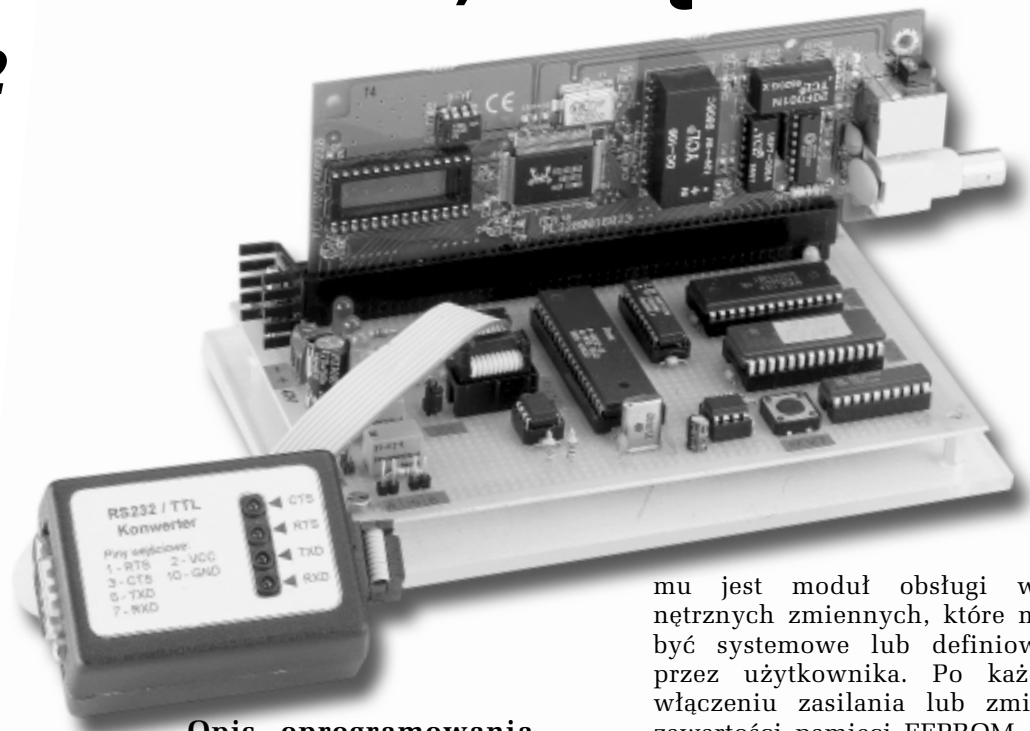


# System NET51, część 2

## AVT-5062



### Opis oprogramowania

Schemat blokowy oprogramowania systemu NET51 przedstawiono na rys. 4. Jak widać, program został podzielony na szereg modułów, które mogą być zestawiane ze sobą na różne sposoby. Takie rozwiązanie stwarza możliwość łatwego dostosowania NET51 do konkretnych potrzeb.

Najważniejszym elementem jest rdzeń systemu, zawierający podstawowe procedury obsługi pamięci, wewnętrznych zasobów mikrokontrolera oraz interfejsów i układów peryferyjnych. Następnym ważnym składnikiem syste-

*W drugiej części artykułu poświęconego modułowi NET51, który zapewnia dostęp do Internetu mikrokontrolerowi '51, przedstawimy jego oprogramowanie systemowe i omówimy sposób posługiwania się modułem.*

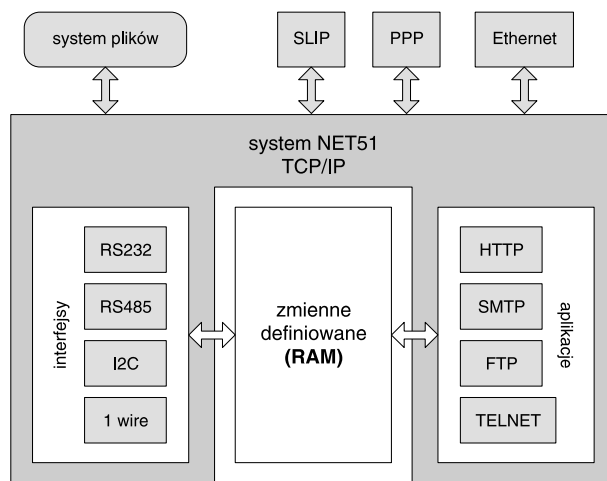
*Zaletą prezentowanego rozwiązania jest programowa implementacja stosu TCP/IP, dzięki czemu wykonanie interfejsu jest stosunkowo*

mu jest moduł obsługi wewnętrznych zmiennych, które mogą być systemowe lub definiowane przez użytkownika. Po każdym włączeniu zasilania lub zmianie zawartości pamięci EEPROM, system NET51 odczytuje definicje zmiennych i tworzy ich odpowiednią strukturę w pamięci danych. Takie rozwiązanie pozwala na wymianę definicji bez konieczności modyfikacji oprogramowania systemu.

W oparciu o rdzeń został zbudowany stos TCP/IP. Nie będę przytaczał tutaj metody implementacji tych protokołów, gdyż wykracza to poza ramy artykułu. Zainteresowanych odsyłam do lektury dokumentów RFC dostępnych w Internecie oraz literatury [1] i [2]. Ze stosem TCP/IP współpracują programy aplikacyjne, dzięki którym możliwa jest wymiana informacji zgodnie z popularnymi protokołami takimi jak FTP lub HTTP.

Serwer FTP wykorzystuje funkcje rdzenia związane z systemem plików znajdującym się w pamięci, gdzie przechowywane są pliki przeznaczone dla aplikacji systemowych (między innymi dla serwera HTTP). Właśnie zawartość tej pamięci może być zdalnie zmieniana przez użytkownika za pomocą dowolnego programu obsługującego protokół FTP.

Serwer HTTP zaimplementowany w NET51 pozwala na tworzenie dynamicznej zawartości odsyłanych stron w zależności od war-



Rys. 4 Schemat blokowy oprogramowania

**Tab. 1. Operatory używane w systemie NET51**

Typ zmiennej	Operator	Opis
bez znaczenia	=	przypisanie wartości np. <i>temp=12</i>
bez znaczenia	:	indeks od prawej np. gdy <i>temp</i> wynosi 1234 to <i>temp:1</i> zwróci 3
liczba	+	zmienna = zmienna + wartość
liczba	-	zmienna = zmienna - wartość
liczba	*	zmienna = zmienna * wartość
liczba	/	zmienna = zmienna / wartość
liczba		zmienna = zmienna or wartość (operacja bitowa OR)
liczba	^	zmienna = zmienna xor wartość (operacja bitowa XOR)
liczba	&	zmienna = zmienna and wartość (operacja bitowa AND)
liczba	\$	zwraca ciąg „checked” gdy (zmienna and wartość) == wartość

tości wewnętrznych zmiennych. Zmiany zawartości lub odczytu tych zmiennych można dokonywać zarówno z poziomu przeglądarki internetowej jak i interfejsu szeregowego obsługiwanego przez mikrokontroler. Takie rozwiązanie umożliwia, za pośrednictwem sieci Internet, wizualizację stanu otoczenia serwera lub zdalne sterowanie dowolnym urządzeniem, które posiada tylko port RS232.

System NET51 potrafi interpretować polecenia, które są przesyłane przez interfejs RS232 w postaci ciągu znaków zakończonych sekwencją <CR><LF>, jak niżej: <polecenie><CR><LF>

W ten właśnie sposób możliwe jest odczytanie lub modyfikacja wartości zmiennych oraz sterowanie niektórymi funkcjami systemu przez współpracujące urządzenie. Polecenie posiada ściśle określony format

```
<nazwa_zmiennej><operator>
[wartość].
```

Jedynie pole *wartość* jest opcjonalne. Dla przykładu, aby odczytać zawartość zmiennej przez port szeregowy należy wysłać polecenie:

```
<nazwa_zmiennej>?
```

i w przypadku poprawnego wykonania, serwer zwróci ciąg zna-

ków także zakończonych sekwencją <CR><LF>. Natomiast w celu zmiany wartości wybranej zmiennej należy wysłać polecenie z odpowiednio dobranymi polami „operator” oraz „wartość”, których zastosowanie jest zależne od typu zmiennej (np. *temp+10* lub *data=12-01-2001*). Znaczenie poszczególnych operatorów przedstawiono w **tab. 1**. W przypadku wystąpienia błędu (np. nieznaną zmienną lub nieprawidłowy operator), system nie wykona polecenia, a do użytkownika nie zostanie zwrócony żaden komunikat. Możliwe jest jednocześnie modyfikowanie zawartości kilku zmiennych za pomocą sekwencji poleceń złożonej z pojedynczych oddzielonych znakiem &, np.:

```
<polecenie_1>&<polecenie_2>&
<polecenie_3>
```

Niestety, w ten sposób nie można odczytać danych z kilku zmiennych za pomocą pojedynczego zapytania.

Opisany wyżej sposób operowania zmiennymi przy użyciu poleceń przesyłanych przez port szeregowy jest zastosowany (z pewnymi ograniczeniami) dla poleceń zawartych w żądaniach napływających do serwera HTTP. Aby przesłać dane do serwera, przeglądarka wykorzystuje metodę GET z protokołu HTTP (opisany w RFC1945), w której parametry dla serwera są przesyłane razem z żądaniem (parametry występują za znakiem ?). Aby wysłać polecenie do serwera, w pasku adresu przeglądarki należy wpisać ciąg: *http://adr.ess.erw.era/bieżąca\_strona.html?<polecenie>*

Podobnie jak w przypadku RS232, można wysłać kilka poleceń jednocześnie (np. *http://192.168.1.100/index.html?temp=13&fan=1*). W przypadku odwoływania się do zmiennych łańcuchowych należy zaznaczyć, że protokół HTTP nie pozwala na użycie w składni żądania pewnych znaków (muszą one być kodowane jako %ASCII\_HEX). Serwer HTTP „rozumie” tak zakodowane dane, więc bez problemów można używać formularzy udostępnianych przez język HTML, co ułatwia tworzenie stron dla NET51. Niestety formularze nie umożliwiają używania innych

operatorów oprócz przypisania, co zmusza użytkownika do korzystania ze skryptów języka Java.

## Konfiguracja i użytkowanie

Na początku należy ustawić podstawowe parametry sieciowe systemu NET51 za pomocą terminala. W tym celu należy zawiązać zworę Z1 oraz włączyć zasilanie systemu, który nie zainicjalizuje części sieciowej i ustawi port szeregowy na 9600 bd. Następnie podajemy wartości parametrów: adres IP oraz maska podsieci itp. (powinny być uzgodnione z administratorem lokalnej sieci). Parametry sieciowe są traktowane jako wewnętrzne zmienne i są modyfikowane w sposób opisany w poprzedniej części artykułu. Oto poszczególne polecenia jakie należy wysłać do NET51:

```
ip=<adres IP np.192.168.1.100>
mask=<maska podsieci np.
255.255.255.0>
gate=<adres IP lokalnej bramy>
cmd=wrcfg
```

Ostatnie polecenie spowoduje przepisanie konfiguracji do pamięci EEPROM i przy ponownym włączeniu zasilania system skorzysta z wartości tam zapisanych.

W kolejnym kroku tworzone są strony HTML, w których dla „dynamicznej” zawartości wykorzystywane są zmienne zdefiniowane przez użytkownika. Aby w oknie przeglądarki pojawiła się zawartość danej zmiennej, należy w kodzie HTML umieścić nazwę tej zmiennej w nawiasach kwadratowych. Na przykład kod:

```
<P> Bieżąca temperatura wynosi
[temp] stopni. </P>
```

spowoduje wyświetlenie wartości aktualnej temperatury identyfikowanej zmienną *temp*. W nawiasach kwadratowych można podać dowolne poprawnie sformułowane polecenie, którego wynik będzie wstawiony w tym miejscu. Przykładem takiego użycia może być

**Tab. 2. Zestawienie typów zmiennych**

Typ zmiennej	Opis
DB	8-bitowa liczba bez znaku (zakres 0...255)
DW	16-bitowa liczba bez znaku (zakres 0...65535)
DS	łańcuch znaków o maksymalnej długości 128 znaków

konstruowanie pól formularzy (typu „radio“ i „check“) za pomocą operatora \$ np.

```
<INPUT TYPE="radio"
NAME="nazwa" VALUE=1
[nazwa$1]>
<INPUT TYPE="radio"
NAME="nazwa" VALUE=2
[nazwa$2]>
<INPUT TYPE="radio"
NAME="nazwa" VALUE=3
[nazwa$3]>
```

Innym przykładem może być dynamiczna wymiana grafiki na stronie w zależności od wartości zmiennej. I tak kod

```
<IMG SRC="led[ioa].gif"
BORDER=0>
```

spowoduje wyświetlenie pliku *led0.gif* lub *led1.gif* na podstawie stanu wejścia oznaczonego IOA.

Aby móc wykorzystać system NET51 należy dostarczyć definicje zmiennych oraz danych dla serwera HTTP, zawartych w specjalnym pliku o nazwie *image.dat*. Wymieniony plik jest wynikiem pracy programu *net51link.exe*, który dokonuje „kompilacji“ projektów definiowanych przez użytkownika. W celu utworzenia nowego projektu, należy na komputerze osobistym (w katalogu z danymi do projektu) utworzyć plik opisu projektu: *nazwa\_projektu.prj* o podanej niżej strukturze:

```
// tak wygląda komentarz
// dla całej linii
// tu zaczyna się sekcja
// zmiennych
[var]
deklaracja zmiennej 1
deklaracja zmiennej 2
deklaracja zmiennej 3
...
// a tu zaczyna się sekcja
// plików
[file]
// komentarz może być
// umieszczony w dowolnych
// liniach projektu
nazwa pliku 1
nazwa pliku 2
nazwa pliku 3
...
```

Jak widać jest on podzielony na dwie sekcje: zmiennych oraz pli-

## Dodatek A

### Zestawienie wewnętrznych zmiennych NET51

Zmienna	Dostęp	Typ	Opis
server	B	DS	serwera HTTP - adres IP (do pracy za firewallem)
request	R	DS	serwer HTTP - ostatnie żądanie skierowane do serwera
baud	B	DB	konfiguracja - prędkość portu szeregowego 1-9600; 2-19200; 3-57600 (aby ustawić port na wskazaną prędkość należy wydać komendę <i>cmd=setup</i> )
ip	B	DS	konfiguracja - adres IP systemu NET51 (np. 192.168.1.100)
mask	B	DS	konfiguracja - maska podsieci
gate	B	DS	konfiguracja - adres lokalnej bramy (do komunikacji z WEB)
smtp	B	DS	konfiguracja - adres serwera pocztowego
cmd	B	DS	system - sterowanie funkcjami serwera. Dostępne parametry <i>wrcfg</i> - zapis konfiguracji; <i>reset</i> - restart systemu; <i>setup</i> - ustawienie prędkości pracy UART na podstawie zmiennej <i>baud</i> .
asca	W	DS	system - port szeregowy dostępny z przeglądarki
rela	B	DB	system - przekaźnik A
relb	B	DB	system - przekaźnik B
ioa	B	DB	system - uniwersalne IO A
ioB	B	DB	system - uniwersalne IO B

ków. Sekcja plików składa się z nazw plików znajdujących się na dysku komputera PC, które powinny być dostępne dla serwera HTTP systemu NET51 (pliki \*.html, \*.gif, \*.jpg, \*.js itp.). Maksymalna liczba nazw plików wynosi 64, a ich łączny rozmiar nie może przekroczyć 30kB (z uwagi na pojemność pamięci EEPROM). Długość nazwy pliku nie może przekroczyć 12 znaków, licząc razem z rozszerzeniem i separującą kropkę. Natomiast sekcja zmiennych zawiera deklarację o następującej składni: *<typ>\_<dostęp>\_<nazwa>[wartość początkowa]*,

gdzie symbol „\_“ oznacza spację, a użycie wartości początkowej jest opcjonalne (domyślna wartość wynosi 0). Nazwa zmiennej może mieć maksymalnie 12 znaków i musi składać się wyłącznie z małych liter. Opisywany system obsługuje zmienne liczbowe traktowane jako 8- lub 16-bitowe liczby bez znaku oraz zmienne będące łańcuchami o maksymalnej długości 128 znaków - zestawienie typów zawiera **tab. 2**.

Dla każdej zmiennej można przypisać metodę dostępu oraz wartość początkową, która będzie wpisana po sprzętowym zerowaniu systemu. Metoda dostępu określa, czy zmienną można odczytać lub zapisać z poziomu przeglądarki internetowej (zestawienie w **tab. 3**).

Kompilacji projektu opisanego w powyższy sposób dokonujemy

wydając z linii komend polecenie *net51link.exe nazwa\_projektu.prj*, przy czym zakładamy, że *net51link.exe* znajduje się w tym samym katalogu co plik \*.prj. W wyniku wykonania powyższego polecenia w katalogu danych powstaje plik o nazwie *image.dat*, który następnie powinien być załadowany do NET51 za pomocą dowolnego programu FTP. Dodatkowo jest generowany plik *report.txt*, zawierający opis przebiegu kompilacji oraz spis napotkanych błędów. Po załadowaniu pliku należy zakończyć sesję FTP oraz dokonać restartu systemu ręcznie lub zdalnie za pomocą polecenia *cmd=reset* (zdalne zerowanie nastąpi po ok. 5 sekundach). Po tych operacjach NET51 jest gotowy do pracy.

**Paweł Ciesłowski**  
**pcieslowski@wp.pl**

#### Literatura

- „TCP/IP Illustrated Volume 1, The Protocols“, W. Richard Stevens
- „TCP/IP Illustrated Volume 2, The Implementations“, W. Richard Stevens, Gary Wright
- <http://www.ietf.org/rfc.html> - zbiór dokumentów RFC

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/czerwiec02.htm> oraz na płycie CD-EP06/2002B w katalogu PCB.

**Tab. 3. Zestawienie metod dostępu**

Metoda	Opis
R	zmienna tylko do odczytu
W	zmienna tylko do zapisu
B	połączenie R oraz B