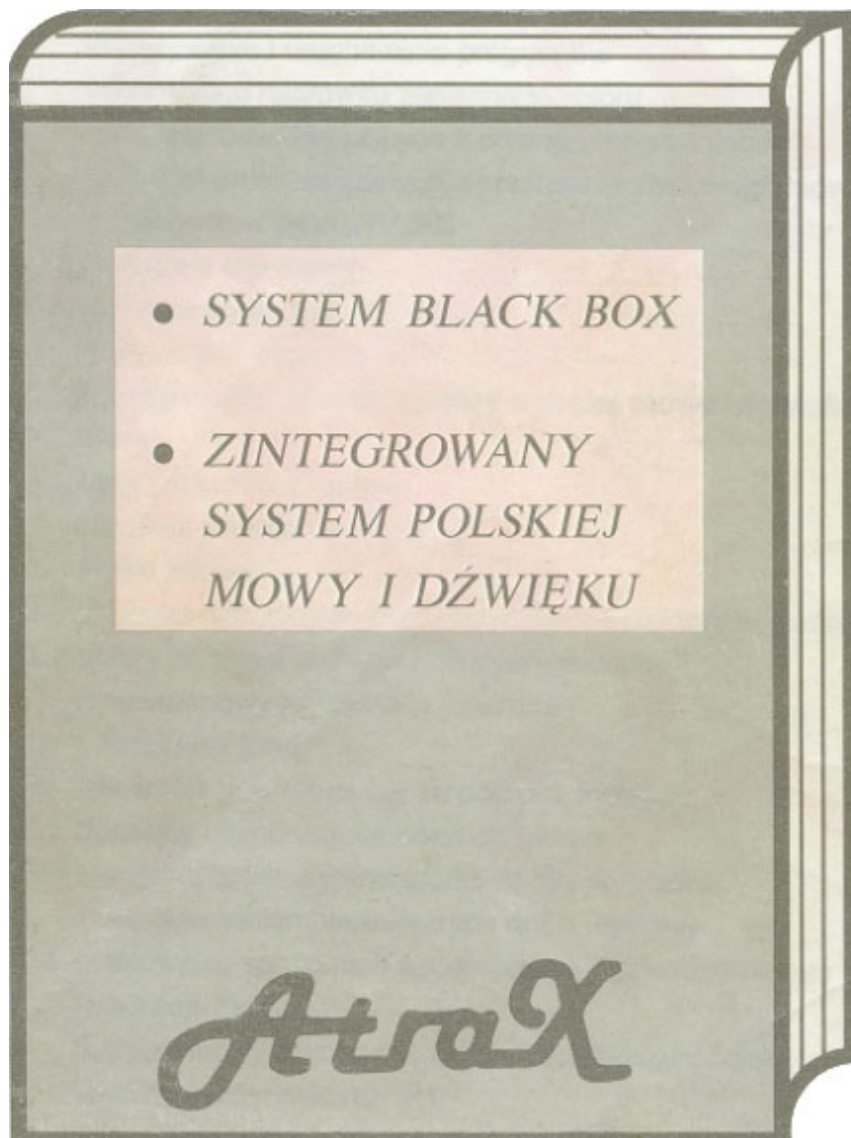


# BLACK BOX

## V. 8



## Spis treści

1. Informacje wstępne. ....	3
2. Uruchomienie cartridge. ....	3
3. SYSTEM BLACK BOX. ....	3
3.1. Użycie przycisku Reset. ....	3
3.2. Wczytywanie i uruchamianie programów. ....	3
3.3. Informacja o nagłówku zapisywanego zbioru. ....	3
3.4. Lista rozkazów związanych z obsługą magnetofonu. ....	4
3.5. Lista rozkazów związanych z przetwarzaniem programów napisanych w języku BASIC. ....	6
3.6. Pozostałe dyrektywy. ....	7
3.7. Klawisze funkcyjne. ....	8
3.8. Assembler – monitor. ....	8
4. ZINTEGROWANY SYSTEM SYNTEZY POLSKIEJ MOWY I DŹWIĘKU. ....	12
4.1. Wstęp. ....	12
4.2. Lista rozkazów syntezy mowy z opisem. ....	12
4.3. Czytanie polskich tekstów. ....	14
4.3.1. Polski alfabet. ....	14
4.3.2. Czytanie cyfr, działań matematycznych i niektórych skrótów. ....	14
4.3.3. Uwagi na temat akcentu i przestankowania. ....	15
4.4. Synteza mowy za pomocą fonemów. ....	15
4.4.1. Oznaczenia fonemów. ....	16
4.4.2. Realizacja polskiej mowy za pomocą fonemów. ....	16
4.4.2.1. Sposoby otrzymywania polskich głosek. ....	16
4.4.2.2. Uwagi na temat wypowiedziania tekstu po polsku, z uwzględnieniem ważniejszych norm wymowy. ....	17
4.4.2.3. O akcencie, sposobach akcentowania i przestankowaniu. ....	19
4.5. Realizacja śpiewu. ....	21
4.6. Synteza mowy uruchamiana językiem maszynowym. ....	22
4.7. Lista rozkazów muzycznych. ....	22

## 1. Informacje wstępne.

Black Box v.8 to cartridge dwusystemowy zawierający Black Box jako system podstawowy oraz ZINTEGROWANY SYSTEM SYNTEZY MOWY I DŹWIĘKU. Uruchomienie drugiego systemu możliwe jest przez rozkaz  $\Leftarrow\Leftarrow\Leftarrow$  lub reset z przytrzymanym klawiszem RUN/STOP albo klawisz funkcyjny F4.

## 2. Uruchomienie cartridge.

Cartridge należy włożyć do gniazda „expansion port” przy wyłączonym zasilaniu komputera, właściwą stroną (etykietką) do góry. Niezastosowanie się do tych zaleceń może spowodować uszkodzenie cartridge’a lub komputera. Po krótkiej chwili od momentu włączenia zasilania pojawia się plansza. Po przyciśnięciu spacji powinien się zgłosić system „C-64 BASIC IMPROVED BY BLACK BOX V8”. Wszystkie rozkazy BLACK BOX rozpoczynające się strzałką są teraz dostępne. Jeżeli jednak wraz z włączeniem zasilania przyciśniemy klawisz „COMMODORE” i przytrzymamy go przez około sekundę, to zgłosi się od razu system BLACK BOX V8. Włączenie zasilania z jednoczesnym przetrzymaniem klawisza RUN/STOP powoduje przejście do drugiego systemu. Podobnie, gdy wraz z włączeniem zasilania przyciśniemy spację i przytrzymamy ją przez około jedną sekundę, to zostanie wywołana procedura „normalnego” resetu i zgłosi się system „COMMODORE 64 BASIC V2”. Komputer zachowuje się teraz tak, jak gdyby w ogóle nie było cartridge’a. Jeśli jednak wykonamy SYS 277, to od tego momentu rozkazy „LOAD”, „SAVE” i „VERIFY” będą działać w systemie TURBO. Przyciśnięcie RUN/STOP + RESTORE przywraca normalny sposób działania tych rozkazów.

## 3. SYSTEM BLACK BOX.

### 3.1. Użycie przycisku Reset.

Cartridge ma zamontowany z tyłu przycisk „RESET”. Możliwych jest pięć sposobów wykonania resetu:

1. Użycie tylko przycisku „RESET” powoduje pojawienie się planszy i krótkiej informacji o cartridge’u. Przyciśnięcie spacji powoduje wejście do systemu BLACK BOX. Ten sposób resetu kasuje zawartość pamięci programu komputera.

2. Użycie przycisku „RESET” wraz z klawiszem „COMMODORE” powoduje wywołanie procedury szybkiego, poprawionego resetu, a następnie zgłasza się system „C-64 BASIC IMPROVED BY BLACK BOX V8”. Ten sposób resetowania nie niszczy programów maszynowych znajdujących się w pamięci, a programy napisane w Basicu łatwo odzyskać wydając rozkaz  $\Leftarrow U$  (unnew); jest on, więc szczególnie zalecany.

3. Reset z przytrzymanym klawiszem RUN/STOP powoduje przejście do drugiego systemu.

4. Reset z przytrzymaną spacją powoduje inicjalizację systemu „COMMODORE BASIC V2”. Od tej chwili komputer zachowuje się jak gdyby nie było cartridge’a. Możliwe jest tylko przyspieszenie procedur „LOAD”, „SAVE” i „VERIFY” poprzez SYS 277 (patrz dyrektywa  $\Leftarrow E$ ).

5. Jeżeli w momencie użycia przycisku „RESET” przytrzymamy klawisz CTRL, to nastąpi wejście do gry lub innego programu uruchamiającego się przez RESET, np. EXTRA-64, SYNTHIMAT, MUSIC MACHINE, QUANGO, EXTENDED BASIC LEVEL 2, itp. (oczywiście, jeżeli tego rodzaju program był poprzednio wczytany).

### 3.2. Wczytywanie i uruchamianie programów.

Programy można wczytywać za pomocą rozkazu  $\Leftarrow L$ , standardowej instrukcji LOAD lub klawisza funkcyjnego F5 (sposób zalecany).

Uruchomienie programów następuje przy użyciu standardowej instrukcji RUN lub poprzez klawisz funkcyjny F3. Zaleca się uruchamianie gier za pomocą klawisza funkcyjnego F3.

### 3.3. Informacja o nagłówku zapisywanego zbioru.

Na początku każdego zapisywanego na taśmie programu znajduje się nagłówek (header), w którym jest umieszczona nazwa programu, jego adres startowy i końcowy oraz typ nagłówka (headertype).

Adresy (startaddress i endaddress) wskazują na obszar pamięci, z którego dany zbiór (program) został zapisany na taśmę. Adres startowy oznacza zawsze pierwszą komórkę należącą do tego obszaru pamięci, a adres końcowy – pierwszą komórkę występującą już poza zapisywanym obszarem (tzn. ostatni bajt zapisywanego zbioru ma numer równy endaddress – 1).

Większość programów napisanych w BASIC’u, a także programy maszynowe (gry) startujące z BASIC’u (tj. uruchamiające się rozkazem RUN) mają zwykle adres startowy równy 2049. Pozostałe programy maszynowe, niektóre programy napisane w BASIC’u (przy zmienionych wskaźnikach w komórkach 43 i 44) i inne zbiory mogą mieć adresy startowe różniące się od tej wartości.

Typ nagłówka (headertype) może mieć dwie wartości „1” albo „3”. Jeżeli typ nagłówka równa się „3”, to zapisany zbiór będzie się zawsze wczytywać pod właściwy adres startowy, bez względu na sposób wczytywania. Ten typ nagłówka powinny mieć w zasadzie wszystkie programy maszynowe, których adres startowy nie jest równy 2049.

Jeżeli typ nagłówka ma wartość „1”, to lokalizacja wczytywanego programu będzie zależeć od sposobu wczytywania. W ten typ nagłówka powinny być wyposażone wszystkie programy napisane w BASIC’u; headertype i mogą mieć również programy maszynowe startujące z BASIC’u (np. wszelkie gry uruchamiające się rozkazem RUN).

„Headertype 3” uzyskuje się zapisując program z tzw. „adresem wtórnym”, np.  $\Leftarrow$ S”NAZWA”,1,1 lub SAVE”NAZWA”,1,1. Typ nagłówka „1” można otrzymać zapisując dany zbiór bez adresu wtórnego, np.  $\Leftarrow$ S”NAZWA” lub SAVE”NAZWA”.

### 3.4. Lista rozkazów związanych z obsługą magnetofonu.

$\Leftarrow$ @ - COPY – kopiowanie na taśmę ostatnio wczytanego zbioru z taśmy lub z dysku. Z taśmy można skopiować prawie wszystkie istniejące programy zapisane w turbo (również maszynowe) z wyjątkiem programów z autostartem. Z dysku można w ten sposób przenosić na taśmę wszystkie programy jednoczęściowe uruchamiające się rozkazem RUN o długości do 152 bloków. Wczytany zbiór zostanie zapisany na taśmie pod tą samą nazwą. Możliwe jest kopiowanie wielokrotne. Kopiowanie powinno następować bezpośrednio po wczytaniu. Przed rozpoczęciem kopiowania i przed każdą zmianą sposobu wczytywania należy wyresetować komputer. Rozkaz COPY jest przyporządkowany klawiszowi funkcyjnemu F7.

$\Leftarrow$ HF – HEAD FIT – dyrektywa uruchamia program do ustawiania głowicy. Do magnetofonu należy włożyć taśmę z wzorcowym nagraniem, wcisnąć „PLAY” i wybrać odpowiedni system. Głowicę należy ustawić tak, aby linie ukazywały się tylko w dwóch sektorach i miały minimalną grubość.

$\Leftarrow$ L – LOAD – odczytywanie programów z taśmy. Rozkaz działa „z klawiatury” i „z programu” (z autostartem). W czasie wyświetlania danych z nagłówka można przerwać procedurę przyciskając klawisz RUN/STOP. W pozostałych sytuacjach należy użyć RUN/STOP + RESTORE. Sposoby zastosowania rozkazu:

$\Leftarrow$ L – jest to najbardziej typowy sposób ładowania programów (zbiorów) do komputera. Pierwszy napotkany na taśmie zbiór będzie wczytany do pamięci komputera. Jeżeli zbiór ten ma „headertype 1” to będzie zawsze wczytany do obszaru pamięci przeznaczonego dla BASIC’u (zwykle począwszy od adresu 2049), bez względu na adres startowy i końcowy. Jeżeli „headertype” programu wynosi 3, to program zostanie wczytany do obszaru pamięci wyznaczonego przez „startaddress” i „endaddress”.

$\Leftarrow$ L”NAZWA” – program o zadanej nazwie zostanie wczytany do pamięci komputera, w sposób jak wyżej. Zamiast pełnej nazwy można podać skrót, opuszczając znaki od końca (nigdy od początku !), np.  $\Leftarrow$ ”NAZWA”,  $\Leftarrow$ ”NA”,  $\Leftarrow$ L”N”.

$\Leftarrow$ L”NAZWA”,1,1 – dyrektywa ta powoduje wyszukanie i wczytanie programu zapisanego na taśmie zawsze do obszaru pamięci określonego przez adres startowy i adres końcowy, bez względu na headertype. Możliwe jest skracanie nazwy w sposób podany poprzednio lub niepodawanie nazwy:  $\Leftarrow$ L””,1,1. Rozkaz w tej postaci jest przyporządkowany klawiszowi funkcyjnemu F5.

⇐M – MERGE – ta dyrektywa powinna być stosowana tylko do programów napisanych w BASIC’u i zapisanych na taśmie z headertype 1. Instrukcja działa podobnie jak ⇐L, z tą różnicą, że jeśli w pamięci komputera znajdował się jakiś program napisany w BASIC’u, to nie zostanie on usunięty, ale do tego programu zostanie dołączony następny doczytywany program. Wskazane jest, aby program doczytywany miał numery linii wyższe, niż poprzedni, w przeciwnym razie należy dokonać przenumerowania (p. instrukcja ⇐R – RENUMBER). Sposób zastosowania: ⇐M lub ⇐M”NAZWA”.

⇐V – VERIFY – weryfikacja programu. Sposoby użycia rozkazu:

⇐V – weryfikacja pierwszego napotkanego programu. Jest to najbardziej typowy sposób sprawdzania poprawności zapisu na taśmie. Można go stosować do wszystkich programów (zbiorów), których adres startowy wynosi 2049, bez względu na headertype oraz do wszystkich programów (zbiorów), których headertype wynosi 3, bez względu na adres startowy.

⇐V”NAZWA” – rozkaz powoduje wyszukanie na taśmie zbioru o nazwie „NAZWA” i weryfikuje go. Sposób i zakres działania rozkazu – jak wyżej.

⇐V”NAZWA”,1,1 – zbiór o nazwie „NAZWA” będzie wyszukany i zweryfikowany. Dyrektywa ta jest najbardziej uniwersalna, ponieważ weryfikuje wszystkie możliwe zbiory, bez względu na headertype, startaddress i endaddress.

⇐£ - VERIFY – weryfikacja ostatnio zapisanego programu. Jest to najszybszy, najpewniejszy i najlepszy sposób weryfikacji. Ten rozkaz jest przyporządkowany klawiszowi funkcyjnemu F8.

⇐S – SAVE – zapisywanie programów. Rozkaz działa „z klawiatury” i „z programu”. Sposoby użycia:

⇐S”NAZWA” – zbiór o nazwie „NAZWA” zostanie zapisany na taśmie. Jest to najbardziej typowy sposób zapisu programów. Uzyskuje się zapis z „headertype 1”. W ten sposób należy zapisywać i kopiować wszystkie programy napisane w BASIC’u, jak również wszelkie gry (i inne programy maszynowe) startujące „z BASIC’u”, tj. uruchamiające się rozkazem „RUN”. Rozkaz ten jest przyporządkowany klawiszowi funkcyjnemu F6.

⇐S”NAZWA”,1,1 – zbiór o podanej nazwie zostanie zapisany na taśmie z „headertype 3”.

⇐W - WRITE – zapisywanie programów maszynowych, ciągu bajtów, obrazów grafiki HI-RES i innych zbiorów. Rozkaz jest użyteczny szczególnie dla programujących w języku maszynowym. Za pomocą tej dyrektywy można zarejestrować dowolny fragment lub całą pamięć komputera (pełne 64kB). Przy zapisywaniu obszaru od komórki nr 0 do 817 (wskaźniki systemu operacyjnego, stos, wektory) wskazana jest ostrożność i dobra znajomość systemu operacyjnego komputera. Sposoby wykorzystania:

⇐W, adres startowy, adres końcowy „NAZWA” – zapisywanie zbioru określonego przez adres startowy (nr pierwszej komórki zbioru) i adres końcowy (nr pierwszej komórki poza zbiorem; nr ostatniego zapisywanego bajtu jest, więc o jeden mniejszy). Nagłówek jest typu „1”.

⇐W, adres startowy, adres końcowy ”NAZWA”,1,1 – podobnie jak poprzednio, z tym, że headertype wynosi „3”.

**UWAGA!** Adres końcowy powinien być liczbą całkowitą o wartości do 65535, większą niż adres startowy. Od tej reguły należy odstąpić tylko w jednym przypadku, jeżeli chcemy zapisać obszar zawierający ostatnią komórkę pamięci o numerze 65535, to wówczas wyjątkowo jako adres końcowy wpisujemy zero.

Przykłady:

⇐W, 49152, 53248 ”PROGRAM MASZ”,1,1 – zapisanie programu maszynowego,

⇐W, 40960, 49152 "TEKST",1,1 – zapisanie RAM'u w obszarze adresowanym interpretera BASIC'u,  
⇐W, 57344, 0 „OBRAZEK”,1,1 – zapisanie grafiki HIRES w obszarze adresowym KERNALA,  
⇐W, 1024, 2024 „EKTRAN”,1,1 – zapisanie (z programu) obszaru ekranu LORES (znakowego),  
⇐W, 1024, 0 „MEMORY”,1,1 – zapisanie całego użytecznego obszaru pamięci komputera.

### 3.5. Lista rozkazów związanych z przetwarzaniem programów napisanych w języku BASIC.

⇐D – DELETE – usuwanie wierszy z programów napisanych w BASIC'u. Sposób wykorzystania: ⇐D, dolna granica usuwanego obszaru, górna granica obszaru. Np. ⇐D, 200, 780 – wiersze o numerach mieszczących się w przedziale od 200 do 780 będą usunięte, ⇐D, 0, 400 – wszystkie wiersze o numerach nie większych niż 400 będą usunięte.

⇐F – FIND WORD – poszukiwanie zadanego słowa. Sposób wykorzystania: ⇐F „SŁOWO”. Będą podane wszystkie numery wierszy zawierające SŁOWO (jeśli istnieją). W ten sposób można poszukiwać wyrazy zawarte w łańcuchach, po REM'ach, a także zmiennych.

⇐KR – KILLER OF REM – dyrektywa ta usuwa wszystkie teksty występujące po REM w programach napisanych w BASIC'u, przez co zmniejsza się długość programu. Jeżeli dany wiersz zawiera tylko REM, to jest usuwany w całości. Do takiego wiersza nie należy kierować skoków typu GOTO lub GOSUB. Podawana jest liczba bajtów (długość) programu przed i po operacji. Wykonanie kolejno ⇐KR, ⇐KS, ⇐P znacznie skraca programy.

⇐KS – KILLER OF SPACE – rozkaz usuwa wszystkie zbędne spacje (również w tekstach po REM), przez co zmniejsza się długość programu. Nie są usuwane spacje w łańcuchach. Wykonanie kolejno ⇐KR, ⇐KS, ⇐P znacznie skraca program.

⇐X – EXPEND – przywracanie spacji (odwrotnie do ⇐KS).

⇐P – PACKAGE – dyrektywa zwiększa gęstość upakowania programu, przez co skraca się jego długość. Program skraccany może zawierać maksymalnie 255 skoków typu GOTO lub GOSUB. Bardziej rozbudowane programy można skracać partiami. Rozkaz jest nieodwracalny i po jego wykonaniu nie można już zmieniać tak skróconego programu. ⇐KR, ⇐KS, ⇐P znacznie skraca program.

⇐U – UNNEW – ratowanie (przywracanie) programów napisanych w BASIC'u po wykonaniu instrukcji NEW lub po resecie z klawiszem COMMODORE.

⇐R – RENUMBER – zmiana numeracji programu napisanego w BASIC'u. Procedura renumeracji uwzględnia i zmienia adresy wszystkich skoków typu GOTO lub GOSUB. Sposób użycia rozkazu: wypisujemy ⇐R i przyciskamy RETURN. Program pyta się o numer pierwszej wiersza i o skok (przyrost numeracji). W przypadku wystąpienia błędu wyświetlany jest odpowiedni komunikat.

⇐A – AUTONUMBER – automatyczna numeracja wierszy. Sposób zastosowania: ⇐A, nr 1-go wiersza, skok. Np. ⇐A, 100, 10. Wartość skoku nie może przekraczać 255. Wyłączenie trybu „AUTONUMBER” następuje poprzez wydrukowanie „pustego wiersza” (tj. do kolejnego zaproponowanego wiersza nic nie wpisujemy i przyciskamy RETURN). Program jednak pamięta numer wiersza, przy którym wyszliśmy z trybu „AUTONUMBER”. Wykonanie ⇐A bez podawania liczb spowoduje kontynuację autonumeracji począwszy od numeru tego wiersza.

⇐I – INFO – informacja o ważnych dla BASIC'u adresach. Podawane są kolejno:  
adres początku programu napisanego w BASIC'u,  
adres końca programu napisanego w BASIC'u,  
adres końca obszaru zmiennych nieindeksowanych,  
adres końca obszaru dla zmiennych indeksowanych,

adres początku obszaru zmiennych łańcuchowych,  
adres końca obszaru przeznaczonego dla BASIC'u,  
liczba wolnych bajtów.

### 3.6. Pozostałe dyrektywy.

⇐CO – COLOR – zmiana kolorów ekranu. Sposób wykorzystania:

⇐CO, nr koloru tła, nr koloru ramki; numery kolorów jak w instrukcji do COMMODORE. Np. ⇐CO, 12, 15.

⇐CP – CURSOR POSITION – ustawienie kursora. Sposób wykorzystania: ⇐CP, nr kolumny, nr wiersza. Nr kolumny = 0-39, nr wiersza = 0-24. Ta dyrektywa jest szczególnie użyteczna „z programu”. Po tym rozkazie może nastąpić np. rozkaz PRINT lub INPUT. Np. ⇐CP, 20, 12: PRINT A\$.

⇐HE – HELP – listowanie rozkazów systemu BLACK BOX.

⇐YK, ⇐NK – YES KEY, NO KEY – włączanie i wyłączanie klawiszy funkcyjnych.

⇐MO – włączenie monitora. Opis rozkazów monitora jest przedstawiony w dalszej części niniejszego opracowania.

⇐YR, ⇐NR – YES REPEAT, NO REPEAT – jednokrotne naciśnięcie i przytrzymanie danego klawisza powoduje (YES) lub nie (NO) jego wielokrotne drukowanie.

⇐YS, ⇐NS – YES SCREEN, NO SCREEN – włączanie lub wyłączanie ekranu. Wyłączenie ekranu przyspiesza obliczenia.

⇐YT, ⇐NT – YES TONES OF MUSIC, NO TONES OF MUSIC – włączanie lub wyłączanie klawiatury muzycznej (muzyczny „BIP”). Klawiatura C-64 zamienia się w 3-oktawową klawiaturę umożliwiającą grę jedno-, dwu- lub trójgłosową. Przyciśnięcie SHIFT lub SHIFT LOCK zwiększa wysokość dźwięku o dwie oktawy. W sumie możliwe jest wydobywanie dźwięków pojedynczych w zakresie 5 oktaw oraz większości dwu- i trójdzwięków. Procedura działa na zasadzie przerwa, możliwa jest równoczesna gra i wykonywanie programów napisanych w BASIC'u.

⇐Q – QUIT – wszystkie rozkazy systemu „BLACK BOX V8” zostają wyłączone. Od tej chwili komputer działa tak, jak gdyby w ogóle nie było cartridge'a. Na ogół rozkaz ten nie jest potrzebny. W nielicznych przypadkach należy go wykonać przed uruchomieniem niektórych gier lub rozszerzeń BASIC'u. Powrót do systemu „BLACK BOX V8” możliwy jest poprzez RESET (sposób najpewniejszy) lub SYS 268.

⇐E – EXIT – wszystkie rozkazy systemu „BLACK BOX V8” rozpoczynające się od strzałki zostają wyłączone. Równocześnie standardowe rozkazy „LOAD”, „SAVE” i „VERIFY” działają teraz w systemie TURBO w sposób identyczny jak w ABC-TURBO. Przyciśnięcie RUN/STOP i uderzenie w RESTORE wyłącza ten sposób pracy i przywraca standardowy sposób działania tych rozkazów. SYS 277 powoduje, że rozkazy „LOAD”, „SAVE” i „VERIFY” znowu działają w systemie TURBO. Podczas wykonywania tych rozkazów nie są podawane adresy i typ nagłówka. Zaletą tego rodzaju pracy jest kompatybilność rozkazów z CBM BASIC V2 przy równoczesnym 10-krotnym wzroście szybkości zapisu i odczytu. Ten sposób jest zalecany przy wczytywaniu i uruchamianiu gier. Rozkazy działają „z klawiatury” i „z programu”. Instrukcja „LOAD” wykonywana „z programu” działa z autostartem. W kolejnych autostartach nie jest wykonywany rozkaz CLR, możliwe jest, więc zastosowanie tej instrukcji w tzw. loaderach.

⇐⇐⇐ - przejście do drugiego systemu (syntezy mowy).

### 3.7. Klawisze funkcyjne.

Włączenie klawiszy funkcyjnych następuje za pomocą rozkazu  $\Leftarrow$ YK lub po resecie. Wyłączenie – rozkazem  $\Leftarrow$ NK lub poprzez RUN/STOP + RESTORE. Przed uruchomieniem programów zaleca się wyłączyć klawisze funkcyjne.

Klawiszom tym przyporządkowano następujące rozkazy:

F1	– LIST
F3	– $\Leftarrow$ Q : RUN
F5	– odczyt z taśmy w turbo
F7	– kopiowanie ostatnio wczytanego programu
F2	– monitor
F4	– przejście do drugiego systemu
F6	– zapis na taśmę w turbo
F8	– weryfikacja ostatnio zapisanego programu

Ponadto, gdy włączone są klawisze funkcyjne, możliwe jest wychodzenie z „QUOTE MODE” i „INSERT MODE” poprzez przyciśnięcie RUN/STOP.

### 3.8. Assembler – monitor.

Assembler – monitor włącza się za pomocą rozkazu  $\Leftarrow$ MO lub poprzez klawisz funkcyjny F2. Powrót do systemu BLACK BOX: X i RETURN.

**UWAGA:** Przy wprowadzaniu adresów: adres początkowy to pierwszy adres w interesującym nas obszarze pamięci, adres końcowy to pierwszy następny poza tym obszarem.

Wszystkie parametry mogą być rozdzielane dowolną liczbą spacji z wyjątkiem parametrów w rozkazach FA, L, S, ?

#### Lista rozkazów z opisem

##### A aaaa

Wywołanie assemblera: aaaa – adres startowy (początkowy) programu. Przy wypisywaniu adresów przy rozkazach nie trzeba używać symbolu \$ oznaczającego liczbę heksadecymalną, np.:

```
LDA 1C2B
LDX #12
BNE 1234
LDA (12),Y
LDA (12),X
```

Gdy wystąpi błąd, kursor przeskakuje z powrotem za adres. Jeżeli chcemy zmienić zakończony już wiersz (np. zmienić adres lub instrukcję), musimy ustawić kursor w miejsce, które ma być zmodyfikowane. Po dokonaniu poprawki trzeba nacisnąć RETURN tyle razy, aż kursor dojdzie do miejsca, w którym przerwaliśmy pisanie. Jeżeli mamy wykonać skok, a nie jest jeszcze znany adres docelowy, możemy zamiast adresu wstawić etykietę. Etykieta składa się z litery „Z” i z liczby szesnastkowej od 00 do 3E, np.

```
2005 JMP Z03
```

Gdy „dotarliśmy” już do adresu skoku musimy przed rozkazem zapisać etykietę, np.:

```
2048 Z03 LDA #4
```

Poszczególne bajty, nie będące rozkazami ani adresami rozkazów (np. krótkie tabele) assembler przyjmuje poprzez postawienie kropki przed bajtem, np.:

357A.FF

**UWAGA:** Etykietować można nie tylko skoki bezwzględne typu JSR czy JMP, ale również skoki względne: BPL, BMI, BNE, BVC, BVS, BCC, BCS, BEQ.

**UWAGA:** Gotowy program kończymy literą „F” i przyciskamy RETURN, tym samym będą wyliczone wszystkie długości skoków i adresy.

### B aaaa bbbb

Bajty zawarte w obszarze pamięci od aaaa do bbbb zostaną umieszczone jako wiersze programu napisanego w BASIC’u (w połączeniu z rozkazem DATA). Program jest kreowany automatycznie, pierwszy wiersz danych ma numer 32000. Wszystkie dane są przedstawione jako liczby dziesiętne. Dane te mogą teraz być (lub już są) dołączone do programu napisanego w BASIC’u i (po uruchomieniu tego programu) ponownie wczytane za pomocą instrukcji READ, a następnie umieszczone za pomocą POKE w odpowiednim obszarze pamięci.

Rozkaz ten pozwala na umieszczenie podprogramów maszynowych w programach napisanych w BASIC’u i na start tych podprogramów „z BASIC’u”.

### C aaaa bbbb cccc dddd eeee

Rozkaz służy do zmiany lokalizacji (przesuwania) podprogramów lub całych programów maszynowych. Realizuje dwie funkcje jednocześnie: po pierwsze przepisuje (przesuwa) program z jednego obszaru pamięci do drugiego, po drugie przelicza wszystkie adresy skoków bezwzględnych JSR lub JMP „dopasowując” je do nowego obszaru pamięci, w którym teraz znajduje się program. Znaczenie poszczególnych adresów jest następujące:

aaaa adres początku obszaru (podprogramu), który ma być przesunięty,  
bbbb adres końca tego obszaru (minus jeden),  
cccc adres początku obszaru, do którego ma być przesunięty program,  
dddd adres początkowy całego programu,  
eeee adres końca całego programu.

**UWAGA:** Należy bardzo uważać, aby adres początku obszaru, który ma być przesunięty, był rzeczywiście adresem pierwszego bajtu rozkazu maszynowego. Jeżeli się pomylimy i np. pozostawimy spację (120 w kodzie ASCII) wówczas zostanie ona „zinterpretowana” jako rozkaz JSR i cała „interpretacja” pozostałych bajtów programu, jak również i wszystkie przeliczenia adresów będą błędne.

**UWAGA:** Opisanym sposobem należy przesuwać tylko programy maszynowe (napisane w języku assemblera) bez tabel. Tabele zawierające dane nie będące rozkazami maszynowymi należy przesuwać za pomocą rozkazu „W”.

### D aaaa lub D aaaa bbbb

Disassembler służy do badania RAM’u pod kątem programów napisanych w języku maszynowym. Przedstawia rozkazy assemblera w kodzie mnemonicznym, wyjaśnia również sposób adresowania. Możliwe jest wprowadzenie zmian (poprawek) do disassembledowanego programu.

disassemblacja pojedynczych wierszy – SPACE,  
ciągła disassemblacja/zatrzymanie – RETURN,

zakończenie disassemblacji – RUN/STOP.

**UWAGA:** Jeżeli przy wywoływaniu disassemblera zamiast litery „D” podamy SHIFT + „D”, to disassembled tekst będzie wyprowadzony na drukarkę. Nie należy zapomnieć jednak przedtem ustawić drukarki za pomocą rozkazu „P”.

#### F ss...ss aaaa bbbb

Poszukiwanie określonej kombinacji bajtów w obszarze od aaaa do bbbb, np. F C3 C2 CD 38, 30, 3000 A000 – poszukiwanie CBM 80.

**UWAGA:** Po F musi być spacja, a przed adresem – przecinek.

Po znalezieniu szukanej kombinacji bajtów drukowany jest adres pierwszego bajtu. Gdy przeszukujemy wg wielu kryteriów, nieistotne części bajtów (lub całe bajty) można zastąpić gwiazdką (lub dwoma gwiazdkami), np.: F A\* \*\* B\* AC, gdy szukamy A0 FF B4 AC, A5 78 B5 AC, itd.

#### Faxxxx aaaa bbbb

Dyrektywa ta powoduje wyszukiwanie wszystkich 3-bajtowych rozkazów, których operandami jest określony adres xxxx, w obszarze od aaaa do bbbb. Znalezione rozkazy są disassembledowane. Np. FA1234 20000 2100 będzie poszukiwać i disassembledować rozkazy.

LDA 1234 STA 1234,X CMP 1234,Y JSR 1234, JMP (1234), itd. W obszarze od aaaa do bbbb.

**UWAGA:** Po FA nie może być spacji. Części bajtów z adresu xxxx można zastąpić gwiazdką, np. FAB\*\*\*\*7A00 CFFF

#### FZxx aaaa bbbb

Poszukuje rozkazów zaadresowanych wg formuły ZERO PAGE, np. FZ12 1200 1337.

**UWAGA:** Po FZ nie ma spacji.

#### FRxxxx aaaa bbbb

Poszukuje skoków względnych (np. BEQ, BNE itp.) do określonego adresu xxxx. Przykłady: FR350A 3100 3600, FR51\*\* 5080 6233.

**UWAGA:** Po FR nie ma spacji.

#### Fixx aaaa bbbb

Poszukuje operandów z operandem typu „immediate” (np. LDA #2C) o wartości xx. Po FI nie może być spacji.

#### FT aaaa bbbb

Listuje wszystkie nie dające się disassembledować bajty, w obszarze od aaaa do bbbb. Ostatecznie otrzymuje się dokładne adresy początkowe i końcowe tabel.

**UWAGA:** We wszystkich rozkazach zaczynających się na literę „F”, jeżeli zamiast „F” wypiszemy SHIFT + „F”, to nastąpi wyprowadzenie na drukarkę.

## G aaaa

Rozkaz startu programu maszynowego podobny do SYS z tym, że na końcu programu powinien być rozkaz BRK (a nie RTS), co zapewnia powrót do monitora.

L'NAZWA',08 (lub L'NAZWA',09) Ładowanie programu ze stacji dysków.

## M aaaa lub M aaaa bbbb

Listowanie obszaru pamięci począwszy od aaaa lub od aaaa do bbbb w postaci liczb szesnastkowych. Po prawej stronie wyprowadzane są dodatkowo znaki ASCII, ale tylko wówczas, gdy chodzi o literę, liczbę lub znak matematyczny.

Po wydaniu tego rozkazu wyprowadzony zostanie tylko jeden wiersz, SPACE listuje następne wiersze, aż do puszczenia klawisza. RETURN albo dowolny inny klawisz – ciągle listowanie, aż do ponownego przyciśnięcia klawisza.

Po przyciśnięciu RUN/STOP monitor melduje się migającym kursorem. Można wówczas wprowadzać modyfikacje (zmieniać bajty) w poszczególnych wierszach.

SHIFT + „M” – wyprowadzenie na drukarkę.

## O aaaa bbbb xx

Wypełnienie obszaru pamięci RAM od aaaa do bbbb wartością xx, np.: O 2031 2109 EA – obszar od 2031 do 2109 będzie wypełniony wartością EA, która oznacza rozkaz NOP.

## Pxx

Ustawienie wyjścia na drukarkę: xx może przyjmować wartości 04 albo 11. Jeżeli xx=04, to komputer steruje drukarką z wejściem szeregowym, kompatybilną z systemem „COMMODORE”, dołączoną do gniazda SERIAL BUS. Jeżeli xx=11, to komputer steruje drukarką z wejściem CENTRONICS dołączoną do gniazda USER PORT.

Po wejściu do monitora wystarczy raz ustawić drukarkę.

## B

Wyświetlenie zawartości rejestrów procesora:

PC – licznik programu,  
SR – rejestr stanu,  
AC – akumulator,  
XR – rejestr X,  
YR – rejestr Y,  
SP – wskaźnik stosu.

S'NAZWA',08, aaaa, bbbb Zapis obszaru od aaaa do bbbb-1 na dyskietkę. Numer stacji może być również 09. Po S nie ma spacji.

## W aaaa bbbb cccc

Rozkaz powoduje przepisanie bajtów zawartych w obszarze od aaaa do bbbb do obszaru, którego początkowy adres jest równy cccc. Np. W 2000 2800 4000 przepisuje obszar 2000 do 27FF w obszar 4000 47FF.

## X

Opuszczamy monitor i powracamy do systemu BLACK BOX.

# aaaa bbbb

Porównywanie, sprawdzanie czy dwa obszary pamięci są identyczne. Podaje się tylko adresy początkowe obszarów. Monitor wykazuje pierwszy adres, gdzie bajty są różne.

Ponadto assembler-monitor pozwala przeliczać liczby dziesiętne na szesnastkowe i odwrotnie, a w przypadku liczb 8-bitowych 0-255 (0-FF) – również na postać binarną.

#xx (xx jest zawarte w granicach od 0 do 255) – przelicza liczbę dziesiętną na postać heksadecymalną i binarną.

#xxxx (xxxx jest większa od 255) przelicza liczbę dziesiętną na szesnastkową.

\$xx – przelicza liczbę szesnastkową 8-bitową na liczbę dziesiętną i liczbę w postaci binarnej.

%00101101 – przelicza liczbę w postaci binarnej na liczbę dziesiętną i heksadecymalną.

?xxxx+yyyy – dodawanie liczb szesnastkowych.

?xxxx-yyyy – odejmowanie liczb szesnastkowych.

**UWAGA:** Po „?” i po xxxx nie może być spacji.

## 4. ZINTEGROWANY SYSTEM SYNTEZY POLSKIEJ MOWY I DŹWIĘKU.

### 4.1. Wstęp.

Cartridge umożliwia wygłaszanie tekstów napisanych po polsku (zgodnie z regułami polskiej pisowni), pozwala również na syntezę mowy za pomocą fonemów. Możliwy jest także śpiew (z akompaniamentem lub bez). Program syntezy polskiej mowy ma również regulację tempa recytacji, wysokości, a nawet barwy głosu.

Program syntezy polskiej mowy jest w rzeczywistości zmodyfikowanym programem „SAM/RECITEL”. Zasadnicza zmiana polega na zastąpieniu reguł wymowy obowiązujących w języku angielskim – regułami języka polskiego.

Do dyspozycji są również rozkazy sterowania dźwiękiem, co umożliwia pisanie programów wykonywania muzyki (mono i polifonicznej) oraz tworzenie różnych efektów akustycznych.

Aby rozkazy syntezy mowy i sterowania dźwiękiem stały się dostępne należy wydać rozkaz  $\leftarrow\leftarrow\leftarrow$  lub wykonać RESET z przytrzymanym klawiszem RUN/STOP. W pierwszym przypadku zgłosi się program demonstracyjny, w drugim nie.

Jeżeli chcemy zakończyć demonstrację przytrzymujemy przez pewien czas SPACJĘ. Jeżeli chcemy wylistować program demonstracyjny przyciskamy klawisz RUN/STOP, a potem wykonujemy rozkaz LIST.

### 4.2. Lista rozkazów syntezy mowy z opisem.

]SAM skrót ]S.A.

Rozkaz ten włącza program S.A.M. Software Automatic Mouth, który umożliwia syntezę mowy bezpośrednio za pomocą fonemów.

]RECITER skrót ]RE

Rozkaz ten włącza program RECITER i umożliwia czytanie (wygłaszanie) dłuższych lub krótszych tekstów, napisanych po polsku, zgodnie z zasadami pisowni, a także czytanie cyfr I niektórych skrótów.

]PITCH xx skrót ]PI xx lub POKE 39439, xx

Parametr xx może przyjmować wartość 0-255 i określa wysokość głosu. Program bezpośrednio po uruchomieniu nadaje temu parametrowi wartość 64, co odpowiada głosowi męskiemu średniej wysokości. Wartości poniżej 20 są niepraktyczne.

#### ]SPEED xx skrót ]SP xx lub POKE 39438, xx

Rozkaz ten określa tempo recytacji. Parametr xx może się zmieniać w granicach 0-255. Bezpośrednio po uruchomieniu cartridge'a zostaje ustawiona wartość ]SP 72. Wartości poniżej 20 nie są praktycznie używane.

#### ]KNOBS xx, yy skrót ] xx, yy lub POKE 38880, xx : SYS 38882: POKE 38881, yy : SYS 38882

Rozkaz ]KNOBS określa pewne cechy głosu, jak np. jego barwę. Parametr xx określa te własności mowy (głosu), o których decyduje krtań, natomiast parametr yy określa własności mowy (głosu, na które ma wpływ jama ustna). Oba parametry mogą przyjmować wartości 0-255. Dla języka polskiego najbardziej odpowiednie jest ustawienie ]KN 145, 140. Używając POKE można parametry xx i yy regulować niezależnie.

#### ]ERROR skrót ]ER

Rozkaz ten jest użyteczny w trakcie prób i eksperymentów realizacji mowy przy użyciu fonemów (gdy włączony jest program S.A.M. za pomocą rozkazu ]SA). Użyty bezpośrednio po rozkazie SAY powoduje sygnalizację błędu. Na ekranie zostaje wyświetlony tekst z negatywem błędnego znaku. Jeżeli tekst jest błędny, to rozkaz ]ERROR nie zostanie wykonany. Przykład zastosowania SAY A\$ : ]ERROR.

#### ]LIGHT x skrót ]LI x

Parametr x może przyjmować tylko dwie wartości: 0 lub 1. Jeżeli x=0, wówczas w czasie wykonywania rozkazu SAY zostaje wyłączony obraz TV, a cały ekran przybiera kolor ramki. Dźwięk jest wówczas wysokiej jakości. Jeżeli x=1, obraz jest cały czas wyświetlany, ale dźwięk jest zniekształcony (głos, chociaż zrozumiały, jest mocno „zachrypnięty”). Zaleca się, jeżeli nie zachodzi konieczność innego rozwiązania, wyłączać ekran w czasie mowy komputera.

#### SAY „ŁAŃCUCH ZNAKÓW” lub SAY A\$

Tym rozkazem realizowana jest mowa, np. SAY „MAO6VAA”, jeżeli realizowany jest program S.A.M. lub SAY „DZIĘKUJĘ BARDZO”, jeżeli wykonywany jest program RECITER. Liczba znaków w łańcuchu podanym bezpośrednio po rozkazie SAY nie może przekraczać 35. Jeżeli używa się zmiennej łańcuchowej nieindeksowanej A\$, wówczas może ona zawierać 255 znaków łącznie ze spacjami. Jednakże nie zaleca się używać jednorazowo zbyt dużej liczby znaków, ze względu na możliwość pojawienia się niespodziewanych krótkich przerw w mówieniu. Wydaje się, że optymalna liczba znaków zawiera się w granicach 20-70. Nie można używać bezpośrednio zmiennych indeksowanych. Jeśli zachodzi potrzeba mówienia długich tekstów wówczas można ułożyć program wg następującego wzoru:

```
100 FOR I = 0 TO N : B$ = A$(I) ; SAY B$ : NEXT
```

Długość tekstów wypowiedzianych wg powyższego wzoru jest ograniczona jedynie pojemnością RAM'u komputera.

**UWAGA:** Program syntezy mowy ma inną składnię w porównaniu ze standardowym BASIC'iem w jednym i tylko jednym przypadku:

```
BASIC:  
IF A$ = „TAK” THEN PRINT „DOSKONALE”
```

```
Program syntezy mowy:  
IF A$ = „TAK” THEN : SAY „BARDZO DOBRZE”
```

Dodatkowy dwukropek występuje tylko wtedy, gdy po THEN umieszczony jest rozkaz SAY.

## ]QUIT skrót ]QU

Dodatkowe rozkazy BASIC'a są otrzymywane techniką zwaną „wedge”, która polega na zmianie procedury (podprogramu) CHRGET. Rozkaz ]QU powoduje wyłączenie tych dodatkowych rozkazów. Od tej chwili sterowanie programem syntezy mowy jest możliwe za pomocą języka maszynowego lub z BASIC'a przy użyciu POKE i SYS.

Przywrócenie „wedge” jest możliwe za pomocą SYS 38144.

**UWAGA:** W rozkazach ]PITCH, ]SPEED, ]KNOBS, ]LIGHT parametry xx i yy muszą mieć konkretne wartości liczbowe, nie mogą być to zmienne. W razie potrzeby używania zmiennych, należy wykorzystywać POKE i SYS. Np. można tylko użyć ]PI 57, ale poprawne jest POKE 39439, K(I).

**UWAGA:** W czasie wykonywania rozkazu SAY nie wolno dokonywać przerwania za pomocą klawiszy RUN/STOP + RESTORE. Można jednak wcisnąć klawisz RUN/STOP (bez RESTORE) i trzymać go tak długo (w praktyce, co najwyżej kilka sekund), aż skończy się wykonywanie rozkazu SAY. Wówczas dopiero program zostanie przerwany.

### **4.3. Czytanie polskich tekstów.**

Program RECITER, który umożliwia wygłaszanie polskich tekstów, jest włączany rozkazem ]RE. Realizacja mowy jest uzyskiwana za pomocą rozkazu SAY „ŁAŃCUCH ZNAKÓW” lub SAY A\$. Łańcuch znaków lub zmienna łańcuchowa A\$ powinny zawierać tekst (zdanie) napisane po polsku z uwzględnieniem zasad pisowni, cyfry, niektóre działania matematyczne lub niektóre skróty.

#### **4.3.1. Polski alfabet.**

Niektóre znaki Commodore 64 zostały użyte do otrzymywania polskich liter. Są to:

#	(SHIFT i 3)	do realizacji „ę”
\$	(SHIFT i 4)	do realizacji „ą”
&	(SHIFT i 6)	do realizacji „ś”
‘	(SHIFT i 7)	do realizacji „ć”
@	(znak między P i *)	do realizacji „ń”
	SHIFT i V	do realizacji „ł”
	SHIFT i X	do realizacji „ż”
	SHIFT i Q	do realizacji „ź”
	SHIFT i U	do realizacji „ó”

Polskie znaki na ekranie są drukowane tylko wówczas, gdy włączone są tylko duże litery i znaki graficzne. Jeżeli potrzebne są znaki #, \$, &, ... itd., wówczas należy przełączyć się na duże i małe litery. Nie ma to jednak żadnego wpływu na wykonywanie rozkazu SAY; tak samo będzie wykonywany rozkaz SAY „RĘKA” jak i rozkaz SAY „R#KA”. Używanie polskich liter nie jest, więc konieczne, jeżeli czytany tekst nie będzie pokazywany na ekranie, może być jednak pomocne w fazie układania tekstu.

#### **4.3.2. Czytanie cyfr, działań matematycznych i niektórych skrótów.**

Program czyta po polsku wszystkie cyfry. Liczby wielocyfrowe wypowiada podając kolejno cyfrę po cyfrze. Wyczytuje również kropkę rozdzielającą część całkowitą liczby od części ułamkowej.

]RE : SAY „1,2,3,4,5,6,7,8,9,0” : SAY „124.45”

Ponadto program RECITER pozwala na wymawianie następujących symboli w następujący sposób:

+	– plus,
*	– razy,
/	– dzielone przez,
↑	– do potęgi,
<	– jest mniejsze niż,
>	– jest większe niż,
=	– równa się,
%	– procent,
itd.	– i tak dalej,
tj.	– to jest,
np.	– na przykład,
nr	– numer.

Przykład : SAY „2\*7” : SAY „2\*3+1-7”

### 4.3.3. Uwagi na temat akcentu i przestankowania.

Pełnię możliwości akcentowania i stosowania pauz logicznych zapewnia tylko program S.A.M. (p. p. 4.4.2.3.). Program RECITER realizuje neutralny akcent wyrazowy (przycisk) dla większości wyrazów. Niestety, nie jest możliwa realizacja akcentu tonicznego (p. 4.4.2.3.), możliwości uzyskania akcentu zdaniowego są bardzo ograniczone. Dlatego też celowe jest korzystanie z programu S.A.M., zwłaszcza do realizacji dłuższych tekstów, co w znaczący sposób może podnieść ich jakość i zrozumiałość. Używanie programu RECITER jest konieczne tylko wówczas, gdy wypowiedane treści nie są z góry znane (np. nieznany tekst jest odbierany przez modem, a następnie wypowiedany głosem przez komputer, albo nieznany tekst jest wprowadzany z klawiatury przez osobę używającą komputera).

Komputer wymawia większość polskich wyrazów z akcentem na drugiej sylabie od końca. Nieprawidłowy akcent może się pojawić w wyrazach, które zawierają następujące litery: ą, ę, ś, ć, ń.

Również te wyrazy, których akcent wypada na trzecią sylabę od końca, będą czytane nieprawidłowo. Np. „PROSTOKĄT”, „MATEMATYKA” będą czytane nieprawidłowo. Możliwe są jednak pewne zabiegi, które przynoszą poprawę akcentu, jak np. rozdzielenie sylab lub dwukrotne wypisanie akcentowanej samogłoski (tj. samogłoski, na której jest oparta akcentowana sylaba), np. „MATEMATY KA”, „PROS TOKĄT” lub „PROSTOOKĄT” brzmią już dużo lepiej.

Program RECITER interpretuje znaki przestankowe w sposób następujący:

- : ; . ! – dłuższa przerwa z obniżeniem tonu głosu przy wypowiedaniu sylaby poprzedzającej ten znak.
- ? – dłuższa przerwa z podwyższeniem tonu głosu sylaby poprzedzającej ten znak.
- , – dłuższa przerwa neutralna.
- – krótka przerwa neutralna pod warunkiem, że znak ten jest rozdzielony spacjami od innych części zdania, w przeciwnym razie znak ten jest ignorowany.

Np. SAY „OTO JEST PYTANIE!” : SAY „CZY TO JEST PYTANIE?” : SAY „BYĆ, ALBO NIE BYĆ” : SAY „BYĆ – ALBO NIE BYĆ”.

### 4.4. Synteza mowy za pomocą fonemów.

Program S.A.M. wykorzystuje ponad 50 fonemów i umożliwia realizację mowy w wielu językach, umożliwia również śpiew; przy zmianie wysokości głosu w granicach około dwóch oktaw. Za pomocą fonemów można tworzyć zarówno pojedyncze wyrazy, jak i długie teksty; można tworzyć także inne dźwięki wydawane głosem, jak np. różne rodzaje śmiechu, okrzyki przerażenia itp. Użycie fonemów jest najdoskonalszym sposobem syntezy mowy na Commodore 64. Tylko wówczas możliwe jest zagwarantowanie prawidłowego akcentu wyrazowego i zdaniowego we wszystkich przypadkach, tylko użycie fonemów pozwala na wprowadzenie wszystkich najważniejszych reguł wymowy do wypowiedzenia tekstu.

Program syntezy mowy bezpośrednio za pomocą fonemów jest uruchamiany poprzez skrót JSA.

#### 4.4.1. Oznaczenia fonemów.

Fonemy użyte w programie S.A.M. są oznaczone za pomocą następujących znaków:

Samogłoski	– IY, IH, EH, AE, AA, AH, AO, OH, UH, UX, ER, AX, IX.
Dwugłoski	– EY, AY, OY, AW, OW, UW, YX, WX.
Spółgłoski dźwięczne	– R, RX, L, LX, W, WH, Y, M, N, NX, B, D, DX, G, J, Z, ZH, V, DH.
Spółgłoski bezdźwięczne	– *, S, SH, F, TH, P, T, K, CH, /H, /X.
Fonemy specjalne	– UL, UM, UN, Q.

Po każdej z wymienionych powyżej głosek dźwięcznych (w praktyce po każdej samogłosce) można umieścić cyfrę od 1 do 8, np. AE5. Dodanie cyfry umożliwia uzyskanie akcentu wyrazowego, może też nadać wypowiedzianemu wyrazowi zabarwienie emocjonalne.

Realizacja mowy za pomocą fonemów polega na łączeniu fonemów w ciągi (łańcuchy) i użyciu rozkazu SAY, np. SAY „DVAA5 RAA6ZIXDVAA – RUH6VBAA SHEHUH CHQTAE6RIX”, A\$ = „TSYX5FRAA” : SAY A\$.

Zastosowanie oznaczeń niezgodnych z podanymi spowoduje niewykonanie rozkazu SAY i dźwiękową sygnalizację błędu. Jeśli rozkaz SAY zostanie użyty w połączeniu z rozkazem ]ERROR, wówczas oprócz sygnalizacji błędu nastąpi jego pokazanie (będzie wydrukowany na ekranie negatyw błędnego znaku), np. SAY A\$ : ]ERROR. Jeśli ciąg fonemów zawarty w zmiennej A\$ będzie bezbłędny, rozkaz ]ERROR zostanie zignorowany.

Podany zbiór fonemów zawiera ich nadmiar, niektóre z nich można zrealizować za pomocą innych (dotyczy to zwłaszcza dwugłosek), np. AY jako AAYX, OY jako AOYX, UL jako AXL itd.

#### 4.4.2. Realizacja polskiej mowy za pomocą fonemów.

##### 4.4.2.1. Sposoby otrzymywania polskich głosek.

Ponieważ język polski zawiera tylko 37 fonemów, więc nie wszystkie podane fonemy są potrzebne do wymawiania polskich słów. Z drugiej strony niektóre polskie fonemy nie mają swojego odpowiednika w podanym wykazie i można je „konstruować” tylko w sposób przybliżony.

Poniżej zostaną podane polskie głoski, bądź grupy polskich głosek i sposób ich uzyskiwania, będący wynikiem doświadczeń autora instrukcji. Autor zachęca wszystkich użytkowników komputera Commodore 64 do własnych eksperymentów.

Głoskę lub grupę głosek	Jak w wyrazie	Można otrzymać za pomocą
a	mama	AA
e	wesele	AE
o	kolor	AO
u, ó	kultura, dopóki	W
uł, ół	artykuł, wół	UXW
łu, łó	łuk, łódka	WUX
i	kino	IY
y	pytanie	YX
ę	się, geś	EHUH
ę	tępić	AEM
ę	preł	AEN
ą	wąwóz, piszą	AOUH
ą	raćbać	AOM
ą	pieniądze	AON
b	baran	B
b	babka, Rabka	P
c	cena	TS
ci, ć	ciało, miłość	Q*
ci	cisza	Q*YI
cz	czapka	ch

d	dom	D
d	radca, kładka, czad	TQ
drz	drzewo	DZH8SH
dź	dżuma	J*
dź	dźwigać	DY*
dzi	działka	DY*IX, DY*
dzi	dziwny, dzisiaj, ludzi	DY*IY
dz	władza, sadza	DDZ7
dz	przeprowadzka, rydz	TS
dc	zarządca	T-TS
f	fabryka	F
g	gość	G
g	próg	K,QG
h, ch	huk, chmura	/H, /X
j	jama	Y
ij	kij, bij	IY5Y7
k	kasza, mak	K
k	akt, fakt	KQ
l	kalka, malarz	LX
ł	łapka, kładka	W
m	mowa	M
n	noc	N
ni	nic	NXIY
ni	koniec, świnia	NIY, NX
ń	koń, bańka	NX
p	puszcza, pismo	P
p	łapka, mapka	PQ
r	rok, pora	RX, R
s	słoń, pasmo	S
rz, ź	rzut, żaba	ZH6, ZHZH, JS H
rz, sz, ź	krzak, przód, szum, wąż	SH
si	siła, prosić	SHIY
si, ś	siano, kość, ściana	SH
t	tato, fakt	T, QT
trz	trzeba, patrz	TSH, CH
w	wizyta	v
w	Wtorek, Wrocław	F
z	zupa, pazur	Z
z	pokaz, rozkosz	S
zi, ź	ziarno, źródło	ZH3, ZH4
zi	zima	ZH3IY
ź	paź	SH

#### 4.4.2.2. Uwagi na temat wypowiedzania tekstu po polsku, z uwzględnieniem ważniejszych norm wymowy.

Omawianie wszystkich reguł wymowy przekracza ramy niniejszego opracowania, nie mniej jednak warto zapoznać się z odpowiednią literaturą, ponieważ może to mieć znaczny wpływ na uzyskiwane efekty. W tej instrukcji podane zostaną tylko takie reguły, których uwzględnienie jest najbardziej rażące i pogarsza zrozumiałość wypowiedzi.

##### Wymowa samogłosek: a, e.

Lokalizacja w wyrazie	a	e
przed spółgłoskami szczelinowymi:		
f, w, s, z, sz, ź, rz, ś, ź, h, ch	AOUH	EHUH
na końcu wyrazu	AOUH	EHUH, AE
przed l, ł	AO	AE
przed t, d, c, dz, cz, dź, k, g	AON	AEN
przed ć, dź	AONX	AENX
przed b, p	AOM	AEM

## Upodobnienia spółgłosek pod względem dźwięczności

Specyfikacją języka polskiego jest istnienie w nim grup spółgłoskowych nie poddzielanych od siebie samogłoskami. Właśnie owe grupy powodują, że nasz język wydaje się obcokrajowcom „szeleszczący”. W grupach spółgłoskowych istnieje tendencja do upodabniania się poszczególnych elementów tych grup, zwłaszcza pod względem dźwięczności. Najczęściej ostatnia spółgłoska wpływa upodobniająco na poprzednie. Np. w wyrazie „babka” dźwięczne „b” upodabnia się do bezdźwięcznego „k” i słowo to wymawia się „bapka”. Podobnie – pisze się „liczba”, a wymawia się „lidźba”, następuje tu udźwięcznienie „cz”. Spółgłoski dźwięczne i ich odpowiedniki bezdźwięczne tworzą następujące pary:

b – p, c – dz, d – t, f – w, g – k, cz – dź, rz – sz, ż – sz, ś – ź, ć – dź

Spółgłoski „w” i „rz” zachowują się odmiennie niż pozostałe, tracą bowiem dźwięczność po lub przed spółgłoską bezdźwięczną, np. „chrzan” wymawia się „chszan”, „przykład” – „pszykłat”, „czwarty” – „czfarty”, „ławka” – „łafka”, „gorzko” – „goszko”.

Spółgłoski b, d, g, dz, dź, dż, w, z, ż, ź znajdujące się na końcu wyrazu, na ogół tracą dźwięczność, np. „staw” wymawia się „staf”, chyba, że następują upodobnienia międzywyrazowe.

Doświadczenia z komputerem Commodore 64 wskazują, że nie można całkowicie pominąć zjawiska upodobnień śródwyrazowych, chociaż wydaje się, że nieuwzględnienie udźwięcznienia jest mniej rażące, niż nieuwzględnienie utraty dźwięczności. Natomiast można bez większej szkody dla zrozumiałości mowy pominąć upodobnienia międzywyrazowe.

Podane w punkcie 4.4.2.1. sposoby uzyskiwania polskich głosek uwzględniają przypadki ubezdźwięcznienia głosek dźwięcznych.

O tym, że uwzględnienie utraty dźwięczności jest istotne, przekonać mogą następujące przykłady:

Słowa	Bez uwzględnienia u. d.	Z uwzględnieniem u. d.
zaraz przyjdę	ZAA6RAAZ PZH6YX6YDEUHH	ZAA6RAAS PSHYX6DEHUH
przed	PZH6AED	PSHAET
wtorek	VTAD6RAEK	ETAO6RAEK

## Rozdzielanie spółgłosek

Program S.A.M. realizując mowę na podstawie zadanego ciągu fonemów wykazuje tendencję do „sklejania” dźwięków realizowanych przez sąsiadujące ze sobą fonemy, przy czym dotyczy to zarówno samogłosek jak i spółgłosek. Użycie spacji nie przeciwdziała temu sklejanemu. Przykład: „AAAA” oraz „AA AA” brzmią tak samo, jako jeden nieprzerwany dźwięk. Takie „sklejanie” jest potrzebne do prawidłowego tworzenia mowy, jest również niezbędne do realizacji śpiewu. Umożliwia łączenie i płynną wymowę spółgłoski i samogłoski, a także grup spółgłoskowych. Jednakże w licznie występujących w języku polskim zespołach spółgłoskowych zdarzają się takie pary spółgłosek, że fonemy użyte do ich dźwiękowej realizacji trzeba rozdzielić, a nie skleić. Może zachodzić potrzeba rozdzielania takich spółgłosek, z których jedna kończy wyraz, a druga zaczyna następny. Najczęściej dotyczy to jednak grup wewnątrzwyrazowych. Jaskrawym przykładem nieprawidłowego „zlewania się” fonemów w jedną całość jest połączenie „ŚĆ”, jak np. w wyrazie „kość”. Jeżeli ten wyraz zrealizuje się jako „KAOSH\*”, to sekwencja „SH\*” będzie słyszana jako jeden dźwięk. Konieczne jest, więc wytworzenie przerwy (choćby bardzo krótkiej) między „SH” i „\*”.

Do realizacji takiej przerwy służy specjalny fonem „Q”, w rzadkich przypadkach celowe jest wywołanie nieco dłuższej przerwy za pomocą znaku „-”. Przykłady SAY „AAQAA” oraz SAY „AA-AA” pokazują, jak krótkie są to przerwy.

Poniżej zostaną podane przykłady wykazujące celowość rozdzielania niektórych spółgłosek:

Wyraz	Realizacja wyrazu bez rozdzielania spółgl.	Realizacja wyrazu z rozdzieleniem spółgl.
ściana	SH*AA6NAA	SHQ*AA&NAA
zawiść	ZAA6VIYSH*	ZAA6VIYSHQ*
kładka	KWAA6TKAA	KWAA6TQKAA
traktor	TRAXAA5KTAOR	TRAXAA5KQTAOR
radca	RAA6TTSA	RAA6TQTSAA
		RAA6T-TSAA

Niektóre przypadki rozdzielania spółgłosek zostały już uwzględnione w wyrazie podanym w punkcie 4.4.2.1.

Niekiedy zachodzi potrzeba rozdzielenia spółgłoski od samogłoski, dotyczy to zwłaszcza spółgłoski „t”, np. „materia” bez rozdzielania ma postać „MAATAE6RXIYAA”, a z rozdzieleniem „MAAQTAE6RXIYAA”. Fakt ten został również częściowo uwzględniony w wykazie.

#### 4.4.2.3. O akcencie, sposobach akcentowania i przestankowaniu.

Wyniki prób syntezy mowy za pomocą komputera Commodore 64 wskazują, że prawidłowy akcent oraz właściwe przestankowanie mają fundamentalne znaczenie dla zrozumiałości mowy. Są one prawdopodobnie ważniejsze, niż prawidłowa wymowa głosek. Każdy język żywy, w tym również i język polski, zawiera dość duży nadmiar informacji. Oznacza to, że jeśli jakiś wyraz wypowiedzianego tekstu nie będzie zrozumiany, to jego znaczenie może być jednak odgadnięte po wysłuchaniu całego zdania lub dłuższego fragmentu tekstu, na podstawie logicznej analizy treści. Taka analiza dokonuje się w mózgu odbiorcy automatycznie i często bezwiednie. Do prawidłowego odbierania mowy „jako całości” przyczynia się właściwy akcent (zwłaszcza zdaniowy) oraz właściwie umieszczone pauzy logiczne.

##### Rodzaje akcentu.

Pod względem miejsca występowania wyodrębnia się akcent wyrazowy i zdaniowy. W języku polskim akcent wyrazowy przypada najczęściej na drugą sylabę od końca. W wyrazach wielosylabowych występuje dodatkowo akcent poboczny na pierwszej zgłosce. Akcent zdaniowy polega na wyróżnieniu jakiegoś wyrazu w zdaniu. Nie istnieją żadne formalne reguły dla akcentu zdaniowego, jest on ruchomy i miejsce jego postawienia zależy wyłącznie od logiki wypowiedzianych treści, od tego, któremu wyrazowi chcemy nadać największe znaczenie. Dlatego też zaprogramowanie w komputerze reguł akcentu zdaniowego jest praktycznie niemożliwe. Nie mniej jednak synteza mowy za pomocą fonemów (i tylko taka) umożliwia wprowadzenie akcentu wyrazowego do każdego konkretnego tekstu, którego treść jest z góry znana. Nie można tego zrobić w stosunku do tekstu odbieranego np. przez modem i następnie mówionego przez komputer, ponieważ treść tego tekstu jest dla komputera zmienną losową.

Pod względem sposobu otrzymywania rozróżnia się akcent „przyciskowy” i akcent „toniczny”. Akcent przyciskowy polega na głośniejszym i czasem nieco dłuższym wymawianiu wyróżnionej sylaby. Akcent toniczny – na podwyższaniu lub obniżaniu tonu głosu przy jej wymawianiu. Akcent wyrazowy w języku polskim jest przyciskiem. Akcent toniczny jest używany rzadziej i to przeważnie jako akcent zdaniowy.

##### Sposoby uzyskiwania akcentu w programie S.A.M.

Podstawowym sposobem uzyskiwania akcentu jest umieszczenie cyfry od 1 do 8 bezpośrednio po samogłosce, na której jest zbudowana akcentowana sylaba. Sposób ten jest najodpowiedniejszy dla uzyskiwania akcentu wyrazowego. Umieszczenie cyfry 6 daje akcent bez podwyższania bądź obniżania tonu głosu, czyli przycisk. Umieszczenie cyfr od 5 do 1 realizuje akcent toniczny, przy czym im mniejsza wartość liczbowa, tym większe podwyższenie tonu głosu. Umieszczenie cyfr 7 lub 8 powoduje obniżenie tonu głosu (cyfra 8 – największe obniżenie). W języku polskim, do realizacji akcentu wyrazowego najczęściej należy użyć cyfry 6 lub rzadziej 5. Daje to niewielki, choć zwykle wystarczający efekt. W razie potrzeby, w celu nadania wypowiedzi zabarwienia emocjonalnego, można użyć innych cyfr.

Przykłady akcentowania sylab w wyrazach.

Wyraz	Realizacja bez akcentu	Realizacja z akcentem
materia	MAAQTAERXYAA	MAQTAE6RXYAA
antena	AANTAENAA	AANTAE6NAA
konieczność	KAONXYAECHNAOSHQ*	KAONXYAE6CHNAOSHQ*

Innym, bardziej radykalnym sposobem, jest przedłużenie czasu brzmienia samogłoski poprzez wielokrotne (np. podwójne) jej wypisanie. Taki sposób jest uzasadniony tylko w niektórych przypadkach, najczęściej daje on efekt przesadzony. Może być zastosowany np. przy realizacji samogłoski „u” za pomocą fonemu „w”, jeżeli oczywiście ta samogłoska występuje w sylabie akcentowanej. Dodanie cyfry zwykle tu nie wystarcza. Np. słowo „matura” zrealizowane za pomocą „MAATW6RXAA” nie brzmi prawidłowo. Dopiero „MAATWWRXAA” jest zadowalające.

Akcent zdaniowy polega na wyróżnieniu jakiegoś wyrazu w zdaniu, zwykle wystarczy zastosowanie mocniejszego akcentu tonicznego w akcentowanym wyrazie. Można również zmienić czas wypowiedzania tego wyrazu za pomocą rozkazu ]SP, wspomagająco mogą również działać pauzy umieszczone w odpowiednim miejscu.

Rozważmy jako przykład następujące zdanie: „Jan wyjeżdża za granicę”. Możemy zaakcentować każdy z wyrazów wchodzących w skład tego zdania. Jeżeli chcemy podkreślić, że to właśnie Jan wyjeżdża, to układamy następujący ciąg fonemów:

„YAA3N VYXYAE6ZH6J\*AA ZAA GRXAANXIY6TSEHUH”,

Jeżeli chcemy zaakcentować fakt wyjazdu Jana wówczas piszemy:

„YAAN VYXYAE3ZH6J\*AA7 ZAA GRXAANXIY6TSEHUH”,

Wyróżnienie zagranicznej podróży przynosi następujący efekt:

„YAAN VYXYAE6ZH6J\*AA-ZAA GRAANXIY2TSEHUH”.

Akcent toniczny może być użyty również do realizacji różnych głosów nie będących mową, jak np. okrzyków, śmiechu itp. Ilustrują to przykłady:

„/HAA1 /HAA3 /HAA5AA6AA7AA8”

„AO1AO3AO5AO7AO8”

„AA7KW3W3 KW4W4W4”

### Przestankowanie

Poszczególne zdania, grupy wyrazów, a nawet poszczególne wyrazy, muszą być oddzielone od siebie dłuższymi lub krótszymi pauzami. Są to pauzy logiczne, których usytuowanie i czas trwania zależą od wypowiedzianych treści, a ich celem jest przede wszystkim zwiększenie zrozumiałości wypowiedzi. Pauzy logiczne są częściowo oznaczone w tekście znakami przestankowymi. Niestety, liczba potrzebnych pauz logicznych jest zwykle dużo większa, niż liczba przecinków, jakie by można postawić zgodnie z zasadami pisowni. Np. w zdaniu: „Jutro przyjdzie do mnie moja nowa narzeczona” wskazana jest pauza między „mnie” i „moja” i zdanie to powinno być wypowiedziane następująco: „Jutro przyjdzie do mnie, moja nowa narzeczona”.

W programie S.A.M. do realizacji przerw służą następujące znaki: Q, -, ., ?, ).

„Q” – bardzo krótka przerwa,

„-” – krótka przerwa,

„.” – dłuższa przerwa,

„?” – dłuższa przerwa z obniżeniem tonu głosu sylaby poprzedzającej tę przerwę,

„?” – dłuższa przerwa z podwyższeniem tonu głosu sylaby poprzedzającej przerwę.

Tempo recytacji można regulować poprzez rozkaz ]SP xx, gdzie xx jest liczbą w granicach 0-255, najczęściej przyjmuje się od ]SP 60 do ]SP 80.

Przykłady stosowania przerw:

„Być, albo nie być – oto jest pytanie!”  
„BYX4Q\*, AA6LXBAO-NXYAE5 BYX7Q\*., AO40TAO5 YAE5S0T PYXTAA7NXYAE.”  
„Czy to jest pytanie ?”  
„CHYX5 QTAO YAESQT PYXTAA7NXYAE?”

#### 4.5. Realizacja śpiewu.

Śpiew uzyskuje się poprzez zmianę tonu i czasu brzmienia samogłosek występujących w śpiewanym tekście.

Do zmiany wysokości tonu można użyć rozkazu ]PI xx, lub POKE 39439, xx gdzie xx jest liczbą od 0 do 255. W praktyce przedział tych wartości jest mniejszy i wynosi od 25 do 100 (poza tym przedziałem dźwięk brzmi nienaturalnie). Regulując wysokość tonu za pomocą POKE można w miejsce xx wstawić nie tylko konkretną liczbę, ale i zmienną (indeksową lub nieindeksowaną), np. POKE 39439, K(I).

Zmiana czasu brzmienia głosek jest możliwa poprzez zmianę liczby powtarzających się samogłosek, np. LXAA, LXAAAA, LXAAAAAA (jeżeli mowa jest uzyskiwana za pomocą fonemów) lub LA, LAA, LAAAA (jeżeli mowa jest realizowana za pomocą programu RECITER).

Przykłady:

]RE : ]KN 130,130 : ]SP 70 : ]PI 62 : SAY „LAA” : ]PI 52 : SAY : “LAA” : ]PI 46 : SAY “LAA” : ]PI 37 : SAY “LAAAAAAA”.

Właściwą wysokość głosu najlepiej uzyskiwać „na słuch”, metodą prób i błędów, tym bardziej, że dla różnych samogłosek różne liczby określają tę samą wysokość tonu. Niestety, dokładność ustawienia częstotliwości jest (zwłaszcza dla wysokich dźwięków) ograniczona i dlatego nie wszystkie tony brzmią „czysto”. Dokładniejsze dostrojenie można otrzymać zmieniając dodatkowo parametr xx rozkazem ]KN xx, yy ale program śpiewu staje się przez to bardziej skomplikowany.

Godząc się z faktem, że komputer Commodore 64 nie może stanowić konkurencji dla Jana Kiepurę, możemy posłużyć się następującą tabelą:

Samogłoska odpowiadająca	a	o	i	y	u	e
Odpowiadający fonem	AA	AO	IY	YX	W	AE
Nuta						
C	62	59	54	56	55	61
C#	60	57	52	55	52	59
D	57	54	50	53	50	56
D#	55	52	47	50	47	54
E	52	49	44	47	44	52
F	50	47	42	44	42	48
F#	48	45	40	42	40	47
G	46	43	39	41	38	45
G#	44	41	36	38	36	43
A	42	39	35	36	34	41
A#	40	37	32	34	32	39
H	39	36	31	33	31	38
C	37	35	29	31	29	36

**UWAGA:** Powyższa tabela jest prawdziwa tylko dla ]KN 130, 130 !

Przykład 1:

```
10 ]S.A. : ]KN 130, 130 : ]SP 70
20 ]PI 59 : SAY „DAOAO” : ]PI 56 : SAY „RXAEAE” : ]PI 44 : SAY „MI-YIY”
30 ]PI 48 : SAY “FAAAA” : ]PI 43 : SAY “SAOAOLX” : ]PI 42 : SAY “LXAAAA”
40 ]PI 31 : SAY “SIYIY” : ]PI 35 : SAY “DAOAO”
```

Przykład 2:

```
10 DIM X(15), A$(15)
20 FOR I=0 TO 15 : READ X(I), A$(I) : NEXT
30 ]RE : ]KN 130,130 : ]SP 60
40 FOR I=0 TO 15 : POKE 39439, X(I) : B$=A$(I) : SAY B$ ; NEXT
50 DATA 50, SZŁAAAA, 40, DZIEE, 34, WEEEEE, 38, CZKAA, 38, DOOOO, 42, LAA
60 DATA 40, SEEE, 44, CZKAA, 47, DOOOO, 49, ZIEE, 54, LOOOO, 54, NEE, 47, GOOOO
70 DATA 35, AA, 41, HAAAA, 50, HAA
```

Komputer Commodore 64 wykorzystuje do syntezy mowy tylko jeden generator nr 1 i tylko wtedy, gdy wykonywany jest rozkaz SAY. Oznacza to, że można wykonywać śpiew z akompaniamentem za pomocą pozostałych dwóch generatorów. Można również realizować muzykę na wszystkich trzech generatorach, jeśli w czasie gry nie będzie syntezy mowy.

Szczegółowe opisywanie programów i wyjaśnianie sposobów realizacji śpiewu z akompaniamentem przekracza założone ramy niniejszego opracowania. Należy tylko zaznaczyć, że zawsze najpierw należy uruchomić generatory akompaniujące nr 2 i nr 3, a dopiero potem wydać rozkaz SAY.

#### 4.6. Synteza mowy uruchamiana językiem maszynowym.

Najpierw należy wprowadzić do RAM’u łańcuch znaków w kodzie ASCII stanowiących mówiony tekst. Pierwszy znak należy wprowadzić do komórki pamięci o adresie \$9A15, następny do \$9A16 itd. Liczba znaków (łącznie ze spacjami) nie może przekraczać \$9B. Jeżeli wprowadziliśmy łańcuch złożony z fonemów i chcemy uruchomić S.A.M., to wydajemy rozkaz JSR \$9A03. Jeżeli wprowadziliśmy polski tekst i chcemy zrealizować mowę według programu RECITER, wówczas wydajemy rozkaz \$9A09. Możliwe jest wykonanie wszystkich pozostałych rozkazów, trzeba tylko użyć maszynowych odpowiedników POKE i SYS.

#### 4.7. Lista rozkazów muzycznych.

Możliwe jest użycie rozkazów w pełnym brzmieniu lub dwuliterowych skrótów. W skrócie rozkazu użyto przecinka, który oznacza, że następna litera musi być poprzedzona SHIFT’em.

SOUNDCLEAR                    S’O

Inicjalizuje wszystkie rejestry pamięci komputera ustawiając w nich wartości początkowe (startowe) i kasując równocześnie wszystkie poprzednie ustawienia (wyłącza wszystkie generatory, częstotliwości ustawia na wartości zerowe itp.). Inicjalizacja dotyczy SID.

VOLUME xx                    V’O

Ustawia głośność dla wszystkich 3 generatorów; wartość xx można zmieniać w granicach od 0 do 15, np. VOLUME 15 oznacza pełną gotowość.

OSCILLATE x, yy              O’S

Rozkaz ustawia częstotliwość yy dla generatora x. Częstotliwość powinna być podana bezpośrednio w Hercach, np. OSCILLATE 2, 1000 oznacza, że częstotliwość generatora nr 2 jest ustalona na 1000 Hz. Maksymalna wartość częstotliwości wynosi 3848 Hz. Rozkaz stanowi alternatywę wobec TUNE.

TUNE x, A\$ T'U

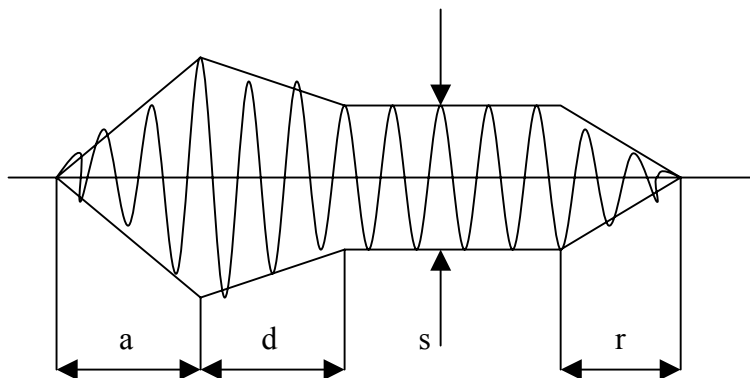
Rozkaz daje inną możliwość ustalania wysokości dźwięku generowanego przez generator x. Łącuch A\$ powinien zawierać nutę zapisaną w sposób podobny, jak w instrukcji do „COMMODORE 64”, z tą różnicą, że łącznik „-”, powinien być pominięty. Np. TUNE 3, „D4” : TUNE 1, „F#5”, generator 3 będzie grać nutę D w oktawie 4, a generator 1 – nutę Fis w oktawie 5.

ENVELOPE x, a, d, s, r E'N

Rozkaz ten umożliwia formowanie obwiedni dla dźwięku z generatora x (x = 1,2,3). Znaczenie pozostałych symboli jest następujące:

- a – attack – określa czas początkowego narastania amplitudy („nabrzmiwania”); a=0-15,
- d – decay – czas opadania amplitudy od momentu zakończenia fazy „attack” do momentu rozpoczęcia fazy „sustain”,
- s – sustain – określa wartość amplitudy generatora (głośność generowanego dźwięku) od momentu zakończenia fazy „decay” do momentu rozpoczęcia fazy „release”. O czasie trwania tej fazy decyduje rozkaz PLAY,
- r – release – określa czas końcowego opadania amplitudy od wartości określonej przez fazę „sustain” do zera. Faza „release”, rozpoczyna się wówczas, gdy skończy się czas trwania dźwięku określony przez rozkaz PLAY lub, gdy wydany zostanie rozkaz PLAY x, OFF.

Wartości a, d, s, r można przyjmować w granicach od 0 do 15. Wartości te określają (z wyjątkiem s) czasy trwania poszczególnych faz i są szczegółowo omówione w instrukcjach podanych przy omawianiu rozkazu TUNE.



WAVE x, y (,zz) (,1) W'A

Rozkaz ten służy do określenia kształtu przebiegu generowanego: x oznacza numer generatora i przyjmuje wartość 1, 2, 3. Zmienna y oznacza numer kształtu fali i przyjmuje następujące wartości:

- 1 – trójkąt (piła symetryczna),
- 2 – przebieg piłozębny (piła niesymetryczna),
- 3 – prostokąt,
- 4 – szum.

Zmienna (,zz) występuje tylko wówczas, gdy generowany jest przebieg prostokątny i oznacza współczynnik wypełnienia prostokąta (impulsu prostokątnego). Przyjmuje wartość od 0 do 100 procent.

Następnym parametrem (,1) jest (są) litera (litery):

OFF – ten parametr może być zastosowany tylko w odniesieniu do generatora nr 3 (x=3). Użycie OFF powoduje niepodawanie sygnału z 3-go generatora bezpośrednio na wyjście małej częstotliwości. Generator ten może być włączony i wytwarzać jakiś przebieg, ale nie będzie on słyszany bezpośrednio. Przebieg ten jednak może być wykorzystany do innych celów, np. do modulacji jakiegoś parametru innego generatora, synchronizacji, modulacji kołowej itp.

S – generator określony przez parametr x będzie synchronizowany z innym generatorem, 1 z 3, 2 z 1, 3 z 2. Synchronizacja polega na powiązaniu dwóch przebiegów wytwarzanych przez parę generatorów za pomocą operacji logicznej „and”.

B – przebieg wytwarzany przez generator x (powinien być to trójkąt) jest powiązany z przebiegiem z innego generatora za pomocą modulacji kołowej (1 z 3, 2 z 1, 3 z 2).

Przykłady zastosowania instrukcji WAVE:

WAVE 2, 2 – generator 2 wytwarza przebieg piłozębny,  
WAVE 1, 3, 60 – generator 1 wytwarza przebieg prostokątny o współczynniku wypełnienia 60%,  
WAVE 3, 1, OFF – generator 3 wytwarza trójkąt, który jednak nie może być bezpośrednio słyszalny,  
WAVE 1, 3, 40, S – generator 1 wytwarza prostokąt o współczynniku wypełnienia 40%, który jest synchronizowany z przebiegiem generatora nr 3.

PLAY x, yy (1) P'L

Rozkaz ten powoduje włączenie generatora x, a więc ostatecznie decyduje o tym, czy dany dźwięk z tego generatora będzie słyszany, czy też nie. Drugi parametr może być liczbą yy lub może być utworzony przez literę (1).

W przypadku podstawienia za drugi parametr wartości liczbowej yy, generator x zostanie włączony na czas określony przez tą wartość. Wartość maksymalna parametru yy wynosi 65535 i oznacza grę przez około 46 sekund. Jeżeli zostanie zastosowana instrukcja PLAY z parametrem liczbowym, to program zapisany w BASIC'u zatrzymuje się aż do zakończenia gry generatora (wypełnienia czasu określonego przez ten parametr), po czym biegnie dalej. Alternatywnym rozwiązaniem jest zastosowanie parametru literowego ON i OFF. Instrukcja PLAY x, ON powoduje włączenie generatora x na czas nieokreślony (tj. aż do wyłączenia), przy czym program nie czeka na wyłączenie i biegnie dalej. Rozkaz ten stanowi, więc podstawę do gry wielogłosowej. Umożliwia również jednoczesne wykonywanie jakiegoś programu i wydawanie dźwięków.

Przykłady zastosowania instrukcji PLAY: PLAY 2, 5000 : PLAY 3, ON.

Wyłączenie generatora x następuje poprzez PLAY x, OFF.

FILTER 1, xx, yy, x (,y) (,z) F'I

Włącza filtr dla jednego lub większej liczby generatorów.

1 – określa sposób filtracji:  
H – górnoprzepustowy,  
B – środkowoprzepustowy,  
T – dolnoprzepustowy,  
N – środkowozaporowy.

xx – określa częstotliwość graniczną filtru w Hercach (od 30 do 11800).

yy – określa rezonans (podbicie charakterystyki częstotliwościowej) przy częstotliwości granicznej, określonej przez parametr xx; parametr yy przyjmuje wartości od 0 (brak rezonansu) do 15 (pełny rezonans).

x,y... – może składać się z 4 cyfr rozdzielonych przecinkami (liczba cyfr nie może przekraczać 4, może być jednak mniejsza niż 4). Cyfry oznaczają numery generatorów, których przebiegi będą filtrowane. Cyfra 4 oznacza wejście „audio”.

Przykłady zastosowania instrukcji FILTER:

FILTER H, 2000, 0, 1, 2 – przebiegi generowane przez generatory 1 i 2 zostaną poddane filtracji poprzez filtr górnoprzepustowy o częstotliwości granicznej 2000 Hz.

FILTER T, 1000, 15, 1, 3, 4 – przebiegi generowane przez generator 1, 3 oraz sygnał akustyczny dostarczany z zewnątrz poprzez wejście „audio” są filtrowane poprzez filtr dolnoprzepustowy o częstotliwości granicznej 1000; częstotliwość graniczna jest podbita poprzez rezonans określony maksymalną wartością (15).

Inne przykłady zastosowania opisanych rozkazów:

VOLUME 15 : ENVELOPE 1, 0, 15, 0 : WAVE 1, 2 : OSCILLATE 1, 500 : PLAY 1, 100 : PLAY 1, ON : FOR I=1 TO 1000 : OSCILLATE 1, I : PLAY 1, OFF.

Przykład programu z zastosowaniem modulacji kołowej:

```
10 SOUNDCLEAR : VOLUME 15
20 ENVELOPE 1, 0, 15, 0 : ENVELOPE 3, 0, 0, 15, 0
30 WAVE 1, 1, R : WAVE 3, 3, 30
40 PLAY 1, ON : PLAY 3, ON
50 FOR I=1 TO 3000 STEP 100 : FOR X=1 TO 3000 STEP 50
60 OSCILLATE 3, X : NEXT : OSCILLATE 1, I : NEXT
70 SOUNDCLEAR
```

Przykład programu z zastosowaniem synchronizacji:

```
10 SOUNDCLEAR : VOLUME 15
20 ENVELOPE 1, 0, 0, 15, 0
30 WAVE 1, 3, 40, S : WAVE 3, 1, OFF
40 OSCILLATE 3, 250
50 PLAY 1, ON : PLAY 3, ON
60 FOR I=100 TO 3000 STEP 3
70 OSCILLATE 1, I
80 NEXT
90 SOUNDCLEAR
```

**UZUPEŁNIENIA:** Częstotliwość w rozkazie OSCILLATE oraz współczynnik wypełnienia prostokąta w rozkazie WAVE nie muszą być liczbami całkowitymi. O dokładności decyduje liczba miejsc po przecinku, np. OSCILLATE 2, 432,7.

**UWAGA:** W programach napisanych w BASIC’u nie należy umieszczać dodawania łańcuchów! Można oczywiście dodawać zmienne łańcuchowe:

```
100 PRINT A$ + B$ - PRAWIDŁOWO
200 PRINT “KOMPUTER “ + “COMMODORE” – błędnie
300 PRINT „KOMPUTER “ , „COMMODORE“ – prawidłowo
```