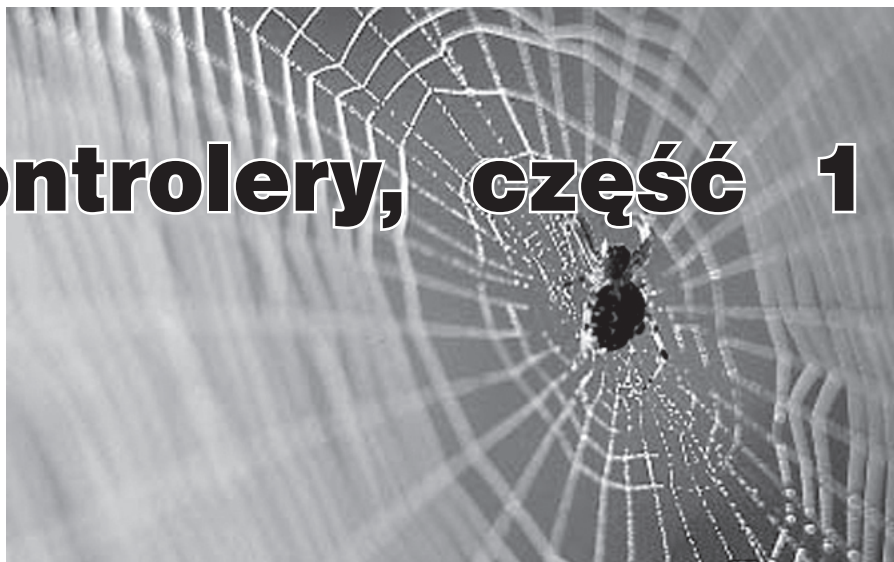


TCP/IP i mikrokontrolery, część 1



*Jak mówi stare przysłowie:
„co dwie głowy to nie jedna”.
A gdy głów tych będą tysiące,
a nawet miliony? Aż strach
pomyśleć. Tymczasem taką
globalną siłę intelektu mamy
dziś w zasięgu ręki, ba – nawet
z niej korzystamy. Wszystko
za pośrednictwem terabajtów
informacji, które w każdej
sekundzie przesyłane są
olbrzymią siecią informatyczną
oplatającą całą kulę ziemską.*

Upowszechnienie się Internetu było tylko kwestią czasu. Dziś prawie każdy użytkownik komputera ma w swoim domu do niego dostęp. Coraz więcej urządzeń domowego użytku pozornie nie mających najmniejszego związku z globalną siecią (np. tuner, lodówka) ma możliwość dostępu do Internetu. Celem niniejszego artykułu jest przedstawienie metod dołączania do Internetu własnych, mniej rozbudowanych urządzeń. W szczególności przedstawione zostaną aspekty programowe wykorzystania różnych protokołów sieciowych we własnych aplikacjach.

Informacje podstawowe

TCP/IP jest rodziną wielu protokołów (TCP, UDP i innych). Wszystkie standardy protokołu TCP/IP są opublikowane w dokumentach RFC dostępnych na stronie www.ietf.org/rfc.html. Więcej informacji można uzyskać z książek na temat TCP/IP.

Dostosowanie aplikacji mikroprocesorowych do współpracy z Internetem czy ogólnie siecią Ethernet jest stosunkowo łatwe przy wykorzystaniu układu W3100A firmy Wiznet. Układ ten posiada 4 niezależne kanały, co stwarza możliwość dołączenia 4 klientów. Wbudowana pamięć RAM pełni funkcję bufora dla otrzymywanych i wysyłanych danych. Układ W3100A jest dostępny w obudowie SMD, ale wykorzystując gotowy moduł IIM7000A można bez

większych problemów zastosować go we własnych aplikacjach. Wszystkie programy zawarte w niniejszym artykule zostały przetestowane na zestawie uruchomieniowym Easy TCP/IP, w którym zastosowano mikrokontroler ATmega162. Przykładowe programy zostały napisane w Bascomie AVR. Efekty działania niektórych aplikacji można obserwować na wyświetlaczu LCD zestawu, komputerowym terminalu komunikującym się z zestawem przez łącze RS232 oraz za pośrednictwem programu *easycp-ip.exe*. Do poprawnej kompilacji programów wymagana jest biblioteka *tcpip.lbx*. Wykorzystywane z tej biblioteki instrukcje będą sukcesywnie opisywane w trakcie przedstawiania przykładowych aplikacji. Będą to:

- aplikacja testująca poprawność konfiguracji zestawu Easy TCP/IP,
- aplikacja umożliwiająca wysyłanie wiadomości e-mail po naciśnięciu przycisku,
- aplikacja umożliwiająca wysyłanie wiadomości e-mail z autoryzacją po naciśnięciu przycisku,
- aplikacja umożliwiająca odbieranie wiadomości e-mail z informacją o włączeniu bądź wyłączeniu diody LED,
- aplikacja umożliwiająca pracę jako serwer, do którego może się dołączyć do 4 klientów,
- aplikacja umożliwiająca pracę jako klienta łączącego się do serwera,
- aplikacja komunikująca się przez protokół bezpołączeniowy UDP,
- aplikacja serwera HTTP, umożliwiająca odczyt temperatury oraz sterowanie diodą LED przez przeglądarkę stron www.

Protokoły TCP/IP oraz UDP

TCP to skrót od *Transmission Control Protocol*, natomiast IP to skrót od *Internet Protocol*. Każde urządzenie w sieci może mieć przypisany adres IP, który będzie go identyfikował (dba o to bezpołączeniowy protokół IP). W TCP/IP do komunikacji pomiędzy aplikacjami używane są tak zwane gniazda, które można porównać do kanałów komunikacyjnych. Aby skomunikować się z danym serwerem lub klientem należy w pierwszej kolejności otworzyć gniazdo. Należy także wspomnieć o portach, które są punktami końcowymi połączeń związanymi z aplikacjami. Numer portu może się zawierać w przedziale od 0 do 65535, przy czym wiele portów jest używanych przez konkretne usługi (np. HTTP używa portu 80). Serwerem jest aplikacja, która oczekuje na połączenie klientów, natomiast klient jest aplikacją łączącą się do serwera. Na przykład klientem jest aplikacja łącząca się do serwera poczty, natomiast serwerem jest aplikacja HTTP, do której klienci łączą się przez przeglądarkę. TCP jest protokołem gwarantującym dostarczenie danych. W tym protokole oczekiwane jest potwierdzenie odbioru przesyłanych danych przed rozpoczęciem transmisji kolejnych. Inaczej jest w protokole UDP (*User Datagram Protocol*). Jest to protokół bezpołączeniowy. Nie gwarantuje on dostarczenia danych. Nie ma w nim żadnego mechanizmu sprawdzającego czy dana aplikacja otrzymała dane. Przesyłanie danych za pośrednictwem UDP generuje więc mniejszy ruch w sieci niż przesyłanie przez TCP. W ramach przykładu z wykorzystaniem UDP zostanie przedstawiony prosty chat.

List. 1.

```
'Program PING - konfiguracja modułu TCP/IP
'Easy TCP/IP
$regfile = „M162def.dat”
$crystal = 4000000
$xa
$hwstack = 50
$swstack = 50
$framesize = 50
'stałe dla modułu TCP/IP
Const Sock_stream = $01
Const Sock_dgram = $02
Const Sock_ipL_raw = $03
Const Sock_macL_raw = $04
Const Sel_control = 0
Const Sel_send = 1
Const Sel_recv = 2
Const Sock_closed = $00
Const Sock_arp = $01
Const Sock_listen = $02
Const Sock_syntent = $03
Const Sock_syntent_ack = $04
Const Sock_syntent_recv = $05
Const Sock_established = $06
Const Sock_close_wait = $07
Const Sock_last_ack = $08
Const Sock_fin_wait1 = $09
Const Sock_fin_wait2 = $0a
Const Sock_closing = $0b
Const Sock_time_wait = $0c
Const Sock_reset = $0d
Const Sock_init = $0e
Const Sock_udp = $0f
Const Sock_raw = $10
$lib „tcpip.lbx”
Enable Interrupts
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Pinb.0 , Db5 = Pinb.1 , Db6 = Pinb.2 , Db7 = Pinb.3 , E = Pinb.4 , Rs = Pinb.5
linii wyświetlacza LCD
Config TcpiP = Int0 , Mac = 12.128.12.34.56.78 , Ip = 192.168.1.8 , Submask = 255.255.255.0 , Gateway = 192.168.1.1 , Localport = 1000 , Tx = $55 , Rx = $55
Cursor Off
Cls
Lcd „IP = 192.168.1.8”
Do
Loop
End
```

'typo mikrokontrolera
'czestotliwosc zegara
'wlaczenie dostepu do zewnetrznej magistrali mikrokontrolera
'wartosc stosu
'wartosc ramki

'Tcp
'Udp
'Ip Layer Raw Sock
'Mac Layer Raw Sock
'Confirm Socket Status
'Confirm Tx Free Buffer Size
'Confirm Rx Data Size
'Status Of Connection Closed
'Status Of Arp
'Status Of Waiting For Tcp Connection Setup
'Status Of Setting Up Tcp Connection
'Status Of Setting Up Tcp Connection
'Status Of Setting Up Tcp Connection
'Status Of Tcp Connection Established
'Status Of Closing Tcp Connection
'Status Of Closing Tcp Connection
'Status Of Closing Tcp Connection
'Status Of Closing Tcp Connection
'Status Of Closing Tcp Connection
'Status Of Closing Tcp Connection
'Status Of Closing Tcp Connection
'Status Of Socket Initialization
'Status Of Udp
'Status of IP RAW
'zalaczenie biblioteki TCP/IP
'odblokowanie sprzetowych przerwan
'konfiguracja rozdzielczosci wyswietlacza LCD

'konfiguracja

'kursor wytlaczony
'czyszczenie LCD
'wyswietlenie na LCD numeru IP przypisanego Easy TCP/IP
'poczatek nieskonczonej petli do-loop

'koniec programu

Konfiguracja

Aby było można komunikować się z aplikacją wykorzystującą moduł IIM7000A, należy odpowiednio skonfigurować sieć. Od tego będzie zależeć także dostęp do Internetu. Podczas testów przykładowych aplikacji zestaw Easy TCP/IP podłączono do routera z wbudowanym przełącznikiem. Do podłączenia zestawu do przełącznika można użyć typowego przewodu sieciowego bez przeplotu. Do przełącznika podłączono także komputer, którego adres IP ustawiono na 192.168.1.2, maskę: 255.255.255.0, a bramę na 192.168.1.1. Zestaw Easy TCP/IP skonfigurowano tak, aby posiadał adres IP: 192.168.1.8, maskę 255.255.255.0 oraz bramę 192.168.1.1. Taka konfiguracja umożliwiła zestawowi TCP/IP dostęp nie tylko do sieci Ethernet, ale i do Internetu.

Aplikacja testująca poprawność komunikacji z modułem TCP/IP

Do sprawdzenia poprawności komunikacji z modułem TCP/IP można wykorzystać dowolny program *ping*, który wysyła pakiet kontrolny do układu o danym adresie IP. Na list. 1 przedstawiono program konfiguracyjny modułu TCP/IP. Aby była możliwa komunikacja pomiędzy mikrokontrolerem, a modułem TCP/IP, należy włączyć dostęp do zewnętrz-

nej obsługi pamięci. Umożliwia to dyrektywa *\$xa*. Dyrektywy *\$hwstack*, *\$swstack* i *\$framesize* ustalają wartości stosów i ramki. W dalszej części programu zostają zdefiniowane stałe, które ułatwiają czytelność programu. Określonym nazwom zostają przypisane wartości liczbowe. Aby był dostęp do procedur obsługi modułu TCP/IP, należy załączyć potrzebną bibliotekę *tcpip.lbx*. Umożliwia to dyrektywa *\$lib*. Następnie zostaje odblokowany globalny system przerwań. Moduł TCP/IP korzysta tylko z jednego źródła zewnętrznego przerwania. W pierwszej kolejności konfigurowany jest wyświetlacz LCD zestawu Easy TCP/IP. W dalszej kolejności następuje konfiguracja modułu TCP/IP instrukcją *config tcpip*. Pierwszym parametrem tej instrukcji jest określenie rodzaju przerwania, z którego korzysta moduł TCP/IP. W tym przypadku wykorzystywana jest zewnętrzna linia przerwania INT0. Drugim parametrem jest adres MAC (*Media Access Control*) modułu. Jest to adres fizyczny danego urządzenia. W tym przypadku MAC będzie adresem układu TCP/IP. MAC jest zazwyczaj złożony z 12 cyfr szesnastkowych (6 bajtów). Adresy MAC urządzeń pracujących w danej sieci nie mogą być identyczne. W module TCP/IP adres MAC zapisano w postaci dziesiętnej. Parametr IP określa adres IP modułu w sieci,

który także powinien być unikatowy. Tak więc moduł TCP/IP będzie posiadał adres IP 192.168.1.8. Kolejny parametr SUBMASK określa maskę sieci, która w tym przypadku wynosi 255.255.255.0. Następny parametr GATEWAY to adres bramy (wykorzystywanej do dostępu do Internetu), którą ustalono na 192.168.1.1. Adres bramy jest zależny od dostawcy Internetu. Parametr LOCALPORT to domyślna wartość portu, którą ustalono na 1000. Domyślna wartość portu może być wykorzystywana podczas pracy modułu w trybie klienta. Za pomocą parametrów TX oraz RX ustalane są wielkości bufora odpowiednio nadawczego i odbiorczego dla każdego z 4 gniazd. Całkowity bufor w module TCP/IP wykości 8 kB. Dla każdego z gniazd w bajcie określającym wielkość bufora, przypisano dwa bity. Dwa najmniej znaczące bity określają wielkość bufora dla gniazda 0, a dwa bardziej znaczące bity wielkość bufora dla gniazda 3. Wartość bitów 00 powoduje ustawienie bufora w rozmiar 1024 bajtów, 01 na 2048 bajtów, 10 na 4096 bajtów, a 11 na 8192 bajtów. Cały dostępny bufor można rozdzielić na każde z gniazd. Czyli dla danych odbieranych i nadawanych na każde z gniazd, będzie przypadła bufor 2048 bajtów. Aby było to możliwe parametry TX oraz RX powinny mieć przypisane war-

```

Microsoft Windows XP [Wersja 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Marcin>ping 192.168.1.8

Badanie 192.168.1.8 z użyciem 32 bajtów danych:

Odpowiedź z 192.168.1.8: bajtów=32 czas=1ms TTL=64
Odpowiedź z 192.168.1.8: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 192.168.1.8: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 192.168.1.8: bajtów=32 czas<1 ms TTL=64

Statystyka badania ping dla 192.168.1.8:
Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0 (0% straty).
Szacunkowy czas bładzenia pakietów w milisekundach:
Minimum = 0 ms, Maksimum = 1 ms, Czas średni = 0 ms

C:\Documents and Settings\Marcin>_

```

Rys. 1.

tości &H55 (&B01010101). Umożliwi to korzystanie ze wszystkich 4 dostępnych gniazd. Oczywiście przy wykorzystywaniu mniejszej liczby gniazd istnieje możliwość zwiększenia bufora dla danego gniazda kosztem bufora innego gniazda.

W dalszej kolejności w programie na wyświetlaczu LCD jest wyświetlany adres IP modułu TCP/IP,

po czym program przechodzi do wykonywania nieskończonej pętli. Poprawność konfiguracji i komunikację można sprawdzić programem *ping*, który jest dostępny w prawie każdym systemie operacyjnym. Program *ping* wysyła pakiet pod określony adres IP. Adresowane urządzenie powinno zwrócić otrzymane dane z powrotem. Aby sprawdzić

poprawność komunikacji z modulem TCP/IP, należy wydać następujące polecenie:

```
ping.exe 192.168.1.8
```

Na rys. 1. przedstawiono rezultat działania programu *ping* z prawidłową odpowiedzią modułu TCP/IP. W przypadku braku odpowiedzi z modułu TCP/IP, problem może być ze źle skonfigurowaną kartą sieciową w komputerze, w złym przewodzie połączeniowym lub z samym układem wykorzystującym moduł TCP/IP. W przypadku otrzymania odpowiedzi istnieje 99% pewności, że moduł TCP/IP z dołączonym do niego mikrokontrolerem działa poprawnie.

Omówieniem przykładowych aplikacji zajmiemy się w kolejnych częściach artykułu.

Marcin Wiązania, EP
marcin.wiazania@ep.com.pl

Dodatkowe informacje:
<http://www.ietf.org/rfc.html>
<http://www.mcselec.com>
<http://www.i2chip.com>
<http://psoc.prv.pl>

ARM7TDMI®
ADuC7000

...układy i narzędzia uruchomieniowe - już dostępne !

ANALOG DEVICES

ALFINE
uznany dostawca rozwiązań dla
ANALOG DEVICES
teraz, jako partner

THE DSP COLLABORATIVE
ANALOG DEVICES

www.analog.com/dsp/3rdparty

ALFINE P.E.P.
ul. Poznańska 30-32 • 62-080 Tarnowo Podgórne
tel.: (61) 89 66 934, 89 66 936 • fax: (61) 81 64 414, 81 64 076
e-mail: analog@alfine.pl • http: //www.alfine.pl

Designed by *Elektron-Forum* • reklama_ELE_30