadzają się do tego, co fizycznie zostało zaimplementowane w procesorze. Nie mamy więc takich możliwości jak w przypadku procesora FULLFIP2.

3.1. Charakterystyka układu FULLFIP2

Układ FULLFIP2 przeznaczony jest do budowy bardzo rozbudowanych interfejsów sieciowych, do pracy jako arbiter magistrali lub stacja abonencka obsługująca dużą liczbę zmiennych. Wymaga on dołączenia zewnętrznej pamięci, w której przechowywane są dane oraz mikrojądro systemu operacyjnego FIPCODE. Do pracy wymaga również zewnętrznego procesora nadrzędnego, który realizuje funkcje górnej warstwy aplikacji użytkownika.

Układ FULLFIP2:

- umożliwia nadzorowanie do 4000 identyfikatorów zmiennych,
- posiada możliwość wyboru prędkości transmisji danych po magistrali: 31.25kB/b, 1MB/s, 2.5MB/s, 5MB/s,
- posiada wbudowany blok do nadzoru transmisji na dwu magistralach FIP,
- posiada 8-bitowy interfejs do wymiany informacji z nadrzędnym procesorem,
- jest kompatybilny ze standardem FIP: NFC 46 602/603/604 oraz IEC 1158-2,
- posiada większość zaimplementowanych w nim protokołów warstwy aplikacji, łącza oraz „górną" część warstwy fizycznej,
- posiada zaimplementowany dekoder kodu Manchester 2 oraz FCS_16[1],
- wykonuje równolegle zadania przychodzące od strony sieci FIP i procesora nadrzędnego.

3.2. Charakterystyka układu MicroFIP

Układ MicroFIP jest w pełni kompatybilny ze standardem sieci FIP NFC 46 602/603/604 oraz IEC 1158-2.

W odróżnieniu do procesora FULLFIP2 nie ma możliwości pracy w trybie arbitra magistrali. Zmniejszona została (do niezbędnego minimum) liczba obsługiwanych zmiennych. W zamian w układzie zaimplementowany został uproszczony odpowiednik programu FIPCODE oraz blok redundancji medium transmisyjnego. MicroFIP nie wymaga stosowania dodatkowych elementów zewnętrznych w celu stworzenia zewnętrznej pamięci EPM[2], a w trybie pracy bez zewnętrznego kontrolera uaktywniana jest bardzo prosta warstwa aplika-

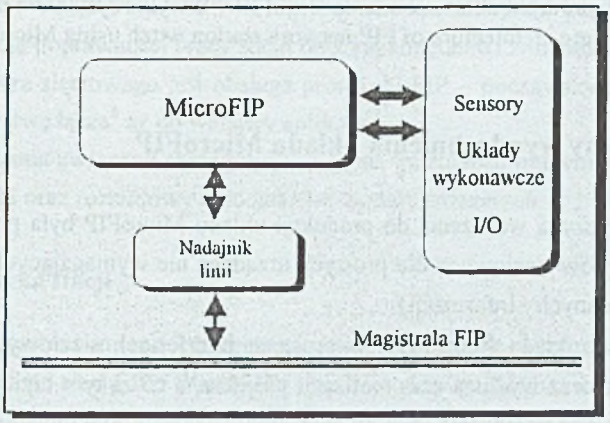
[1] FCS_16 – suma kontrolna ramki.

cji użytkownika, obsługująca dwie zmienne systemowe, jedną konfiguracyjną oraz dwie zmienne odpowiadające za obsługę portów I/O.

Układ MicroFIP:

- obsługuje do 8 zmiennych[3] - 4 zmienne „produkowane" oraz 4 „konsumowane",
- umożliwia nadawanie i odbieranie komunikatów o długości do 128 bajtów,
- posiada możliwość pracy z szybkościami transmisji 31.25kB/s, 1MB/s, 2.5MB/s,
- posiada interfejs do bezpośredniego podłączenia, najczęściej używanych, mikrokontrolerów jednoukładowych (Intel: 80C51, 80C251, Motorola: 68HC11, itd.).
- posiada dwa programowalne 8-bitowe porty I/O, umożliwiające podłączenie układów kontrolno-pomiarowych (czujników, układów wykonawczych, I/O).

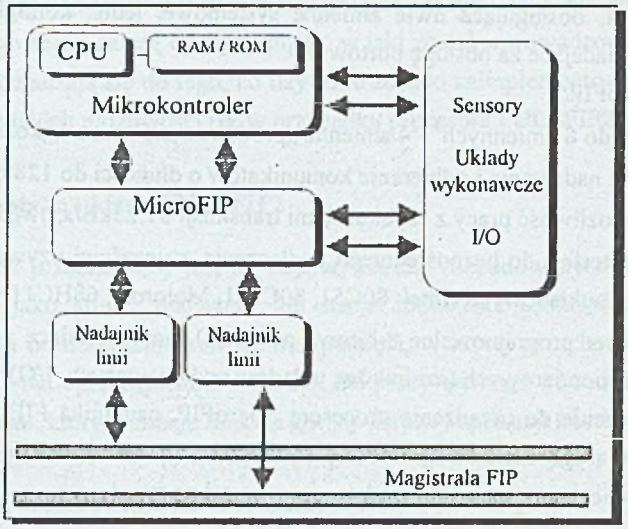
Przez dołączenie do urządzenia procesora MicroFIP, nadajnika FIELDRIVE oraz transformatora izolującego FIELDTR, możemy zbudować najprostszą wersji interfejsu sieciowego. Przykład takiego rozwiązania przedstawiony został na rys. 1.



Rys. 1. Najprostsza architektura abonenta sieci FIP z wykorzystaniem koprocesora MicroFIP
Fig. 1. Simplify architecture of FIP network station witch using MicroFIP coprocessor

Typowe rozwiązanie interfejsu sieciowego z wykorzystaniem procesora MicroFIP oraz dołączonym do niego mikrokontrolerem przedstawia rys. 2.

[2] EPM – zewnętrzna pamięć danych i programu (ang. External Private Memory).
[3] W trybie pracy z zewnętrznym kontrolerem MicroFIP obsługuje do 8 zmiennych, które mogą zajmować pamięć o pojemności do 120 bajtów.



Rys. 2. Kompletna architektura abonenta sieci FIP - z wykorzystaniem procesora MicroFIP
Fig. 2. Complete architecture of FIP network station witch using MicroFIP coprocessor

4. Przyczyny wyodrębnienia układu MicroFIP

Główną przyczyną wdrożenia do produkcji układu MicroFIP była potrzeba zmniejszenia kosztów interfejsów sieciowych dla prostych urządzeń nie wymagających obsługi dużej liczby zmiennych (danych / informacji).

Zastosowanie układu FULLFIP2 w prostych interfejsach sieciowych zwiększa stopień skomplikowania oraz wydłuża czas realizacji projektu, a co za tym idzie – niepotrzebnie podraża koszty całego urządzenia. Jest on zbyt drogi w fazie projektowania i oprogramowywania warstwy aplikacji użytkownika dla konkretnego urządzenia (oczywiście pomijając urządzenia, które wymagają zastosowania układu o tak dużych możliwościach).

Uproszczenie konstrukcji koprocesora sieciowego MicroFIP pociągnęło za sobą zmniejszenie jego możliwości odnośnie do liczby obsługiwanych zmiennych i trybów pracy[4], ale takie było zamierzenie konstruktorów.

Powstał więc prosty w implementacji i programowaniu układ prawie natychmiast gotowy do podłączenia dowolnego urządzenia do sieci FIP.

Jeszcze jedną przyczyną, dla której powstał ten koprocesor, jest chęć zwiększenia przez jej pomysłodawców popularności sieci FIP.

[4] Warstwa fizyczna i łącza nie uległy zmianie, zmodyfikowano jedynie warstwę aplikacji.

Jest to nowoczesna sieć polowa, o gruntownie przemyślanych założeniach co do mechanizmów jej działania i co najważniejsze – posiadająca normę, w oparciu o którą powstał cały szereg narzędzi oraz podzespołów umożliwiających praktyczną implementację w dowolnym urządzeniu. Sieć ta jest ciągle rozwijana – powstają coraz to nowsze produkty programowe oraz sprzętowe. Dowodem na to mogą być prace prowadzone nad następcą układu MicroFIP oraz cała lista nowych narzędzi pracujących pod kontrolą systemu Windows 95 i NT, do uruchamiania i testowania sieci FIP [14].

5. Praktyczna implementacja układu MicroFIP

W wyniku prac prowadzonych na przełomie lat 96/97 powstał projekt stacji zdalnych wejść-wyjść binarnych i analogowych, zbudowanej w oparciu o koprocesor sieciowy MicroFIP firmy CEGELEC oraz mikrokontroler 68HC11 firmy Motorola.

Zastosowanie mikrokontrolera umożliwiło uruchomienie warstwy aplikacji użytkownika, nadzorowanie i kontrolę poprawności pracy stacji oraz zapamiętanie i zmianę konfiguracji.

Zadaniem kontrolera sieciowego jest obsługa protokołu FIP – począwszy od warstwy fizycznej[5], poprzez warstwę łącza[6] aż do warstwy aplikacji[7].

W fazie projektowania zwracano szczególną uwagę na możliwość implementacji prostych algorytmów sterowania oraz rozbudowy o dodatkowe moduły rozszerzeń.

5.1. Charakterystyka stacji

Stacja RM_I/O-32 jest tanim oraz prostym w instalacji i obsłudze urządzeniem do zbierania i przesyłania danych po sieci FIP / WorldFIP.

---

[5] Warstwa fizyczna protokołu FIP realizowana jest przez układ MicroFIP wraz z układem FIELDRIVE: MicroFIP dokonuje obróbki cyfrowej sygnału, FIELDRIVE zamienia sygnał z postaci cyfrowej na analogową i odwrotnie, sprawdza poprawność nadawania i testuje magistrale na ewentualność wystąpienia konfliktu.

[6] Warstwa łącza protokołu FIP dokonuje wykrywania i korekcji błędów na poziomie ramek (FCS_16), określa rozmiar wysyłanego pakietu danych i jego przeznaczenie oraz zapobiega próbie transmisji danych przez dwie lub więcej stacji w tym samym czasie.

[7] Warstwa aplikacji protokołu FIP obejmuje procedury realizujące wymianę danych pomiędzy siecią FIP a nadrzędnym kontrolerem lub portami I/O.

Tabela 1

Standardy wejść / wyjść obsługiwanych przez stację

Nazwa	Liczba	Opis	Uwagi
Wejścia binarne	16	Izolowane galwanicznie w standardzie 12/24V DC POS/NEG, podział na 2 sekcje po 8 wejść	
Wyjścia binarne	16	Izolowane galwanicznie w standardzie 12/24V DC POS lub NEG, podział na 2 sekcje po 8 wyjść	
Wejścia analogowe	1	Izolowane galwanicznie w standardzie 0÷5mA, 0÷20mA, 0÷10V, Czas aktualizacji stanu 5÷50 ms	Opcja
	4/8	0÷20mA, 0÷10V, czas aktualizacji stanów 1ms	
Wyjścia analogowe	2	Izolowane galwanicznie w standardzie 0÷20mA, 4÷20mA 0÷10V	Opcja
Łącze szeregowe	1	RS-232 lub 485 do komunikacji z urządzeniami zewnętrznymi	Opcja

Na płycie czołowej umieszczono diody LED sygnalizujące stany wejść i wyjść binarnych, stany magistral sieci FIP i tryb pracy urządzenia.

Tabela 2

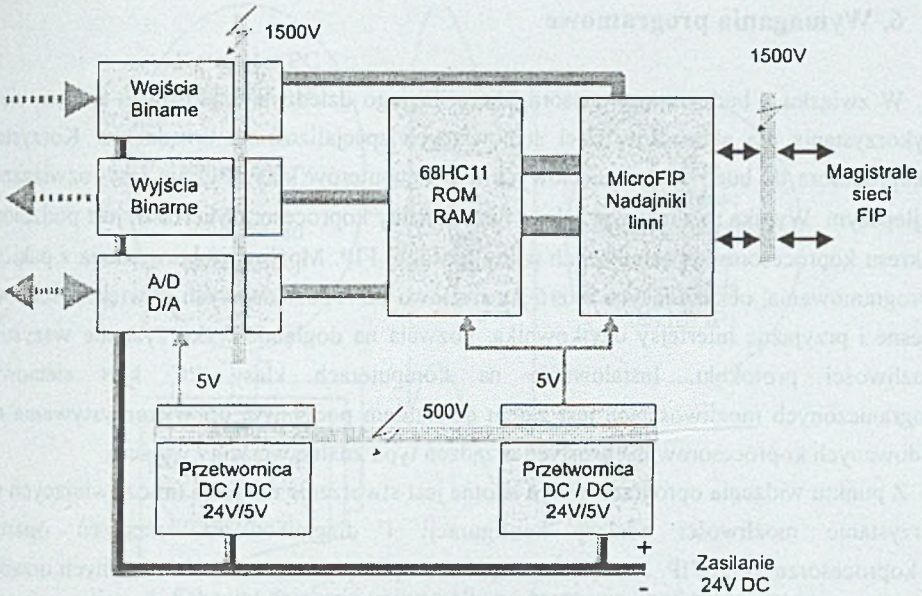
Opis zmiennych obsługiwanych przez RM_IO-32

Identyfikator	Opis zmiennej
# 14xy[8]	Zmienna produkowana przez stację, informująca o obecności stacji w sieci FIP
# 10xy	Produkowana przez stację – zmienna identyfikacyjna
# 07xy / # Xyxy[9]	Zmienna konsumowana przez stację RM_I/O-32 - jej zawartość ustawia wyjścia binarne OUT [15:0]
# 06xy	Zmienna produkowana przez stację - zawartością są stany logiczne z wejść IN [15:0]
# 05xy	Zmienna konsumowana przez stację - jej zawartość ustawia wyjścia analogowe Aout1 i Aout2
# 04xy	Zmienna produkowana przez stację RM_I/O-32 zawartością są stany wejścia analogowego AIN1 lub wejść analogowych AIN1÷AIN4 / 8
# 03xy	Zmienna konfiguracyjna konsumowana przez stację RM_I/O-32
# 02xy	Zmienna konfiguracyjna produkowana przez stację RM_I/O-32 odpowiednik ID_DAT 03xy

Wszystkie bloki: wejść i wyjść analogowych i binarnych, oraz interfejsu sieciowego posiadają izolację galwaniczną >1500V, blok zasilacza >500V.

[8] xy - oznacza numer ustawiony na przełączniku S1 służącym do wyboru numer abonenta w sieci FIP.

[9] Xyxy - oznacza globalny identyfikator zmiennej odpowiadającej za obsługę wyjść binarnych, niezależny od ustawień na przełączniku S1.



Rys. 3. Schemat blokowy RM_IO-32
Fig. 3. Block scheme of RM_IO 32

5.2. Konfiguracja trybów pracy

Po załączeniu zasilania z pamięci wewnętrznej mikrokontrolera odczytywane są dane, w oparciu o które konfigurowany jest koprocesor sieci FIP oraz przetwornik A/D.

Istnieje możliwość zmiany i zapamiętania konfiguracji w wewnętrznej pamięci EEPROM mikrokontrolera. Konfiguracja parametrów pracy stacji RM_IO-32 odbywa się przez sieć FIP z użyciem, specjalnie do tego celu opracowanego przez firmę PROLOC, oprogramowania, pracującego pod kontrolą systemu KRONOS lub innego oprogramowania, za pomocą którego możemy przesłać odpowiednio sformatowaną ramkę o identyfikatorze #03xy i zawartości opisanej w dokumentacji technicznej [17].

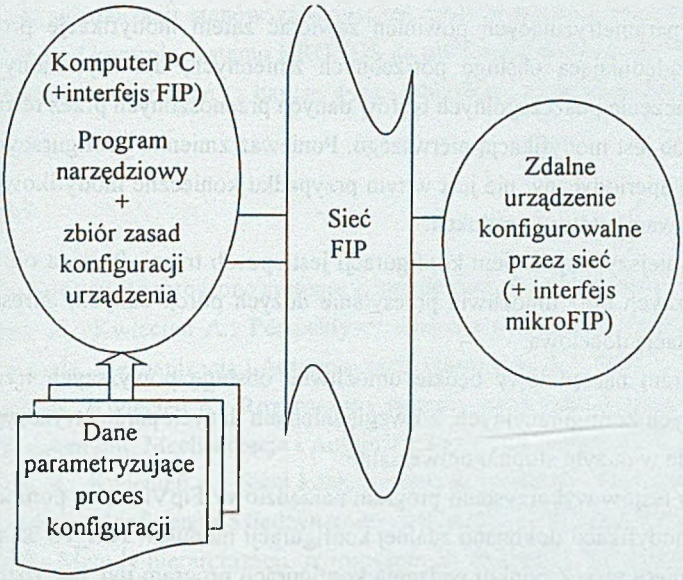
Za pomocą zmiennej #03xy możemy dokonać następujących zmian dotyczących:

- globalnego identyfikatora zmiennej #07xy / #XYxy,
- stanów awaryjnych wyjść binarnych (Safety position),
- czasu milczenia (Silence time),
- czasów odświeżania zmiennych #07xy, #06xy, #05xy, #04xy, #03xy, #02xy,
- danych identyfikujących stację,
- parametrów przetwarzania przetwornika A/D.

6. Wymagania programowe

W związku z budową koprocesora MicroFIP jego dziedzina zastosowań zawęża się do wykorzystania dla abonentów sieci stanowiących specjalizowane urządzenia. Korzystanie z koprocesora do budowy kart sieciowych dla komputerów klasy PC nie jest rozwiązaniem najlepszym. Wynika to z faktu, iż zakres funkcjonalny koprocesora MicroFIP jest podzbiorem zakresu koprocesorów obsługujących pełny protokół FIP. Możliwość korzystania z pakietów oprogramowania, obsługujących interfejsy sieciowe FIP i posiadających w większości nowoczesne i przyjazne interfejsy użytkownika, pozwala na dogłębne wykorzystanie wszystkich możliwości protokołu. Instalowanie na komputerach klasy PC kart sieciowych o ograniczonych możliwościach jest zatem działaniem podobnym do wykorzystywania rozbudowanych koprocesorów dla prostych urządzeń typu zdalne wejścia / wyjścia.

Z punktu widzenia oprogramowania istotne jest stworzenie narzędzi umożliwiających wykorzystanie możliwości zdalnej konfiguracji i diagnozowania urządzeń opartych na koprocesorze MicroFIP. Narzędzie takie może być dedykowane dla konkretnych urządzeń lub może być uniwersalne przy założeniu stałego sposobu pracy. Rozwiązanie dedykowane jest dość kosztowne, gdyż wymaga tworzenia specjalnych programów dla konkretnych typów urządzeń lub budowy kolejnych programów obsługi w kolejnych wersjach narzędzi programistycznych. Bardziej uniwersalne jest rozwiązanie, w którym zakłada się jednolitą metodę zdalnej współpracy urządzenia, a konfiguracja i diagnozowanie podlegają parametryzacji. Uzyskuje się wówczas jedno narzędzie i n plików konfiguracyjnych (rys. 4). Jednolita metoda zdalnej współpracy wydaje się niemożliwa ze względu na mnogość producentów sprzętu i co za tym idzie – rozwiązań konstrukcyjno-programowych. Należy jednak pamiętać, iż mamy do czynienia z protokołem sieci mającym swój standard i z jasno postawionym problemem polegającym na skomunikowaniu ze sobą warstwy aplikacji konfigurowanego urządzenia i konfiguratora. Należy ponadto założyć, że znamy działanie interfejsu zdalnego warstwy aplikacji konfigurowanych urządzeń. Na tej podstawie można opracować grupę metod, które mogą być używane do zdalnej konfiguracji, i dla tej grupy stworzyć narzędzie programowe.



Rys. 4. Schemat działania uniwersalnego narzędzia konfiguracyjnego
Fig. 4. Schematic of universal configuration tool work

Stworzone według tej idei narzędzie będzie uniwersalne tylko w pewnym stopniu, a mianowicie dla grupy urządzeń charakteryzujących się tym samym zbiorem zasad konfiguracji. Przez zbiór zasad konfiguracji należy rozumieć metodologię zdalnej współpracy urządzenia na płaszczyźnie wymiany informacji serwisowej. Część usług serwisowych jest definiowana przez sam protokół. Należą do nich przede wszystkim: testowanie obecności oraz zdalna identyfikacja abonenta. Umożliwiają one sprawdzenie, czy abonent o zadanym adresie fizycznym jest obecny w sieci, oraz sprawdzenie, jakiego typu jest to urządzenie. Do usług niezdefiniowanych przez protokół, a niezbędnych do konfiguracji i diagnostyki, należy zaliczyć przesył informacji konfiguracyjnej oraz zwrotny transfer informacji o konfiguracji bieżącej. Istotne jest, aby obie warstwy aplikacji zarówno stacji konfigurującej, jak i konfigurowanej potrafiły się porozumieć co do tych dwóch typów transferów. Analizując protokół, można wyodrębnić trzy metody przesyłania tych informacji:

- transfer przy użyciu zmiennych cyklicznych,
- transfer przy użyciu transakcji aperiodycznych zmiennych,
- transfer przy użyciu transakcji aperiodycznych komunikatów.

Pierwszy ze sposobów zakłada, że istnieją w systemie specjalne zmienne cykliczne związane z procesem konfiguracji. Identyfikatorów tych zmiennych nie może używać sieć do przenoszenia informacji od obsługiwanego procesu. Zmienne mają określony czas ważności, na podstawie którego wypracowywany jest status produkcji zmiennej [7] służący warstwie aplikacji urządzenia konfigurowanego do rozpoznania ważnych danych konfiguracyjnych.

Zbiór danych parametryzujących powinien zawierać zatem modyfikację programu arbitra magistrali uwzględniającą obsługę potrzebnych zmiennych, definicje samych zmiennych i określenie znaczenia poszczególnych bajtów danych przenoszonych przez te zmienne.

Drugi sposób jest modyfikacją pierwszego. Ponieważ zmienne konfiguracyjne obsługiwane są w sposób aperiodyczny, nie jest w tym przypadku konieczne modyfikowanie programu arbitra ani testowanie statusu produkcji.

Najefektywniejszym sposobem konfiguracji jest sposób trzeci. Posiada on zalety transferów aperiodycznych oraz umożliwia przesyłanie dużych porcji danych, adresując przy tym bezpośrednio stację docelową.

Jeżeli program narzędziowy będzie umożliwiał obsługę powyższych trzech sposobów transferów danych konfiguracyjnych, z uwzględnieniem danych parametryzujących, to narzędzie takie będzie w dużym stopniu uniwersalne.

Dla potrzeb testów wykorzystano program narzędziowy FipView, za pomocą którego bez żadnych jego modyfikacji dokonano zdalnej konfiguracji modułów RM_IO 32 przy wykorzystaniu metody pierwszej. Z punktu widzenia konfiguracji program ten, bez rozbudowy o moduł interpretacji danych parametryzujących, jest dość niewygodny w użyciu dla użytkowników nie zaawansowanych w dziedzinie obsługi sieci FIP. Alternatywą dla programu FipView było wykorzystanie systemu wizualizacyjno-nadzorczego KRONOS [18]. Stworzenie konfiguratora na bazie tego systemu okazało się dość proste. System KRONOS jako narzędzie udostępnia m.in. mechanizmy tworzenia własnych okien dialogowych, obsługi sieci oraz programów użytkownika. Wykorzystując te mechanizmy stworzono program umożliwiający zdalną konfigurację urządzeń zgodnie z danymi parametryzującymi zapisywanymi w plikach tekstowych na bazie metody pierwszej. Program ten po zaimplementowaniu pozostałych metod będzie mógł posłużyć jako uniwersalne narzędzie konfiguracyjno-diagnostyczne, oczywiście dla grupy urządzeń ze zdalnym interfejsem warstwy aplikacji pracującym na bazie powyższych metod.

7. Wnioski

Zastosowanie koprocesora MicroFIP znacznie upraszcza konstrukcję interfejsu sieciowego, obniża koszty jednostkowe oraz skraca czas realizacji projektu. Dodatkowo przy swojej prostocie MicroFIP posiada dość znaczne możliwości, szczególnie w trybie pracy z zewnętrznym kontrolerem.

Opracowana i wdrożona do produkcji stacja RM_I/O-32 z sukcesem została zastosowana w Elektrociepłowni Miechowice do zbieranie stanów zabezpieczeń transformatorów 6/0.6kV zasilających przemienniki częstotliwości dużej mocy z serii GD 2000E [16].

Aplikacja do wizualizacji stanów zabezpieczeń transformatorów oraz arbiter magistrali sieci FIP pracują pod kontrolą systemu KRONOS na platformie systemowej Windows NT4.0. Całość pracuje bez zastrzeżeń i z bardzo dużą szybkością i niezawodnością.

LITERATURA

[1] Praca zbiorowa: Rozproszone systemy komputerowe. Pronet, Gliwice 1994.

[2] Grzywak A., Kwiecień A.: Perspektywy rozwoju zastosowań sieci komputerowych w górnictwie. Mechanizacja i Automatyzacja Górnictwa nr 5, 6/94.

[3] Grzywak A., Kwiecień A.: Rozproszone systemy sterowania i zarządzania procesami technologicznymi. Mechanizacja i Automatyzacja Górnictwa nr 8/94.

[4] Grzywak A., Kwiecień A.: Sieci komputerowe w systemach sterowania i zarządzania w górnictwie. Konferencja Międzynarodowa ICAMC'95 World Mining Congress.

[5] Cupek R.: Metody hierarchizacji rozproszonych procesów przemysłowych. ZN Pol. Śl. s. Informatyka z. 28, Gliwice 1995.

[6] Bigewski Z.: Optymalizacja pracy sieci przemysłowych. ZN Pol. Śl. s. Informatyka z. 28, Gliwice 1995.

[7] Kwiecień A.: Sieć rozległa FIP. ZN Pol. Śl. s. Informatyka z. 24, Gliwice 1993.

[8] Kwiecień A., Gaj P.: Sieć FIP, wstęp do analizy czasowej. ZN Pol. Śl. s. Informatyka z. 28, Gliwice 1995.

[9] Kwiecień A., Gaj P., Grzywak A., Mrówka Z.: Rozwiązania sprzętowe i programowe sieci przemysłowej FIP. ZN Pol. Śl. s. Informatyka z. 30, Gliwice 1996.

[10] Kwiecień A., Gaj P., Mrówka Z.: Optymalizacja wymian w sieci FIP. ZN Pol. Śl. s. Informatyka z. 32, Gliwice 1997.

[11] Cupek R., Fojcik M.: Budowa modułów komunikacyjnych stacji nadzorczej z sieciami przemysłowymi. ZN Pol. Śl. s. Informatyka z. 32, Gliwice 1997.

[12] FIPCODE Software v. 5.0 User Reference Manual; DPS 50263 b–en, Clamart 1994.

[13] FIP NETWORK General Introduction; DPS 50249 aA, Clamart 1990.

[14] Klubowe materiały szkoleniowe; WorldFip, Clamart 1995, 1996, 1997, 1998.

[15] FIP STARTER KIT materiały klubowe WorldFip; Clamart 1995.

[16] GD2000E User Manual; DPS 51374 c–en, Clamart 1997.

[17] Dokumentacja techniczna stacji RM_IO 32; Proloc, Katowice 1997.

[18] System KRONOS v.2.0 – podręcznik użytkownika; Proloc, Katowice 1997.

Recenzent: Dr hab. inż. Adam Mrózek,Prof. Politechniki Śląskiej

Wpłynęło do Redakcji 21 stycznia 1998 r.

Abstract

The paper describes the issue of simplification of FIP network interface and its practical implementation. The new coprocessors are able to build easy and cheap network interface for remote device like remote IO modules. In article is described example of small coprocessor CEGELEC's MicroFIP.

The one part of article describes the functionality of MicroFIP and its practical implementation in remote input / output module RM_IO 32. Second part is dedicated to question remote subscriber configuration and diagnostic. Article trying gives the way of construction for programming tools using to remote device configuration.

Authors based on early publications [[7], [8], [9], [10]] and own experience with building, using and servicing FIP network remote device.