Raport Badawczy
Research Report

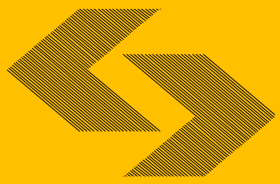
RB/54/2010

Mikrostosy TCP/IP w zastosowaniach

P. Biegus

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



MIKROSTOSY TCP/IP W ZASTOSOWANIACH

Piotr Biegus
Studia Doktoranckie IBS PAN

Connecting different devices to the Internet is more and more popular today. In a past it was quite expensive to connect device to the internet considering comparison between a price of connected devices and a necessary interface itself. Today the price of necessary interfaces is quite reasonable and enable connection of even ordinary devices like for example central heating.
It could be more interesting because house-automation subject is very hot topic today and prices of different types of energy are growing and will grow in the future. It means remote monitoring can became an important issue in the near future. I am trying to compare two different solution of the network interfaces.

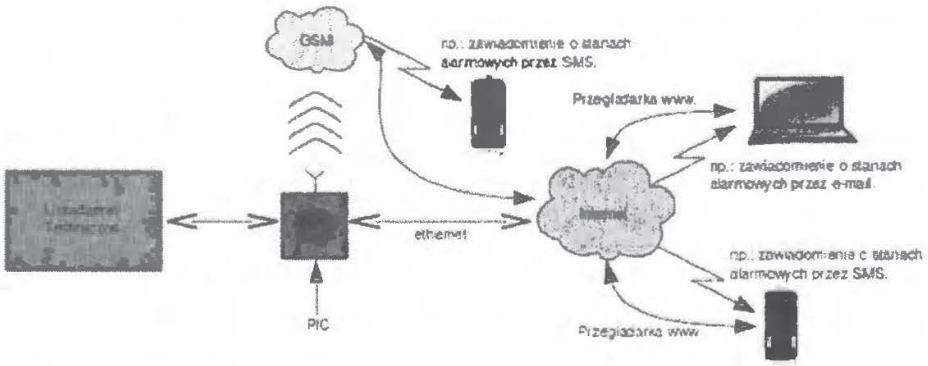
Wstęp

Dawniej tylko komputery typu „workstation" i duże serwery miały możliwość komunikacji za pomocą protokołu TCP/IP. Obecnie dzięki ciągłej miniaturyzacji, zwiększeniu integracji układów scalonych, jak również obniżce cen, możliwe jest dołączenie mikrosystemu mieszczącego się w jednej kości do sieci TCP/IP. Daje to zupełnie nowe możliwości i rzuca nowe światło na rozległą sieć Internet. Możliwe jest bardzo tanie podłączenie praktycznie każdego urządzenia technicznego do sieci Internet, takiego jak np. piece CO, klimatyzatory, oświetlenie, itp. Daje to nowe możliwości dla rożnego rodzaju projektów (np. bardzo modnego obecnie „inteligentnego budynku"). Kilka lat temu połączenie dowolnego urządzenia technicznego było zwykle realizowane za pomocą modułów typu „embedded Linux" (małe systemy mikroprocesorowe, na których można było uruchomić Linux'a). Jednak tego typu rozwiązania są, w mojej ocenie, mimo wszystko drogie. Mogło by się wydawać, że jest inaczej, bo jak wszyscy twierdzą „Linux jest za darmo". Jednak z ekonomicznego punktu widzenia jest to drogie rozwiązanie, dlatego że wszystkie dotychczas produkowane chipsety „SOC" (mogące unieść Linuxa) potrzebują dodatkowej kości pamięci DRAM oraz FLASH, co z kolei pociąga za sobą konieczność zastosowania czterowarstwowego PCB. Koszt wykonania czterowarstwowego PCB jest raczej wysoki, jeśli produkcja nie jest ustalona na poziomie setek tysięcy sztuk.

Biorąc pod uwagę koszty modułu z embedded Linuxem są raczej drogie. Dodatkowo kość DRAM i Flash, szczególnie DRAM powoduje znaczny przyrost poboru prądu. Zasilanie systemu z akumulatora lub baterii staje się bardziej kłopotliwe. Reasumując dzięki Linux'owi zyskujemy możliwość skorzystania z całego dorobku GNU, ale jest to okupione dużo wyższą ceną sprzętu.

Tam gdzie cena gra rolę pierwszorzędną, jak również pobór energii, bardzo ciekawym rozwiązaniem jest użycie modułu opartego o chipsety firmy Microchip. Mam tu na myśli całą serię układów PIC18F66-97J60-65 oraz układ ENC 28J60, który może być wykorzystany jako bardzo prosty sprzęg pomiędzy siecią Ethernet a istniejącym systemem mikroprocesorowym wyposażonym w wyjście SPI.

W układach serii PIC18F66-97J60-65 w jednej kości mieści się cały system mikroprocesorowy oraz ETH-PHY. Maksymalny rozmiar wewnętrznej pamięci Flash to 128KB. Nie jest to wartość duża, ale wystarczająca. Ilość pamięci SRAM to 3808 bajtów. Czasami mogą pojawić się problemy, bo przy większej ilości aplikacji działających równocześnie może brakować pamięci. Mikrosystem zbudowany w oparciu o chip-set pobiera tylko około 0,315W. Jest to wartość nie do osiągnięcia dla systemów, na których może działać embedded Linux. Pozwala to na zasilanie systemu z baterii, akumulatora lub nawet małego ogniwa solarnego, co ma duże znaczenie marketingowe.



Rysunek 1. Przykładowe zastosowanie

Powstaje coraz więcej projektów stosów TCP dla małych systemów mikroprocesorowych. Można te projekty znaleźć w sieci. Także producent „Microchip" oferuje bazę software'owa w postaci stosu TCP/IP, jak również

kilka przykładowych aplikacji, z których najważniejszą jest mini-HTTP-server. Stanowi to dobrą bazę deweloperską dla własnych projektów.

1. Mikrostosy TCP/IP

1.1 Protokoły obsługiwane przez stos firmy „Microchip"

a) Warstwa Link:

- ETHERNET,
- ARP.

b) Warstwa Internet:

- ICMP (Uwaga tylko „ping-pong„),
- IP (Ipv4).

c) Warstwa Transportowa:

- TCP (limitowana liczba otwartych gniazd),
- UDP (limitowana liczba otwartych gniazd).

d) Warstwa Aplikacji:

- DNS,
- HTTP,
- TFTP,
- SMTP,
- SNTP,
- SNMP,
- TELNET.

1.2 Budowa stosu TCP/IP dla małych systemów mikroprocesorowych

Dzisiaj trudno sobie wyobrazić wielu osobom napisanie programu, jeśli nie mamy do dyspozycji procesora z zegarem co najmniej kilkaset megaherców i wielu megabajtów pamięci RAM. Całkiem poważne projekty można realizo-

wać w oparciu o małe procesory (np. PIC18F97J65 wykonuje maksymalnie 10 mln operacji na sekundę, a niektóre operacje potrzebują więcej niż jeden cykl maszynowy). Cykl maszynowy trwa cztery cykle zegarowe, czyli dla zegara 41,666 Mhz mamy 10 416 500 cykli maszynowych na sekundę.

Cały mikrostos napisany jest w języku C, tylko niewielka cześć kodu napisana jest w asemblerze. Prymitywna jednostka centralna CPU nie obsługuje wielozadaniowości i posiada bardzo mało miejsca na stos (np. nie można stosować rekurencyjnego wywołania procedur). Brak obsługi wielozadaniowości wymusza na programiście rygorystyczny styl pisania.

Należy zauważyć, ze obsługa pakietów UDP jest raczej prosta, natomiast obsługa pakietów TCP jest dużo bardziej skomplikowana, ponieważ trzeba odwzorować cala maszynę stanów, która jest niezbędna przy obsłudze protokółu TCP (RFC 793)

W. Richard Stevens fig.: 18.12, 18.13 oraz obsługę timerów:

- TCP Timeout and Retransmition,
- TCP Persist Timer,
- TCP Keepalive Timer.

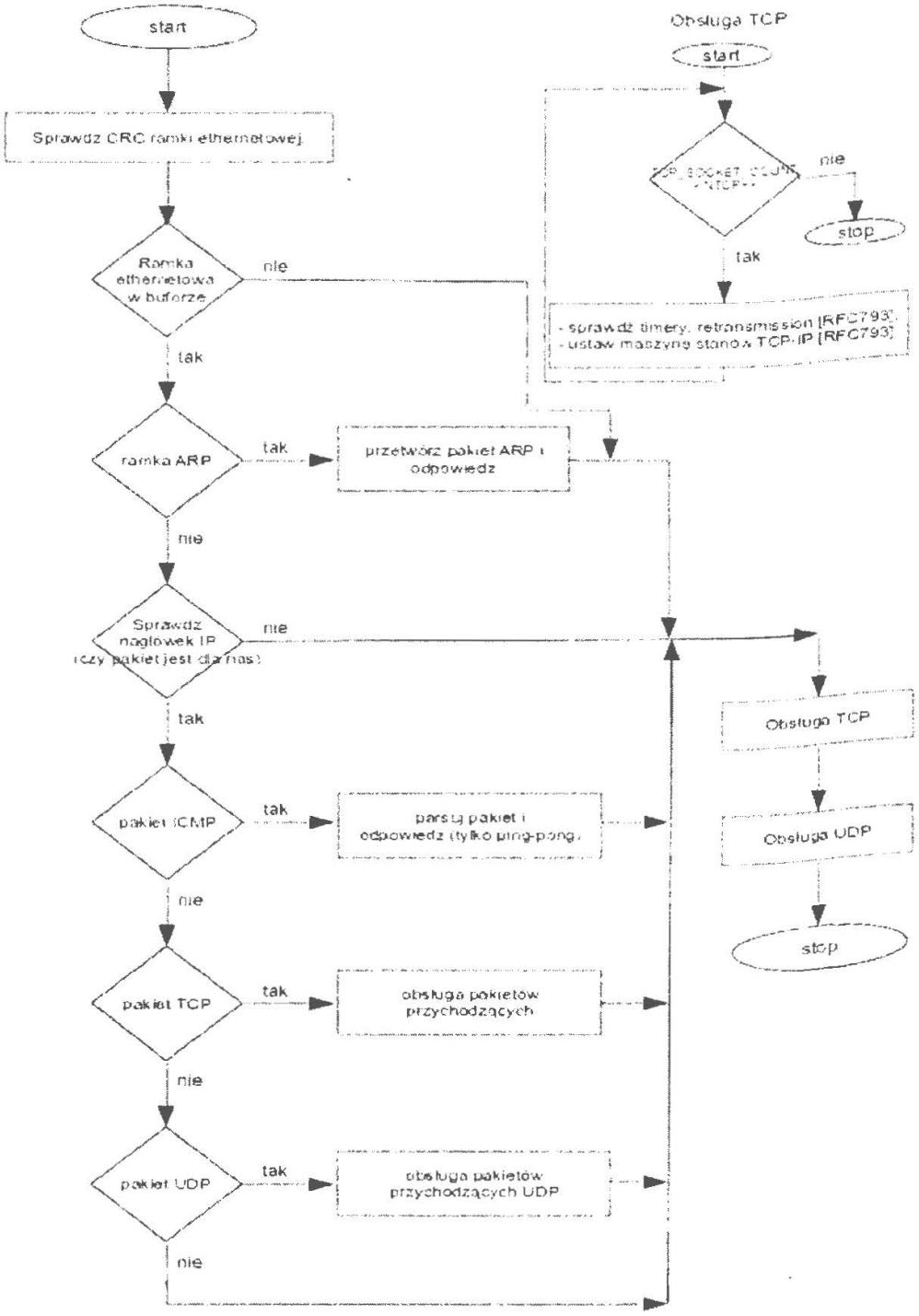
Wymagany jest reżim czasowy jak dla każdego systemu czasu rzeczywistego dlatego istnieje potrzeba implementacji wewnętrznego timera, który odlicza liczbę „ticków", aby moc zrealizować timery wymagane przez protokół TCP. Ze względu na limitowaną ilość pamięci SRAM można otworzyć maksymalnie 5 gniazd (socket) TCP.

Obsługa pakietów ICMP jest bardzo uboga, obsługiwany jest tylko ping-pong (ICMP pakiet numer 8).

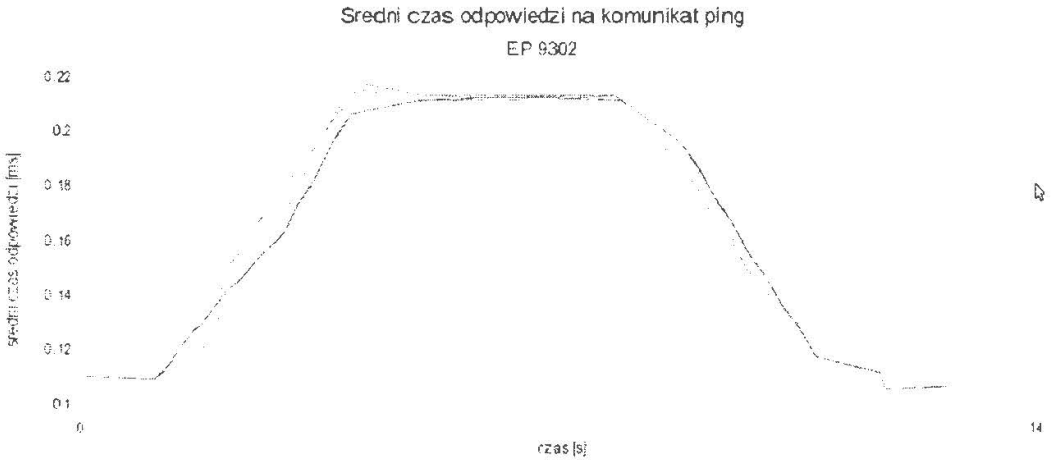
Porównanie dwóch systemów mikroprocesorowych, na bazie których można zbudować interface umożliwiający połączenie dowolnego urządzenia do sieci Ethernet:

- pierwszy oparty o mikrokontroler Microchip PIC18F97J65,
- drugi oparty o mikrokontroler Cirrus Logic EP 9302.

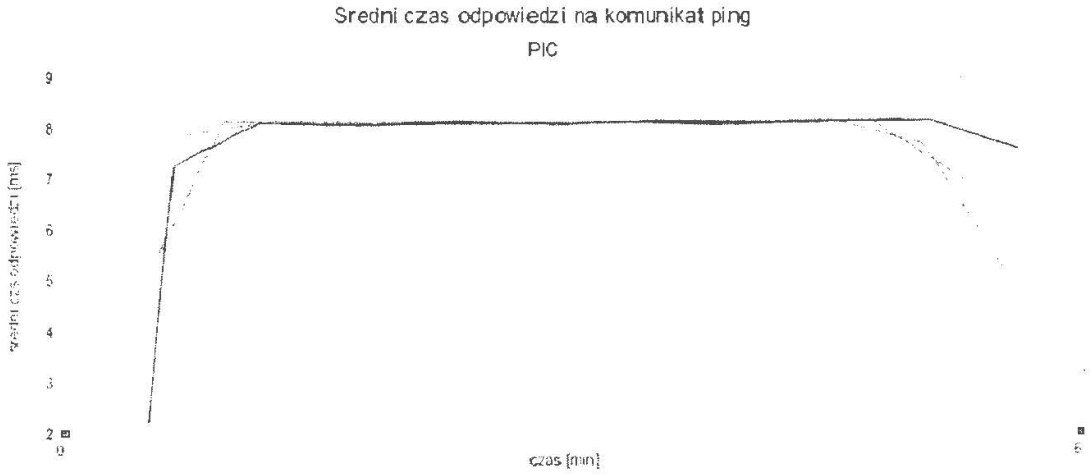
Rysunki 3 i 4 przedstawiają średni czas odpowiedzi systemu na komunikaty ping. Średnia wyciągnięta została dla 5000 komunikatów przesłanych do systemu przez 5 procesów z odstępem uruchomienia.



Rysunek 2. Uproszczony algorytm działania mikrostosu



Rysunek 3. Średnie czasy odpowiedzi na komunikaty ping dla systemu EP 9302



Rysunek 4. Średnie czasy odpowiedzi na komunikaty ping dla systemu PIC18F97J65

Poniższa tabela przedstawia porównanie funkcjonalności obu systemów.

	PIC18F97J65	EP 9302
Transfer danych (HTTP) [bajty]:		
•3533	31 [ms]	15 [ms]
•4942	42 [ms]	19 [ms]
•5652	47 [ms]	22 [ms]
•7066	59 [ms]	28 [ms]
Średni pobór mocy.	0,315 [W]	2 [W]
Orientacyjna cena systemu.	Około 150zl	Około 400zl
Dostępne interfejsy elektryczne:		
•UART	tak	tak
•GPIO	tak	tak
•A/D	tak	tak
•PWM	tak	tak
•SPI	tak	tak
Częstotliwość taktowania procesora.	41,66 [Mhz]	200 [Mhz]
Minimalna liczba obwodów scalonych potrzebna do zbudowania w pełni funkcjonalnego systemu.	1	4 (Microcontroler + DARM + FLASH + ETH_PHY)

Tabela 1. Porównanie systemów zbudowanych na procesorach: PIC18F97J65 i ARM 9 (EP 9302)

Słowniczek:

•UART – interfejs szeregowy.

•GPIO – General Purpose Input/Output – porty wejścia wyjścia ogólnego przeznaczenia [http://en.wikipedia.org/wiki/GPIO],

•A/D – przetworniki analogowo cyfrowe [http://en.wikipedia.org/wiki/A/D].

•PWM – pulse with modulation – modulacja szerokości impulsu [http://en.wikipedia.org/wiki/Pulse-width_modulation].

•SPI – Serial Periferial Interface. Magistrala szeregowa do komunikacji różnych komponentów mikroprocesorowych. Mikrokontroler może komunikować się np. z kartą SD, modułem GSM. [http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus]

Literatura

[1] W. Richard Stevens (1994): *TCP/IP Illustrated, Vol. 1: The Protocols*, Addison-Wesley Publishing Company, Inc.

[2] RFC 793: Transmission Control Protocol, *The Internet Engineering Task Force*, 1981 (http://www.ietf.org/rfc/rfc0793.txt).