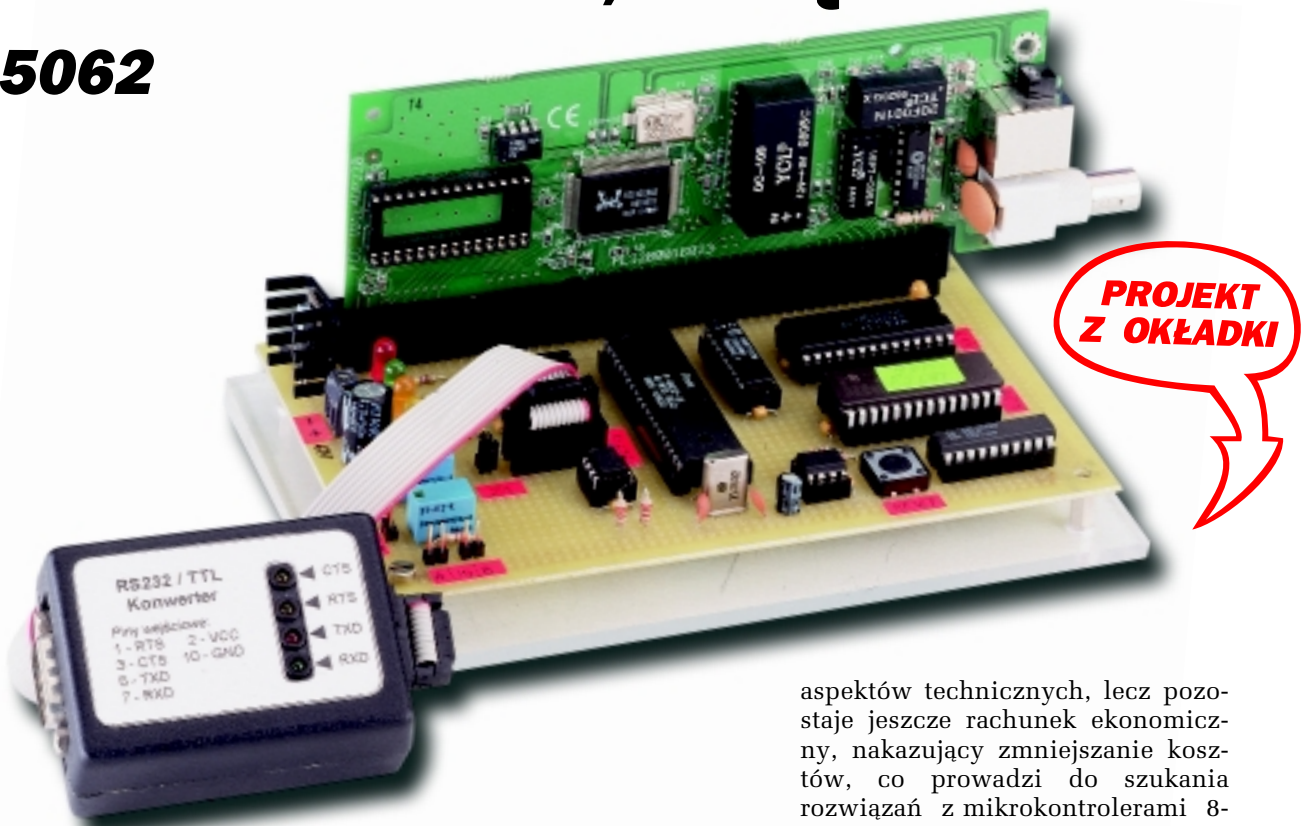


System NET51, część 1

AVT-5062



Sprzętowo zrealizowany stos TCP/IP przedstawiliśmy w projekcie okładowym w marcu tego roku. Do tematu wracamy - tym razem prezentując przykład programowej realizacji takiego stosu komunikacyjnego. Jest to znany internetowym fanom interfejs NET51, który w ostatnich dniach marca wzbudził olbrzymie emocje na elektronicznych listach „newsowych“.

W ciągu ostatnich lat obserwuje się trend podłączania różnorodnych urządzeń do Internetu. Praktycznie każdy z liczących się producentów mikroprocesorów przedstawił już swoje autorskie rozwiązanie. Powstały także niezależne firmy, które lansują rozwiązania zarówno sprzętowe jak i programowe w tej dziedzinie. Niektóre z tych rozwiązań były opisywane na łamach Elektroniki Praktycznej w dziale poświęconym nowościom sprzętowym. Przykładowo można przytoczyć opis opracowania modułu z układem *iChip* (EP3 i 4/02).

Z używaniem Internetu do wymiany informacji jest związanych wiele korzyści, z których najważniejszą jest globalny zasięg, przy względnie niskim koszcie połączenia. Do niedawna zastosowanie łączności przy użyciu protokołów TCP/IP wymagało wykorzystania komputerów z procesorami 16-sto lub 32-bitowymi, co dla docelowej aplikacji oznacza duży koszt części sprzętowej, rozbudowane płytki z obwodami drukowanymi oraz znaczny pobór energii.

Postęp w technologii w elektronice spowodował rozwiązanie większości problemów dotyczących

aspektów technicznych, lecz pozostaje jeszcze rachunek ekonomiczny, nakazujący zmniejszanie kosztów, co prowadzi do szukania rozwiązań z mikrokontrolerami 8-bitowymi. W przypadku systemów mikroprocesorowych udostępnienie możliwości pracy w sieci najczęściej odbywa się na trzy sposoby:

1. Wydzielony komputer (np. PC) ma połączenie do Internetu i pełni funkcję bramy internetowej dla pozostałych urządzeń, które są do niego podłączone (poprzez RS232 lub RS485).

2. Wykorzystanie dedykowanego układu lub modułu do łączności internetowej, który dla współpracującego mikrokontrolera jest elementem peryferyjnym.

3. Bezpośrednia implementacja stosu TCP/IP razem z oprogramowaniem aplikacyjnym mikrokontrolera.

Jakiś czas temu, obserwując w Internecie różnorodne próby implementacji stosu TCP/IP na „małych“ mikrokontrolerach, postanowiłem też spróbować swoich sił w tej dziedzinie i tak powstał system NET51 (należący do kategorii 2 lub 3), który opisuję w tym artykule.

Trochę teorii

Stos TCP/IP składa się z szeregu protokołów umieszczonych w różnych warstwach. W uproszczeniu system TCP/IP przedstawiany jest na czterech warstwach

Warstwy stosu TCP/IP	Protokoły
Warstwa aplikacji	HTTP, FTP, TELNET
Warstwa transportowa	TCP, UDP
Warstwa sieciowa	IP
Warstwa łącza	Ethernet

Rys. 1. Warstwy stosu TCP/IP

(rys. 1), chociaż model OSI przewiduje siedem warstw (pozostałe warstwy są używane w celu zwiększenia poziomu abstrakcji danych, czyli inaczej mówiąc ułatwienia obsługi systemu sieciowego).

Najwyżej jest umieszczona warstwa aplikacji, na której są realizowane protokoły obsługujące usługi takie jak HTTP, FTP, TELNET itd. Warstwa ta zajmuje się wyłącznie wymianą oraz interpretacją informacji w sposób zgodny z wymaganiami danej aplikacji. Warstwy leżące poniżej nie zawierają informacji o cechach współpracującej aplikacji, gdyż ich zadaniem jest tylko zapewnienie usług poprawnej komunikacji. W warstwie transportowej umieszczony jest protokół TCP, który odpowiada za niezawodne dostarczanie danych. Zazwyczaj TCP dzieli strumień danych z aplikacji na mniejsze fragmenty zwane segmentami, które następnie wysyła do warstwy sieciowej. Na drugim końcu połączenia odbiornik TCP dba o retransmisję zagubionych segmentów, dokonuje potwierżeń i ustala poprawną kolejność odebranych segmentów, a następnie przekazuje dane do warstwy aplikacji. Takie działanie TCP zwalnia aplikację od zajmowania się szczegółami transmisji.

Zdecydowanie prostszym protokołem warstwy transportowej jest UDP, lecz ta usługa warstwy transportowej nie gwarantuje dostarczenia pakietów danych (zwanymi datagramami) do odbiorcy. Zadaniem warstwy sieci, inaczej nazywanej warstwą internetową, jest obsługa ruchu pakietów w sieci. Między innymi, warstwa ta świadczy usługę określania właściwej trasy do odbiorcy pakietu (*IP routing*). W tej części stosu działają protokoły IP oraz ICMP. Ostatnią warstwą jest warstwa łącza danych zajmująca się wszystkimi szczegółami związanymi z interfejsem sprzętowym (sterowniki)

oraz połączeniem fizycznym (Ethernet lub modem).

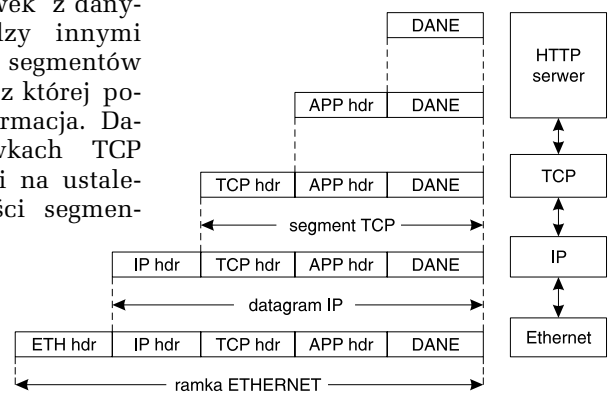
Pomiędzy warstwami są przekazywane informacje w formie ramek składających się z pola danych oraz nagłówek o ściśle określonym formacie. Informacja o budowie nagłówka umożliwia odbiorcy wydzielenie właściwego pola danych. Formaty nagłówków poszczególnych protokołów są zdefiniowane w ogólnie dostępnych w Internecie dokumentach RFC (np. IP - RFC791, TCP - RFC793, UDP - RFC768). Dowolna warstwa może komunikować się tylko ze swoimi sąsiadami za pomocą wywołań funkcji interfejsowych udostępnianych przez poszczególne elementy składowe stosu. Za ich pomocą dane są przekazywane w dół stosu, aż osiągną formę strumienia bitów w interfejsie fizycznym. Każda warstwa dodaje własne informacje do otrzymanych danych, a następnie przesyła je do warstwy niższej. Taki proces jest określany terminem pochodzącym z języka angielskiego jako enkapsulacja (bardziej po polsku - kapsułkowanie).

W celu przybliżenia czytelnikom omawianego zagadnienia posłużyć się rys. 2, przedstawiającym proces przepływu danych przez stos TCP/IP. Przykładowo, na żądanie klienta serwer HTTP wysyła do warstwy transportowej DANE, które mogą stanowić plik graficzny lub tekstowy. Zanim to nastąpi, procedury serwera dołączają do bieżącej informacji nagłówki aplikacji, zawiera informację określającą rodzaj pliku (tekst/obraz/aplikacja) oraz jego rozmiar. Następnie moduł TCP, jeżeli zachodzi taka potrzeba, dzieli otrzymaną informację na segmenty i do każdego dołącza własny nagłówek z danymi dotyczącymi między innymi kolejności przesyłania segmentów oraz rodzaju aplikacji, z której pochodzi przesyłana informacja. Dane zawarte w nagłówkach TCP pozwalają odbiornikowi na ustalenie właściwej kolejności segmentów, gdyż z uwagi na charakter przepływu danych w Internecie możliwe jest, że kolejno nadane segmenty zostaną odebrane w innym porządku. Na

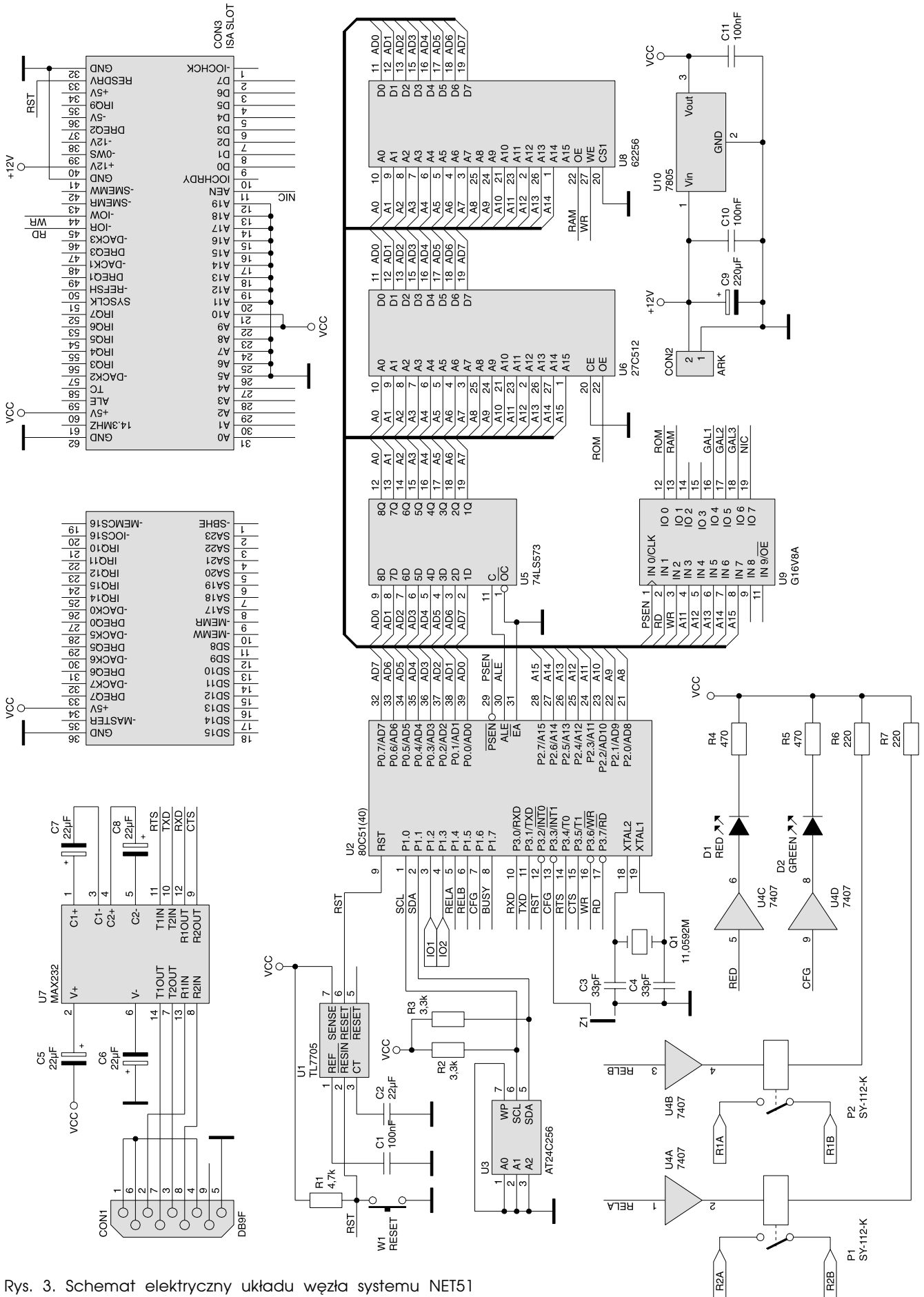
swej dalszej drodze segmenty TCP trafiają do warstwy sieciowej, gdzie protokół IP dodaje własny nagłówek z adresami IP nadawcy i odbiorcy oraz informację dotyczącą protokołu transportowego (TCP czy UDP) - powstaje *datagram*. Na podstawie adresu odbiorcy, protokół IP dokonuje wyboru dalszej trasy datagramu korzystając z tabeli tras i kieruje go do właściwego sterownika w warstwie łącza. Należy dodać, że tabela tras może się zmieniać dynamicznie, ponieważ zawarte w niej dane tracą ważność w przypadku awarii węzła przekażującego pakiety na trasie. Informacje o stanie sieci oraz poszczególnych węzłów są przekazywane przy użyciu protokołu ICMP, który także został umieszczony w warstwie sieciowej. Ostatnim etapem „podróży” danych przez stos jest warstwa łącza danych, w której sterownik interfejsu dołącza kolejne informacje związane z fizycznym przekazem danych (np. suma kontrolna CRC) i dokonuje wysłania ramki w Wielką Sieć.

Opis układu

Schemat elektryczny układu przedstawiono na rys. 3. W systemie NET51 został wykorzystany mikrokontroler 8051 w wersji podstawowej taktowany sygnałem zegarowym o częstotliwości 11,0592MHz, lecz nic nie stoi na przeszkodzie, aby użyć „większych braci” z rodzin produkowanych przez firmy Dallas lub Atmel. Mikrokontroler U2 pracuje w typowym układzie aplikacyjnym z zewnętrzną pamięcią programu i danych. Układ U5 jest zatrzaszkiem, który zapamiętuje osiem młodszych bitów adresu, wystawianych przez U2 na multipleksowanej magistrali sygnałów ad-



Rys. 2. Proces enkapsulacji danych



Rys. 3. Schemat elektryczny układu węzła systemu NET51

resowych i danych. Mikrokontroler współpracuje z pamięcią danych U6 o pojemności 32kB, której większa część została wykorzystana na system buforów danych dla stosu TCP/IP oraz współpracujących z nim aplikacji. Oprogramowanie systemowe jest umieszczone w pamięci programu U8 o pojemności 64kB, którą jest układ pamięci EPROM typu 27C512. Kod źródłowy opisywanego modelu serwera zajmuje około 45kB, a więc pozostaje fragment pamięci na programy użytkownika lub zaimplementowanie dodatkowych protokołów.

W otoczeniu mikrokontrolera znajduje się układ U3 typu AT24C256, który jest pamięcią EEPROM z interfejsem I²C o pojemności 32kB. W tej pamięci są umieszczone dane konfiguracyjne serwera NET51 (m.in. adres IP), a także przechowywana jest struktura prostego systemu plików.

Nietypowym elementem jest układ PLD typu GAL16V8, pełniący funkcję dekodera adresowego. Odpowiednie skonfigurowanie matrycy PLD pozwala utworzyć w przestrzeni adresowej pamięci połączenia pamięci programu i danych (uzyskujemy możliwość dynamicznej modyfikacji programu), lecz ta właściwość nie jest wykorzystywana w omawianym systemie.

Kolejnym elementem układu serwera jest generator sygnałów zerujących typu TL7705. Układ ten monitoruje napięcie zasilania. Jeżeli jego wartość zmniejszy się poniżej określonego progu, wyjściowe sygnały przechodzą w stan aktywny. Po powrocie napięcia zasilającego do właściwej wartości, wyjścia zerujące pozostają w stanie aktywnym przez czas określony stałą czasową wynikający z pojemności kondensatora C2 oraz wewnętrznego rezystora układu U2. Element TL7705 dysponuje wejściem ręcznego zerowania, z wbudowanym układem eliminującym drgania zestyków, które w połączeniu z niewykorzystanym wyprowadzeniem mikrokontrolera może być użyte do zdalnego zerowania przez sieć. W systemie NET51 wykorzystane jest tylko wyjście z aktywnym poziomem wysokim, które służy do zerowania, zarówno głównego mikrokontrolera, jak i kontrolera karty sieciowej.

Najważniejszym elementem całego systemu, umożliwiającym łączność za pośrednictwem sieci Ethernet, jest karta sieciowa ze złączem ISA. W serwerze została użyta, z uwagi na niską cenę oraz dużą popularność, 16-bitowa karta firmy „D-Link“ z kontrolerem typu RTL8019. Układ ten jest całkowicie zgodny ze standardem NE2000, jednak oprogramowanie NET51 wykorzystuje specyficzne właściwości tego kontrolera. Drugim argumentem przemawiającym za użyciem karty ze złączem ISA jest bezpośrednie sprzęgnięcie z magistralą systemową mikrokontrolera (opis można odnaleźć w EP10/93), w przeciwieństwie do kart PCI, które wymagają już specjalizowanego sterownika.

W systemie NET51 kontroler RTL8019 pracuje w trybie 8-bitowym (najstarsze bity szyny danych złącza ISA nie są wykorzystywane) i zajmuje 15 bajtów w przestrzeni adresowej 8051. Zatem do sterowania pracą karty używanych jest pięć linii adresowych A0...A4. Sygnał AEN (aktywny poziom niski) jest sterowany z dekodera adresowego i pełni rolę wejścia CS (*Chip Select*), zezwalającego na operacje we/wy z udziałem RTL8019. Aby używać kartę sieciową w systemie NET51, należy ją odpowiednio skonfigurować. Domyślnie karta pracuje w trybie PnP (*Plug and Play*), a właściwej konfiguracji dokonuje komputer. W tym celu należy umieścić kartę w wolnym slotcie ISA komputera PC i uruchomić program RSET8019.EXE z dyskietki dostarczonej przez producenta karty. Kolejnym krokiem jest wprowadzenie karty w bezworkowy tryb pracy i ustawienie adresu bazowego na 300H. Po zapisaniu zmienionej konfiguracji w pamięci karty i wyłączeniu komputera można ją umieścić w złączu NET51.

Stan serwera jest sygnalizowany za pomocą dwóch diod świecących. Czerwona dioda D1 migocze (z częstotliwością ok. 2Hz), gdy serwer pracuje poprawnie. Natomiast zielona D2 sygnalizuje prawidłowo zakończony proces konfiguracji (np. za pomocą protokołu DHCP). Do komunikacji z otoczeniem system NET51 uży-

WYKAZ ELEMENTÓW

Rezystory

R1: 4,7kΩ
R2, R3: 3,3kΩ
R4, R5: 470Ω
R6, R7: 220Ω

Kondensatory

C1, C10, C11: 100nF
C2, C5...C7: 22µF/16V
C3, C4, C8: 33pF
C9: 220µF/16V

Półprzewodniki

D1: LED czerwony
D2: LED zielony
U1: TL7705
U2: 80C51
U3: AT24C256
U4: 7407
U5: 74LS573
U6: 27C512
U7: MAX232
U8: 62256
U9: G16V8A
U10: 7805

Różne

Q1: 11,0592MHz
CON2: ARK2
CON3: złącze ISA
CON1: DB9
W1: mikroprzełącznik
P1, P2: SY-112-K
Z1: zwora

wa portu szeregowego, dwóch uniwersalnych wejść/wyjść cyfrowych (na schemacie IO1 oraz IO2) oraz dwóch wyjść przekaźnikowych. Do sterowania przekaźnikami oraz diodami świecącymi zostały użyte bufory z otwartym kolektorem (typu 7407), ponieważ wydajność prądowa końcówek portów mikrokontrolera okazała się niewystarczająca. Układ MAX232 odpowiada za konwersję poziomów napięć występujących na stykach portu szeregowego do standardu RS232, co jest wymagane przy transmisji danych do komputera PC. System NET51 wykorzystuje sprzętowo kontrolę transmisji RTS/CTS, dlatego przy połączeniach PPP lub SLIP wymagany jest „pełny“ kabel połączeniowy interfejsu RS232. Natomiast w pozostałych przypadkach może być używany kabel „null modem“. Moduł z układem U7 został zmontowany na osobnej płytce, przez co możliwa jest bezpośrednia współpraca NET51

z innymi mikrokontrolerami posiadającymi port szeregowy.

Ostatnim elementem układu serwera jest prosty zasilacz zbudowany w oparciu o stabilizator U10 typu 7805, który dostarcza napięcia +5V do zasilania elementów cyfrowych. Układ ten jest wymagany, gdyż serwer jest zasilany stabilizowanym napięciem +12V z zewnętrznego zasilacza. Zastoso-

wanie zasilacza o takiej wartości napięcia wyjściowego wynika z konieczności dostarczenia do karty sieciowej napięć +5V oraz +12V.

Paweł Ciesłowski
pcieslowski@wp.pl

Literatura

1. „TCP/IP Illustrated Volume 1, The Protocols“, W. Richard Stevens

2. „TCP/IP Illustrated Volume 2, The Implementations“, W. Richard Stevens, Gary Wright
3. <http://www.ietf.org/rfc.html> - zbiór dokumentów RFC

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/maj02.htm> oraz na płycie CD-EP05/2002B w katalogu PCB.