

F-240

 **DIDAKTIK**

FRANTIŠEK PALKOVIČ

PRÍRUČKA UŽÍVATEĽA
DIDAKTIK ALFA 2

ÚVOD

Pri vývoji mikropočítača Didaktik Alfa 2 sa vychádzalo z mikropočítača Didaktik Alfa. Zároveň sa bral ohľad na kompatibilitu s mikropočítačom PMD 85-2.

Možno povedať, že uvedené mikropočítače sú kompatibilné zdola nahor v tomto poradí:

Didaktik Alfa	PMD 85-1
PMD 85-2	
PMD 85-2+	
Didaktik Alfa 2	

V tejto príručke budú opisované len zmeny vo verziách mikropočítača Didaktik Alfa 2 oproti pôvodnej verzii Didaktik Alfa. Preto treba túto príručku chápať ako doplnok k základným príručkám určeným pre mikropočítač Didaktik Alfa.

1. ZAVEDENIE OPERAČNÉHO SYSTÉMU

V hardware je zmena na počítačovej doske, ktorá umožňuje osádzat do počítača pamäti EPROM typu MHB 8716 o rozsahu 2 kbyty. Tým sa zväčšil využiteľný priestor ROM pamäti na plných 16 kbytov, v ktorých je uložený operačný systém Monitor aj jazyk BASIC 2/2.1. ROM modul, ktorý sa štandardne nedodáva, je možné využiť pre ďalšie programové vybavenie. Možno ho osadiť až 32 kbytami pamäťou EPROM.

Po zapnutí počítača a po každom resetovaní sa vykoná test pamäti ROM a RAM. Do dialógového riadku sa vypíše

Testing memory

Ak je v pamäti ROM chyba, vypíše sa do dialógového riadku

++ Rom error++

a po stlačení klávesu /okrem ↑, STOP a pravého stĺpca klávesov/ sa pokračuje testom pamäti RAM. V prípade chyby sa vypíše do dialógového riadku

++ Ram error ++

a do ľavého horného rohu obrazovky sa vypíše adresa pamäti, na ktorej je chyba.

Ak sa po zahriatí a opätovnom resetovaní chyby opakujú, treba sa obrátiť na servisnú organizáciu.

Zavedenie operačného systému a ďalšieho programového vybavenia sa vykoná stlačením uvedených klávesov takto:

1/ RESET - po prvom zapnutí vynuluje pamäť RAM /nuluje kľúče/ a prejde do operačného systému Monitor.

Po opakovanom použití nemaže kľúče ani pamäť od 7000 do 7F00H, ako pôvodný systém.

Monitor sa ohlásí výpisom

++OS ready++

- 2/ RESET + ↑ /SHIFT/ - zavedie jazyk BASIC do RAM od 0000H do 23FFH a nastaví obsah klúčov č. 0 až 11 takto:

RUN, LIST, LLIST, INPUT, PRINT, GOTO, GOSUB, RETURN, FOR, NEXT, THEN, WINDOW

Obsah týchto klúčov je možné zmeniť.

Jazyk BASIC sa ohlásí výpisom

BASIC \angle /V2.1

- 3/ RESET + ↑ + STOP - zavedie jazyk BASIC a nenastavuje obsah klúčov.
- 4/ RESET + STOP - načíta z ROM modulu prvých 12 bytov do oblasti zápisníka a tento program odštartuje. Týchto 12 bytov by malo obsahovať program, ktorý z ROM modulu načíta potrebnú časť do pamäti a prenesený program odštartuje.

2. POUŽITIE KLÁVESNICE

Po stlačení klávesu /okrem STOP, ↑ a pravého stĺpca klávesov/ sa generuje akustické návestie, pri dlhšom podržaní klávesu sa jeho funkcia opakuje /autorepeat/.

Tabuľka znakov /kapitola 6/ obsahuje znaky podľa kódu KOI-8 čs /písmená s diakritickými znamienkami/ doplnenú o špeciálne znaky. Ak vypisujeme znak, pre ktorý je v tabuľke prázdne miesto, vypíše sa na obrazovku štvorček. Znaky s kódom FCH až FFH sa vypisujú podľa toho, ako sú definované v tabuľke znakov. Ak stlačíme kláves, vypisujú sa veľké písmená, t.j. znak, ktorý je zobrazený na klávese. Ak stlačíme kláves + ↑, vypisujú sa malé písmená /ku kódu klávesu sa pričíta 20H/; napr.

A + ↑ = a ,

B + ↑ = b

Ak stlačíme kláves **STOP**, vypisujú sa malé písmená s diakritickými znamienkami /pričíta sa 80H/; napr.:

A + **STOP** = á , **B** + **STOP** = ß

Ak stlačíme kláves **STOP** + **↑**, vypisujú sa veľké písmená s diakritickými znamienkami /pričíta sa A0H/, napr.:

A + **STOP** + **↑** = Á , **B** + **STOP** + **↑** = Æ

Ak chceme zistiť, ktoré klávesy máme stlačiť pri výpise znakov s kódmi C0H - FAH, postupujeme takto:

a/ od kódu znaku odčítame 80H, rozdiel nám udáva kód klávesu, ktorý máme stlačiť

b/ stlačíme kláves **STOP** a klávesy pre kód rozdielu /t.j. kláves znaku, prípadne so **↑** /. Napr. ä = **Q** + **STOP**



Stlačením klávesu **CAPS LOCK** sa rozsvieti ľavá LED dióda a znaky sa budú vypisovať v režime písacieho stroja /malé písmená, so **↑** veľké písmená/. Opätovným stlačením sa prejde do pôvodného režimu.


Tabuľka znakov je rozdelená na 8 skupín, každá skupina má 32 znakov /každý znak ako 8 bytov, spolu 32x8 = 256 bytov/. Na adresách C0B0H - C0BFH sú uložené začiatkové adresy jednotlivých skupín zväčšené o 8. Adresa FF00H označuje nedefinované znaky, ktoré sa vypisujú ako štvorčeky. Zmenou začiatkových adries skupín a vytvorením vlastnej tabuľky znakov môžeme modifikovať štandardný výstup znaku na obrazovku.


Klávesom **END LINE** sa nastaví kurzor za posledný nemedzerový znak v dialógovom riadku.


Riadiace klávesy majú ďalší význam spolu so stlačeným klávesom **↑** :


↑ **O** - zobrazí sa znak, ktorý nasleduje v tabuľke znakov za tým znakom, pod ktorým je nastavený kurzor;


kurzor sa po zobrazení neposunie. Ak je napr. kurzor pod znakom A a stlačíme   , nad kurzor sa zobrazí znak B. Tak môžeme zobrazovať aj znaky, ktoré nie sú na klávesnici /napr.vlnovka/.


-  MSG - počítač prejde do režimu terminál. Použijeme ho v tom prípade, ak chceme použiť počítač Didaktik Alfa 2 ako terminál počítača SMEP.


-  RCL - návrat do Monitora zo stavu, v ktorom sa očakáva vstup znaku.

-  DEL - skok na adresu 0000H, je to ekvivalent príkazu JUMP 0000. Po prvom zavedení jazyka BASIC ho použijeme na návrat do jazyka BASIC z Monitora.

-  HOME - zmaže pracovnú časť obrazovky.

-  ROL - zmaže pracovnú časť obrazovky a zmení podklad na inverzný.

-  END LINE - rozsvieti sa stredná LED dióda a vypne sa akustická signalizácia. Opätovným použitím sa LED dióda zhasne a akustická signalizácia sa obnoví.





-  ROL - jeho použitím sa cyklicky mení jas vypisovaných znakov.

Pri použití KEY klávesov sú tieto zmeny:

- a/ pri ukladaní obsahu dialógového riadku sa uloží len časť riadku vľavo od kurzora, dialógový riadok sa nemení; môžeme ukladať i reťazce ukončené medzerami.



- b/ po stlačení KEY klávesu s nulovým obsahom sa nič nevypíše, ani hlásenie o chybe.

3. OPERAČNÝ SYSTÉM MONITOR

Zrušil sa príkaz BASIC  znak a príkaz  znak /JUMP 6000/. Zavedenie jazyka BASIC bolo už uvedené /  + . Pri použití ostatných príkazov došlo k niektorým zmenám. U všetkých príkazov nie je medzi príkazom a parametrami povinná medzera.

Príkaz SUB - ak sa nezačajú za adresu žiadne dáta, nevypíše sa hlásenie o chybe a príkaz zostane vypísaný v dialógovom riadku.

Príkaz MEM - po jeho odoslaní sa od začiatku riadku vypíše SUB s príslušnou adresou.

Príkaz DUMP - vypisuje sa obsah pamäti bez prerušenia. Výpis je možné pozastaviť pridržením klávesu  alebo ukončiť klávesom .

Príkaz MGSV - číslo súboru môže byť od 00 až po 99. Okrem pôvodného tvaru príkazu môžeme použiť ďalší tvar:

MGSV číslo; adr.1 - adr.2 : komentár

Dvojbodka značí, že súbor sa bude nahrávať pó blokoch, každý blok obsahuje 512 bytov dát. S každým blokom sa nahrá kontrolný súčet bloku. Blokové nahrávanie je vhodné pri dlhších súboroch, kedy je väčšia pravdepodobnosť výskytu chyby pri prenose.

Ak je v blokovom nahrávaní prvý znak za dvojbodkou znak "*" /hviezdička/, po nahratí súboru do počítača sa automaticky odštartuje program od začiatku adresy súboru.

Príkaz MGEND - jeho vykonávanie je možné prerušiť kedykoľvek

klávesom STOP , aj keď nie je pripojený magnetofón.

Neďá sa použiť pre kontrolu blokového súboru.

Príkaz MGLD - je upravený tak, aby bolo možné nahráť i súbory nahrané na inom type magnetofónu. Pri čítaní pásky sa vypisujú hlavičky súborov, ktoré sa nenahrávajú do počítača, do pracovnej oblasti obrazovky.

Príkazom MGLD 00 sa nahrá do počítača prvý súbor, ktorý bol na pásku nahratý príkazom MGSV.



Ak nahrávame do počítača súbor po blokoch, do dialógového riadku sa vypisujú hlavičky jednotlivých blokov číslované od 1. Pri nahrávaní posledného bloku súboru sa na mieste ôsmeho znaku komentára zobrazí štvorček. Ak niektorý blok nie je načítaný správne, hlavička nasledujúceho bloku sa vypíše do pracovnej časti obrazovky. Vtedy vrátime pásku späť a skúsime načítať chybný blok /počítač ho stále očakáva/. Správne nahraný je vtedy, ak sa nasledujúce hlavičky nevypisujú do pracovnej časti obrazovky.

Po nahratí súboru po blokoch s autoštartom sa nahratý program automaticky odštartuje.

Ak prerušíme príkaz MGLD alebo MGEND klávesom STOP , po stlačení klávesu RCL sa vypíše posledná načítaná hlavička.

Príkaz JOB - zadáva sa skutočná dĺžka časti ROM modulu /pôvodne sa zadávala dĺžka väčšia o 256 bytov/. To isté platí pre využitie rutiny Monitora Prenos /uloženej od adresy 8C00H/.

4. REŽIM TERMINÁL

Do režimu terminál sa dostaneme súčasným stlačením klávesov  . Spojenie s počítačom sa uskutočňuje cez asynchrónne sériové prúdové slučky, pričom obidve linky sú pasívne. Rýchlosť prenosu je štandardne nastavená na 4800 Baudov. Rýchlosť prenosu možno meniť zmenou hodnoty SPEED uloženej na adrese C07CH. Platí, že

$$\text{SPEED} = 128000 / \text{rýchlosť prenosu}$$

Na pripojenie sa použije na strane mikropočítača Didaktik Alfa 2 sériový kanál.

Režim terminál má okrem štandardných funkcií terminálu aj tri nové príkazy, ktoré umožňujú blokový prenos dát medzi nadriadeným počítačom a Didaktikom Alfa 2. Sú to tieto príkazy:

INBLOCK: 04 adr.

pre vstup údajov do počítača Didaktik Alfa 2.
Blok má dĺžku 131 bytov.

Prvý byte má hodnotu 04, ďalšie dva byty určujú adresu v pamäti, od ktorej sa uloží nasledujúcich 128 bytov dát bloku.

OUTBLOCK: 02 adr blok



pre výstup údajov z počítača Didaktik Alfa2.
Prvý byte v príkaze má hodnotu 02, ďalšie dva byty určujú adresu v pamäti, od ktorej sa nasledujúcich 128 bytov vyšle na výstupnú linku. Po nich sa vyšle byte kontrolného súčtu týchto 128 bytov.


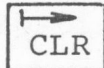

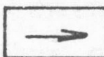


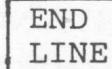

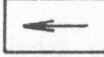




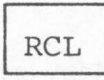

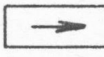







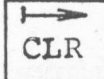

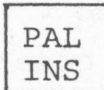
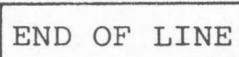
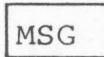






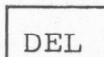
JUMP: 05 adr

skok na určenú adresu v pamäti
Prvý byte má hodnotu 05, ďalšie dva určujú adresu.
Návratová adresa sa uloží na vrchol zásobníka.

V režime terminál sa menia kódy a funkcie riadiacich klávesov.

Kódy riadiacich klávesov sú uvedené v nasledujúcej tabuľke.

 pred názvom klávesu znamená, že treba stlačiť aj kláves .

Kód	Kláves	Kód	Kláves
1	 	17	 
2		18	 
3		19	 
4		20	
5		21	
6	 	22	
7	 	23	
8		24	
9	 	25	
10	 	26	
11		27	
12		28	
13		29	
14		30	
15	 	31	 
16	 	127	

Kódy, ktoré nie sú v tabuľke uvedené, nedajú sa dosiahnuť z klávesnice. Režim terminál bol vyvinutý hlavne pre pripojenie k počítačom radu SMEP, ale možno ho použiť i k iným počítačom. Po spustení režimu terminál sa vysielajú do počítača znaky CTRL Q a CR /kódy 17 a 13/, ktoré slúžia na prípadné odblokovanie prenosovej linky pri počítačoch SMEP.

V ďalšej činnosti funguje Didaktik Alfa ako terminál.

Pri rolovaní obrazovky vysielá terminál kód CTRL S /19/ pred začiatkom rolovania a kód CTRL Q /17/ po ukončení.

Od počítača sa vyžaduje, aby po každom znaku CR posielal niekoľko /3-5/ výplňových znakov /kód Ø/.

Softwaerové pripojenie na počítače radu SMEP pod operačným systémom DOS RV-V2 je možné takto:

Predpokladajme, že hardwaerovo je Didaktik Alfa pripojený ako TTX: /X je číslo terminálu/. Ak je pripojený cez AMUX, rýchlosť treba nastaviť riadiacim príkazom DOS RV:

SET/SPEED = TTX:4800:4800

Ak je pripojený cez ASAD, je treba nastaviť na ASAD-e prepínačmi rýchlosť 4800 Bd.

Potom je nutné nastaviť tieto charakteristiky terminálu:


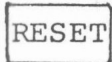
SET/C2111 = TTX:

SET/CRT = TTX:

SET/BUF = TTX:48 .

Charakteristika C2111 zabezpečuje vysielanie výplňových znakov. V prípade nastavovania charakteristík v obraze systému na disku pomocou programu VMR sa namiesto C2111 píše LA30S.



5. JAZYK BASIC ↵ /V2.1

Rozšírením možností jazyka BASIC sa zväčšil aj jeho rozsah. Zavedením jazyka BASIC /   / sa presunie 9 kbyte z oblasti ROM do pamäti RAM /0000H - 23FF/, pričom časť jazyka BASIC je využívaná v oblasti ROM. Po zavedení jazyka BASIC sa zmaže obrazovka a v ľavom hornom rohu sa vypíše

BASIC ↵ /V2.1

BASIC ↵ /V2.1 bol vyvinutý tak, že všetky programy pre Didaktik Alfa, ktoré využívali len podprogramy Monitora uvedené v príručke užívateľa, budú pracovať aj v jazyku BASIC ↵ /V2.1.

5.1. Drobné úpravy a zlepšenia

- a/ Za grafickými príkazmi môžu nasledovať v tom istom riadku ďalšie príkazy.
- b/ Premenné X0, X1, X2, X3, Y0, Y1, Y2, Y3 je možné používať ako bežné premenné súčasne s grafickými príkazmi.
- c/ Funkcia INKEY spracúva aj KEY kláves č. 11.
- d/ Číslo riadku môže byť len do 0 do 32767.
- e/ V príkazoch ON výraz GOTO a ON výraz GOSUB môže byť hodnota výrazu od -32768 do 32767.
- f/ Príkaz MONIT bol zrušený. Je nahradený stlačením   .
- g/ Je možné zadávať čísla v šestnástkovej sústave. Zadáme ho tak, že pred číslom dáme apostrof '/'.
Napr. odoslaním

```
PRINT '100  
      vypíše počítač
```

256 ,

lebo 100H = 256D.

Zadávané číslo môže byť v rozsahu 0 - FFFFH.

V desiatkovej sústave ho počítač vypisuje v tvare dvojkového doplnku. Napr.:

```
PRINT 'FFFF
```

dáva ako výsledok -1.

h/ Dĺžka vypisovaného riadku /počet znakov/ pre príkazy PRINT a LIST je nastavená na 48 znakov /pôvodne bolo 80/.


Môžeme ju zmeniť príkazom

```
POKE '2E, dĺžka
```

Príkazom

```
POKE '2E,20
```

nastavíme dĺžku riadku na 20 znakov.

i/ Vykonávanie programu alebo výpis programu je možné pozastaviť kedykoľvek pridržením klávesu  . Ak kláves pustíme, vykonávanie programu alebo výpis programu pokračuje.

j/ Názvy premenných môžu obsahovať i malé písmená a znak @ .

5.2. Rozšírenie existujúcich príkazov a funkcií

Príkaz POKE

Príkaz POKE môže mať tvar

```
POKE adr, výraz 1, výraz 2, ..., výraz N
```

Hodnoty výrazov 1, 2, ..., N sa uložia postupne do pamäti od adresy adr.

U tohoto príkazu je výhodné použiť možnosť zadávať čísla v šestnástkovej sústave. Príkaz

```
POKE '60000,20,30,40
```

je ekvivalentný príkazom

```
POKE '60000,20: POKE '6001,30: POKE '6002,40.
```

Príkaz ROM

Má tvar

ROM výraz

kde hodnota výrazu $/\emptyset - 31/$ určuje poradie bloku pamäti
o rozsahu 1 kbyte v ROM module.

Príkaz BEEP

Môže mať ďalší tvar

BEEP výška, dĺžka

kde aritmetické výrazy výška $/2\emptyset - 32767/$ a dĺžka $/1 - 225/$
zadávajú výšku a dĺžku tónu.

Jednotka dĺžky tónu je asi $1/33$ sekundy.

Výška tónu odpovedá zhruba frekvencii v Hz.

Príkaz PLOT

Ak chceme spojiť viac bodov, môžeme použiť nasledujúci
tvar príkazu PLOT:

PLOT $X_1, Y_1, P_1; X_2, Y_2, P_2; \dots; X_N, Y_N, P_N$

kde X_i, Y_i sú súradnice jednotlivých bodov, P_i určuje,
či sa bod spojí s predchádzajúcim alebo nie.

Výraz P_i nemusí byť samozrejme zadáný.

Príkazom

PLOT $\emptyset, \emptyset, 1; 0, 5; 5, 5, 1; 5, \emptyset$

nakreslíme dve zvislé strany štvorca.

Príkaz LIST

Pri vykonávaní príkazu LIST sa riadky, ktoré majú viac
znakov, než ako je nastavená dĺžka vypisovaného riadku,
vypisujú do viac riadkov obrazovky. Ďalšie riadky progra-
mového riadku sa vypisujú až za úroveň čísla riadku /pod
číslom riadku je voľné miesto/.

Príkaz LLIST

Kurzor je po výpise riadku nastavený pod prvou číslicou, ale nie je zobrazený. Ak ideme riadok opravovať a stlačíme niektorý kláves, vykoná sa príslušná činnosť a kurzor sa zobrazí.

Príkaz PRINT

Je rozšírený o funkcie AT a INK.

Funkcia AT má tvar

AT riadok, stĺpec

Výraz riadok /Ø - 26/ určuje, na ktorý riadok sa nastaví kurzor.

Výraz stĺpec /Ø - 47/ určuje, na ktorý stĺpec sa nastaví kurzor.

Príklad:

PRINT AT 11,11;"POCET";

Vypíše na zadané miesto obrazovky reťazec "POCET".

Kurzor zostane nastavený za posledným znakom reťazca.

Funkcia INK umožňuje ovládať jas vypisovaných znakov.

Má tvar

INK (výraz)

Hodnotu výrazu dostaneme súčtom týchto parametrov:

Ø - 25% jas /normál/

1 - 50% jas

2 - 75% jas

3 - 100% jas

4 - inverzný výpis

Zavedením jazyka BASIC je nastavená hodnota Ø.

Po funkcii INK ostáva jej hodnota v platnosti pre výpis znaku až dokiaľ jej hodnotu nezmeníme.

Príkazom

PRINT AT11,11;INK (4) "ABC"INK (3) "DEF"

sa vypíše na zadané miesto obrazovky reťazec "ABC" negovane /na bielom podklade/ a za ním reťatec "DEF" v 100% jase.

Príkaz PRINT môžeme použiť taktiež pre výstup znakov na niektorý výstupný kanál v tvare

PRINT # k; zoznam

Príkazom

PRINT # 43;"ABECEDA"

vypíšeme na tlačiareň reťazec "ABECEDA".

Príkaz OUTPUT

režim práce pre kanál č. 4 je nastavený takto:

43 - výpis 8 bitov /CENTRONICS/

45 - výpis 8 bitov negovaných /DZM, IRPR/

46 - ako 43, ale pred vyslaním znaku sa vysielajú postupnosť znakov ESC R1 pre národnú abecedu

47 - ako 46, dáta negované

Pre uvedené hodnoty nie je treba výstupný obvod nastavovať príkazom CONTROL.

Príkaz INPUT

Môžeme ho použiť na načítanie hodnoty zo vstupného kanála v tvare

INPUT # k;zoznam premenných

Príkazy GOTO a GOSUB

Na mieste čísla riadku môže byť výraz /Ø-32767/ v tvare:

a/ celé číslo

b/ aritmetický výraz, ktorý nezačína číslicou

Ak začína aritmetický výraz číslicou, príkaz spracuje len celé číslo, ktoré touto číslicou začína. Napr.:

GOTO 2Ø * A

je chápané ako GOTO 2Ø.

Môžeme ale napísať

GOTO (2Ø * A) alebo GOTO A * 2Ø

Vtedy sa vykoná príkaz podľa hodnoty výrazu.
Takto môžeme jednou zmenou hodnoty premennej zmeniť
vykonávanie programu na viacerých miestach.

Príkaz AUTO

Môže mať tvar

AUTO výraz

kde hodnota výrazu určuje prvý riadok, ktorý sa príkazom
AUTO vypíše. Ďalšie riadky sú číslované s krokom 1Ø.

Príkaz NEW

Môže sa použiť na mazanie časti programu. Má tvar:

a/ NEW C1,C2

Zruší riadky programu od C1 do C2 včítane.

b/ NEW C1

Zruší riadky programu od C1 do konca programu.

c/ NEW Ø,C2

Zruší riadky programu od začiatku do riadku C2

d/ NEW RETURN

Ak sme zrušili celý program alebo jeho koniec, môžeme
ho ihneď po zrušení príkazom NEW RETURN obnoviť.

Príkaz SAVE

Číslo súboru môže byť Ø až 99.

Príkaz SAVE obsahuje jednoduchú možnosť chránenia programu.
Ak chceme program chrániť proti prezeraniu, nahráme ho
na pásku príkazom

SAVE číslo" *komentár

Ak sa takýto program prečíta príkazom LOAD, ihneď sa odštartuje a nedá sa zastaviť klávesom STOP . Ak prerušíme

program resetovaním, program sa zmaže.

Príkaz LOAD

Príkazom

LOAD Ø

sa načíta /prípadne odštartuje/ prvý program z pásky
v jazyku BASIC.

Príkazom

LOAD CODE číslo

sa načíta z pásky binárny súbor, ktorý bol na pásku
uložený príkazom MGSV. Nie je teda potrebný prechod
do Monitora a naspäť. Takto sa však nedajú načítať blokové
súbory.

Veľmi užitočné je ďalšie rozšírenie príkazu LOAD, ktoré nám
umožní spojiť viac programov do jedného programu.

Príkaz LOAD môže mať tvar

LOAD END C

Tento príkaz nahrá program číslo C z pásky a uloží
ho za program, ktorý bol doteraz v pamäti. Číslo prvého
riadku programu na páske musí byť väčšie ako posledné
číslo riadku programu v pamäti.

Ak nie je, použijeme príkaz LOAD v tvare

LOAD END C1,C2

Tento príkaz prihrá program číslo C1 z pásky a všetky
riadky prihrávaného programu zväčší tak, aby prvý
riadok prihrávaného programu mal číslo C2 a všetky nasledu-
júce riadky sú príslušne posunuté.

Pri nesprávnom použití vypíše počítač hlásenie o chybe:

+++ File small+++ - prvý riadok prihrávaného programu
nemá vyššie číslo ako posledný riadok
programu v pamäti.

+++Pg too big+++ - prihrávaný program sa do pamäti
nezmestí

+++File bound+++ - pokus prihrať chránený program

Funkcia VAL

Argumentom funkcie VAL môže byť znakový reťazec, ktorý obsahuje aritmetický výraz jazyka BASIC. Funkcia VAL vyráta hodnotu tohto výrazu.

Príklad:

```
10 PRINT "ZADAJ FUNKCIU PREMENEJ X"
20 INPUT F$
30 X = 2:PRINT F$, VAL (F$):GOTO 10
```

Program po zadaní funkcie vypíše funkciu a jej hodnotu pre $X = 2$. Ak zadáme napr. $X + 3$, počítač vypíše

$X + 3$

5

5.3. Nové funkcie

Funkcia HEX\$

Má tvar

HEX\$ (výraz)

kde výraz je aritmetický výraz s hodnotou od -32768 do 32767. Funkcia HEX\$ prevádza celú časť svojho argumentu do tvaru reťazca v šestnástkovej sústave. Výsledkom je vždy reťazec dĺžky 4.

Príkaz

```
PRINT HEX$ (100), HEX$ (-1)
vypíše
```

0064

FFFF

Pre prevod čísla zo šestnástkovej sústavy do desiatkovej môžeme použiť nasledujúci program:

```
10 PRINT "ZADAZ CISLO HEXA":INPUT C$
20 C = VAL (" " + C$):PRINT C
30 GOTO 10
```

Funkcia APEEK

Má tvar

APEEK(adr)

Kde adresa je aritmetický výraz určujúci adresu v pamäti.
Výsledkom funkcie APEEK je hodnota dvoch bytov v pamäti
v tvare dvojkového doplnku.

Funkcia ADR

Má tvar

ADR (premenná)

Jej výsledkom je adresa premennej v pamäti.

- a/ Ak premenná je číselná, výsledkom je adresa, od ktorej sú uložené 4 byty hodnoty číselnej premennej.
- b/ Ak premenná je reťazcová, výsledkom je adresa, od ktorej sú uložené 4 byty informácie o hodnote reťazcovej premennej.
- Prvý byte je dĺžka /Ø - 255/, tretí a štvrtý byte udávajú adresu hodnoty reťazcovej premennej v pamäti.

Pomocou funkcie ADR môžeme ľahšie predávať parametre medzi programom v jazyku BASIC a programom v strojovom kóde.

Taktiež ju môžeme použiť na vyhradenie časti pamäti.

Vieme, že každá číselná premenná je uložená na 4 bytoch.

Ak potrebujeme vyhradiť napr. 1000 bytov, urobíme to takto:

```
100 DIM B(249)
```

```
110 A = ADR(B(0))
```

Hodnota A určuje začiatočnú adresu časti pamäti dlhú

1000 bytov, ktorú máme vyhradenú /pole B nebudeme používať, aby sa do tejto oblasti neukladali hodnoty prvkov poľa/.

5.4. Nové príkazy

Príkaz PEN

Používa sa nastavenie spôsobu zobrazovania pre grafické príkazy a pre príkazy PRINT a LIST. Má tvar

PEN výraz

kde hodnotu výrazu dostaneme súčtom týchto parametrov:

0 až 4 - ako pre funkciu INK

8 - vykreslenie negovaním

16 - vykreslenie mazaním

Ak nie je v súčte ani 8 ani 16, vykresľuje sa nastavovaním, t.j. body sa nastavujú bez ohľadu na predchádzajúci stav.

Ak je v súčte 8 aj 16, berie sa do úvahy len 16.

Vykresľovanie negovaním znamená, že vykresľovaný bod mení stav zo svietiaceho na nesvietiaci a naopak.

Pri vykresľovaní mazaním sa body zmažú vždy.

Pri resetovaní počítača je nastavená hodnota 8.

Príkazy DEG a RAD

Týmito príkazmi je umožnená voľba zadávať argumenty goniometrických funkcií v stupňoch alebo v radiánoch.

Príkazom

DEG

sa chápu zadávané argumenty v stupňoch.

Príkazom

RAD

sa chápu zadávané argumenty v radiánoch.

Po spustení programu príkazom RUN a použitím príkazu CLEAR sa nastaví zadávanie argumentov v radiánoch.

Príkaz APOKE

Má tvar

APOKE adr, výraz

kde adr a výraz sú aritmetické výrazy v rozsahu

od -32768 do 32767. Hodnota výrazu /2 byty/ sa uloží od adresy adr.

Príkaz WINDOWG

Používa sa na voľbu grafického okna, t.j. okna, v ktorom sa budú vykonávať grafické príkazy. Má tvar

WINDOWG X1,X2,Y1,Y2

kde výrazy X1,X2 zadávajú ľavú a pravú hranicu okna / \bar{X} - 48/, výrazy Y1,Y2 zadávajú hornú a dolnú hranicu okna / \bar{Y} - 255/.

Presnejšie povedané, ľavá hranica okna začína na ľavom kraji zóny č.X1 pre výpis znaku, pravá hranica okna končí na ľavom kraji zóny X2.

Príkazom WINDOWG sa celá plocha obrazovky transformuje na grafické okno. Grafické príkazy, ktoré využívajú nastavenie súradníc príkazom SCALE, sa budú vykonávať len v tomto okne /sú to príkazy MOVE, PLOT, AXES/.

Pre hodnoty výrazov musí platiť:

a/ $X1 < X2$, $X2 - X1 < 43$

Ak $X2 - X1 > 42$, potom $(X2 - X1) * 6 > 256$ a pravá hranica okna sa "zdegeneruje" tak, že bude na ľavej strane obrazovky

b/ $Y2 - Y1 > 11$

Príkazom

WINDOWG

bez parametrov je nastavené grafické okno tak isto, ako po resetovaní, t.j. ako pre pôvodný Didaktik Alfa.

Príklad:

1 \bar{X} GCLEAR

2 \bar{X} SCALE \bar{X} ,1 \bar{X} , \bar{X} ,1 \bar{X}

3 \bar{X} WINDOWG 1 \bar{X} ,2 \bar{X} , 1 \bar{X} \bar{X} , 2 \bar{X} \bar{X}

4 \bar{X} AXES \bar{X} , \bar{X} :AXES 1 \bar{X} ,1 \bar{X}

Na obrazovku sa vykreslí obdĺžnik, ktorý ohraničuje grafické okno. Grafický kurzor môžeme nastaviť iba do tohto okna.

Príkaz WINDOWA

Používa sa na nastavenie textového /alfanumerického/ okna, v ktorom sa budú vykonávať príkazy PRINT a LIST. Má tvar WINDOWA X1,X2,Y1,Y2

kde výrazy X1, X2, Y1, Y2, majú ten istý význam pre textové okno ako bolo uvedené v popise príkazu WINDOWG.

Pre ich hodnoty musí platiť:

a/ $X1 < X2$, X1,X2 sú v rozsahu 0 - 48

b/ $Y2 - Y1 > 11$, Y1,Y2 sú v rozsahu 0 - 255

Príkazom

WINDOWA

bez parametrov sa nastaví textové okno na celú obrazovku.

Príklad:

10 GCLEAR

20 WINDOWA 10,30, 100,200

30 FOR I = 33 TO 127: PRINT CHR\$(I);:NEXT

40 GOTO 30

Program vypisuje znaky do definovaného textového okna.

Ak po prerušení programu klávesom STOP zadáme príkaz

LIST, výpis programu sa vykoná opäť len do zadaného textového okna. Ak sa riadok programu nezmestí do jedného riadku okna, pokračuje výpis na ďalšom riadku.

Príkaz GCLEARG

Používa sa na zmazanie grafického okna. Má tvar

GCLEARG

Príkaz GCLEARA

Používa sa na zmazanie textového okna. Má tvar

GCLEARA

Príkaz REN

Príkaz REN umožňuje prečíslovať riadky programu. Má tvar

REN C1,C2,CN,K

kde C1, C2, CN, K sú celé čísla.

Príkaz REN prečísluje úsek programu od riadku C1 po riadok C2, pričom prvý riadok prečíslovaného úseku bude mať číslo CN a ďalšie riadky sa prečíslujú s krokom K. Prečíslovaním sa môže prenášať celý úsek programu ale iba do voľného úseku programu.

Ak zadáme $K = 0$, prečíslovaný úsek sa len posunie vzhľadom na prvý riadok úseku.

Ak niektorý parameter vynecháme, platia preň nasledujúce hodnoty:

C1 = prvý riadok programu

C2 = posledný riadok programu

CN = ak je C1 uvedený, má jeho hodnotu, inak 10

K = 10

Príklad:

10 REM 10

20 REM 20

30 REM 30

40 REM 40

Príkazom

REN 10,30,100,5

sa prečísluje program na

40 REM 40

100 REM 10

105 REM 20

110 REM 30

Ďalším príkazom

REN , , ,

sa prečísluje program na

1Ø REM 4Ø
2Ø REM 1Ø
3Ø REM 2Ø
4Ø REM 3Ø

Príkaz REN spolu s príkazmi LOAD END a NEW dovoľujú jednoducho spájať ľubovoľné časti rôznych programov.

Príkaz ON ERR

Umožňuje potlačiť výpis hlásenia o chybe a ošetriť chyby užívateľským programom v jazyku BASIC. Má tvar

ON ERR príkaz

Vykonaním tohto príkazu sa nastaví režim nezastavovania na chybe a zvyšok programového riadku sa ignoruje. Od vykonania tohto príkazu platí, že ak sa vyskytne chyba, najprv sa vyprázdni zásobník návratových adries a zásobník rozrobených cyklov. Potom sa pokračuje vykonaním príkazu uvedeným za ON ERR. Pritom sa však zruší stav nezastavovania na chybe, t.j. pri ďalšej chybe sa program preruší. Preto je vhodné po ošetroení chyby opäť použiť príkaz ON ERR, aby sa opäť nastavil režim nezastavovania na chybe. Informácie o chybe získame takto:

a/ číslo riadku, na ktorom vznikla chyba, je uložené na adrese 5E5DH na dvoch bytoch. Jeho hodnotu zistíme príkazom

CR = APEEK('5E5D)

b/ kód chyby je uložený na adrese 0026H, jeho hodnotu zistíme príkazom

KD = PEEK ('26)

Kódy chyby, odpovedajúce druhu chyby, sú uvedené v nasledujúcej tabuľke.

Kód chyby	Druh chyby
1	Subscr.rng
2	Arr.alloc.

3	Fnc.param.
4	Only in pg
5	No for stm
6	Data exhau
7	Pg too big
8	Overflow
9	Syntax err
10	Return err
11	Numb.nonex
12	Dv by zero
13	Can't cont
14	Strng long
15	No str.spc
16	Str.algrth
17	Type conv.
18	File small
19	Input err
20	Field lost
21	File bound
22	Stop
23	File error

Užívateľský program môže na základe týchto údajov oznamovať užívateľovi, akej chyby sa dopustil a na ktorom mieste.

Príklad:

```
100 ON ERR GO TO 100
20 PRINT "ZADAJ CISLO"
30 INPUT C:PRINT C/(C-5)
40 GOTO 20
100 E = PEEK ('26):R = APEEK ('5E5D)
110 PRINT:PRINT "CHYBA";E;"UV RIADKU";R;":"
120 IF E = 8 THEN PRINT "PRETECENIE,";GOTO 150
130 IF E = 12 THEN PRINT "DELENIE 0,";GOTO 150
140 IF E = 19 OR E = 20 THEN PRINT "NEPOZNAS CISLO?";
```


15Ø PRINT "ZADAJ ZNOVA"

16Ø ON ERR GOTO 1ØØ

17Ø GOTO 3Ø

6. TABUĽKA ZNAKOV PRE DIDAKTIK ALFA 2

Rád 1 kódu znaku

	φ	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
φ	NUL		SP	φ	@	P	`	p					ſ	ô	ž	ô
1			!	1	A	Q	a	q					á	ä	Á	Ä
2			"	2	B	R	b	r					ß	ř	ŕ	Ř
3			#	3	C	S	c	s					č	š	Č	Š
4			\$	4	D	T	d	t					ď	ť	Ď	Ť
5			%	5	E	U	e	u					ě	ú	Ě	Ú
6			&	6	F	V	f	v					í	ı	Ř	ſ
7			'	7	G	W	g	w					ch	é	CH	É
8	BS		(8	H	X	h	x					ü	à	Ü	À
9)	9	I	Y	i	y					í	ý	Í	Ý
A	LF		*	:	J	Z	j	z					û	ž	Û	Ž
B			+	;	K	[k	{					í	Á	Í	
C		FS	,	<	L	\	l						ř	ŕ	Ř	
D	CR		-	=	M]	m	}					ö	Č	Ö	
E			.	>	N	^	n	~					ň	Ď	Ň	
F			/	?	O	_	o	■					ó	Ě	Ó	

Rád φ kódu znaku