

Basic dla mikrokontrolerów MCS-51, część 2



Kontynuujemy prezentację Tiny BASIC-a dla mikrokontrolerów rodziny MCS-51. W tej części artykułu omówione zostały dostępne polecenia, sposób obsługi portów I/O, najprostsze metody archiwizowania programów oraz wiele innych informacji niezbędnych do wydajnego korzystania z możliwości oferowanych przez interpreter.

Dostęp do portów i pamięci

Tiny Basic umożliwia prosty dostęp do portów, rejestrów procesora oraz każdego z typów pamięci: wewnętrznej RAM procesora, pamięci programu - wewnętrznej i zewnętrznej oraz zewnętrznego RAM-u. Dostęp ten jest możliwy dzięki zastosowaniu dodatkowych zmiennych ośmiobitowych (**tab. 1**). Za pomocą *DBYTE* można odczytywać i modyfikować wewnętrzną pamięć RAM procesora. Na przykład program:

```
10 FOR A=0 TO 127
20 PRINT A, DBYTE A
30 NEXT A
```

spowoduje wyświetlenie zawartości tej pamięci wraz z adresami. Gdy argument *DBYTE* będzie z zakresu 128..255, uzyskujemy dostęp do portów i rejestrów specjalnych. Podane adresy są identyczne z używanymi przez CPU w adresowaniu bezpośrednim. Tak

więc, na przykład wpisanie *PRINT DBYTE 144* spowoduje wyświetlenie na konsoli zawartość portu *P1*.

Z kolei polecenie *RBIT* umożliwia dostęp do pojedynczych bitów. Adresowanie jest następujące: jeśli adres portu *P1* dla *DBYTE* wynosi 144, to adres jego bitu o numerze zero (*P1.0*) wynosi właśnie 144, dalej *P1.1* ma adres 145, *P1.2* - 146 itd. Na przykład, wykonanie *RBIT 144 = RBIT 145 AND NOT RBIT 146* spowoduje

ustawienie *P1.0* tylko wtedy, gdy *P1.1=1* i *P1.2=0*.

Zmienna *XBYTE* umożliwia czytanie i zapisywanie zewnętrznej pamięci RAM (do 64 KB). Np. *XBYTE 16384=0,1,2,3,4,5* zapisze podanymi wartościami sześć kolejnych adresów RAM-u. W identyczny sposób *CBYTE* daje dostęp do pamięci programu (np. tablice stałych).

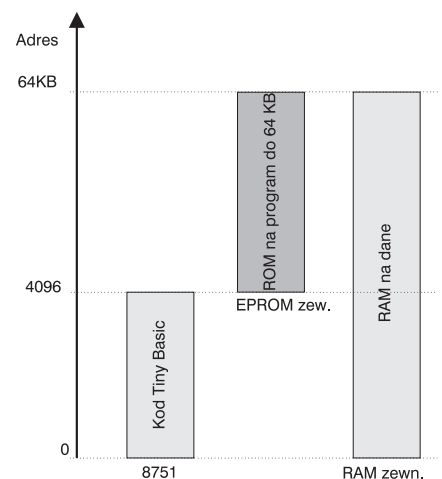
Instrukcje

Pełna lista instrukcji została zamieszczona w **tab.2**. W zasadzie ci Czytelnicy, którzy znają któryś z dialektów Basic-a nie będą mieli żadnych kłopotów z ich stosowaniem, gdyż składnia jest typowa.

Warto jednak krótko wspomnieć o dodatkowych możliwościach. Wykonanie polecenia *LET A=1,2,5,A,B,B+C* spowoduje przypisanie zmiennej *A* wartości 1, *B* wartości 2, *C* wartości 5, *D* wartości 1, *E* wart. 2 i *F* wart. 7. Jest to tzw. przypisanie sekwencyjne - wygodne i szybkie.

Łańcuchy znaków instrukcji *PRINT* mogą być ograniczane również apostrofem zamiast tradycyjnego górnego cudzysłowu. Tak więc komendy *PRINT "ALFA"* i *PRINT 'ALFA'* są równoważne. Aby wy-

BASIC 51



Rys. 1.

Tabela 1. Zależność pomiędzy nazwami rejestrów 8051 a wartościami zmiennej DBYTE.

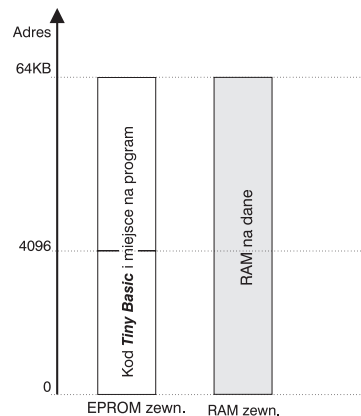
Nazwa rejestru	P0	P1	P2	P3	TCON	TMOD	TLO	TL1	TH0	TH1	SCON	SBUF	IE	IP
wartość DBYTE	128	144	160	176	136	137	138	139	140	141	152	153	168	184

Tabela 2. Zestawienie poleceń Tiny BASIC-a '51.

POLECENIE	OPIS
REM tekst	Komentarz, tekst za słowem REM jest ignorowany
LET zmienna = wyrażenie	Przypisuje wartość zmiennej, np. LET a=123-5*B.
PRINT wyrażenie, "tekst"	Wypisuje na ekranie terminala podany tekst lub obliczoną wartość wyrażenia, np. PRINT 2+3*A,"abcdef"
INPUT zmienna, zmienna	Wyświetla na ekranie dwukropek i czeka na wprowadzenie danych, przypisując je później w kolejności wskazanym zmiennym, np. INPUT A,B
INPUT "tekst", zmienna, "tekst", zmienna	Zamiast dwukropka, komenda wyświetla jako znak zachęty podany tekst, np. INPUT "Podaj dana A",A,"A teraz B",B
IF warunek THEN polecenie: polecenie	Testuje logiczną wartość warunku, gdy wynik jest prawdziwy, wykonywane są polecenia podane za słowem THEN, w kolejności aż do końca linii. Jeśli wartość logiczna warunku jest fałszem, polecenia za słowem THEN są ignorowane. Na przykład: IF B<A THEN PRINT "B mniejsze od A": INPUT B
GOTO wyrażenie	Powoduje przejście wykonywania programu do linii wyliczonej w polu wyrażenie. Adres skoku może być podany w formie jawnej, np. GOTO 125 lub niejawnej GOTO A*20+B. Uwaga polecenie GOTO musi być ostatnim poleceniem w linii programu.
GOSUB wyrażenie	GOSUB działa identycznie jak GOTO, z tym, że BASIC zapamiętuje numer linii, w której znajduje się ta instrukcja.
RETURN	Wykonanie RETURN powoduje powrót wykonywania programu do następnej linii, w stosunku do zapamiętanej w poleceniu GOSUB. Polecenie to musi być ostatnim w linii. Wywołania GOSUB - RETURN mogą być zagnieżdżane, głębokość zagnieżdżenia jest limitowana jedynie pojemnością stosu.
FOR zmienna = wyrażenie1 TO wyrażenie2	Rozpoczyna pętlę. Zmiennej przypisywana jest wartość wyrażenia 1 oraz zapamiętywana jest wartość wyrażenia 2. W następnej kolejności wykonywane są instrukcje zapisane za rozpoczęciem pętli, aż do napotkania rozkazu NEXT.
NEXT zmienna	Polecenie sprawdza wartość zmiennej będącej argumentem. Jeśli nie została przekroczona wartość zapamiętana po obliczeniu wyrażenia 2, wartość zmiennej zostaje powiększona o 1 i ciąg instrukcji zawartych między FOR a NEXT jest wykonywany ponownie. Przykład: 10 FOR A=1 TO B*2: PRINT A 20 NEXT A
END	Jest równoważnikiem ostatniej linii programu. Powoduje zatrzymanie pracy programu i powrót do pracy interakcyjnej. Polecenie to musi być ostatnim w linii. Ponadto wykonywanie programu może być przerwane przez naciśnięcie klawiszy CTRL-C a wyświetlanie danych może być chwilowo zatrzymane za pomocą CTRL-S.
CALL liczba	Umożliwia uruchomienie i wykonanie własnego podprogramu napisanego w kodzie maszynowym 8051 (wstawki) i umieszczonego w pamięci programu pod adresem "liczba". Program maszynowy musi kończyć się rozkazem RET (22H). Dla użytkownika udostępnione są (wewnątrz procesora): obszar RAM 20H do 27H, banki rejestrów 2 i 3 (ale nie w trybie <i>single chip</i>).
ROM	Po tej komendzie aktywacji ulega program zapisany w wewnętrznej pamięci ROM procesora. Ponieważ Tiny Basic zajmuje 3 KB w pozostałym kilobajcie można umieścić jakiś własny, zawsze potrzebny program. W wersji oryginalnej zawarty jest tam prosty przykład - gra. Takiego programu nie można oczywiście zmodyfikować.
RAM	Tryb dostępny po resecie. Program wpisywany z klawiatury umieszczony jest w zew. RAM od adresu 2000H
PROM	Działa podobnie jak ROM, z tym, że aktywuje program zawarty w zewnętrznym EPROM-ie. Początek programu użytkownika zaczyna się od adresu 1080H i może on mieć do 60 KB długości. Ten program jest automatycznie uruchamiany po resecie, gdy nóżka RxD procesora jest zwarta do masy.
HEX	Włącza tryb rozpoznawania i wyświetlania liczb w postaci szesnastkowej. Dodatkowo, zakres liczb, jakie są dostępne, zmienia się z zakresu -32767 .. +32767 do zakresu 0 ... 65535!
DECIMAL	Przywraca tryb dziesiętnej reprezentacji i zakres -32767.. 32767.
RESET	Wykonanie polecenia jest równoważne sprzętowemu resetowi sterownika.

Skrócona postać poleceń dopuszczalna w Tiny BASIC-u

POLECENIA					
C. CALL	D. DECIMAL	E. END	F. FOR	G. GOTO	GOS. GOSUB
H. HEX	I. IF	IN. INPUT	L. LET	LI. LIST	N. NEXT
NEW NEW	P. PRINT	PRO. PROM	R. RETURN	RA. RAM	RES. RESET
RO. ROM	RU. RUN	REM REMARK	T. TO	T. THEN	
FUNKCJE I ZMIENNE SPECJALNE					
A. ABS	A. AND	C. CBYTE	D. DBYTE	M. MOD	N. NOT
O. OR	R. RBIT	RN. RND	X. XBYTE	X. XOR	



Rys. 2.

konanie komendy PRINT nie powodowało przejścia do następnego wiersza, należy zakończyć ją przecinkiem (np. PRINT 'ALFA').

Linia programu w Basicu musi być poprzedzona numerem linii. Dopuszczalne są wszystkie wartości z zakresu 1..32767. W jednej linii programu można umieścić kilka instrukcji, pamiętając jednak, że instrukcje takie jak: GOTO, END, FOR i RETURN muszą być ostatnimi w danej linii.

Błędy

Komunikaty o błędach są najprawdopodobniej najsłabszą stroną Tiny Basica. Należy jednak pamiętać, iż całość kodu programu zajmuje niewiele ponad 3 KB, gdzie zatem musiały pojawić się ograniczenia - sygnalizowane są jedynie trzy typy błędów.

Komunikat WHAT? oznacza, iż użytkownik nie jest rozumiany przez program. Najczęstszą przyczyną jest złe wpisanie jakiejś komendy (np. PRITN 'ALFA').

Z kolei HOW? to komunikat informujący, że Tiny Basic nie potrafi wykonać jakiejś z operacji (skok do nieistniejącej linii, wynik obliczeń większy niż 32767 itp.).

Ostatni z komunikatów o błędach - SORRY - pojawia się, gdy zabraknie pamięci do wykonania żądanego polecenia.

Źle wpisane komendy można poprawić kasując źle napisane litery za pomocą klawisza DEL (ASCII 127). Maksymalna długość jednej linii programu wynosi 32

```
Listing 1. Ilustracja sposobu zapisania programu w pamięci EPROM.
10 REM CHOINKA
20 LET A=RND (255): DBYTE 144=A
30 GOTO 20
```

Tabela 3.

adres	0	1	2	3	4	5	6	7	...	12	13	14	15	16	17	18	...	44	45	46	47	48	49	50	51	52	53
wartość	0	10	82	69	77	32	67	72	...	65	13	0	20	76	69	84	...	0	30	71	79	84	79	32	50	48	255
znak			R	E	M		C	H	...	A				L	E	T	...			G	O	T	O		2	0	

znaki, gdyż taka jest wielkość bufora wejściowego.

Mini - czyli sam procesor

Minimalna konfiguracja sterownika sprowadza się do samego procesora 8751 lub 8752. Taka praca, bez dodatkowej pamięci RAM, wprowadza następujące ograniczenia:

1. Nie ma możliwości buforowania programu, nie można zatem w takiej konfiguracji tworzyć oprogramowania ani go poprawiać. Pojedyncze linie programu mogą jednak być wykonywane po wpisaniu ich z klawiatury. Ograniczenie to nie przeszkadza w automatycznym wykonywaniu programu umieszczonego w EPROM-ie.
2. Zmienne programu zostają umieszczone w pamięci RAM procesora w przestrzeni adresowej 10H..28H.

Taki program najlepiej jest umieścić w pamięci EPROM od adresu 1080H. Po odłączeniu terminala i zwróceniu do masy wprowadzenia RxD będzie on automatycznie uruchamiany po wyzerowaniu sterownika. Sposób zapisania programu przedstawiony został w tab. 3 (i list. 1).

Kłopoty z pamiętaniem

Uważni Czytelnicy zorientowali się już zapewne, że w przedstawionym zestawie instrukcji nie ma żadnych poleceń dotyczących zapamiętania programu, nad którym się aktualnie pracuje. Rzadko zdarza się, aby kompletne dzieło powstało podczas jednej sesji.

Najprostsze rozwiązanie jest banalne lecz męczące - nie wolno wyłączać zasilania sterownika. Można także użyć podtrzymania baterijnego dla zewnętrznego RAM-u.

Jeśli jednak chcemy program zapisać na dyskietce lub wydrukować (nawet po to, aby w końcu zaprogramować EPROM), należy skorzystać z możliwości, jakie oferują terminale. Większość z nich (ja używam TELIX-a) posiadają opcję *capture* (przechwyć), która pozwala zapisać do pliku to wszystko, co przyjdzie do terminala po złączeniu

RS. Wystarczy zatem włączyć przechwytywanie, wywołać komendę *LIST* w Basicu, zakończyć przechwytywanie i zapisać plik. Później dowolny edytor tekstów bez trudu usunie niepotrzebne znaki i w efekcie da gotowe dzieło.

W drugą stronę transmisję można zrealizować podobnie, trzeba jednak ustawić bardzo małą szybkość pracy terminala, aby sterownik nadażył z odbiorem danych.

W przypadku innych komputerów niż PC, problem wymiany danych wygląda podobnie. Nie ma przeszkód, aby program realizujący transmisję umieścić na stałe w pozostałym 1kB pamięci zawartej w procesorze, zastępując przykładową grę zawartą w oryginalnym opracowaniu.

Przykładowy sterownik i program

Gotowy sterownik można wykonać na dwa sposoby. W pierwszym kod Basicu umieszcza się w pamięci wewnętrznej procesora 8751, dodając ewentualnie w razie potrzeby zewnętrzny RAM (na dane i zmienne) i ROM (na gotowy program).

Mapę pamięci takiego sterownika przedstawia rys. 1. Można również użyć

kości 8752 i program użytkownika zapisać w jej ROM-ie (większym). Drugi sposób (według mnie najwygodniejszy w fazie testów i uruchamiania) polega na wykorzystaniu procesora 8031, do którego podłącza się minimum 8kB EPROM i RAM (patrz rys. 2). Można również skorzystać z kitu AVT-222, do którego na płycie uniwersalnej AVT-222/1 dołączymy RAM i układ MAX 232.

Na list. 2 przedstawiono program prostej gry. Aby ją uruchomić wystarczy w odpowiedzi na znak zachęty napisać dwie komendy: ROM i RUN. Miłej zabawy!

Robert Magdziak, AVT

Interpreter Tiny Basic jest na płycie CD-EP1, dostępnej w AVT w sprzedaży wysyłkowej.

Listing 2.

```

2101 PR."HERE'S HOW YOU PLAY ACEY-DEUCEY:"
2102 PR."I'LL DEAL TWO CARDS FACE UP."
2103 PR."YOU BET ON WHETHER THE NEXT CARD WILL HAVE"
2105 PR."A VALUE BETWEEN THE FIRST TWO."
2110 E=100
2190 PR.:PR."YOU NOW HAVE ";E;" DOLLARS."
2260 PR.:PR."HERE ARE YOUR NEXT TWO CARDS...":PR.:GOS.4000
2280 A=D:GOS.4000
2290 B=D:IF A>B B=A:A=D
2300 IF B-A<2 PR."MISDEAL":GO.2260
2310 IF B-A=12 PR.:PR."ACEY-DEUCEY - GOOD ODDS!!"
2650 PR.:IN."PLACE YOUR BET";F
2660 IF F>E PR."YOU'VE ONLY GOT ";E;" BUCKS LEFT,
TURKEY":GO.2650
2670 IF F=0 PR."CHICKEN SHIT!!":PR.
2680 IF F<0 F=-F:PR."I'LL ASSUME YOU MEANT $";F;".00"
2730 PR."AND YOUR LAST CARD IS A ... ";:GOS.4000
2910 IF D>A IF D<B E=E+F:PR."YOU WIN!!":GO.2190
2970 E=E-F:PR."SORRY, YOU LOSE.":IF E>0 GO.2190
3010 PR."FRIEND, YOU BLEW YOUR WAD."
3040 PR.:IN."TYPE 1 TO PLAY AGAIN",D:IF D=1 GO.2110
3060 END
4000 D=RND(13)+1:IF D<11 PR.D
4040 IF D=11 PR."JACK"
4050 IF D=12 PR."QUEEN"
4060 IF D=13 PR."KING"
4070 IF D=14 PR."ACE"
4080 RET.

```