

Internetowy interfejs dla mikrokontrolera, część 3

AVT-5055



Jest to ostatnia część opisu projektu, który wzbudził ogromne zainteresowanie naszych Czytelników. Sprzętowa realizacja stosu TCP/IP, choćby w oparciu o układ S-7600A, zapewnia nowoczesne i bardzo praktyczne rozwiązanie - warto więc zapoznać się z programowaniem i konfiguracją tego układu.

Najbardziej dostępną i znaną metodą połączenia się z Internetem przez modem jest wykorzystanie numeru dostępowego TP S.A. Wystarczy wysłać do modemu komendę ATDT 0202122, a do PAP String sekwencję: 0x03 PPP0x03PPP i czekać na poprawne połączenie. Pierwsze próby, jakie robiłem nie dały jednak pozytywnego rezultatu. Powstawały zmiany w programie, kolejne próby i... niestety nic z tego. Zrobiłem wiele takich prób, straciłem mnóstwo czasu na modyfikację programu i wyczytywanie pakietów PPP wymienianych pomiędzy S-7600A a serwerami TP S.A. Okazało się, że nie jest możliwe ustanowienie połączenia pomiędzy S-7600A a serwerami TP S.A. Po zmianie na numer dostępowy firmy Polbox wszystko zaczęło działać od razu! Wszelkie próby dotarcia do kogoś z TP S.A., kto mógłby pomóc w wyjaś-

nieniu dlaczego tak jest skończyły się tak samo, jak próby łączenia się przez 0202122. Pozostawię to bez komentarza...

Jak już wspomniałem, wpisanie jedynek do PPP_En w rejestrze PPP Control and Status powoduje rozpoczęcie negocjowania połączenia przez sprzętowy stos PPP. Wcześniej trzeba wpisać jedynekę na pozycji bitu SCTL w rejestrze Serial Port Configuration/Status. Umożliwi to przejście kontroli nad portem szeregowym przez stos sprzętowy. Przy okazji trzeba też ustawić (wpisać jedynekę) bit DSR/HWFC i włączyć mechanizm hardware flow control.

Ze stosem PPP związane są jeszcze cztery rejestry Our IP Address. Pod adresem 0x10 znajduje się mniej znacząca część numeru IP, a pod 0x13 bardziej znacząca. Można tam wpisać adres IP serwera dostępowego lub nic nie wpisywać (domyślnie są

Tab. 11. Rejestr Config Status Low

Bit	Nazwa	Dostęp	Opis
7	TO	R	TCP Timeout - ten bit sygnalizuje, że wystąpił timeout w czasie ustanawiania połączenia TCP lub czekania na pakiet TCP po ustanowieniu połączenia. 0 = normalna praca 1 = nastąpił timeout
6	Buff_Empty	R	Ten bit sygnalizuje, czy bufor danych wyjściowych jest pusty czy też nie. Bit jest jedynką, jeżeli bufor jest pusty. Jest zerowany, jeżeli kieszeń danych wyjściowych nie jest pusta i taka pozostaje. 1 = bufor pusty 0 = bufor nie jest pusty
5	Buff_Full	R	Ten bit sygnalizuje, czy przestrzeń jest dostępna do zapisu danych. Może również wyzwać przerwanie, kiedy bufor jest pełny i bit <i>Buff_Full_En</i> w rejestrze <i>Socket Interrupt Mask Low</i> (0x2a) jest ustawiony. Rejestr <i>Data Register</i> nie powinien być zapisywany, kiedy <i>Buff_Full</i> =1. 0 = bufor dostępny 1 = bufor nie jest dostępny
4	Data_Avail/RST	R/W	Wyzerowanie tego bitu ustawia wszystkie parametry kieszeni do wartości domyślnych. Jest samozerający i nie potrzebuje zerowania dla właściwej operacji. Przed wyzerowaniem należy się upewnić, że bit <i>Snd_Bsy</i> w rejestrze <i>Socket Status High</i> (0x3A) jest wyzerowany. Przeczytany określa, czy są dostępne dane w kieszeni.
2:0	Protocol_Type	R/W	Te bity są używane do ustawiania protokołu kieszeni. 010 = TCP Client Mode 101 = UDP Mode 110 = TCP Server Mode

tam wpisane same zera). W tym drugim przypadku serwer sam prześle swój adres i zostanie on wpisany do rejestrów *Our IP Address* (*floating IP address* negocjowany podczas sesji PPP).

Poprawne zakończenie negocjacji układ S-7600A sygnalizuje ustawiając bit *PPP_UP/SRst* w *PPP Control and Status*. Można wtedy odczytać rejestry *Our IP Address*.

Po ustawieniu *PPP_En* program sterujący czeka w pętli na ustawienie bitu *PPP_UP/SRst*. Jeżeli to nastąpi, to na wyświetlaczu (w górnym wierszu) pojawi się „PPP_ok.“, a w dolnym będą wyświetlone hexadecimalnie cztery bajty adresu IP.

W tym momencie S-7600A może rozpocząć nawiązywanie połączenia TCP ze stacją docelową. Do rejestrów *Their IP Address* (0x3c...0x3f) trzeba wpisać adres IP tej stacji. Przed dostępem do indeksowanych rejestrów kieszeni musi być zaprogramowany rejestr indeksowy o adresie 0x20. Kolejną czynnością jest wyzerowanie bitu *Data_avail/RST* (wyzerowanie kieszeni) w rejestrze *Config Status Low* (adres 0x22). Oprócz adresu IP potrzebne jest jeszcze określenie numerów portów źródła i przeznaczenia. Prezentowane urządzenie ma pracować jako klient poczty. W takim przypadku

port źródła ma mieć losowo wybraną wartość z zakresu 1024...65535. Port przeznaczenia dla aplikacji używającej protokołu POP3 ma standardowy numer 110. Do rejestrów *Our Port registers* (0x38...0x39) wpisywana jest wartość 0x9200 (37376 dziesiętnie),

natomiast do rejestrów *Their Port Registers* (0x37...0x37) jest wpisywana wartość 0x006e (110 dziesiętnie). Obie te wartości, tak jak i adres IP, muszą być ustawione przed aktywacją kieszeni. Po wpisaniu tych ustawień w rejestrze *Config Status Low* trzeba określić typ protokołu kieszeni na *TCP Client Mode*.

Teraz jest już wszystko gotowe i można rozpocząć połączenie TCP (aktywacji kieszeni) przez wpisanie dowolnej wartości do rejestru o adresie 0x30 (*Data Send and Buffer Length*). Jeżeli połączenie dojdzie do skutku, to w rejestrze *Socket Status Mid* (tab. 11) bit *ConU* zostanie ustawiony, a w polu *TCP State* powinna pojawić się wartość 2 (*established*).

Na wyświetlaczu w górnym wierszu pojawi się wtedy komunikat „Polaczenie TCP/IP ok“, natomiast w dolnym „port 110“. Jest to moment, w którym S7600A może wysyłać i odczytywać z serwera pocztowego dane. Dane są wysyłane w momencie wpisania do rejestru *Socket Data* (adres 0x2e). Odczytanie tego rejestru powoduje kolejne wczytywanie danych przychodzących z pamięci aktywnej kieszeni. Przez odczytanie rejestrów *Data Send and*

Bit	Nazwa	Dostęp	Opis
7	URG	R	Ten bit sygnalizuje przychodzące pilne dane. Wpisanie 1 na pozycji bitu URG w <i>Socket Interrupt High Register</i> (0x2d) zeruje ten bit 0 = nie ma pilnych danych 1 = są pilne dane
6	RST	R	Ten bit sygnalizuje, że kieszeń odebrała sygnał RST ze strony TCP 0 = nie odebrano sygnału RST 1 = odebrano RST
5	Term	R	Ten bit sygnalizuje, że kieszeń odłącza się od źródła i wyzwała przerwanie, jeżeli bit <i>Term_En</i> jest ustawiony w rejestrze <i>Socket Interrupt Mask High</i> (0x2b). Ustawienie maski przerwania nie powoduje braku przesyłania tego bitu. 0 = praca normalna 1 = odłączenie kieszeni od źródła ten bit ustawia się, kiedy S-7600A odbierze segment z flagą FIN. To oznacza, że zdalna strona żąda zamknięcia połączenia TCP.
4	ConU	R	Ten bit sygnalizuje, że kieszeń ustanowiła połączenie z hostem. 0 = połączenie nieustanowione 1 = połączenie ustanowione
3:0	TCP State	R	Te bity sygnalizują bieżący stan TCP 0 = CLOSED 1 = SYN_SENT 2 = ESTABLISHED 3 = CLOSE_WAIT 4 = LAST_ACK 5 = FIN_WAIT1 6 = FIN_WAIT2 7 = CLOSING 8 = TIME_WAIT 9 = LISTEN A = SYN_RECVD

Buffer Length można określić wielkość bufora wejściowego.

Wymiana informacji może się odbywać tylko w ramach standardowego protokołu. Do odczytywania poczty stosuje się protokół POP3. Bez wdawania się w szczególności, umożliwi on uwierzytelnienie użytkownika przez wysłanie nazwy i hasła, sprawdzenie liczby wiadomości i zajmowanej przez nie pamięci. Można też odczytać kolejne wiadomości i skasować wiadomość o konkretnym numerze. Po wysłaniu każdej komendy POP3, serwer odpowiada komunikatem, który zaczyna się „+ok.” dla sytuacji kiedy komenda jest zaakceptowana i poprawnie wykonana lub „-err” jeżeli wystąpi błąd. Jeżeli wystąpi błąd, to program sterujący wyświetla odebrany komunikat i rozłącza połączenie przez wyzerowanie bitu *PPP_En* w rejestrze *PPP Control and Status*. W przypadku kiedy cała sekwencja odczytywania poczty jest poprawna, to można odczytać na wyświetlaczu całą przesłaną wiadomość. Zawiera ona oprócz przesyłanej treści - takiej, jaka pojawia się w oknie programu do odczytu poczty - szereg innych istotnych wiadomości normalnie niewidocznych (list. 3) - można tam znaleźć informację od kogo nadeszła wiadomość, dokładną datę i godzinę wysłania, a także dane identyfikacyjne serwera, z którego wiadomość została wysłana.

Oczywiście są tam też takie podstawowe informacje, jak temat i sama treść wiadomości. Ponieważ nawet najkrótsza wiadomość, to kilkaset znaków ASCII - w przedstawionym przypadku 979 znaków - to przeglądanie jej na ekranie wyświetlacza możliwe jest w sekwencjach zawierających 40 znaków. Pierwsze 40 znaków wyświetla się po odebraniu, a kolejne po przyciśnięciu przycisku SW1.

Podsumowanie

Przedstawiony tutaj projekt jest dość nietypowy. Nie jest to projekt zamkniętego urządzenia, ale raczej „szkieletu” służącego do opracowania własnych konkretnych aplikacji.

Starałem się przedstawić w artykule, w jak najbardziej przej-

```
List. 3
+OK 979 octets
Return-Path: <tom50@poczta.onet.pl>
Received: from ghost3.onet.pl (ghost3.onet.pl [213.180.128.18])
by mach4.polbox.pl (8.10.2/8.10.2) with ESMTMP id f9I6KKK9K052319
for <tomasz.jablonski@ep.com.pl>; Thu, 18 Oct 2001 20:14:09 +0200
Received: from pa24.zulavs.cvx.ppp.tpnet.pl ([213.77.198.24]:4100 "HELO
world")

by ghost3.onet.pl with SMTP id <S1078111AbRJRJGON>;
Thu, 18 Oct 2001 20:14:13 +0200
Message-ID: <001d01c157bledd8b0080$0334521fjj9@world>
From: "Tomek" <tom50@poczta.onet.pl>
To: <tomasz.jablonski@ep.com.pl>
Subject: POP3 test
Date: Thu, 18 Oct 2001 20:13:19 +0200
MIME-Version: 1.0
Content-Type: text/plain;
charset="iso-8859-2"
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 4.72.3110.5
X-MimeOLE: Produced By Microsoft MimeOLE V4.72.3110.3
Status: RO

Witaj S7600A!
-
```

rzysty sposób, jak można wykorzystać S-7600A w aplikacji internetowej. Program sterujący działa poprawnie w sytuacji, kiedy połączenie jest realizowane właściwie i nie występują żadne problemy. Program, który zawierałby procedury obsługi sytuacji krytycznych byłby zdecydowanie bardziej rozbudowany, a co za tym idzie dużo mniej zrozumiały. Celowo też wybrałem w warstwie aplikacji protokół POP3. Można było wybrać np. HTML i ze strony WWW sterować jakimś elementem - np. przekaźnikiem. Jednak w takim przypadku jednocześnie do sieci musiałyby być podłączone: nasz moduł i komputer z otwartą stroną WWW, a to z kolei mogłoby zniechęcić wielu potencjalnych eksperymentatorów do prób. W przypadku poczty można najpierw wysłać na swoją skrzynkę wiadomość, a potem odczytać ją za pomocą modułu. Podczas testów opisywany moduł łączył się w zasadzie bez problemu z serwerem dostępowym Polboxu. Zdarzały się problemy przy logowaniu na serwerze POP3 - serwer odpowiadał komunikatem o zajętości. Jednak w czasie niewielkiego natężenia ruchu w Internecie połączenia następowały szybko i bezproblemowo. Do modułu podłączony był standardowy modem Zoltrix FM366 za pomocą firmowego kabla RS232. Do połączenia

się ze stacją w Internecie potrzebny jest jej adres IP. Dostawcy usług internetowych zmieniają adresy IP zachowując tą samą nazwę domeny. Jeżeli aplikacja wykorzystuje system DNS do wyszukiwania adresów IP, to wszystko jest w porządku. W naszym przypadku tak nie jest i ważne jest by znać aktualny adres IP serwera pocztowego. Przy znajomości datagramów protokołu IP można ten adres wyczytać podczas sprawdzania poczty za pomocą np. Outlook Express. Można w tym celu wykorzystać dowolny terminal znakowy. Odpowiedni jest na przykład *Terminal Emulation z Nortona Commandera*. Adres IP serwera dostępowego może być wyświetlany na ekranie modułu po nawiązaniu połączenia PPP. Odpowiedni fragment programu w pliku *tcp.c* (opublikowany na płycie CD-EP3/2002B) jest zapisany w postaci komentarza i wystarczy go odpowiednio zmodyfikować, aby taka informacja mogła być wyświetlana. W pliku *tcp.c* należy też wpisać nazwę i hasło swojej skrzynki pocztowej, wpisać numer telefonu i całość skompilować. Program został napisany dla kompilatora C firmy Keil. W przypadku innych kompilatorów należy go odpowiednio zmodyfikować.

Mimo wielu sceptycznych głosów wydaje się, że systemy ste-

rowania czy akwizycji danych oparte na mikrokontrolerach nieodwracalnie wchodzą w świat wielkiej sieci. Przed konstruktorami otwierają się olbrzymie możliwości przesyłania danych na duże odległości. Dzięki S7600A, i na pewno doskonalszym następcom, aplikacje internetowe stają się łatwe i tanie. Nawet przedstawione tutaj rozwiązanie może służyć na przykład do okresowej

zmiany parametrów sterowania oddalonych od siebie sterowników. Wystarczy za pomocą e-mail'a wysłać zbiór tekstowy zawierający potrzebne dane. Nic nie stoi na przeszkodzie, aby zaimplementować protokół SMTP i wysyłać do serwera wiadomości o zaistniałych zdarzeniach lub jakiś inny protokół np. HTTP.

Tomasz Jabłoński, AVT
tomasz.jablonski@ep.com.pl

Uwaga!

Kody źródłowe do projektu internetowego interfejsu opublikowaliśmy na CD-EP3/2002B.

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.