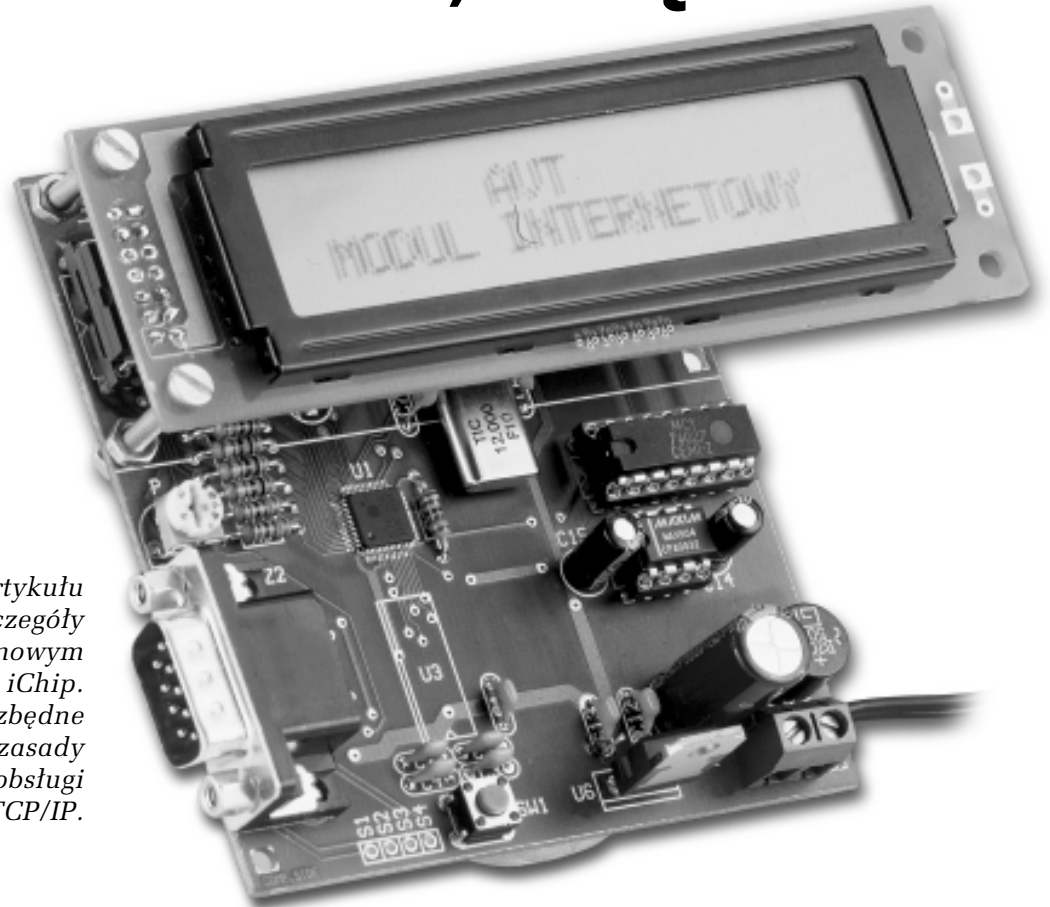


# Internetowy interfejs dla mikrokontrolera, część 2

## AVT-5055



W drugiej części artykułu przedstawiamy szczegóły związane z programowym sterowaniem układu iChip. Informacje te są niezbędne do zrozumienia zasady działania i obsługi sprzętowego stosu TCP/IP.

Jak już wspomniałem do złącza Z2 trzeba dołączyć dowolny zewnętrzny modem, a układ zasilić napięciem stałym lub przemiennym o wartości ok. 8V. Po włączeniu zasilania mikrokontroler rozpoczyna procedurę inicjalizacji. Linie *cs* i *resetx* są zerowane, inicjowany jest licznik T1 (używany do odliczania opóźnień), oraz wyświetlacz LCD. Odblokowywany jest również system przerwań. Po wykonaniu tych czynności linia *resetx* jest ustawiana na „1” i do rejestrów *Baud Rate Divider* (adresy 0x1c...0x1d) wpisywane są wartości określające prędkość transmisji. Tą prędkość oblicza się według wzoru:

$$\text{wartość programowa} = \frac{f_{clk}}{pr\_transmisji} - 1$$

gdzie: *fclk* to częstotliwości zegara S-7600A (uwaga: 0x0003 jest najmniejszą wartością, jaka może być wpisana do tych rejestrów!).

Przy obliczaniu wartości wpisywanych do rejestrów trzeba pamiętać, że dla niektórych prędkości błąd może być dość spory. Na przykład dla częstotliwości zegara 250kHz i prędkości transmisji 38400bd obliczona wartość do wpisania do rejestrów wyniesie:  $250\text{kHz}/38,4\text{kb} = 6,5$ . Oczywiście możemy wpisać 6 albo 7. Dla wartości 6 rzeczywista prędkość wyniesie  $250\text{kHz}/6 = 41,666\text{kb}$ , a dla wartości 7 -  $250\text{kHz}/7 = 35,714\text{kb}$ . Widać, że rzeczywiste wartości różnią się znacznie od żądanych. Dla częstotliwości taktowania 250kHz optymalnym rozwiązaniem będzie prędkość transmisji 19200bd -  $250\text{kHz}/19,2\text{kb} = 13,02$ . Rzeczywista prędkość transmisji wyniesie  $250\text{kHz}/13 = 19,23\text{kb}$  - jest to wartość, którą można zaakceptować. W naszym rozwiązaniu taka właśnie prędkość została wybrana. W momencie, kiedy płytką już

**Tab. 5. Funkcje bitów rejestru Serial Port Configuration/Status (0x08)**

Bit	Nazwa	Dostęp	Opis
7	S_DAV	R/W	<i>Serial Port Data Available</i> - czytany określa czy dane z portu szeregowego są dostępne. Powinno się tam wpisać 0.
6	DCD	R/W	<i>Carrier detect</i> - ten bit odzwierciedla stan linii DCD portu szeregowego. Jest on niezależny od ustawienia bitu SCTL.
5	DSR/HWFC	R/W	<i>Data Send Ready/Hardware flow Control</i> - kiedy ten bit jest czytany, to określa bieżący stan linii DSR portu szeregowego. Kiedy jest zapisywany: 0 - <i>Hardware flow control</i> jest nieaktywny 1 - <i>Hardware flow control</i> jest aktywny
4	CTS	R	<i>Clear to send</i> (bit tylko do odczytu) - odzwierciedla stan bieżący linii CTS portu szeregowego.
3	RI	R	<i>Ring Indicator</i> - bit (tylko do odczytu) odzwierciedla stan linii RI portu szeregowego. Jest on niezależny od ustawienia bitu SCTL.
2	DTR	R/W	<i>Data Terminal Ready</i> - podczas odczytu określa bieżący stan linii DTR portu szeregowego. Mikrokontroler może kontrolować stan tego bitu poprzez wpisywanie.
1	RTS	R/W	<i>Request to send</i> - podczas odczytu określa bieżący stan linii RTS portu szeregowego. Mikrokontroler może kontrolować stan tego bitu poprzez zapisywanie.
0	SCTL	R/W	<i>Serial Port Control</i> - określa kto kontroluje port szeregowy. Wyzerowanie tego bitu (domyślnie) powoduje, że kontrolę przejmuje mikrokontroler. Ustawienie - sprzętowa kontrola przez stos sieciowy. 0 - MPU 1 - Kontrola sprzętowa

**Tab. 6. Rejestr Serial Port Interrupt (0x09)**

Bit	Nazwa	Dostęp	Opis
7	PT_INT	R	<i>Port Transport Interrupt</i> - ten bit sygnalizuje, kiedy przerwanie od portu szeregowego jest aktywne. Jest to zależne od stanu bitów PINT_EN i DSINT_EN w rejestrze Serial Port Interrupt Mask. Kiedy PINT_EN jest jedynką przerwanie wystąpi w momencie, kiedy dana jest dostępna w buforze FIFO portu (S_DAV w Serial Port Configuration/Status jest też jedynką). Kiedy DSINT_EN jest jedynką przerwanie będzie aktywne kiedy CPU zapisuje do Serial Port Data Register, by transmitować daną. Jeżeli oba te bity są jedynkami, to przerwanie występuje w obu przypadkach.

Pozostałe bity nie są używane i po odczytaniu rejestru mają wartość 0.

**Tab. 7. Rejestr Serial Port Interrupt Mask (0x0a)**

Bit	Nazwa	Dostęp	Opis
7	PINT_EN	R/W	<i>Port Interrupt Enable</i> - zezwala na przerwanie od portu szeregowego
6	DSINT_EN		<i>Data Send Interrupt Enable</i> - zezwala na przerwanie od wysyłania danych.

Pozostałe bity nie są używane i po odczytaniu rejestru mają wartość 0.

była gotowa udało mi się zdobyć rezonator o częstotliwości 2,4576MHz. Po podzieleniu przez cztery przez przerzutniki powstaje przebieg zegarowy o częstotliwości 614400Hz. Po wewnętrznym podzieleniu przez 32 uzyskujemy dokładnie 19200. Po podzieleniu przez 16 można uzyskać dokładnie 38400bd. Okazało się jednak, że przy częstotliwości przebiegu zegarowego 614,4kHz układ przestał prawidłowo pracować. Najprawdopodobniej spowodowane jest to zbyt dużą częstotliwością sygnałów jakie wystąpiły na magistrali S-7600A - większe częstotliwości zegara są przeznaczone dla szybkich procesorów. Pewnym rozwiązaniem tego problemu by-

łoby dodatkowe podzielenie częstotliwości zegara przez 2 za pomocą dodatkowego przerzutnika, ale jak wspomniałem płytka już była gotowa, więc pozostałem przy dotychczasowym rozwiązaniu, tym bardziej, że 19200bd jest prędkością zupełnie wystarczającą dla większości zastosowań.

W następnym kroku zapisywane są rejestry *Clock Divider* (0x1c...0x1d). Rejestry te służą do konfigurowania wewnętrznego zegara o częstotliwości 1kHz. Jest on używany do różnych funkcji czasowych w S-7600A: Licznik Podziału=(fclk/1kHz)-1, gdzie fclk to częstotliwości zegara S-7600A.

Dla f=250kHz 250kHz/1kHz-1=249, co w zapisie szesnastkowym

odpowiada wartości 0x00f9. Pod adres 0x1c wpisywana jest wartość 0xf9, a pod adres 0x1d wpisywane jest zero. Procedurę inicjalizacji kończy skonfigurowanie portu szeregowego poprzez wpisanie odpowiedniej wartości do rejestru *Serial Port Configuration/Status* (tab. 5).

Rejestr ten pozwala na sterowanie pracą portu szeregowego UART. Przez jego odczytanie można monitorować stany linii sterujących portu. Port szeregowy może być kontrolowany przez mikrokontroler lub przez sprzętowy wewnętrzny stos (bit SCTL). Wpisana w procedurze inicjalizacji wartość do rejestru *Serial Port Configuration/Status* powoduje, że kontrolę nad portem przejmuje mikrokontroler oraz włączony jest mechanizm *Hardware Flow Control* (bit DSR/HWFC = 1). Jeżeli mechanizm ten jest wyłączony, to dane są transmitowane niezależnie od poziomu sygnału CTSX. Jeżeli mikrokontroler kontroluje port to może on testować stan linii sterujących czytając rejestr *Serial Port Configuration*. Ma też kontrolę nad liniami DCD, DSR, DTR i RTS, a co za tym idzie nad wysyłaniem i odbieraniem danych.

Włączenie *Hardware Flow Control* powoduje, że zaczyna działać pełne sprzętowe potwierdzanie RTS/CTS. Kiedy port wykryje, że sygnał CTS przestał być aktywny, przestaje transmitować dane do momentu, aż ponownie stanie się on aktywny. W momencie kiedy bufor FIFO odbiornika portu jest zapełniony do połowy, to nieaktywny staje się sygnał RTS. Jest to sygnalizacja dla nadawcy, że ma zatrzymać nadawanie danych. Sygnał RTS ponownie staje się aktywny, jeżeli zostaną przeczytane dane z bufora wejściowego portu. Wpisanie danej do rejestru *Serial Port Data* kiedy port jest pod kontrolą mikrokontrolera powoduje, że dana ta zostanie wysłana z wcześniej zaprogramowaną prędkością transmisji. Zakończenie wysyłania sygnalizowane jest ustawieniem bitu *PT\_INT* w rejestrze *Serial Port Interrupt* (tab. 6). Jednak żeby tak się stało, musi być ustawiony bit *DSINT\_EN* oraz *PINT\_EN* w rejestrze *Serial Port Interrupt Mask* (tab. 7).

Odczytywanie danych z portu odbywa się poprzez czytanie tego

**Tab. 8. Rejestr PPP Control and Status (0x60)**

Bit	Nazwa	Dostęp	Opis
7	PPP_Int	R/W	<i>Przerwanie PPP</i> - ten bit sygnalizuje, że zostało wyzwolone przerwanie od stosu PPP. Trzeba odczytać PPP interrupt code register dla stwierdzenia przyczyny. Zapisanie 1 do tego bitu kasuje przerwanie.
6	Con_Val	R/W	<i>Connection Valid</i> - ten bit sygnalizuje warstwie sieciowej, że połączenie „pod nią” jest poprawne 0 = połączenie niepoprawne 1 = połączenie poprawne
5	Use_PAP	R/W	Ten bit odblokowuje autoryzację PAP w protokole PPP. Jeżeli jest odblokowana, to dostęp PAP jest dołączany po negocjacji autoryzacji PAP. Łańcuchy znakowe PAP są wprowadzane poprzez rejestr 0x64 0 = PAP zablokowany (domyślnie) 1 = PAP odblokowany
4	TO_Dis	R/W	<i>Timeouts Disabled</i> - ten bit blokuje blok PPP z timeoutami dla celów diagnostycznych. 0 = timeouts odblokowane (domyślnie) 1 = timeouts zablokowane
3	PPP_Int_En	R/W	<i>PPP Interrupt Enable</i> - ten bit odblokowuje przerwanie PPP 0 = zablokowane (domyślnie) 1 = odblokowany
2	Kick	W	<i>PPP kick start</i> - kiedy wpisana jest 1 to bit ten wystartuje PPP, jeżeli zostało przerwane przez timeout. Bit ten jest samozerający.
1	PPP_En	R/W	<i>PPP enable</i> - ten bit odblokowuje warstwę PPP i musi być ustawiony przed jakąkolwiek transmisją. 0 = PPP zablokowany (domyślnie) 1 = PPP odblokowany
0	PPP_UP/SRst	R/W	Przy odczycie wskazuje, kiedy warstwa PPP ustanowi połączenie 0 = brak połączenia 1 = połączenie ustanowione Przy zapisie bit ten zeruje stos PPP. Jest samozerający i nie potrzebuje zerowania podczas normalnej pracy 0 = normalna praca PPP 1 = reset PPP

samego rejestru *Serial Port Data*. Dane są gotowe do odczytu kiedy bit S\_DAV w rejestrze *Serial Port Configuration/Status* jest jedynką.

Narzuca się pytanie: po co stosować tryb kontroli mikrokontrolera nad portem, skoro S-7600A może sam przejąć nad nim kontrolę? Otóż może to zrobić wówczas, gdy modem połączony do urządzenia i modem z drugiej strony ustalą między sobą połączenie. Przedtem jednak trzeba wysłać do modemu komendy sterujące, które zmuszą go do wybrania numeru dostępowego. Mikrokontroler musi odczytać z modemu informację o poprawnym zestawieniu połączenia modemowego, a może to zrobić tylko wtedy, gdy ma kontrolę nad portem. Modemy mogą wymagać specyficznej obsługi i jej sprzętowa realizacja może być trudna. Dla sprzętowego stosu protokołu PPP informacja, że połączenie w warstwie fizycznej jest poprawne oznacza, że można negocjować połączenie w warstwie sieciowej PPP. Właśnie wtedy S-7600A powinien przejąć kontrolę nad portem szeregowym.

Wróćmy jednak do modemu i sterowania jego pracą. Do komu-

nikacji ze współcześnie stosowanymi modemami stosuje się standard potocznie nazywany „komendy AT”. Nazwa ta utrwaliła się dlatego, że każda komenda wysyłana do modemu musi się zaczynać od znaków AT. Zestaw tych komend jest dość duży. W prezentowanym rozwiązaniu zastosowano dwie: ATVO i ATDT. Pierwsza z komend przedstawia modem na potwierdzenie komend w postaci numerycznej. Modem może odpowiadać po wykonaniu komendy takim właśnie kodem cyfrowym, lub ciągiem znaków ASCII będącym opisem wyniku działania komendy. Odpowiedź w postaci kodu wydaje się być prostsza w obsłudze i dlatego modem został przestawiony w taki właśnie tryb. Druga komenda powoduje, że modem wybiera numer wpisany jako parametr. Numer ten może być wybierany tonowo, lub impulsowo (ATDP). Komenda ta może mieć na przykład postać: ATDT 0202122. Wygląda znajomo - tak jest to numer dostępowy TPSA. Niestety, jak się dalej okaże, wysyłanie takiej komendy nie jest najlepszym pomysłem! Opis większości komend można

znaleźć w instrukcjach obsługi modemów lub w Internecie.

Po wykonaniu komendy ATDT modem wybiera numer i łączy się z modemem połączonym z serwerem dostępowym. Jeżeli wysyłamy do modemu znaki komend AT z prędkością 19200bd, to modem próbuje nawiązać łączność z taką właśnie prędkością. Jeżeli nie jest to możliwe, zmniejszana jest prędkość transmisji, aż do uzyskania poprawnego połączenia. Po nawiązaniu łączności nasz modem odsyła kod określający prędkość transmisji wynegocjowaną w czasie połączenia i na taką prędkość należy przestawić S-7600A. W naszym rozwiązaniu zakładamy, że modem łączy się zawsze z prędkością 19200bd i jeżeli jest inaczej, to połączenie jest rozłączane.

Odebranie kodu po komendzie ATDT kończy fazę realizowania połączenia w warstwie fizycznej. W rejestrze *PPP Control and Status* o adresie 0x60 (tab. 8) można ustawić bit *Con\_Val*, S-7600A może przejąć kontrolę nad portem i rozpoczyna się faza połączenia w warstwie sieciowej za pomocą stosu PPP.

Protokół PPP (*Point to Point Protocol*) jest tu używany do ustanowienia połączenia pomiędzy naszym urządzeniem a serwerem świadczącym usługi dostępowe. Warstwa PPP stanowi połączenie pomiędzy warstwą siecią IP a warstwą fizyczną. Za pomocą PPP urządzenia na obu końcach łączy modemowego mogą wynegocjować sposób przesyłania ramek informacyjnych. Protokół dopuszcza negocjowanie wielu opcji i może się zdarzyć, że urządzenia nie będą mogły, mówiąc obrazowo, porozumieć się. Aby zapobiec takiej sytuacji wprowadzono pojęcie wartości domyślnych. Każda stacja MUSI przyjąć i potwierdzić opcje z tymi wartościami. Układ S-7600A podczas negocjacji połączenia PPP używa właśnie wartości domyślnych - powinien więc, przynajmniej teoretycznie, nawiązać połączenie z każdym serwerem.

Oprócz konfiguracji kanału transmisyjnego, PPP może być używany do potwierdzania autentyczności użytkownika w trakcie logowania się w serwerze (autoryzacja PAP - *Password Authenti-*

**Tab. 9. Rejestr PAP String (0x64)**

Bajt 0	0x05 Liczba znaków nazwy użytkownika
Bajt 1	Znak "t"
Bajt 2	Znak "o"
Bajt3	Znak "m"
Bajt 4	Znak "e"
Bajt 5	Znak "k"
Bajt 6	0x05 liczba znaków hasła
Bajt 7	znak "a"
Bajt 8	znak "l"
Bajt 9	znak "a"
Bajt 10	znak "m"
Bajt 11	znak "a"

tion Protocol). Podczas logowania przesyła się nazwę użytkownika oraz hasło. Wpisanie do bitu *PPP\_En* jedynek powoduje włączenie PAP. Trzeba wtedy do rejestru *PAP String* (adres 0x64, **tab. 9**) wpisać kolejno znaki określające nazwę użytkownika i hasło. Pierwszy wpisywany do *PAP String* bajt określa liczbę znaków ASCII nazwy użytkownika. Następnie trzeba po kolei wpisywać te znaki. Po wpisaniu wszystkich wpi-

**Tab. 10. Rejestr PPP Interrupt Code (0x61)**

Kod błędu	Definicja
0x00	Zarezerwowane
0x01	Niepowodzenie inicjalizacji fazy LCP protokołu PPP
0x02	Niepowodzenie negocjacji fazy NCP
0x03	Niespodziewane zakończenie LCP
0x04	Odebranie termination Request
0x05	Niepowodzenie negocjacji PAP

suje się bajt określający liczbę znaków hasła i, tak jak w przypadku nazwy użytkownika, znaki hasła.

Po wpisaniu hasła i nazwy użytkownika pozostaje tylko wyzerowanie stosu PPP przez wpisanie do *PPP\_UP/SRst* jedynek i odblokowanie warstwy PPP przez wpisanie jedynek do *PPP\_En*.

Od tego momentu nie mamy żadnego wpływu na proces negocjowania połączenia PPP. Sprzętowy stos robi wszystko sam i pozostaje tylko czekać aż bit *PPP\_UP/SRst* ustawi się na „1” i w ten sposób S-7600A zasygnalizuje poprawne połączenie.

Gdy to połączenie nie może być zrealizowane, S-7600A może zgłaszać przerwanie przez wpisanie jedynek do rejestru *PPP Control and Status* na pozycji bitu *PPP\_Int*. Trzeba wtedy przeczytać rejestr *PPP Interrupt Code* (adres 0x61) i określić przyczynę niepowodzenia (**tab. 10**).

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

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/kwiecien02.htm>.