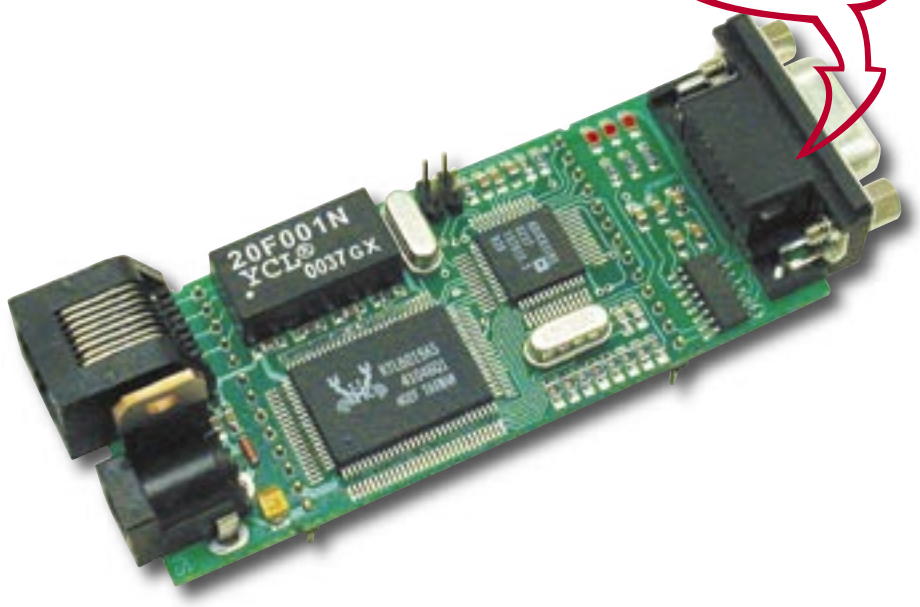
Embedded Ethernet, część 1

PROJEKT Z OKŁADKI



Do czego służy Internet wiedzą wszyscy. Używamy go prawie codziennie do przeglądania poczty e-mail, wyszukiwania informacji bądź oglądania stron WWW. O tym, co znajduje się z drugiej strony naszego okna Internet Eksplorera, wie znacznie mniej osób, chociaż „postawienie" własnego serwera nie jest już sprawą zbytnio skomplikowaną. Natomiast zbudowanie własnego urządzenia, udostępniającego dynamiczne strony HTML, jest nadal pewnym problemem. Jeśli dodatkowo ma ono kosztować mniej niż 50 zł i mieścić się w pudełku od zapałek, to z pewnością niewielu elektroników podejmie trud wykonania takiego „serwerka". Temu, jak to zrobić, będzie poświęcony cykl artykułów, zakończony (w trzeciej części) opisem praktycznie zrealizowanego projektu.

Rekomendacje:
Artykuł polecamy wszystkim zainteresowanym łącznością poprzez ethernet, którzy chcieliby zapoznać się podstawami działania sieci i zdobycie tej wiedzy uwieńczyć samodzielnym wykonaniem mini-serwera sieciowego.

Aby zrozumieć działanie sieci Ethernet, na której opiera się współczesny Internet, należy poznać dokładnie jej podstawy. Zacznijmy więc od kabla.

Co w kablu piszczy?

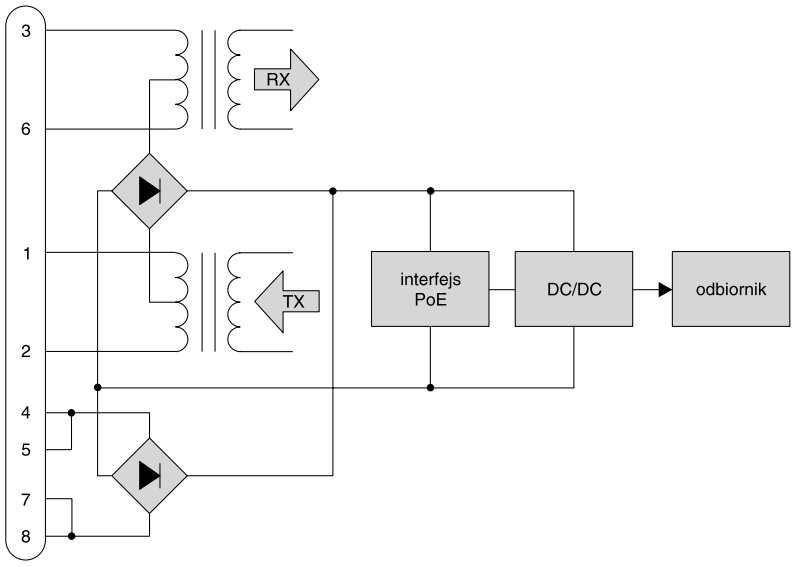
Sieć Ethernet (obecnie najszerzej stosowana w wersji II, zdefiniowanej w normie IEEE802.3) wykorzystuje medium transportowe w postaci szynowej, do którego mogą być przyłączane 2 lub więcej urządzenia. Medium tym może być kabel koncentryczny, światłowód lub para skręconych przewodów. W zależności od parametrów nośnika i urządzeń do niego przyłączonych możliwe do uzyskania szybkości zazwyczaj zawierają się w przedziale 10 Mbd...1 Gbd. W większości obecnie użytkowanych komputerów klasy PC wykorzystuje się karty sieciowe systemu 100BaseT, działające na 2 parach skręconych przewodów. Standard ten umożliwia pracę także z mniejszą szybkością – 10 Mbd (10BaseT), po wykonaniu procesu negocjacji sesji. W przypadku wielodostępu do sieci konieczne jest rozstrzyganie konfliktów, które występują podczas próby jednoczesnego nadawania przez więcej niż jedno urządzenie. Nad prawidłowym przebiegiem tego procesu czuwa technologia CSMA/CD (*Carrier Sense Multiple Access/Collision Detect*), zaimplementowana w strukturze kontrolera sieci.

W kablu sieciowym, łączącym ze sobą urządzenia sieci Ethernet, przemieszczają się ramki komunikacyjne, których celem jest prawidłowe dotarcie do odbiorcy. W treści ramki jest „zaszyty" także adres nadawcy (**rys. 1**) tak, aby wiedzieć kto wyemitował daną serię danych. O spójności danych wewnątrz ramki świadczy suma kontrolna, dołączana na końcu całego ciągu bitów. Ramki sieci Ethernet mają ściśle określony rozmiar – nawet jeśli nie niosą odpowiedniej ilości danych, i tak są rozszerzane do minimum 64 bajtów. Powyższa ramka uzupełniana jest na początku sekwencją startową (*preamble*), w celu zsynchronizowania ze sobą nadawcy i odbiorcy.

Kabel łączący dwie końcówki standardu 10/100BaseT może być zgodny lub krzyżowy, w zależności od typu urządzeń, jakie są nim sprzęgnięte. Sposób jego wykonania jest zasadniczo ściśle określony

adres odbiorcy 6 bajtów	adres nadawcy 6 bajtów	typ ramki 2 bajty	dane 46...1500 bajtów	suma kontrolna 4 bajty

Rys. 1. Ramka IEEE802.3



Rys. 2. Typowy obwód wejściowy PoE

(EIA568A/B), chociażby ze względu na łatwość identyfikacji jego typu na podstawie kolorów przewodów. Szczegółom wykonania i testowania takich kabli poświęcony był artykuł w EP12/2003.

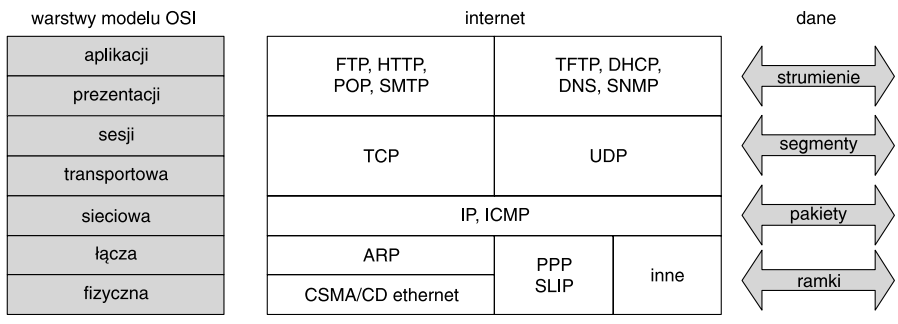
W typowym kablu sieciowym UTP do transmisji wykorzystywane są tylko 2 z 4 istniejących par przewodów sygnałowych. Pozostałe przewody pozostają niepodłączone. Można je więc użyć do zasilania urządzenia, które komunikuje się przez sieć. Ale należy to zrobić prawidłowo! Nowo przyjęta norma IEEE802.3af *Power Over Ethernet* (PoE) specyfikuje, które z przewodów RJ45 mogą być używane do przesyłania zasilania i jak należy to zrobić. Zazwyczaj na ten cel przeznacza się wolne linie 4-5 i 7-8, chociaż można także użyć linii sygnałowych 1-2 i 3-6, które typowo są przyłączone do wejść różnicowych transformatora (**rys. 2**). Zasilane urządzenie (PD) powinno umieć poinformować zasilacz (PSE), że jest zgodne z normą i gotowe do odebrania energii. W skrócie proces ten przebiega następująco (po włożeniu wtyczki RJ45 do gniazda):

– PD przyłącza minimalne obciążenie w postaci rezystora 25 kΩ,
– PSE sprawdza obecność tego rezystora i steruje napięcie 48 V,
– PD uaktywnia przetwornicę DC/DC i odbiera nie więcej niż 13 W mocy.

Opcjonalnie może być także wykonany wybór klasy, do której należy PD. Pozwala to elastycznie dystrybuować moc w ilości zgłaszanej przez zasilane urządzenie. PD musi cały czas pobierać minimalny prąd obciążenia, inaczej PSE odłączy źródło napięcia 48 V i zacznie od nowa testować obecność rezystora 25 kΩ. Stosowanie przetwornicy DC/DC jest konieczne celem zapewnienia izolacji galwanicznej między PD a PSE.

Warstwy, protokoły i stosy

Zapewne każdy, kto miał kontakt z siecią Ethernet, zetknął się z pojęciem stosu TCP/IP bądź modelem OSI (**rys. 3**). Próbując zrozumieć te pojęcia, należy uzmysłowić sobie, co tak naprawdę dzieje się podczas wysyłania ramki od nadawcy do odbiorcy. Prawdę mówiąc, do prawidłowej komunikacji siecią Ethernet wcale nie są potrzebne żadne dodatkowe struktury bądź protokoły, poza oczywiście tą, która definiuje samą ramkę. Bez problemu można bowiem zrealizować połączenie punkt-punkt, korzystając wyłącznie z fizycznego adresu nadawcy i odbiorcy. W krańcowym przypadku, przy bezpośrednim połączeniu tylko dwóch urządzeń ich adresy mogą być nawet całkowicie zignorowane!



Rys. 3. Model OSI

Adres MAC

W rozległej sieci nie można jednak pozwolić sobie na taką swobodę. Każde urządzenie musi mieć przypisany unikalny w skali globalnej adres fizyczny, nazywany adresem MAC. Jest on przydzielany przez niezależną instytucję, która dba o jego właściwy rozdział i stosowanie. Rozmiar adresu MAC wynosi 6 bajtów, przy czym 3 pierwsze z nich są traktowane jako kod producenta, zaś pozostałe jako kolejne w zakresie jego wyrobów. Adres ten rzadko bywa zaszyty na stałe w kontrolerze sieciowym. Zwykle jest umieszczany w zewnętrznej pamięci EEPROM i ustawiany na właściwą wartość na etapie produkcji. Specyficznym adresem jest FF:FF:FF:FF:FF:FF. Jest on używany do rozsyłania informacji do wszystkich odbiorców (*broadcast*), przy czym zazwyczaj odpowiadają tylko te zainteresowane.

Model OSI

Komunikacja między urządzeniami podpiętymi do sieci opiera się na wymianie danych, przesyłanych w odpowiednim formacie. Do tego celu został skonstruowany i ogólnie przyjęty warstwowy model sieci OSI (*Open System Interconnection*), logicznie dzielący system sieciowy na warstwy (poziomy). W każdej z nich może być używany jeden lub więcej protokołów, formatujących dane do postaci zrozumiałej dla warstw sąsiadujących. W zależności od potrzeb liczba warstw może zostać zredukowana (np. TCP/IP do 5 – **rys. 3**), zaś ich rzeczywista implementacja może trafić nawet w całości do kontrolera sieci.

Tab. 1. Podstawowe parametry PoE

Parametr	Min.	Typ.	Max.
Napięcie wyjściowe PSE [V]	44	48	57
Rezystancja doprowadzeń [Ω]	–	–	20
Moc pobierana z PSE [W]	–	–	15,4
Średni prąd obciążenia PD [A]	–	–	0,35

Z każdą warstwą powiązana jest właściwa nazwa strumienia danych, wymienianego pomiędzy węzłami sieci. Zazwyczaj wymianę informacji inicjuje warstwa aplikacji, która przekazuje swoje dane do warstwy niższej. Ta z kolei dodaje własne dane np. nagłówek (*header*) i przekazuje dalej. Ostatnim etapem jest wysłanie ciągu bitów po kablu UTP. Węzeł, który go odebrał, dokonuje odwrotnego procesu – usuwa zbędne nagłówki i przekazuje do warstwy wyższej.

Implementacja modelu OSI (np. części odbiorczej) zdaje się być więc prosta – należy odczytać dane z ramki ethernetowej, a następnie sukcesywnie sprawdzać, jaki protokół się w nich znajduje. Do jednoznacznej identyfikacji jego typu jest wykorzystywany nagłówek, który zwykle niesie także szereg dodatkowych informacji (np. długość pola danych, sumę kontrolną itp.). Sprawa zaczyna się nieco komplikować, jeśli odebrane dane dla jakiejś warstwy (np. TCP) są powiązane z danymi poprzednio wysyłanymi lub odbieranymi. Jeśli liczbę takich stanów da się przedstawić w skończonym, opisywalnym grafie przejść, to możemy wówczas mówić o stosie.

Adres IP

Ważną rolę w procesie komunikacji odgrywają adresy logiczne IP. Nie są one na stałe powiązane z konkretnym kontrolerem sieci i mogą być dynamicznie zmieniane. Daje to ogromną zaletę podczas wymiany uszkodzonego węzła, gdyż wystarczy poznać jego nowy adres fizyczny MAC, a następnie przypisać mu poprzednią wartość adresu logicznego. Od tego momentu nowy węzeł będzie adresowany jak jego poprzednik.

Adres IP bazuje na 4- lub 6-bajtowej liczbie (IPv4/v6), przy czym jej wartości są różnie rozpoznawane przez urządzenia pracujące w sieci. Część adresów, np. z zakresu 192.168.0.0 do 192.168.255.255, służy do budowy lokalnych intranetów, zaś pakiety z takimi wartościami nie są dystrybuowane (*routowane*) na zewnątrz.

Co nieco o protokołach

Funkcjonalnie protokoły odpowiadają za prawidłowe przedstawienie i dostarczenie danych istniejących w danej warstwie. Nie wnikając zbytnio w szczegóły ich działania (do tego celu służą dyrektywy RFC), należy jednak krótko przedstawić kilka najważniejszych z nich.

ARP

Protokół ten umożliwia pobranie adresu fizycznego MAC od urządzenia, dla którego znany jest adres logiczny IP. W tym celu pytający wysyła ramkę z adresem fizycznym FF:FF:FF:FF:FF:FF oraz docelowym adresem IP. Posiadacz takiego adresu odpowiada własnym MAC, podając jako adres przeznaczenia adres nadawcy. Stworzona w ten sposób para MAC-IP umożliwia późniejsze fizyczne wysyłanie ramek z adresem logicznym.

IP

Jest to protokół bezpołączeniowy, co oznacza, że nie wymusza potwierdzania każdego wysłanego pakietu danych. Urządzenie odbierające taki pakiet może go przesłać dalej, obsłużyć lub też zignorować, jeśli wymaga tego warstwa wyższa, której dane są właśnie transportowane. Jeśli dane przesyłane tym protokołem nie zmieszczą się w jednym pakiecie, to mogą one zostać podzielone na szereg mniejszych datagramów i być wysłane oddzielnie. Nie świadczy to jednak o tym, że dotrą one do miejsca przeznaczenia we właściwej kolejności! Na szczęście nad prawidłowym scalaniem pakietów czuwa proces defragmentacji protokołu IP.

ICMP

Częściej kojarzony z komendą ping protokół ten ma szereg użytecznych funkcji. Pozwala przesyłać po sieci informacje o aktywnych lub nieosiągalnych hostach, testować ścieżkę przesyłu danych bądź też powiadamiać o zaniku datagramów (np. przekroczony czas życia pakietu).

UDP

Ten protokół jest także bezpołączeniowy – jak IP z tym, że umożliwia komunikację między różnymi odbiorcami w obrębie tego samego węzła. Do ich rozróżnienia używa 16-bitowej liczby – portu, traktowanej praktycznie jako rozszerzenie adresu IP. Niektóre z numerów portów są zarezerwowane (*well-known-ports*) i pozwalają na uzyskanie informacji np. o bieżącej dacie i godzinie. Główną cechą tego protokołu jest szybkość, zaś podstawową wadą brak potwierdzenia wysłanych danych. Ale niedobór ten można łatwo uzupełnić wyższą warstwą aplikacji. Tego protokołu używa między innymi TFTP, służący do transferu plików.

TCP

Protokół wykorzystuje mechanizm potwierdzania transmitowanych danych, przez co zapewnia istnienie połączenia logicznego między odległymi węzłami. Wbudowanie mechanizmu retransmisji segmentów zwalnia także wyższe warstwy aplikacji z troski o pewność dostarczenia danych. Daje to ogromne ułatwienie przy tworzeniu programów odpornych na błędy transmisji i pozwala skupić troskę programisty na warstwie aplikacyjnej.

Algorytm funkcjonowania stosu TCP zakłada wzajemną symetryczność połączenia. Zarówno serwer, jak i klient muszą monitorować ciąg wymienianych danych tak, aby zapewnić ich prawidłowy przepływ. Do tego celu służą dwa wskaźniki – Seq oraz Ack. Pierwszy z nich niesie informację o względnym początku danych właśnie przesyłanych, zaś drugi oznacza liczbę danych ostatnio odebranych.

Każda ze stron połączenia posiada własny zestaw Seq i Ack, przez co może wykrywać fakt zagubienia segmentu lub jego nadejścia w niewłaściwej kolejności. Do synchronizacji stanu połączenia używa się zestawu flag, które pozwalają zidentyfikować typ segmentu właśnie obsługiwanego. Flaga SYN umożliwia zainicjowanie połączenia, ACK potwierdza segment danych, zaś FIN żąda zakończenia sesji. Odebranie RST powoduje zerwanie połączenia.

Stany, w jakich może się znajdować stos TCP, najlepiej jest przedstawić w postaci grafu możliwych

przejść. Zmiana aktualnego stanu odbywa się na skutek odebrania flagi lub też przeterminowania czasu połączenia.

Nawiązanie połączenia klient-serwer odbywa się przez wysłanie segmentu z ustawioną flagą SYN. Aktywny serwer w trybie LISTEN nasłuchujący na aktywnym porcie potwierdza odebranie żądania i jednocześnie żąda połączenia z klientem. Ten z kolei potwierdzając go własnym ACK, ustanawia połączenie. W stanie ESTABLISHED obydwie strony mogą dowolnie wymieniać dane. Nie jest konieczne natychmiastowe potwierdzanie odebrania danych – można to zrobić później, pod warunkiem że łączna liczba danych nie przekroczy szerokości okna (*window*). Strona, która chce zakończyć połączenie, wysyła w segmencie aktywną flagę FIN. Odpowiadając z własnym FIN połączenie między węzłami (*sockets*) zostaje zakończone.

Podobnie jak w protokole UDP, tak i tutaj można nawiązać połączenie na wybranym porcie. Część z nich wzajemnie się pokrywa (np. *echo*, *daytime*), lecz generalnie przyjmują one różne, całkowicie niezależne wartości. Warstwy aplikacji wykorzystują predefiniowane porty do realizacji standardowych zadań, wymagających pełnej kompatybilności między różnymi systemami.

HTTP

Protokół warstwy aplikacji wykorzystujący port TCP o wartości 80 do dwustronnej wymiany danych w postaci tekstu, bitmap i innych predefiniowanych formatów. Łatwość implementacji tego protokołu bazuje na prostym przetwarzaniu komunikatów tekstowych, które po formatowaniu tworzą zawartość okna przeglądarki internetowej.

Przykładowe przesłanie ciągu:

```
HTTP/1.0 200 OK\r\nContent-type:
text/html\n\n<html>\n<body>\n<b>To
proste\n</b>\n</body>\n</html>\n
```

spowoduje wyświetlenie pogrubionego komunikatu „*To proste*” w oknie IE. Przesyłanie danych binarnych np. plików *.wav* wymaga jedynie poprzedzenia bloku danych właściwym nagłówkiem. Inne z protokołów aplikacyjnych, np. Telnet, TFTP lub DHCP, umożliwiają realizację bardzo użytecznych programów, przy minimalnej wiedzy potrzebnej do ich implementacji. Bazując wyłącznie na specyfikacji RFC protokołu FTP, można z łatwością stworzyć interfejs między systemem plików Windows a mikrokontrolerem. Standardowe operacje typu *drag and drop*, wykonywane między oknami, mogą wówczas bez problemu przyjąć za cel pamięć wewnętrzną mikrokontrolera. O technicznych możliwościach stworzenia prostego systemu typu *embedded ethernet* dowiecie się w drugiej części naszego cyklu.

Grzegorz Oleszek
grzegorz@savo.pl