

## TECHNOLOGIE INTERNETOWE W SYSTEMACH POMIAROWO – STERUJĄCYCH

*W referacie przedstawiono aktualne tendencje w zakresie rozwiązań komunikacyjnych stosowanych w systemach pomiarowo – sterujących (SPS). Zarysowano możliwości i warunki wykorzystania technologii internetowych do budowy SPS. Przedstawiono charakterystykę dedykowanych serwerów WWW i zaprezentowano korzyści, jakie daje ich wykorzystanie w tworzeniu otwartej struktury SPS. Opisano część sprzętową i środowisko programowe dedykowanych serwerów WWW. Przedstawiono sposób konfigurowania parametrów protokołu TCP/IP, modyfikacji oprogramowania wbudowanego serwera WWW oraz sposób ładowania stron WWW. Dyskutowane są zagadnienia dotrzymania ograniczeń czasowych oraz spełnienie wymogów bezpieczeństwa. Zwrócono uwagę na inicjatywę utworzenia Internetu 0 dla urzędów.*

### 1. STAN WIEDZY I TENDENCJE ROZWOJOWE

Dostęp do informacji pomiarowo - sterującej z różnych miejsc i poziomów struktury informacyjnej staje się obecnie coraz ważniejszym aspektem współcześnie opracowywanych systemów pomiarowo – sterujących. Stosowanie technologii internetowej umożliwia realizację tego postulatu i pozwala na osiągnięcie wymiernych korzyści ekonomicznych wynikających z możliwości wykorzystania istniejącej infrastruktury komunikacyjnej (sieci LAN/WAN), istniejącego sprzętu komputerowego uniezależniając jednocześnie użytkowników od dedykowanego oprogramowania użytkowego poza powszechnie stosowanymi przeglądarkami internetowymi. Rozwój sieci komputerowych i technologii internetowych doprowadził do stanu, w którym występowanie infrastruktury komunikacyjnej oraz narzędzi do tworzenia aplikacji informatycznych jest powszechne. Sytuacja ta ma znaczący wpływ na obserwowane tendencje wprowadzania tych technologii do inteligentnych urządzeń pomiarowych, sterujących i systemów pomiarowo - sterujących. Wykorzystanie w systemach pomiarowo - sterujących rozwiązań sprzętowych i programowych, z sukcesem stosowanych w sieciach komputerowych (Ethernet, TCP/IP, HTTP, WWW itp.), prowadzi do uproszczenia wielopoziomowej struktury tworzącej model systemu informacyjnego przedsiębiorstwa. Uproszczenie struktury przejawia się w tym, że zarówno dostęp do informacji pomiarowych jak i przesyłanie informacji sterujących elementami wykonawczymi lub danych konfiguracyjnych jest możliwe z poziomu przeglądarki internetowej, która stała się standardowym wyposażeniem komputera osobistego niezależnie od stosowanej platformy systemowej.

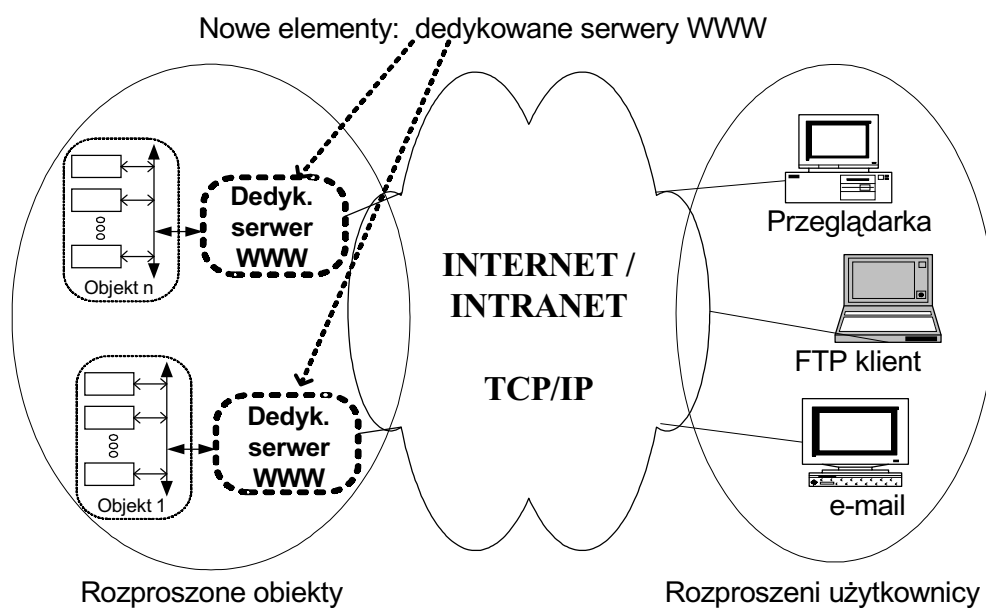
Podstawową zaletą wynikającą ze stosowania technologii internetowych jest rezygnacja z konieczności stosowania dedykowanego oprogramowania pracującego po stronie użytkownika, co może w sposób znaczący obniżyć koszt funkcjonowania aplikacji, zwłaszcza przy dużej liczbie osób z niej korzystających. Dzięki zastosowaniu stosu protokołowego TCP/IP możliwe jest wprowadzenie na poziom sieci przemysłowych technologii internetowych. Stosowanie technologii internetowych pozwala na znaczne uproszczenie interfejsu użytkownika przy dostępie do informacji pomiarowo – sterującej (przeglądarka, e-mail, FTP) oraz wpływa na obniżenie kosztów przygotowania oprogramowania (JVM, Java) i zwalnia z konieczności stosowania drogich i specjalizowanych systemów projektowo -

uruchomieniowych. Skróceniu ulega czas opracowywania i uruchamiania oprogramowania, które stanowi obecnie największy czynnik kosztów opracowywania nowego urządzenia.

Zastosowanie technologii internetowych do budowy urządzeń pomiarowych i sterujących może być najprostszą metodą standaryzacji środowiska komunikacyjnego w sieciach przemysłowych na poziomie sieci typu *Fieldbus* i *Devicebus*. Rozwiązanie takie umożliwia budowanie systemów otwartych, w których projektanci lub użytkownicy mogą zestawiać urządzenia pochodzące od różnych producentów i tworzyć oprogramowanie aplikacyjne metodą komponentową z wykorzystaniem przeglądarki internetowej.

## 2. DEDYKOWANE SERWERY WWW

Internetowe systemy pomiarowo – sterujące tworzą trzy podstawowe elementy: system komunikacyjny ze stosem protokołowym TCP/IP, dedykowany serwer WWW, do którego bezpośrednio lub za pośrednictwem sieci przemysłowej podłączone są czujniki, przetworniki i elementy wykonawcze oraz stacje klienckie z przeglądarkami internetowymi, przy użyciu których użytkownik uzyskuje dostęp do danych pomiarowych i ma możliwość oddziaływania na obiekt. W systemach tych informacje pomiarowe i sterujące pomiędzy serwerem WWW a użytkownikiem przesyłane są w postaci tekstowych stron WWW napisanych w języku znacznikowym HTML. Na rysunku 1 przedstawiono strukturę przykładowego internetowego systemu pomiarowo – sterującego z serwerami WWW.



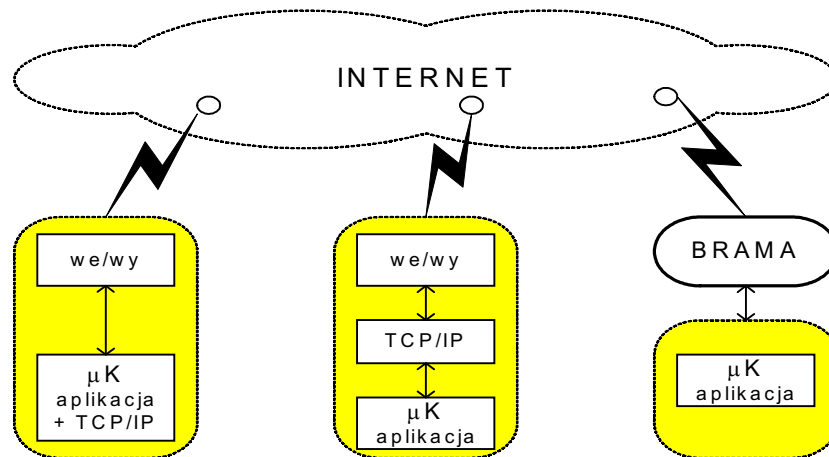
Rysunek 1. Dedykowane serwery WWW w strukturze SPS

Zakres zastosowań internetowych systemów pomiarowo-sterujących może być bardzo szeroki. Zdalne pomiary, diagnostyka, konfigurowanie, załączanie wykonywane z poziomu przeglądarki internetowej standaryzują interfejs użytkownika i znacznie upraszczają dostęp do zasobów dedykowanego serwera WWW. Wykorzystywanie w środowisku systemów pomiarowo – sterujących sprawdzonych i wykorzystywanych w środowisku informatycznym standardowych protokołów komunikacyjnych oraz internetowych technologii informatycznych znacznie ułatwia integrację urządzeń pochodzących od różnych dostawców oraz integrację z systemem informacyjnym firmy. Internetowe systemy pomiarowo-sterujące mają wiele atrakcyjnych zalet, jednak ich stosowanie wymaga bardzo starannego rozwiązania, przede wszystkim zagadnień bezpieczeństwa. Ponadto w takich systemach trudne jest dotrzymanie wygórowanych warunków czasu rzeczywistego.

Ze względu na ograniczenia sprzętowe, wbudowane serwery WWW realizują jedynie niezbędne funkcje stosu TCP/IP. Serwer WWW identyfikowany jest przez unikalny adres IP

lub nazwę domenową. W zależności od sposobu implementacji stosu TCP/IP, wbudowane serwery WWW mogą być wykonane w następujących wariantach (rysunek 2):

- Stos TCP/IP i oprogramowanie aplikacyjne wykonywane są przez ten sam mikrokontroler,
- Stos TCP/IP wykonywany jest przez specjalizowany układ, który połączony jest mikrokontrolerem serwera WWW,
- Stos TCP/IP wykonywany jest w bramie, która połączona jest z mikrokontrolerem aplikacji portem szeregowym.



Rysunek 2. Przykładowe struktury dedykowanych serwerów WWW

W pierwszym wariantcie duże wymagania stawiane są mikrokontrolerom, ponieważ obsługa stosu TCP/IP jest absorbująca i mikrokontrolerom pozostaje niewiele czasu na wykonywanie aplikacji. Budowa serwera WWW z wykorzystaniem specjalizowanych układów (np.: S-7600AS firmy Seiko Instruments) lub dedykowanych bram (np.: iChip firmy Connect One, SmartStack firmy e-Device, EMIT firmy emWare), znacznie odciąża mikrokontroler i zmniejsza wymagania na rozmiar układów pamięciowych. Do specjalizowanych układów lub dedykowanych bram mogą zostać przeniesione funkcje związane z bezpieczeństwem, których obsługa jest również czasochłonna. Dominującą strukturą obecnie projektowanych dedykowanych serwerów WWW jest struktura w wariantcie pierwszym, która prowadzi do rozwiązań typu SoC (ang. System on a Chip). Tę strukturę dedykowanego serwera WWW do budowy bramy internetowej NB-11 w OBRME Metrol w Zielonej Górze. NB-11 stanowi połączenie dedykowanego serwera WWW oraz sterownika PLC.

### 3. CZĘŚĆ SPRZĘTOWA DEDYKOWANEGO SERWERA WWW

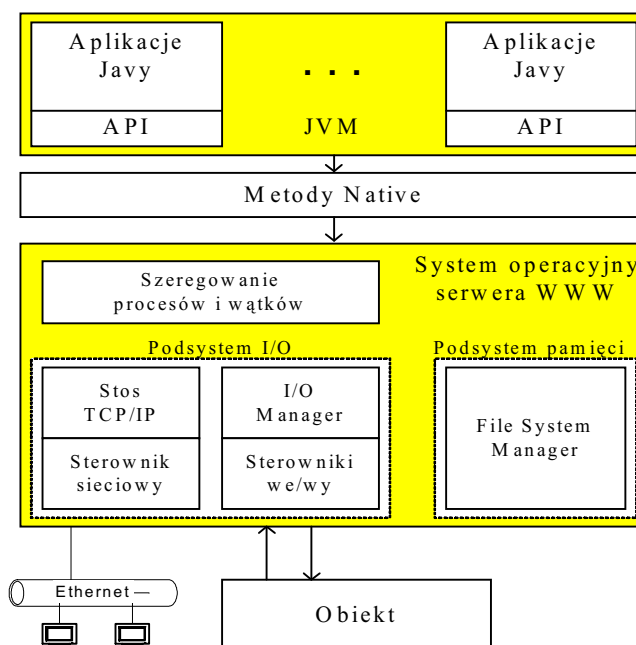
Środowisko sprzętowe i programowe dedykowanych serwerów WWW jest bardzo zróżnicowane. Do realizacji dedykowanego serwera WWW TINi wykorzystano 8-bitowy procesor DS80C4000 o ulepszonej architekturze typu 8051 z zegarem do 50 MHz, cyklem instrukcji 80 nsek i przestrzenią pamięci o adresacji liniowej do 16 MB. Serwer WWW zawiera do 1 MB zewnętrznej pamięci SRAM podtrzymywanej bateryjnie do buforowania danych i do pamiętania stron WWW. Pamięci SRAM i Flash ROM podłączone są do zewnętrznej szyny adresowej i do szyny danych.

Serwer WWW posiada kilka wyjść komunikacyjnych: wyjście do sieci Ethernet, cztery sprzętowe pełno-duplexowe porty szeregowy, wyjście do sieci przemysłowej CAN oraz port szeregowy 1-Wire. Porty szeregowy wykorzystywane są do tworzenia połączeń SLIP lub PPP. Jeden z portów szeregowych wykorzystywany jest do konfigurowania serwera WWW z zewnętrznego komputera. Wyjście do sieci Ethernet zrealizowano z wykorzystaniem układu CS 8900A z wewnętrzną 4K bajtową pamięcią, który zaprogramowano do pracy w trybie 8-bitowym. Obsługa sterowników komunikacyjnych realizowana jest

z wykorzystaniem przerwań. Wykorzystany do budowy serwera WWW procesor posiada maksymalnie 64 wejścia binarne, zatem bezpośrednio do serwera WWW można podłączyć do 64 wejść lub wyjść dwustanowych. Dedykowany serwer WWW wyposażony jest w batteryjnie podtrzymywany układ zegara czasu rzeczywistego.

#### 4. CZĘŚĆ PROGRAMOWA DEDYKOWANEGO SERWERA WWW

Ważnym elementem dedykowanego serwera WWW jest system operacyjny. Zarządza on zasobami (dostęp do pamięci, szeregowanie procesów i wątków, dostęp do wewnętrznych i zewnętrznych elementów sprzętowych) i tworzy środowisko programistyczne dla programów aplikacyjnych. Ze względu na charakter pracy dedykowany serwer WWW powinien być wyposażony w systemem operacyjny umożliwiającą pracę wielozadaniową i posiadający zestaw standardowych protokołów komunikacyjnych i zestaw modułów API pozwalających w fazie uruchamiania skupić się głównie na aplikacji. Dyskutowany w artykule dedykowany serwer WWW TINI TBM400, poza wymienionymi elementami programowymi wyposażony jest w wirtualną maszynę javy (JVM) (rysunek 3) zajmującą jedynie 40kbajłów w pamięci Flash.



Rysunek 3. Struktura blokowa środowiska programowego dedykowanego serwera TINI.

Kompletny system operacyjny zajmuje 384 kbajty pamięci Flash. Programy aplikacyjne dla tego serwera pisane są języku Java. Poza niezależnością od platformy sprzętowej zaletą takiego rozwiązania jest wykorzystywanie bezpłatnych narzędzi uruchomieniowych. Oprogramowanie aplikacyjne dla danego procesora może być przygotowywane również w języku danego procesora, co widoczne jest w strukturze pokazanej na rys. 3 w postaci warstwy „Metody Native”. W odróżnieniu od powszechnie stosowanych apletów Javy, w serwerze TINI aplikacje Javy mają pełen dostęp do zasobów serwera wbudowanego, dzięki czemu możliwa jest szybka obsługa urządzeń zewnętrznych.

#### 5. KONFIGUROWANIE DEDYKOWANYCH SERWERÓW WWW

Do poprawnej pracy dedykowanego serwera WWW niezbędne są parametry protokołu TCP/IP, które ustawia się programem konfiguracyjnym. Najważniejszy parametr to adres logiczny IP. Może on być ustawiony statycznie lub odczytywany dynamicznie z serwera DHCP. Serwer nie ma stałego numeru fizycznego MAC. Można go zmienić i wprowadzić różne wartości. Jeśli w oprogramowaniu serwera wykorzystane będą funkcje obsługi poczty

elektronicznej, to należy skonfigurować ustawienia dotyczące konfiguracji części pocztowej jak: adres nadawcy i odbiorcy, adres serwera pocztowego. Serwer może również pracować z protokołem FTP. Aby to było możliwe, należy włączyć jego obsługę i ustawić nazwę użytkownika i hasło dostępu do serwera.

Modyfikowanie oprogramowania dedykowanego serwera WWW wymaga znajomości programowania w języku Java lub C i znajomości pracy mechanizmów wymiany danych w Internecie. W szczególności wymagana jest znajomość działania protokołów TCP/IP, HTTP i FTP. Oprogramowanie serwera jest budowane w sposób modułarny. Programista ma możliwość wyboru modułów, z których będzie korzystał jego serwer. Wyboru modułów należy dokonać na etapie kompilacji programu serwera WWW. Domyślnie ładowane są wszystkie wymienione powyżej moduły obsługi różnych protokołów.

Serwer WWW przechowuje strony w postaci plików tekstowych w formacie html. Mogą to także być pliki graficzne w dowolnych formatach lub aplety java. Przechowywane one są w pamięci flash serwera w systemie plikowym, który jest utworzony programowo i przypomina system plikowy FAT z komputera PC. Dostęp do tego systemu jest możliwy poprzez FTP. Aby zmienić wygląd stron umieszczonych na serwerze i oglądanych przez użytkowników sieci przez przeglądarki, należy przekopiować nowe pliki do systemu plikowego serwera. Dostęp jest możliwy po podaniu prawidłowej nazwy konta i hasła użytkownika

## **6. BEZPIECZEŃSTWO DEDYKOWANEGO SERWERA WWW**

Zagadnienie bezpieczeństwa jest jednym z kluczowych zagadnień, które należy rozwiązać wówczas, kiedy jakiegokolwiek urządzenie podłączane są do sieci Internet. Ważność tych zagadnień zależy od rodzaju funkcji wykonywanych przez dane urządzenie i od rodzaju informacji, która jest w nich przechowywana.

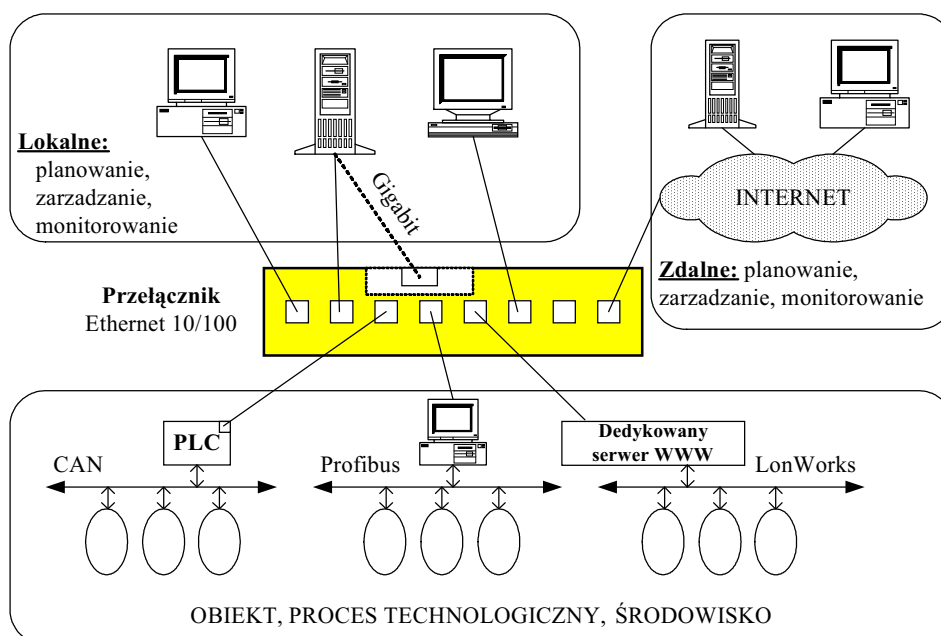
Stosowanie starszych, prostszych wersji protokołów komunikacyjnych nie posiadających żadnych technik zabezpieczających jest podstawowym źródłem zagrożeń i stanowi duże ułatwienie dla potencjalnych hackerów. Protokoły warstwy aplikacji Telnet, FTP i HTTP podczas autentykacji przesyłają nazwę użytkownika i hasło w sposób niezakodowany. Ze względu na ograniczone zasoby, w zbudowanym serwerze WWW zagadnienia bezpieczeństwa nie zostały uwzględnione w sposób szczególny. Przyjęto, że w razie potrzeby istnieje wiele rozwiązań funkcjonujących poza serwerem, pozwalających na zabezpieczenie się przed nieautoryzowanym dostępem do zasobów serwera. Najprostszym sposobem zabezpieczenia się przed nieautoryzowanym dostępem jest udostępnienie aplikacji powyżej TCP/IP jedynie użytkownikowi o danym adresie IP. Można to osiągnąć poprzez podanie adresu IP wówczas, kiedy serwer rozpoczyna obsługę portu np.: TCPopen (1032, 192.168.31.55). Każde odwołanie do portu 1032 od użytkownika innego niż 192.168.31.55 zostanie odrzucone. Rozwiązanie to nie szyfruje przesyłanej informacji, zatem nie jest ono w pełni bezpieczne. Istnieje możliwość przechwycenia zarówno adresu IP jak i przesyłanych danych. Przechwycenie adresu IP pozwala na dostęp do zasobów serwera WWW z innego miejsca blokując ten adres dla właściwego użytkownika, który w sieci DHCP otrzyma inny adres IP uniemożliwiający na dostęp do serwera WWW. Stosowanie szyfrowania informacji wymaga stosowania odpowiednich algorytmów np. SSL.

Dedykowane serwery WWW nie są w stanie obsługiwać zbyt wielu użytkowników jednocześnie, zatem w łatwy sposób można takie serwery zablokować poprzez ataki typu DoS (ang. Denial of Service). Najlepszym sposobem uniknięcia takiej sytuacji jest ograniczenie liczby użytkowników pracujących w tej samej sieci, w której pracuje dedykowany serwer WWW. Jeżeli jest to niemożliwe do zrealizowania, to należy pomiędzy serwerem WWW a użytkownikami zewnętrznymi zastosować, np.: zaporę ogniową lub serwer proxy.

## 7. TECHNOLOGIE INTERNETOWE I ETHERNET PRZEMYSŁOWY

Wykorzystanie w rozproszonych systemach pomiarowo - sterujących rozwiązań sprzętowych i programowych, z sukcesem stosowanych w sieciach komputerowych (Ethernet, TCP/IP, HTTP, WWW itp.), prowadzi do uproszczenia wielopoziomowej struktury tworzącej model systemu informacyjnego przedsiębiorstwa [8,9]. Uproszczenie struktury przejawia się w tym, że zarówno dostęp do informacji pomiarowych jak i przesyłanie informacji sterujących elementami wykonawczymi lub danych konfiguracyjnych jest możliwe z poziomu przeglądarki internetowej, która stała się standardowym wyposażeniem komputera osobistego niezależnie od stosowanej platformy systemowej.

Podstawową zaletą wynikającą ze stosowania Ethernetu przemysłowego i technologii internetowych jest rezygnacja z konieczności stosowania dedykowanego oprogramowania pracującego po stronie użytkownika, co może w sposób znaczący obniżyć koszt funkcjonowania aplikacji, zwłaszcza przy dużej liczbie osób z niej korzystających [5,7,12]. Stosowanie sieci Ethernet w sposób naturalny prowadzi do używania stosu TCP/IP. Dzięki zastosowaniu stosu protokolowego TCP/IP możliwe jest wprowadzenie na poziom sieci przemysłowych technologii internetowych. Stosowanie technologii internetowych pozwala na znaczne uproszczenie interfejsu użytkownika przy dostępie do informacji pomiarowo – sterującej (przeglądarka, e-mail, FTP) oraz wpływa na obniżenie kosztów przygotowania oprogramowania (JVM, Java) i zwalnia z konieczności stosowania drogich i specjalizowanych systemów projektowo - uruchomieniowych. Skróceniu ulega czas opracowywania i uruchamiania oprogramowania, które stanowi obecnie największy czynnik kosztów opracowywania nowego urządzenia. Na rysunek 4 przykładową strukturę SPS, w którym na najniższym poziomie komunikacyjnym wyróżniono klasyczne sieci przemysłowe CAN, Profibus i LonWorks. Węzły pomiarowo – sterujące pracujące w tych sieciach są zainstalowane bezpośrednio przy obiekcie. Na kolejnym poziomie znajduje się przełącznik Ethernet-u przełączającego, który z jednej strony połączony jest poprzez sterownik PLC, stację roboczą lub dedykowany serwer WWW z obiektem oraz z siecią komputerową poziomu biurowego.



Rysunek 4. SPS z Ethernetem przemysłowym i dedykowanym serwerem WWW

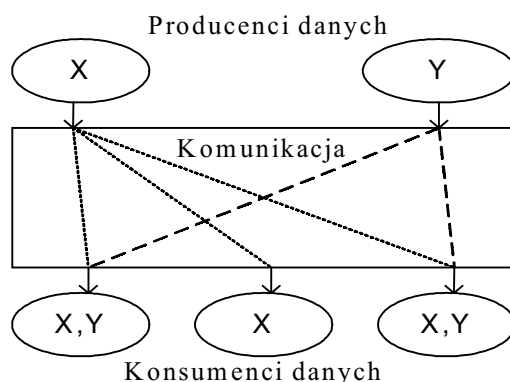
Ta część obiektu, która jest podłączona poprzez dedykowany serwer WWW dostępna jest dla użytkowników lokalnych lub zdalnych z poziomu przeglądarki internetowej. Natomiast dostęp do informacji pomiarowo – sterujących w pozostałych dwóch przypadkach wymaga

zainstalowania po stronie klienta dodatkowego oprogramowania i znacznie utrudnia realizację funkcji zdalnego konfigurowania lub monitorowania obiektu. W ramach projektu celowego realizowanego w Instytucie Metrologii Elektrycznej w Zielonej Górze opracowano urządzenie, w którym połączone zostały funkcje sterownika PLC i dedykowanego serwera WWW.

Zastosowanie technologii internetowych do budowy urządzeń pomiarowych i sterujących może być najprostszą metodą standaryzacji środowiska komunikacyjnego w sieciach przemysłowych na poziomie sieci typu *Fieldbus* i *Devicebus* [8,11]. Rozwiązanie takie umożliwia budowanie systemów otwartych, w których projektanci lub użytkownicy mogą zestawiać urządzenia pochodzące od różnych producentów i tworzyć oprogramowanie aplikacyjne metodą komponentową z wykorzystaniem przeglądarki internetowej.

## 6. ARCHITEKTURA TYPU PS

Wymienionych powyżej wad pozbawiona jest architektura typu PS (ang.: „*Publish – Subscribe*”) (rys. 3) [2,4,5]. W tej architekturze węzły dzielą się na producentów i konsumentów informacji. Producent danych może obsługiwać wielu konsumentów. Każda informacja wysyłana przez producenta posiada nazwę oraz przypisany jej na etapie konfigurowania typ, na podstawie których konsumenci je identyfikują. Przesyłanie informacji inicjowane jest przez producentów w trybie wyzwalanym zdarzeniowo tzn. pojawienie się nowych danych powoduje ich rozesłanie do konsumentów. Taki sposób przesyłania danych jest bardziej wydajny niż stosowane dotychczas architektury klient- serwer takie jak DCOM lub CORBA. Zalety architektury PS ujawniają się zwłaszcza, jeżeli dane są przesyłane w sposób okresowy i nałożone są na nie ograniczenia czasowe.

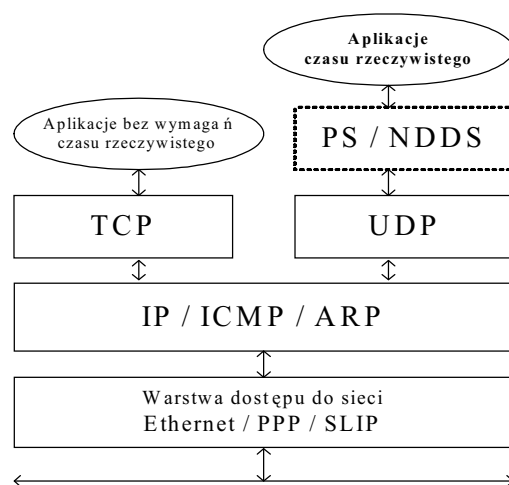


Rys. 3. Architektura typu PS

Architektura ta umożliwia bezpośredni przepływ danych pomiędzy producentami a konsumentami informacji, bez konieczności nawiązywania połączenia, dzięki czemu jest ona bardziej wydajna od architektury „*klient - serwer*”. Nie występują tutaj mechanizmy centralizacji informacji, a jako mechanizm transportowy jest wykorzystywany protokół bezpołączeniowy UDP gwarantujący szybkie dostarczenie informacji. Protokół ten wykorzystuje w warstwie sieciowej, podobnie jak protokół TCP, protokół IP. Takie rozwiązanie nadaje się do przesyłania informacji w systemach z rozproszonymi źródłami informacji i z rozproszonymi odbiorcami informacji, w których występują ograniczenia czasowe. Ze względu na występowanie, na poziomie protokołu UDP, bezpołączeniowych mechanizmów transportowych, ale nie gwarantujących wysokiej niezawodności dostarczenia informacji, rozwiązania te mogą być z sukcesem stosowane w systemach z występowaniem ograniczeń czasowych, ale jedynie w systemach typu „*soft real-time*”. Wzrost zainteresowania tą architekturą w obszarze systemów SCADA wynika ze wzrostu liczby instalowanych tam sieci przemysłowych bazujących na Ethernetie przemysłowym oraz wzrostu rozwiązań bazujących na technologiach internetowych np. dedykowane serwery WWW [4].

## 7. ARCHITEKTURA PS W SYSTEMACH RT

Realizacja wymagań czasu rzeczywistego w architekturze PS wymaga rozszerzenia rozbudowy jej podstawowych funkcji o następujące funkcje: ustalenie okresu generowania wiadomości, możliwość ustawienia ograniczenia czasowego na wiadomości okresowe i sporadyczne, kontrola przepływu wiadomości pomiędzy jej producentem a jej konsumentami, dostępność mechanizmów tolerujących uszkodzenia, możliwość realizacji stempli czasowych, współpraca z systemami RTOS. Implementacja architektury RTPS została zrealizowana w firmie *Real Time Inovations Inc.* i jest dostępna dla wielu systemów operacyjnych czasu rzeczywistego stosowanych zarówno w rozwiązaniach dedykowanych jak i w stacjach roboczych jako protokół NDDS (*Network Data Delivery Service*) [5]. Na rysunku 4 przedstawiono strukturę stosu TCP/IP i umiejscowiono protokół NDDS.



Rysunek 4. Protokół NDDS na tle stosu TCP/IP

W pierwszej, handlowo dostępnej implementacji protokołu NDDS węzeł mógł publikować pojedynczą wiadomość o długości do 1024 bajtów oraz mógł prenumerować do 64 zmiennych o długości do 1024 bajtów. Konsument pozwala na kontrolowanie ograniczenia czasowego oraz minimalnej odległości pomiędzy kolejnymi wiadomościami przychodzącymi od producenta. W modelu RTPS dopuszcza się istnienia dwóch takich samych nazw danych produkowanych przez różnych producentów. Stwarza to możliwości budowy systemu z tolerancją błędów. Pomiedzy producentami danych o tej samej nazwie można utworzyć hierarchię, z której będzie wynikało, który producent będzie generował dane o wyższym , a który z nich będzie producentem rezerwowym i będzie generował dane o niższym priorytecie. Jeżeli do konsumenta dotrą wiadomości o tym samym temacie, to ta o niższym priorytecie zostanie odrzucona.

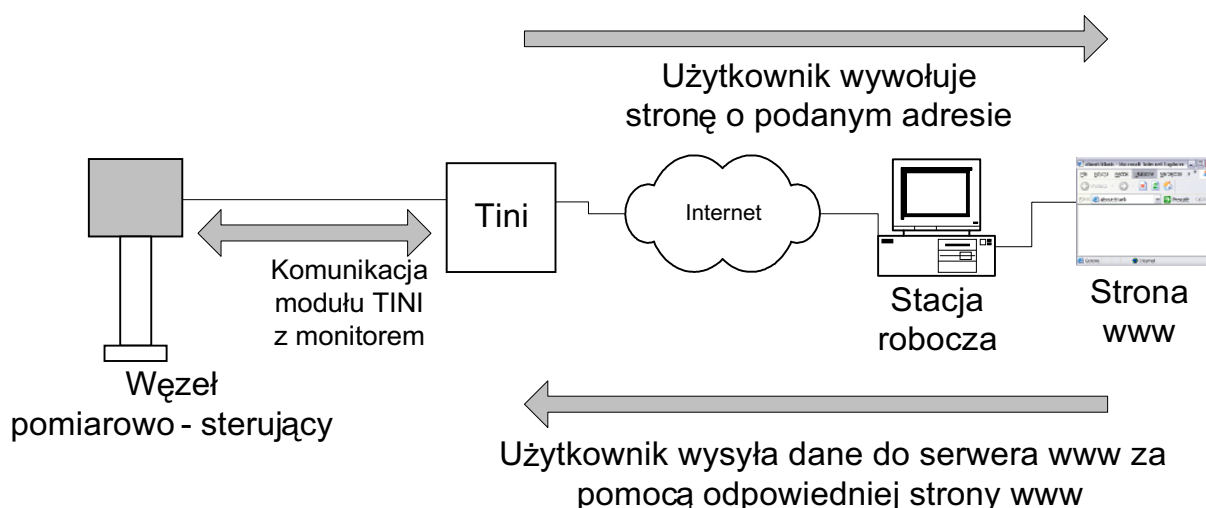
NDDS jest protokołem symetrycznym, nie posiadającym węzła centralnego, serwerowego lub jakiegokolwiek wyróżnionego o prostej konfiguracji. Uszkodzenie jednego z węzłów nie powoduje zawieszenia pracy pozostałej części systemu. NDDS pozwala na dynamiczne konfigurowanie węzłów (dodawanie producentów i konsumentów) w trakcie pracy pozostałych węzłów. Ze względu na niewielką objętość kodu i wykorzystanie protokołu UDP, NDDS jest bardziej efektywny od stosowanych na tym poziomie protokołowych rozwiązań takich jak: DCOM, OPC i CORBA.



## 8. OPROGRAMOWANIE APLIKACYJNE MODUŁU TINI400

Wykorzystanie modułu TINI400 wymaga zbudowania i uruchomienia kilku modułów programowych pracujących w sterowniku TINI400. Ponieważ sterownik ten musi komunikować się z monitorem promieniowania, wymagane jest zbudowanie modułu programowego realizującego transmisję danych poprzez port szeregowy sterownika bramki dozymetrycznej. Dedykowany serwer TINI400 może być podłączony do sieci Internet poprzez port Ethernet 10/100 lub poprzez port szeregowy z protokołem PPP. W serwerze TINI400 zaimplementowana jest obsługa stosu protokolowego TCP/IP, który wymaga skonfigurowania w miejscu zainstalowania bramki dozymetrycznej. Funkcje umożliwiające zdalne monitorowanie lub konfigurowanie realizuje moduł programowy dedykowanego serwera WWW, który komunikuje się z bramką za pomocą protokołu SC-755 lub S.C.-501 i udostępnia dane o wskazaniach monitora w sieci Internet za pomocą interfejsu Ethernet lub poprzez RS-232.

Jeżeli wyjście do sieci Internet następuje poprzez port szeregowy z wykorzystaniem modemu do sieci telefonii stacjonarnej lub komórkowej, to po stronie modułu TINI należy uruchomić funkcje klienta lub serwera protokołu PPP. Komunikacja z monitorem promieniowania poprzez serwer TINI400 umożliwia realizację dwóch rodzajów operacji. Możliwe jest przeglądanie stron statycznych umieszczonych na serwerze WWW za pomocą przeglądarki internetowej (np. Internet Explorer) lub wysyłanie wybranych danych za pomocą specjalnych stron do serwera WWW. Serwer WWW komunikuje się z bramką dozymetryczną i może przysyłać do niej odpowiednie komendy na zlecenie wysłane przez użytkownika w sposób zdalny (rysunek 5).



Rysunek 5. Komunikacja z monitorem za pomocą przeglądarki WWW

Strony WWW zapamiętane są na serwerze w postaci plików tekstowych html. Istnieje możliwość umieszczania plików graficznych, jednak z uwagi na ograniczoną wielkość pamięci serwera (1 MB) strony powinny być skonstruowane w taki sposób, żeby zajmowały jak najmniej miejsca w pamięci.

### 8.1 Charakterystyka modułów programowych dedykowanego serwera WWW TINI

Dedykowany Server WWW TINI wymaga zainstalowania odpowiednich modułów programowych. Do jego poprawnej pracy wymagane jest uprzednie zainstalowanie oprogramowania serwera – systemu operacyjnego **Slush**. Jego wersja musi być zgodna z wersją, dla której kompilowany będzie program. Opis wymaganego oprogramowania znajduje się w kolejnych rozdziałach.

Oprogramowanie komunikacji ze sterownikiem składa się z kilku plików. Składają się one na dwie aplikacje. Aplikację komunikacji, która jest uruchamiana w serwerze TINI po jego starcie – *TiniWeb.tini*. Drugi program to tzw. aplet języka Javy służący do komunikacji użytkownika z serwerem za pomocą przeglądarki WWW – *relAppler.jar*. Dodatkowo aplet może być skompilowany do pliku exe, który umożliwi uruchamianie aplikacji bez udziału przeglądarki. Pliki źródłowe aplikacji i apletu znajdują się na dołączonej płycie CD w katalogu **source**. Do kompilacji programów wymagane jest dedykowane oprogramowanie, które przedstawiono w kolejnym punkcie.

## 8.2 Uruchamianie modułu TINI

### a. Wymagane oprogramowanie

Uruchomienie modułu TINI wymaga uprzedniego zainstalowania dedykowanego oprogramowania. Do uruchomienia modułu należy zainstalować następujące oprogramowanie:

- Tini Software Development Kit
- JDK Java Development Kit
- Java Communications API

W pierwszym kroku należy zainstalować JDK, następnie TiniSDK, a na końcu Communication API. Należy postępować zgodnie z poleceniami zawartymi w opisie instalacji. Bardzo ważne jest ujednoczenie ścieżek dostępu i poprawne ich wpisanie.

### b. Uruchomienie modułu

Moduł TINI wymaga wstępnego skonfigurowania. Zwłaszcza chodzi o ustalenie odpowiedniego adresu IP i ewentualne wgranie odpowiedniej wersji oprogramowania systemowego. W pierwszej chwili adres IP nie jest ustalony i nie można się połączyć z modułem przez sieć Ethernet. Możliwe jest połączenie przez terminal JavaKit, który uruchamia się zgodnie z dokumentem **app note 612**.

**Uwaga:** ważne jest użycie odpowiedniego przewodu RS-232. Powinien to być przewód tzw. jeden do jeden. Przy pomocy programu JavaKit należy połączyć się z modułem TINI, który musi być dołączony do portu szeregowego PC.

## 8.3 Konfiguracja modułu TINI

Ustawianie konfiguracji modułu obejmuje ustalenie jego adresu IP, maski podsieci i bramy internetowej. Po połączeniu się za pomocą JavaKit można wydać polecenie **ipconfig** z klawiatury i ustawić odpowiedni adres IP. Wartość adresu logicznego IP zależy od miejsca, w którym moduł TINI jest instalowany. Moduł serwera TINI jest jednym z hostów w segmencie sieci komputerowej, do której jest podłączany. IP może być pozyskiwane dynamicznie poprzez DHCP lub statycznie z poziomu konsoli. Modułowi serwera TINI można przypisać nazwę domenową, którą wcześniej należy uzgodnić z dostawcą usług internetowych. Wykorzystywanie nazwy domenowej wiąże się z koniecznością dysponowania adresem publicznym przypisanym do tej nazwy.

**Uwaga:** wartości należy wpisać do pamięci **flash** za pomocą przełącznika **ipconfig -C**. W przeciwnym przypadku po ponownym włączeniu modułu ustalone wartości zginą. Kolejnym ustawieniem mającym wpływ na pracę modułu jest zawartość pliku startowego **startup** w katalogu **/etc**. Jego zawartość może wyglądać następująco:

```
#####  
#Autogen'd slush startup file  
setenv FTPServer enable  
setenv TelnetServer enable  
# rem  
# setenv SerialServer enable
```

```

##
#Add user calls to setenv here:
##
initializeNetwork
#####
#Add other user additions here:
java TiniWeb.tini &

```

Znaczniki # symbolizują komentarze w pliku. Linia rozpoczynająca się takim znakiem nie zostanie uruchomiona.

#### 8.4 Kompilacja oprogramowania komunikacji ze sterownikami

Oprogramowanie komunikacyjne może być kompilowane za pomocą kompilatora JDK lub innymi kompilatorami jak JBuilder firmy Borland. Kompilacja za pomocą JDK przebiega w sposób wsadowy za pomocą odpowiedniego polecenia. Do plików źródłowych umieszczonych na płycie CD dołączone są odpowiednie pliki wsadowe, które pozwalają na kompilowanie programów w sposób wsadowy. W plikach należy zmodyfikować ścieżki do kompilatora Javy i SDK Tini. Plik wsadowy **build** w katalogu **source\tini** tworzy program TiniWeb.tini, który jest umieszczany w katalogu głównym serwera TINI.

#### 8.5 Kompilacja apletu przeglądarki

Plik wsadowy **build** w katalogu **source\applet** tworzy plik **relApplet.jar**, który jest również umieszczany w katalogu głównym Tini. Istnieje możliwość kompilacji modułu **relApplet.jar** do postaci exe. Jednak nie można wykonać tego za pomocą plików wsadowych. Taki plik powstaje po użyciu wizarda Archive Builder w programie JBuilderX firmy Borland.

#### 8.6 Konfiguracja i ładowanie stron WWW do modułu TINI

Serwer WWW TINI wymaga skonfigurowania tzw. strony domyślnej. W katalogu głównym Tini znajduje się plik **index.html** i spakowany Applet Javy **relApplet.jar**. Plik **relApplet** powstaje w wyniku kompilacji programu źródłowego. Plik index.html jest statycznym plikiem tekstowym. Jego zawartość określa jak będzie wyglądała strona WWW załadowana do przeglądarki WWW, po podaniu adresu internetowego serwera. W celu zmiany wyglądu tej strony, można użyć dowolnego edytora tekstu (np. notatnika). Możliwe jest umieszczanie większej liczny plików i linków pomiędzy nimi jak również grafiki. Należy mieć jednak na uwadze to, iż zasoby serwera WWW są ograniczone do 1 MB! W przypadku pozostawienia małej ilości miejsca na serwerze WWW może on przestać działać, ponieważ pamięć jest używana jednocześnie do przechowywania plików i uruchamiania programów. Plik **index.html** jak i pozostałe pliki systemu (**TiniWeb.tini** i **relApplet.jar**) można umieścić na serwerze za pomocą programu **ftp**. Należy połączyć się z serwerem używając odpowiedniego adresu sieciowego. System zapyta się o konto i hasło. Domyślnie jest to **root** i **tini**.

### 9 INTERNET - 0

Potrzeby użytkowników w obszarze szkieletowych sieci teleinformatycznych wymuszają jego rozwój w kierunku Internetu 2 tzw. Internetu optycznego o bardzo dużych przepustowościach. Natomiast potrzeby wielu obiektów, na których instalowane są węzły pomiarowo – sterujące nie są aż tak wygórowane i wystarczyłyby dużo mniejsze przepustowości i niezbyt rozbudowany stos komunikacyjny, który mógłby być obsługiwany przez proste mikrokomputer klasy 8051. Zasady dostępu do Internetu nie zawierają szczegółowych wymagań dotyczących wydajności komunikacyjnej podłączanych urządzeń a postęp w dziedzinie sprzętu i oprogramowania nie wymusza zmiany w podstawowej architekturze Internetu. Stanowi to dogodną okoliczność do wykorzystania tej technologii

w obszarze systemów pomiarowo – sterujących a rzeczywiste potrzeby SPS doprowadziły do powstania inicjatywy Internetu 0 (I0) (tzw. Internetu urzędzeń). Celem I0 jest usunięcie ograniczeń wynikających ze złożoności sieci, a nie proste zwiększenie wydajności. Nie ma on zastąpić obecnego Internetu, ale raczej stworzyć pewien obszar kompatybilności będący niżej w hierarchii. Dla I0 przyjęto siedem złotych zasad, które zostaną w dalszej części krótko scharakteryzowane.

Zasada pierwsza. Przyjęto, że każde urządzenie I0 będzie korzystało z protokołu IP, który zostanie ograniczony do pojedynczych kilobajtów. Umożliwi to jego obsługę nawet przez proste procesory i uniknie się konieczności konwersji ramek protokołu sieci przemysłowej na pakiety IP. Protokół IP zwiększa objętość przesyłanych ramek o około 100 bitów, co nie ma znaczącego wpływu na czas odpowiedzi i ilość zużywanej energii.

Zasada druga. Uproszczono wielopoziomowe oprogramowanie komunikacyjne, rezygnując z rozdzielnej implementacji protokołów komunikacyjnych poszczególnych warstw. W urządzeniu I0 rezygnujemy z podziału na poziomy, w pełni wykorzystując znajomość konkretnej aplikacji.

Zasada trzecia. Poza architekturą "Klient-Serwer" wprowadza się architekturę "Peer-to-Peer", co oznacza, że dwa urządzenia I0 nie wymagają kolejnego, aby się ze sobą komunikować. W klasycznym Internecie klient bez serwera jest bezużyteczny.

Zasada czwarta. Urządzenie I0 w pełni identyfikuje obiekt fizyczny, do którego należy i potrafi rozpoznawać nazwy domenowe, które byłyby nadawane automatycznie przez urządzenia w sposób losowy. Powiązanie nazwy funkcjonalnej i sprzętowej będzie następowało podczas konfiguracji systemu.

Zasada piąta. Urządzenia I0 powinny posługiwać się bitami o rozmiarach większych niż sieć. Bit, który jest najczęściej impulsem elektrycznym, radiowym lub świetlnym, ma pewien rozmiar, wynikający z pomnożenia czasu trwania impulsu przez prędkość, z jaką się on rozchodzi. Jeżeli szybkość transmisji jest zbliżona np. do 1 Mbit/sek, to bit rozciąga się on na ok. 300 m, a więc obejmuje dużą część systemu i wówczas rodzaj nośników nie ma dużego znaczenia.

Zasada szósta. Jeżeli bity są duże, pakiety z danymi można formować tak samo dla wszystkich ośrodków fizycznych, w których mają się rozchodzić. Reprezentacja fizyczna małych bitów (ich modulacja), musi być zoptymalizowana pod kątem rodzaju transmisji. Gdy bity są duże, charakterystyki przenoszenia nie są istotne a różne kanały komunikacyjne za każdym razem przenoszą te same dane, z tą samą szybkością, wykorzystując ten sam schemat kodowania. Nie trzeba stosować żadnej translacji.

Zasada siódma. W ramach I0 należy posługiwać się standardami otwartymi.

## 7. PODSUMOWANIE

W referacie przedstawiono strukturę SPS, zbudowanych w oparciu o dedykowane serwery WWW TINi bazujące na mikrokomputerze DS80C400 i stosie protokołowym TCP/IP. Przedstawiono sposób konfigurowania i programowania serwera WWW z programowo realizowanym stosie TCP/IP, zbudowanego na bazie modułu TINi. Przedstawiony w artykule nowy kierunek rozwoju RSPS wykorzystujący technologie internetowe może otworzyć nowe możliwości wykorzystania inteligentnej aparatury pomiarowo – sterującej dzięki wykorzystaniu standardowego interfejsu użytkownika i powszechnie dostępnej internetowej technologii komunikacyjnej.

## LITERATURA

- [1] Automation Research Corporation: Device & Field Network. Market Studies, 1999.
- [2] Castellote G.P., Thiebaut S., Hamilton M., Choi H.: Real-Time Publish-Subscribe Protocol for IP-Based Real-Time Communication. Instrument Society of America, 2001.
- [3] Hirschman Network Systems: Distributed Communication Architecture, 1998.
- [4] Lee K. B.: Schneeman R. D.: Internet-Based Distributed Measurement System and Control Application. IEEE Instr. & Measurement Magazine, June 1999, s. 23-27.
- [5] Lutz A. T.: Using TCP/IP as an Instrument Interface. Sensors Magazine, July, 1998.

- [6] Michta E.: Modele komunikacyjne sieciowych systemów pomiarowo – sterujących. Monografia 99. Politechnika Zielonogórska, Zielona Góra, 2000.
- [7] Michta E.: Ethernet przemysłowy w systemach pomiarowo-sterujących o architekturze sieciowej. Metrologia Wspomagana Komputerowo - MWK 2001: V Szkoła - Konferencja. Rynia k/Warszawy, 2001.- T. 2: Referaty, s. 263–268.
- [8] Michta E.: Technologie internetowe w sieciowych systemach pomiarowo-sterujących. Mezura - Auromecon 2001 : Konferencja. Poznań, 2001, s. 131-138.
- [9] Michta E., Szulim R.: Wbudowane serwery WWW. Studia Informatica. Vol.2, nr 49, Politechnika Śląska, Gliwice, 2002, s. 121-132.
- [10] Oertel J. H.: CAN server for TCP/IP. CAN Newsletter, March 2002, pp. 60-64.
- [11] Pfeiffer O.: Embedded Internetworking with 8- and 16-bit Microcontrollers. Embedded Systems Conference East, Chicago, 2001, s. 421-431.
- [12] Rytting T.: Internet Enables Existing Embedded Process Control Devices. RTC Magazine, Oct. 2000, s. 58-64.