

Embedded ethernet, część 2

O tym, co się dzieje w kablu ethernetowym dowiedzieliśmy się w poprzedniej części cyklu.

Teraz czas na przedstawienie możliwości realizacji systemu, oczywiście pozostających w zasięgu elektronika-hobbysty.

Rekomendacje:

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



Jak?

Możliwości podłączenia własnego urządzenia do sieci Ethernet jest wiele. Wszystko zależy od tego, co chcemy osiągnąć, ile mamy czasu i pieniędzy. Zazwyczaj istotne jest także, jakie elementy chcemy ze sobą połączyć – czy wystarczy nam sporadyczne odczytywanie stanu urządzenia przez WWW, czy też chcemy w czasie rzeczywistym przysyłać wyniki pomiaru na dysk zdalnego komputera. Przeznaczenie skutkuje określeniem minimalnej przepustowości łącza, a tym samym typem interfejsu na jego końcach.

Czarna skrzynka

Jest to najprostsze rozwiązanie i czasami w pełni wystarczające. Nie musimy wnikać w zawiłości protokołów oraz szczegóły działania kontrolera sieciowego. Po prostu wstawiamy dedykowany moduł, ewentualnie instalujemy sterownik na PC i tyle! Wszystko inne jest zaszyte wewnątrz „czarnej skrzynki”, która powinna być mała i nie sprawiać problemów przy uruchomieniu i późniejszym zarządzaniu. Najlepiej byłoby, aby do jej konfiguracji użyć standardowej przeglądarki WWW. Nie musimy się niczego uczyć – po prostu *plug and play!*

Czy coś takiego istnieje? Ależ tak! Wielu producentów oferuje moduły, które są gotowe do natychmiastowego użycia. Niektóre z nich mieszczą się nawet wewnątrz niezbyt wiele powiększonego gniazda RJ45. Funkcjonalnie zapewniają możliwość podłączenia własnego urządzenia poprzez interfejs asynchroniczny lub kilka linii we/wy. Sterowniki programowe zazwyczaj umożliwiają „mapowanie” takiego interfejsu jako zwykły port COM, co umożliwia pracę sieciową za pośrednictwem typowych programów komunikacyjnych np. HyperTerminal.

Fot. 4. Wygląd modułu EM100

Jedyną przeszkodą w swobodnym używaniu tego typu modułów może być cena, która zazwyczaj wynosi ok. 30...50 Euro. W zamian jednak otrzymujemy bardzo funkcjonalny interfejs, z pełną obsługą stron WWW oraz elastyczną konfiguracją linii we/wy. Wydaje się, że w aplikacjach wymagających szybkiej adaptacji istniejącego rozwiązania do potrzeb pracy sieciowej jest to najwygodniejsze podejście.

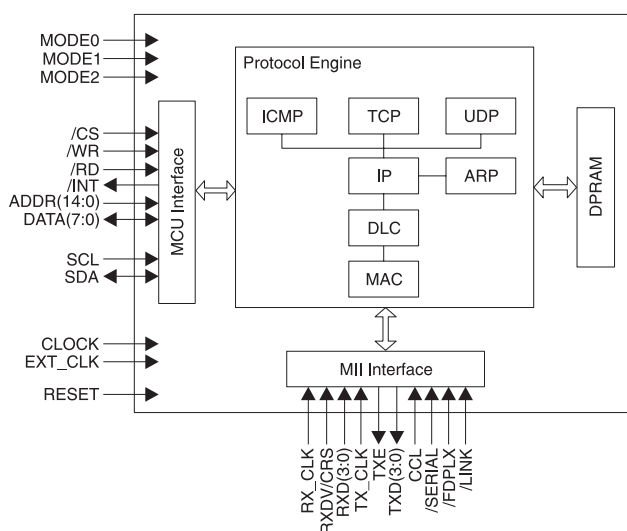
Należy jednak zwrócić uwagę, że nie wszystkie interfejsy w obudowanym gnieździe RJ45 odpowiadają gabarytom standardowego gniazda sieciowego. Czasami może to stanowić pewien problem.

Jeśli planujemy wykorzystanie interfejsu Ethernet-serial, to użyteczne może okazać się zastosowanie modułu bez zamontowanego gniazda RJ45. Przykładem może być EM100 (fot. 4), pracujący z zewnętrznym gniazdem RJ45. Jego wygodną cechą jest możliwość samoczynnego sterowania kierunkiem nadawania i odbioru, co znacznie ułatwia jego użycie w aplikacji RS485.

Stosunkowo duże gabaryty tego modułu w nowszej wersji - EM120 - zostały ograniczone do 35x27x9 mm. Dodatkowo powiększono bufor łącza szeregowego do 4 kB, co znacznie podniosło funkcjonalność tego urządzenia.

Podejście konstrukcyjne

Projektując nowe urządzenie, zazwyczaj dążymy do minimalizacji kosztów jego opracowania i późniejszej produkcji. Oczywistym jest, że właściwe dopracowanie w pełni funkcjonalnego stosu TCP/IP i jego warstw aplikacyjnych nie jest sprawą banalną. Z tego też względu wiele firm oferuje gotowe układy scalone, integrujące w swojej



Rys. 5. Schemat blokowy W3100A

strukturze sprzętową obsługę protokołów i czasami także blok MII (*Media Independent Interface*). Po uzupełnieniu takiego układu o fizyczny sterownik (PHY) i transformator (MAG) uzyskujemy kompletny interfejs Ethernet. Jak jednak można się do niego podłączyć?

Scalone sterowniki funkcjonalne

Przyjaznym pod względem możliwości interfejsowych jest układ W3100A (rys. 5). Układ ten w swojej strukturze zawiera sprzętową obsługę protokołów, z TCP i UDP włącznie. Komunikację z nim można zrealizować jak ze zwykłą pamięcią równoległą lub też potraktować go jako jeden z modułów magistrali I2C. Do pracy potrzebny jest także zewnętrzny układ PHY no i oczywiście transformator.

Od strony interfejsu projektant dostaje zestaw funkcji API, które wykonują zadania wysokiego poziomu. Przykładowy rozkaz *Connect ()* nawiązuje połączenie ze zdalnym serwerem w jednym z 4 możliwych kanałów. Jego aktualny stan można na bieżąco monitoro-



Fot. 6. Programowalny moduł interfejsowy IPC@CHIP

wać, odczytując zawartość rejestru *Socket State Register*.

Układ W3100A jest wygodny w użyciu, zwłaszcza, że producent dostarcza także oprogramowanie w wersji źródłowej do warstw aplikacyjnych (m.in. HTTP, FTP, DHCP). Jeśli nie planujemy wymiany dużej liczby danych przez sieć, to warto się zastanowić nad wykorzystaniem interfejsu I2C, w który jest wyposażony ten moduł. Ograniczy to znacznie liczbę niezbędnych linii oraz umożliwi współpracę z układami nie wyposażonymi w zewnętrzną magistralę. Dla leniwych dostępne są także kompletne moduły – np. IIM7010A, zawierające W3100A, RTL8201BL (PHY) oraz MAG-JACK (gniazdo RJ45 zintegrowane z transformatorem).

Jeszcze bardziej posunięta integracja funkcji sieciowych, a tym samym odciążenie programisty od trudów implementacji warstw sieciowych, występuje w nowym układzie iChip CO710 firmy ConnectOne. Posiada on wbudowaną obsługę także podstawowych aplikacji (HTTP, TELNET, SMTP), dzięki czemu praca z nim jest wyjątkowo ułatwiona. Do komunikacji z *hostem* można skorzystać z łącza asynchronicznego lub interfejsu równoległego. Oprogramowanie bazuje na rozszerzonym zestawie komend *AT+i*, które analogicznie jak w przypadku zwykłego modemu pozwalają w pełni obsługiwać połączenie. Układ jest umieszczony w obudowie BGA-121 o gabarytach 10x10x1,2 mm, przez co nie zajmuje zbyt wiele miejsca. Niestety, aktualnie jest dostępny wyłącznie w dużych ilościach, przez co pozostaje poza zasięgiem dla konstrukcji jednostkowych.

Powyższe rozwiązania zakładają użycie zewnętrznego modułu lub ukła-

du scalonego jako sposobu na podłączenie się do sieci Ethernet. Czasami jednak, zwłaszcza przy tworzeniu nowych urządzeń, można oprzeć się na konstrukcji zawierającej wbudowany interfejs Ethernet (blok MAC oraz czasami PHY) oraz wolne zasoby do realizacji własnych zadań. Jednym z możliwych rozwiązań jest układ DS80C400. Jest to układ bazujący na bardzo szybkim rdzeniu 8051 (75 MHz), wyposażony w blok MAC oraz obsługę stosu TCP/IP (v4/v6) „zaszytą” w pamięci ROM.

Układ posiada bardzo bogate peryferia (CAN, 1-Wire, 3 UART-y), co przy uzupełnieniu o 16 MB przestrzeni adresowej pozwala na zbudowanie bardzo wydajnego systemu. Bezpłatne środowisko TINI daje możliwość swobody programowania, zaś obsługa *apletów* Java wręcz nieograniczone własności użytkowe.

Układy o podobnej funkcjonalności oferują także inni producenci (np. DSTni LX firmy Lantronix, IPC@CHIP firmy Beck - fot. 6), przy czym zakres ich własności musi być każdorazowo dobierany pod kątem konkretnego zastosowania.

A gdyby tak samemu?

Korzystanie z gotowych modułów, oprócz niewątpliwych zalet, zazwyczaj nie umożliwia dogłębnego poznania działania układów sieciowych oraz szczegółów implementacji ich protokołów. Ktoś mógłby powiedzieć – po co sobie utrudniać życie? A jednak, jeśli chodzi o maksymalną redukcję kosztów bądź niestandardowe potrzeby użycia sieci Ethernet, jedynym rozwiązaniem pozostaje zbudowanie i oprogramowanie własnego urządzenia. Cóż więc można zrobić w tym zakresie?

Po pierwsze sprzęt

Do dyspozycji mamy cały szereg układów, które możemy wykorzystać we własnych konstrukcjach. Nie wszystkie umożliwiają prostą implementację w konstrukcjach amatorskich. Przykładowo, chcąc skorzystać z bardzo popularnego układu RTL8139C, występującego w wielu kartach sieciowych 10/100 Mb, musielibyśmy obsłużyć magistralę PCI 2.0, co jest poza zasięgiem prostych mikrokontrolerów. Co więc im pozostaje, aby stać się *Ethernet-ready*?

Bardzo popularne mikrokontrolery rodziny AVR, 8051 oraz PIC zazwyczaj posiadają możliwość obsługi zewnętrznej magistrali adresowej lub chociażby jej emulacji za pośrednictwem portów we/wy. Można je wykorzystać do podłą-

Pracoholicy z HITACHI. Do 8000 godzin wydajnej pracy non - stop w temp. 85 C, a związki zawodowe milczą.



8000 godzin non-stop w temp. + 85 C lub + 105 C.

Kondensator elektrolityczny Hitachi z wyprowadzeniem śrubowym

Standardowy czas pracy kondensatora Hitachi z wyprowadzeniem śrubowym wynosi 8000 godzin przy swojej maksymalnej temperaturze 85 lub 105 stopni C. Dla porównania - u innych producentów czas ten wynosi 3000 lub 2000 godzin. Górna granica napięcia pracy wynosi aż 550V.



4000 godzin non-stop w temp. + 85 C lub + 105 C.

Kondensator elektrolityczny Hitachi z wyprowadzeniem typu SNAP

Standardowy czas pracy kondensatora Hitachi typu SNAP wynosi 4000 godzin przy swojej maksymalnej temperaturze pracy 85 lub 105 stopni C. U innych producentów czas ten wynosi standardowo 2000 godzin.

ROPLA®

Przede wszystkim kondensatory...

Ropla Elektronik, 53-010 Wrocław, ul. Wyścigowa 3. tel +48 (71) 339 7229, fax. 339 7230, info@ropla.pl, www.ropla.pl

czenia i obsługi kontrolera sieci, w skrócie NIC (Network Interface Controller). Układy te posiadają zazwyczaj magistralę 8/16-bitową oraz linie adresowe i sterujące typowe dla pamięci równoległych. Można je więc bez problemu sprząć z systemem mikroprocesorowym.

Na polskim rynku dostępne są m.in. układy firmy Crystal CS8900A oraz Realtek RTL8019AS (rys. 7). Obydwa z nich integrują w sobie kontroler MAC, PHY oraz duży bufor na ramki nadawcze i odbiorcze. Komunikację z nimi można zrealizować w trybie 8-bitowym, zaś liczbę linii adresowych ograniczyć do kilku (dla trybu I/O). Układy te obsługują łącze o szybkości 10 Mb. Po dołączeniu transformatora tworzą kompletny front-end do sieci Ethernet.

Zestaw przydatnych linków

<http://www.poweroverethernet.com>
<http://standards.ieee.org/regauth/>
<http://www.faqs.org/rfcs/std/std-index.html>
<http://www.beck-ipc.com>
<http://www.smc.com/main/catalog/lan91c111.html>
<http://www.realtek.com.tw>
http://www.mcselec.com/easy_tcp_ip.htm
<http://www.cmx.com>
<http://www.opentcp.org/>

Współpraca z układem NIC ogranicza się do zaprogramowania rejestrów konfiguracyjnych oraz wysyłaniu i odbieraniu ramek Ethernet. Cała obsługa protokołów musi znaleźć się po stronie procesora sterującego. Ale nie ma się czego obawiać – w prostej wersji implementacja HTTP ze wszystkimi warstwami modelu OSI może się zmieścić w 6...8 kB pamięci 8051.

Jeśli wymagana jest praca z szybkością 10/100 Mb, to można skorzystać z układu LAN91C111, produkowanego przez firmę SMSC. Pozwala on elastycznie dopasować się do wielu typów magistral (8/16 bit, Intel/Motorola), przy zachowaniu łatwości jego programowania.

Po drugie oprogramowanie

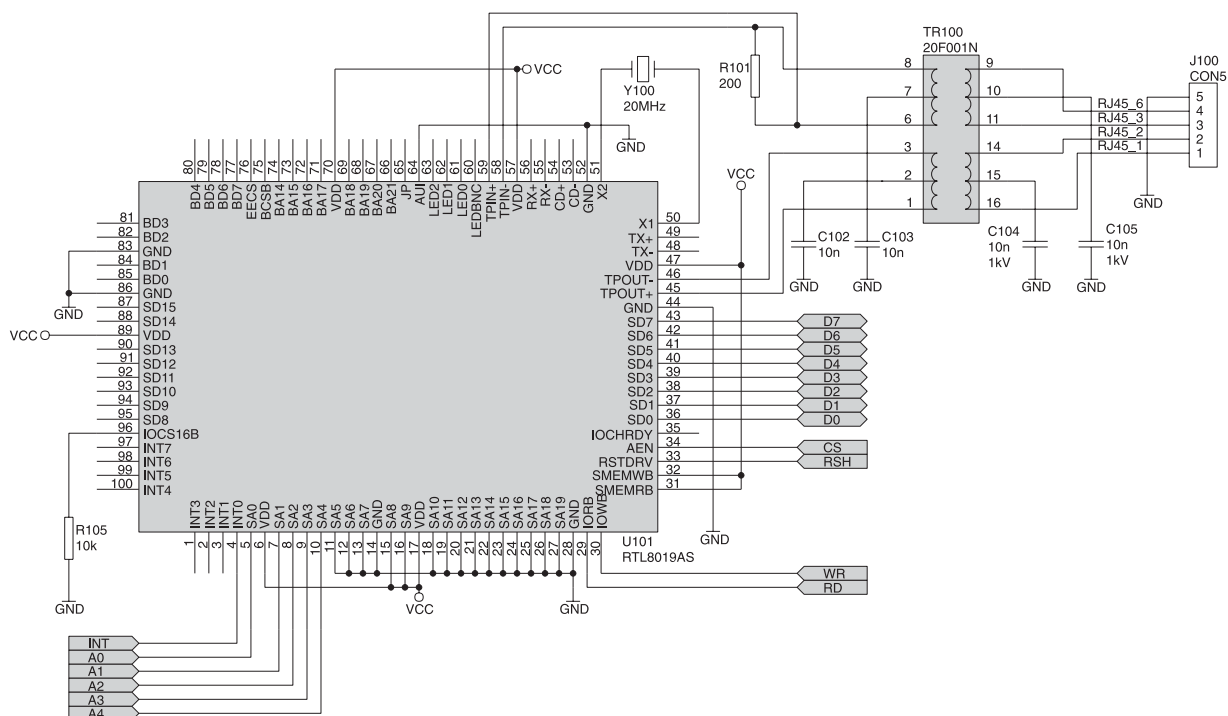
Jak to zwykle bywa w systemach mikroprocesorowych wykonanie sprzętu to znacznie mniej niż połowa sukcesu. Zaimplementowanie modelu OSI dla niewłaściwego programisty może być przeszkodą nie do pokonania. Dlatego zawsze na początku najlepiej jest skorzystać z gotowych rozwiązań i przykładów.

Wiele firm programistycznych oferuje gotowe biblioteki sieciowe, obsługujące większość protokołów i aplikacji.

Szczególnie interesujące są te rozwiązania, które pozwalają zminimalizować wymagania odnośnie wielkości pamięci programu i danych, a tym samym móc je wykorzystywać na popularnych mikrokontrolerach. Jednym z liderów w tej dziedzinie jest firma CMX, oferująca kilka pakietów oprogramowania, zoptymalizowanych pod kątem procesorów 8 i 16-bitowych. Przykładowy CMX-MicroNet zawiera obsługę protokołów TCP/IP, HTTP, TFTP i kilku innych. Użytkownik pakietu dostaje wersje źródłowe w C, przy czym dodatkowo są one wygodnie zarządzane przez program konfiguracyjny. Istotne jest, że finalna aplikacja nie jest obciążona żadnymi dodatkowymi opłatami licencyjnymi. Należy na to zwracać uwagę przy wyborze bibliotek do zastosowań komercyjnych.

Miłośnicy Bascoma także mogą skorzystać z gotowego rozwiązania, przygotowanego dla procesorów AVR. Oferowany pakiet Easy-TCP jest dedykowany do współpracy z modułami opartymi o W3100A.

Jeśli planujemy dogłębne zapoznanie z funkcjonowaniem protokołów sieciowych, realizując własny układ oparty o procesory PIC18, to godny polecenia jest pakiet oprogramowania TCP/IP,



Rys. 7. Przykładowa aplikacja RTL8019AS

udostępniany nieodpłatnie na stronie Microchipsa. Przejrzystość kodu źródłowego wraz z niezbędnymi komentarzami, powoduje, że jest łatwy do zrozumienia, jest nawet przez tych, którzy po raz pierwszy zetkną się z zagadnieniami programowania sieciowego. Zestaw dostępnych aplikacji obejmuje HTTP, FTP, DHCP, wraz z niezbędnymi protokołami niższych warstw – TCP/IP, ARP, UDP. Możliwe jest także skorzystanie z protokołu SLIP, jeśli chcemy wykorzystywać łącze szeregowe do obsługi Internetu. Co jest bardzo istotne – na etapie kompilacji można wybrać funkcję, jaką będzie realizowała finalna aplikacja, co pozwala elastycznie gospodarować zazwyczaj skromnymi zasobami pamięci.

Jeśli chcielibyśmy bazować na innych typach mikrokontrolerów lub układów sieciowych, to możemy skorzystać z bardzo popularnego i przyjaznego zestawu bibliotek uIP oraz lwIP. Zostały one przygotowane z myślą o dowolnych procesorach (ANSI C), zaś ich sprzężenie z typowymi kontrolerami sieci (RTL8019AS, CS8900A) wymaga modyfikacji jedynie kilku funkcji. Większość wykonywanych zadań w tym środowisku opiera się na wywoływaniu funkcji użytkownika (*callback*) po wystąpieniu określonego stanu (np. odebranie ramki, przeterminowanie itp.), co uwalnia procesor od żmudnego przeglądania i analizy stanu danego połączenia sieciowego.

Nieco bardziej rozbudowane wersje bibliotek sieciowych oferuje Viola

Systems. Są one przeznaczone głównie dla procesorów 16-bitowych (M16C, MB90, H8), chociaż można znaleźć także ich adaptację dla popularnej rodziny 8051. Udostępniane aplikacje zawierają obsługę poczty e-mail, telnet i HTTP.

Praktyczny wybór docelowej biblioteki, jeśli oczywiście nie chcemy jej tworzyć samodzielnie, powinien być zwykle poprzedzony realizacją prostego projektu, zazwyczaj bazującego na wersji ewaluacyjnej danego pakietu. Do jego uruchomienia i przetestowania możliwości wygodnie jest użyć prostego zestawu startowego, o szczegółach wykonania którego napiszemy w ostatniej części naszego cyklu.

Grzegorz Oleszek
grzegorz@savo.pl

MONTAŻ SMT

- na paście
- na kleju

PROGRAMOWANIE KONSTRUOWANIE

- sterowników na bazie mikrokontrolerów 8-bitowych, 16-bitowych, 32-bitowych

PROJEKTOWANIE

- układów elektronicznych
- obwodów drukowanych

PONADTO OFERUJEMY:

- montaż mieszany: przewlekany, SMT
- lutowanie na fali lutowniczej SOLTEC MIDI z podwójną falą typu SMART WAVE

MCD Electronics
34-300 Żywiec ul. Lelewela 26
tel/fax: 33/861 60 35
e-mail: smt@mcd.com.pl
http://www.mcd.com.pl