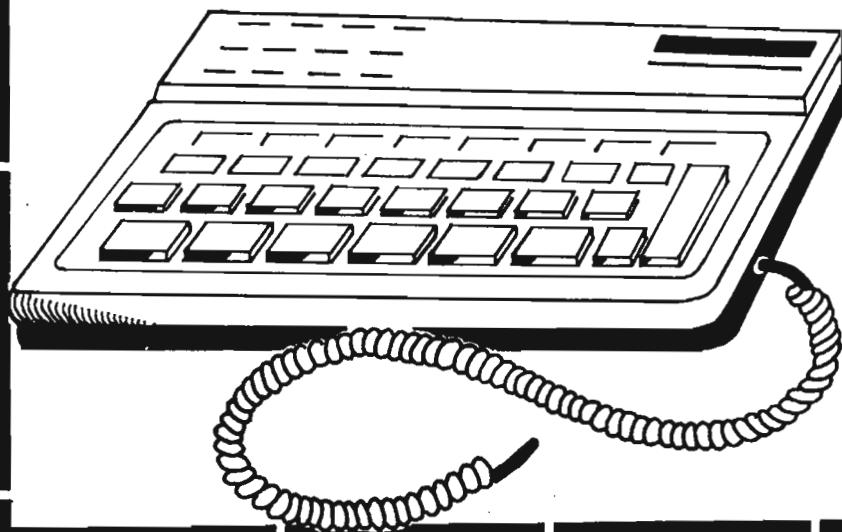


SORO +
SCHNEIDER

600

03



```
*****  
SORD SORD SORD SORD SORD SORD SORD SORD SORD  
SORD SORD SORD SORD SORD SORD SORD SORD SORD  
*****
```

Paměť 64 kB DRAM

Při různých experimentech s pamětí 64 kB DRAM jsem zjistil, že rutiny ve strojovém kódu, které pracují s jemnou grafikou nefungují stejně, jako při práci s originální pamětí EM 5. Jako příklad může sloužit program WALLY (BUDIČ, PYJAMARAMA ...), který se dal bezchybně používat jen s EM 5.

Zjistil jsem, že příčinou je pomalá činnost VDP, který při čtení z VM nedokáže včas poskytnout platná data.

```
LD C, 11H  
OUT /C/,L  
OUT /C/,H  
IN A, /10H/
```

Tuto chybu jsem programově řešil vložením dvou relativně pomalých instrukcí (PUSH AF, POP AF) před vlastní čtení z portu. Myslím si, že hlavní problém je v tom, že EM 5 je asi upravená tak, aby "waitovala" činnost CPU. Přesnější analýzu si netroufám udělat.

Program WALLY jsem si "opravil" takovymto zápisem do CM (verze nezávislá na BF)

```
POKE &B3A5,&CD,&41,&81  
POKE &8141,&F5,&F1,&db,&10,&CD  
POKE &B3BC,&CD,&47,&81  
POKE &B3D2,&CD,&47,&81  
POKE &8147,&F5,&F1,&DB,&10,&23,&C9
```

Zásah nemá žádný vliv na vizuální rychlost kreslení, ale odstraní nežádoucí překreslování části obrázku.

Matej Čurko
Košice

DEFINICE VLASTNÍCH ZAŘÍZENÍ

=====

Tento článek by měl být návodem, jak umožnit, aby různá přídavná zařízení, která se k počítači připojují, mohla být ovládána systémovými příkazy (především příkazem PRINT).

Jedná se například o tiskárny, které nemají styk CENTRONICS (dálnopis, jednojehlová tiskárna), hlasový výstup, obrazovku s 64 znaky na řádek, apod.. Maximálním využitím je pak styk s RAM-DISKem nebo disketovou jednotkou.

Základní pojmy:

PROCEDURA - podprogram ve strojovém kódu, který podle vstupních údajů zajišťuje požadovanou činnost

ZAŘÍZENÍ - vstupní nebo výstupní část počítačového systému ke které existují obslužné procedury pro vstup a výstup ve stanoveném tvaru tak, aby stejný program mohl pracovat s různými zařízeními pouhou změnou názvu zařízení

STANDARDNÍ ZAŘÍZENÍ - zařízení, které je implicitně definováno v BASICu-F (viz tab. 1)

NÁZEV ZAŘÍZENÍ - maximálně 3 písmenná zkratka zařízení následovaná dvojtečkou; v příkazech se může psát malými nebo velkými písmeny

DEFINICE ZAŘÍZENÍ - tabulka v paměti počítače, v které je uložen název zařízení, adresy procedur pro otevření a uzavření zařízení, pro vstup a výstup a některé další údaje (viz tab. 2)

OTEVŘENÍ ZAŘÍZENÍ - soubor operací, které připraví zařízení k činnosti (např. zapnutí motoru dálnopisu, vyhledání nebo zapsání hlavičky záznamu na magnetofon); u některých zařízení se žádné operace neprovádějí

KANÁL - logická cesta, která spojuje program (jmenovitě příkazy pracující se zařízeními) s konkrétním zařízením

OTEVŘENÍ KANÁLU - operace, která přiřadí určitému kanálu určité zařízení; je-li pak v příkazu tento kanál, pracuje se se zařízením určeným při otevření kanálu

Typy příkazů:

Příkazy BASICu-F komunikují se zařízeními důsledně prostřednictvím kanálů. Neplatí to však o systémovém editoru, který pracuje s obrazovkou přímo a dále o většině programů ve strojovém kódu, které spolupracují s obrazovkou nebo tiskárnou. Zatímco v BASICu-G jsou kanály pevně přiděleny zařízením (0=CNS, 1=GRP, 2=PRT, 3=PRI), může v BASICu-F pracovat libovolný kanál s libovolným zařízením.

Kanálů je celkem 16 (číslo 0-15), což je plně dostačující počet. Zvláštní postavení má kanál 0, který je implicitně používán systémem pro styk s obrazovkou (CNS). Číslo kanálu označujeme prefixem # (heš).

Pro snadnější pochopení si rozdělíme příkazy pracující se zařízeními na dva typy:

1. typ - číslo kanálu si volíme sami podle potřeby; jedná se o příkazy OPEN, PRINT, PUT, INPUT, GET, RECORD, CLOSE a případně LINE INPUT; není-li uvedeno číslo kanálu, použije se kanál 0

2. typ - uvádíme pouze typ zařízení a příkaz si již sám vyhledá nejbližší volný kanál, který použije pro styk s tímto zařízením; jedná se o příkazy LIST, CLIST, SAVE, VSAVE, OLD, RUN, VERIFY a CHAIN; není-li uvedeno zařízení, použije se zařízení CNS u příkazu LIST nebo CLIST a CMT nebo FX u ostatních.

Příkazy 1. typu:

Abychom mohli některý kanál použít, musíme nejprve určit, s kterým zařízením bude spolupracovat. To provedeme příkazem OPEN (otevření kanálu):

INPUT

OPEN název (FOR <OUTPUT>) AS #k (RECORD n)

APPEND

název - "název zařízení;název souboru"; název souboru se uvádí pouze u zařízení, která to vyžadují ke své činnosti (magnetofon, disketa)

k - číslo kanálu (0-15)

n - délka záznamu (používá se pro náhodný přístup k souborům na disketě)

INPUT/OUTPUT/APPEND - oznamuje proceduru otevření zařízení, jestli se bude pouze číst / pouze zapisovat / přidávat data; používá se pouze u diskety nebo mgf

PRINT (#k) ((,) (;) (TAB(n)) (výraz)) (...)

Provede výstup výrazů ve znakové formě (čísla jako posloupnost znaků). Je-li v seznamu použita čárka, doplní se počet znaků od začátku řádky (tj. od posledního CR (0D)) mezerami na celistvý násobek 8. Dále lze v příkazu PRINT použít funkci TAB (n), která doplní počet znaků na n. Funkce CURSOR je neúčinná (působí vždy na obrazovku). Není-li příkaz PRINT ukončen ani středníkem, ani čárkou, vystoupí nakonec znak CR (0D). Každému výrazu odpovídá jedno volání procedury print a každé čárce nebo TAB procedury tab.

PUT (Rk) (,) výraz (, ...)

Provede výstup výrazů ve vnitřní formě počítače - číslo INTEGER dvoubajtové (např. číslo 258% vystoupí jako bajty 02 a 01), číslo DOUBLE osmibajtové (např. 0h vystoupí jako 00 00 00 00 00 00 00 00), řetězec podle jeho okamžité délky. Na hranicích mezi výrazy a mezi více příkazy PUT se žádné oddělovače neukládají. Každému výrazu odpovídá jedno volání procedury put.

(LINE) INPUT (Rk:text) (,;) proměnná (, ...)

Provede vstup jednoho záznamu (řetězec ukončený znakem CR) a naplní jím proměnné (po čarce se začíná plnit vždy další proměnná). LINE INPUT dá celý záznam do jedné řetězcové proměnné.

GET (Rk) (,) proměnná (, ...)

Provede vstup znaků podle typu proměnných (do proměnné INTEGER 2 bajty, do proměnné DOUBLE 8 bajtů, do řetězcové proměnné podle její DEKLAROVANÉ délky).

CLOSE (Rk) (, ...)

Provede uzavření vyjmenovaných kanálů a se zařízeními, s kterými kanály pracovaly provede proceduru close (uzavření). Nejsou-li kanály uvedeny, uzavře všechny kanály.

Kromě příkazů CLOSE se automaticke uzavření všech kanálů provádí v těchto případech:

- 1) při vzniku chyby neošetřené příkazem ON ERROR
- 2) při ukončení programu příkazem END nebo vykonáním poslední řádky programu
- 3) při zastavení programu pomocí SHIFT+RESET
- 4) po příkazech RUN a CHAIN

Při zastavení pomocí CTRL+RESET nebo příkazem STOP se kanály nezavírají.

Po CALL @ nebo vypnutí a zapnutí počítače budou sice kanály uzavřeny, ale byly-li předtím otevřeny, neprovede se s příslušnými zařízeními procedura close, což může dělat problémy např. při práci s disketou.

Tabulka 1

Seznam standardních zařízení:

adr	typ	dalsi	název	open	print	input	record
				close	put	get	tab
609D	08	60B3	CNS	5EE5	5EE5	619F	6137 5D6A 6137 6137 613A
60B3	08	60C9	GRP	5EE5	5EE5	61A4	6137 6137 6137 6137 613A
60C9	08	60DF	PRT	5EE5	5EE5	619A	6137 6137 6137 6137 6140
60DF	08	60F5	PRI	5EE5	5EE5	6195	6137 6137 6137 6137 6140
60F5	00	610B	CMT	69BD	6839	6ABF	6AEE 6AD7 680E 6137 6146
610B	78	6121	FX	6C7A	6CA6	6CD2	6CB4 6CE1 6CC3 6CF0 6146
6121	08	73CE	SIO	6BF3	5EE5	6C12	5EE5 6C38 5EE5 6137 6146
73CE	80	0000					

Tabulka 2

Struktura definice zařízení:

ofset	název	bajtů	význam
00	1 typ		bit 7 - 1 = nenásleduje vlastní definice zařízení, ale pouze adresa další definice (tzv. pseudodefinice) bit 6, 5, 4 - 1 = jedná se o originál disketovou jednotku FD-5 bit 3 - 0 = zařízení smí být v každém okamžiku používáno maximálně jedním kanálem - 1 = zařízení smí být používáno více kanály současně
01	2 dalsi		adresa další definice zařízení je-li 0000, jedná se o poslední definici, případně pseudodefinici

(zde končí pseudodefinice)

03	3	název	3 písmenný název zařízení (používají se velká písmena) adresa procedury	používají příkazy:
06	2	open	pro otevření zařízení	OPEN a 2. typu
08	2	close	pro uzavření zařízení	CLOSE, END a 2. typu
0A	2	print	pro výstup textu ve znakové formě	PRINT, LIST, CLIST
0C	2	put	pro výstup hodnot ve vnitřní formě	PUT, SAVE, VSAVE
0E	2	input	pro vstup textu ve znakové formě	INPUT, LINE INPUT, OLD
10	2	get	pro vstup hodnot ve vnitřní formě	GET, OLD, VERIFY, CHAIN, RUN

12	2	record	pro nastavení ukazatele	RECORD
f			dat v souborech s náhodným přístupem	
14	2	tab	pro zjištění počtu vystoupených znaků	TAB, , (čárka)

Z uvedeného je vidět, že se definuje velké množství činností, které umožňují nadefinovat jako zařízení např. disketovou jednotku nebo RAM-DISK (včetně zakládání nových souborů, vyhledávání souborů, náhodného přístupu, apod. s výjimkou mazání souborů, pro které není vhodná procedura).

Definiční 1. zařízení začíná na adrese &609D, další pak následují bezprostředně za sebou; poslední pseudodefinice začíná v oblasti systémových proměnných na adrese &73CE.

Některé procedury:

5EE5 - 'prázdná' procedura (nic se neprovede)
 6137 - ohlásí se Err 105 (nedovolený typ přístupu)
 613A - procedura tab , je-li počet znaků v proměnné &70A7
 6140 - " - " - " - " - &705C
 6146 - " - " - " - " - v tabulce kanálů
 v bajtu s offsetem +05

Tabulka 3

Vstupní a výstupní údaje procedur:

open IN: DA (tj. D*256+A) = (511B) = 5309 bez části FOR
 = (511A) = 2204 INPUT
 = (511C) = 310A OUTPUT
 = (511E) = 370A APPEND

BC - délka záznamu (není-li část RECORD, pak BC = 0100)

IX - bázeová adresa otevíraného kanálu v tabulce kanálů; je možno inicializovat bajt (IX+02) (7120-712F) - údaje o požadovaném nebo zakládaném souboru (jako ve hlavičce mgf - typ souboru, název souboru, atd.); procedura open je aktualizuje podle skutečnosti

print, IN: B - maximální počet znaků na zpracování (pokud put, se dříve nevyskytne znak 00 nebo 0D)

input, E - bajt z tabulky kanálů s offsetem +02

get HL - adresa 1. znaku pro vstup/výstup
 (708E) - bit 1 - 0 = OLD
 1 = VERIFY (pouze při get)
 (7094) - bit 1 - 0 = CTRL kódy se mají vykonat
 1 = CTRL kódy se mají zobrazit (pouze při print)

OUT: B - kolik znaků z max. počtu nebylo zpracováno
HL - adresa za posledním zpracovaným znakem

record IN: BC - číslo, které bylo za RECORD
E - bajt z tabulky kanálů s ofsetem +02

close IN: E - bajt z tabulky kanálů s ofsetem +02

tab je vhodné používat výše uvedených již existujících procedur

Příklad použití

Nyní si provedeme nadefinování jednoduchého zařízení. Bude se chovat jako obrazovka (CNS) s tím, že malá písmena budou změněna na velká a naopak a při otevírání a uzavírání zařízení počítač pípne. Definici přeložíme do assembleru např. EDTASMem (protokol o překladu níže). Zařízení má definovány pouze procedury open, close, print a tab; při pokusu o volání jiných procedur se ohlásí chyba. Po vložení definice a procedury print můžeme se zařízením provádět různé pokusy (příklad níže).

Protokol o překladu:

0010	00100	.RADIX 16	;implicitně hexa
F000	00120	ORG 0F000	;od &F000
F000 00	00140	START DEFB 00	
F001 0000	00160	DEFW 0000	;poslední zařízení
F003 49	00180	DEFB 'INV'	;název zařízení
F004 4E			
F005 56			
F006 7611	00200	DEFW 1176,1176;	pípnutí při ote-
F008 7611			;vření a uzavření
F00A 16F0	00220	DEFW PRINT	;adresa procedury
F00C 3761	00240	DEFW 6137,6137,	6137,6137
F00E 3761			;hlášení Err 105
F010 3761			;při nedovoleném
F012 3761			;typu přístupu
F014 3A61	00260	DEFW 613A	;TAB standardní
F016 7E	00280	PRINT LD A,(HL)	;znak na tisknutí
F017 23	00300	INC HL	;příprav další
F018 B7	00320	OR A	;je to 0 ?
F019 C8	00340	RET Z	;ano - konec
F01A FE0D	00360	CP 0D	;je to CR ?
F01C F5	00380	PUSH AF	
F01D C5	00400	PUSH BC	


```

F01E E5      00420      PUSH HL
F01F CD29F0  00440      CALL ZNAK      ;tiskni znak
F022 E1      00460      POP HL
F023 C1      00480      POP BC
F024 F1      00500      POP AF
F025 C8      00520      RET Z          ;CR - konec
F026 10EE    00540      DJNZ PRINT    ;ještě znak ?
F028 C9      00560      RET          ;ne - konec
F029 CAED10  00580      ZNAK JP Z,10ED    ;CR - odřádkuj
F02C FE41    00600      CP 41        ;je < 'A' ?
F02E 3806    00620      JR C,NEUPR   ;přeskoč
F030 FE80    00640      CP 80        ;je > 'z' ?
F032 3002    00660      JR NC,NEUPR  ;přeskoč
F034 EE20    00680      XOR 20      ;z velkých male
F036 CD8810  00700      NEUPR CALL 1088  ;vytiskniř a naop.
F039 C9      00720      RET          ;návrat
73CF        00740      ORG 73CF
73CF 00F0    00760      DEFW START  ;POKEW &73CF,&F000
0000        00780      END
00000 Total Errors

```

Příklady pokusů:

```

OPEN "INV:" AS #1
PRINT #1 "AHOJ","ahoj"
CLOSE #1
10 CLOSE #0:OPEN "INV:" AS #0
20 PRINT "abcd";TAB (10);"0123ABCD"
30 CLOSE #0

```

RUN

```

LIST "INV:"
CLIST "INV:"

```

Většina uvedených poznatků byla zjištěna empiricky pozorováním chování počítače bez znalosti vnitřní struktury BASICu-F. Je proto možné, že některé jevy nejsou zcela přesně popsány. Při praktickém použití (řadič RAM-DISKu, dálnopis) jsem se však s žádnými problémy nasetkal.

Praha, 11.6.1987

- Jan Jelínek ml. -

DEBUG M-5

=====

Tento program slouží k ladění programů ve strojovém kódu. Tvoří základní prostředek programového vybavení pro práci s mikropočítači na bázi mikroprocesoru Z80.

DEBUG-M5 obsahuje příkazy, skládající se z jednoho nebo dvou znaků a ze seznamu parametrů.

Seznam parametrů se skládá z hexadecimálních čísel oddělených mezerou. Ukončení příkazu musí být provedeno pomocí RETURN. Příkazový řádek je možno editovat. Pokud bude v příkazu chyba, příkaz se neprovede. DEBUG napíše "?" a nový pohotovostní znak ">".

Spuštění programu:

DEBUG-M5 je umístěn od adresy &E000. Pokud přejdete do BASICu, návrat do DEBUGu se provede pomocí CALL&E000.

Parametry příkazů jsou povinné a volitelné. Povinné příkazy jsou v následujícím popisu zapsány v kulatých závorkách, volitelné v závorkách šikmých.

SOUBOR PŘÍKAZŮ:

A ARITHMETIC, sečítá a odečítá dvě čísla v hexadecimálním tvaru
B BRAK, zastavení laděného programu na předem modifikované adrese
D DISPLAY, zobrazení obsahu paměti s možností měnit obsah
G GO, spuštění laděného programu
J JUMP, skok na zvolenou adresu
M MOVE, přesun pamětového bloku
PO PORT, čtení a zápis na vstupní port
R REGISER, čtení a zápis do registrů určených pro ladění
LI LIST, dekompilace obsahu paměti v mnemonice Z80
SM SET MEMORY, zaplnění obsahu pamět zvoleným obsahem
ST STATE, zobrazuje adresu zastavení
V VERIFICATION, srovnává dvě pamětové oblasti
Q QUIT, návrat z D modu

POPIS PŘÍKAZŮ:

>A (číslo1) (číslo2)

>B (adresa) - Nastavení adresy zastavení v laděném programu. Pokud nebude adresa specifikována, bude adresa zastavení vymazána. Po dosažení adresy zastavení se na obrazovce vypíše obsah všech registrů.

- >D (adrea) /délka/-Zobrazuje obsah paměti specifikovaný adresou. Je možno napsat nová data a pokračovat na další paměťovou lokaci, nebo provést písmenem "U" krok zpět. Pokud bude specifikována délka, bude vypsan paměťový blok včetně převodu do ASCII.
- >G /adresa/ - Provede spuštění programu od předem modifikovaného registru PC (příkazem "R"). Pokud bude specifikována adresa, bude tato zavedena do registru PC.
- >J (adresa) - Provede skok na adresu modifikovanou parametrem "adresa".
- >M (cílová adresa) (adresa zdroje) (délka)-Provádí přesun paměťového bloku požadované délky z adresy zdroje na adresu cílovou.
- >PD (adresa)- Provádí čtení a zápis na vstup/výstupní bránu.
- >R /jméno/ - Provede výpis všech registrů. Pokud je specifikován parametr "jméno", je možno každý registr a registrův pár modifikovat.
- >SM (počáteční adresa) (koncová adresa) (data)- Nastavuje specifikovaný úsek paměti obsahem "data".
- >ST - Zobrazuje hodnotu zastavovací adresy "break addr".
- >LI (počáteční adresa) /koncová adresa/-Dekompiluje blok paměti v mnemonice Z80. Pokud je zadána jen počáteční adresa, dekompile se zastaví po naplnění obrazovky. Stisknutím RETURN dekompile pokračuje.
- >V (adresa1) (adresa2) (délka)-Verifikuje dva paměťové bloky, dané specifikovanými parametry. Pokud jsou nalezeny nesrovnalosti, jsou průběžně vypisovány na obrazovku.

== Programy z oblasti medicíny nabízí ke kvalifikovanému využití dr. K. Ježek a MUDr. J. Naidr. Adresy jsou v redakci.

TURBOPASCAL - úpravy pro CP/M plus

Populární překladač TurboPascal vykazuje při použití na počítači CPC 6128 pod operačním systémem CP/M plus některé drobné chyby.

1. Při použití direktivy překladače [!\$U+], (možnost přerušení programu stiskem Ctrl-C), dochází při běhu takto přeloženého programu k podivným jevům, většinou dojde ke zhroucení systému. Příčinou je to, že TurboPascal používá v tomto případě opakovaně instrukci RST #38, která je rezervována pro maskovatelné přerušení. Tuto chybu můžeme opravit pomocí systémových programů SAVE a SID následujícím způsobem:

A>save

A>sid

CP/M 3 SID - Version 3.0

#eturbo.com

NEXT MSZE PC END

7980 7980 0100 CDFE

#s372

0372 38 28

0373 00 .

#s378

0378 39 29

0379 00 .

#s5395

5395 FF ef

5396 32 .

#C

CP/M 3 SAVE - Version 3.0

Enter file (type RETURN to exit): turbo.com

Delete turbo.com? y

Beginning hex address 100

Ending hex address 7980

2. Zvolíme-li v menu TurboPascalu zobrazení direktory disku zjistíme, že udávány počet volných bytů (Bytes remaining) je nesprávný. Příčinou je to, že Turbo používá při výpočtu volných bytů alokační vektor disku. Ten je však v CP/M plus umístěn v jiném banku paměti než TPA. V CP/M plus je k dispozici přímo funkce 46 - Vrať volnou oblast na disku, jejíž použití je vhodnější. Úpravu provedeme následovně:

A>save

A>sid

CP/M 3 SID - Version 3.0

#eturbo.com

NEXT MSZE PC END

7980 7980 0100 CDFE

#s2c42

2C4C 3C 21

2C56 6E cb

2C60 C6 21

2C42 1F 19

2C4D 1F 57

2C57 05 3f

2C61 2C 0

2C43 CD .

2C4E 1F 79

2C58 DD cb

2C62 E5 0

#s2c46

2C4F 1F 5e

2C59 66 1a

2C63 0E 18

2C46 E5 5f

2C50 E6 23

2C5A 06 cb

2C64 1B 12

2C47 DD 0e

2C51 1F 56

2C5B 22 1b

2C65 CD .

2C48 E1 2e

2C52 32 23

2C5C 6F 10

#C

2C49 DD cd

2C53 71 7e

2C5D 7B f8

2C4A 7E 5

2C54 7B 6

2C5E 23 eb

2C4B 03 0

2C55 DD 3

2C5F CD e5

```

CP/M 3 SAVE - Version 3.0
Enter file (type RETURN to exit): turbo.com
Delete turbo.com? y
Beginning hex address 100
Ending hex address 7980

```

František Langmaier

P R O G R A M D L A N (Display Language)

Výrobce: Campbell Software Design Ltd.

Použití: Tisk plakátů, postery, využití obrazovky jako poutač. Program umožňuje zvolit si typ písma, tvořit vlastní typy písmen, má zabudovaný program na tisk obrazovky (hard copy).

Spuštění programu: [CTRL] [SHIFT] [ESC], RUN"DLAN.

Na obrazovce se objeví hlavička s copyrightem a po načtení se změní na pokyn (v angličtině) LOAD/RUN nebo zadej vlastní program z klávesnice. Zároveň je uživatel upozorněn na předefinování numerické klávesnice: [7] RUN program uživatele, [8] vypis programu v módu 2, [9] vypis programu v módu 1. Program musí být nahráván do "čistého" počítače (proto RESET na začátku).

Princip programu:

Program je nahrán v paměti počítače od adresy #3400 (MEMORY #33FF). Pod touto hranicí může být umístěn libovolný program v BASICu, jehož součástí jsou speciální příkazy (viz níže) programu DLAN. Ty jsou umístěny v řádcích označených apostrofem (tj. v rámci BASIC-programu jsou interpretovány jako komentář). Jednotlivé příkazy jsou od sebe odděleny oddělovačem >. Aktivaci programu DLAN lze uskutečnit dvojím způsobem: buď přímo z BASICu voláním CALL #3400,N, kde N je číslo řádku (komentáře), který má být interpretován jako první, nebo stejným voláním (CALL #3400,N) v přímém módu. Je-li N=0 nebo není-li N uvedeno, interpretuje program první apostrofový řádek, na který narazí. Po aktivaci programu prochází DLAN jeden apostrofový řádek, na který narazí komentářový řádek uživatele programu za druhým, a po dosažení konce programu začíná znova od začátku. To umožňuje (spolu s možností uzamčení obrazovky proti přerušení) použít počítače jako reklamní poutač. Texty mohou být ve čtyřech barvách (mód 1). Je-li potřeba zaměnit jeden uživatelský program za jiný, je nutné použít příkaz NEW (RESET by vymazal i program DLAN).

Příklad programu:

```

100 .....
110 REM Začátek programu v kódu DLAN
120 ' wB>tK>=AHOJ>10SL
130 CALL #3400
140 .....

```

Zde w je příkaz pro výběr okna (window) B, t je příkaz pro výběr písmen typu K, = je příkaz pro text, S je příkaz pro posouvání textu (scrolling) směrem doleva (L - left), a to celkem 10krát.

Seznam příkazů programu DLAN

A[vp]... Umožňuje interakci s uživatelem při běhu programu; v je proměnná, kterou uživatel po vybidnutí z obrazovky na- plní textem podle vysvětlujícího textu p (prompt). Kde- koliv se potom dále v textu objeví v, je na tomto mís- tě vytištěn obsah proměnné v.

B[+In]... Nastavuje [mění] barvu okraje obrazovky (border); n je číslo barvy.

C[i]... Maže (clear) nastavené právě aktivní okno [inkoustem i] i=0,1,2,3.
D[n]... Zpoždění (delay) dalšího běhu programu o [n]/6 sekundy. Není-li n udáno, dosadí se 1.

F[f]R]... Kreslí rámeček (frame) typu f okolo zvoleného okna. [R mění (reverse) mezi sebou barvy papíru a pera]; [není-li f udáno, je celé okno k dispozici pro text; předdefinované rámečky jsou typu A až P]. F bez parametrů způsobí vymazání rámečku.

Gs... GOSUB na návěští :s, s lze volit z písmen a-z, A-Z a číslic 0-9.

L[F]... Uzamyká obrazovku [úplně (fully)].

Ni[, [+]c]... Vybírá inkoust i=0,...3 pro pero [s barvou (colour) c [nebo zvyšuje číslo barvy o c]]; c=0,1,...26.

Pi[, [+]c]... Vybírá inkoust i pro papír; význam stejný jako u Ni.

Q... Kopíruje obrazovku na tiskárnu typu EPSON (ne lze použít v případě tiskárny DMP-1).

R... Návrat (return) z podprogramu nebo skok na první příkaz.

S[UDLR]... Posun textu (scroll) v daném oknu nahoru (up), dolů (down), nalevo (left) nebo napravo (right).

Tt[n[, m]]... Vybírá typ písma t (z předem daných typů A až P a X,Y) [s možností volby výšky řádku n [, a vzdáleností mezi písmeny m, daných v pixel-jednotkách]].

Ww[; [n1, n2, n3, n4]]... Vybírá okno (window) w z předem definovaných oken A až Z [nebo definuje nové okno pomocí souřadnic n1, n2, n3 a n4]. Obsah okna není mazán a kurzor je umís- těn na pozici 1,1 okna [, nebo na místo, kde předchozí text skončil].

X... Vystup (exit) z DLANu do BASICu.

=text... Nový řádek s textem.

;text... Zobrazí text bez přechodu na nový řádek.

!text... Zobrazovaný text se posunuje zprava doleva.

H+/-n... Posun kurzoru o n bodů doprava (+) nebo doleva (-). Při nastavení kurzoru pro text musí být v příkazu pro text použito příkazu ; a nikoliv =, který umísťuje kurzor na začátek okna.

V+/-n... Posun kurzoru o n bodů dolů(+) nebo nahoru (-).

:s... Návěští s, na které se skáče z příkazu G.

Všechny příkazy A - Z mohou být psány malými písmeny. Také symboly pro okna, rámečky a typy písmen mohou být psány velkými nebo malými písmeny. Nepovinné parametry jsou psány v hranatých závorkách. Opakování příkazů je určeno číslicí před příkazem, avšak opakování má smysl jen před příkazy D> G> Q> S> => ;> !>

Řídící příkazy.

Není-li obrazovka uzamknuta příkazem L (lock), může být běh programu přerušen těmito řídicími příkazy:

[CTRL]X Vystup (exit) do BASICu.
[CTRL]S Krokování (step) v programu. Následné zmáčknutí libovolné klávesy způsobí posun programu o jeden krok.
[CTRL]D Diagnostika programu zobrazením vykonávaného řádku. Výhodné při současném použití s [CTRL]S.
[CTRL]A Aktivuje příkaz A...>, tj. obrací akci neparametrického A>.
[CTRL]P Kopíruje danou obrazovku na tiskárně.

Seznam předdefinovaných oken:
(souřadnice jsou uváděny stejně jako v BASICU)

A >WA1,40,1,25>, B >WB1,20,1,12>, C >WC21,40,1,12>
 D >WD1,20,13,25>, E >WE21,40,13,25>, F >WF1,40,1,12>
 G >WG1,40,13,25>, H >WH1,20,1,25>, I >WI21,40,1,25>
 J >WJ1,40,1,8>, K >WK1,40,9,16>, L >WL1,40,17,25>
 M >WM1,40,1,16>, N >WN1,40,9,25>, O >WO1,15,1,25>
 P >WP25,40,1,25>, Q >WQ1,25,1,25>, R >WR16,40,1,25>
 S >WS11,30,2,20>, T >WT1,15,1,10>, U >WU26,40,1,10>
 V >WV1,16,16,25>, W >WW26,40,16,25>, X >WX1,40,1,5>
 Y >WY1,40,1,6>, Z >WZ1,40,1,7>.

Typy písmen

U předdefinovaných znaků (písmen) je udána výška řádku a vzdálenost mezi znaky (počítáno v jednotkách pixel módu 1). Údaje o rozměrech znaků nejsou dány, ale jejich výška je úměrná výšce řádku. Výška řádku a vzdálenost může být měněna příkazem T.

A(10,1), B(10,1), C(20,1), D(16,1), E(26,1), F(24,1),
 G(24,1), H(24,1), I(24,2), J(24,2), K(30,1), L(24,1),
 M(30,2), N(36,1), P(14,1), X(10,0), Y(10,1).

Modifikace typů písma programem DLANTM

Spuštění programu DLANTM: [CTRL] [SHIFT] [ESC] RUN"DLANTM
 Jméno kódu: DLANCODE nebo jméno vlastního programu.

Po spuštění programu se na obrazovce objeví MENU:

```
-----
Draw: d=draw      e=erase      n=neutral      C=clear
Char:  g=get       s=store      E=erase
Misc:  t=typeface  l=list       u=use          V=save
-----
```

a pod ním souřadnice kursoru (počítáno od spodního levého okraje). Dále je uveden zvolený typ písma (TYPEFACE=D).

Příkazy programu DLANTM:

d... při posunu kursoru píše na obrazovku e...při posunu maže n... mění souřadnice kursoru C...maže pracovní část obrazovky g...žádá znak z klávesnice, který buď vypíše na obrazovku (je-li obsažen v seznamu vybraného typu znaků) nebo nevypíše; zadany znak lze kurorem modifikovat l...vypisuje seznam znaků daného typu na obrazovce u...žádá o text, který vypíše spolu se seznamem daného typu s...po zodpovězení otázek o šířce a výšce znaku a zadání znaku, na jehož místo bude modifikovaný znak uložen, zobrazí znak ve tvaru, v jakém bude uložen, vedle původně modifikovaného znaku (pro kontrolu), a uloží ho do seznamu znaků daného typu E...maže znak se souboru znaků V...ukládá celý soubor znaků na disk pod zvoleným názvem

Zkušenosti nabyté při používání programu DLANTM:

- rozměry znaků (písmen) v daném typovém souboru lze zjistit objetím znaku kurorem (a odečtením souřadnic z obrazovky);
- po modifikaci znaku pomocí příkazů d,e,n, je nutné nastavit kursor do levého horního rohu obdélníka, který

by bylo možno opsat zmodifikovanému znaku;
 - příkaz s ukládá zmodifikovaný znak na místo cílového (target) znaku klávesnice; tento cílový znak je nutno volit obezřetně, abychom nepřepsali znak, který budeme potřebovat;
 - cílové znaky nemusí být obsaženy ve vybraném souboru znaků; po uložení ho rozšíří.

Příklad:

dopsání háčku nad písmenem C ze souboru znaků C

t (typeface): C;

g (get): C - na obrazovce se objeví C. Souřadnice levého horního rohu opsaného obdélníka jsou (50,125). Písmeno C lze vepsat do čtverce o straně 14. Kursorem (d,e,n) přepíšeme háček. y-souřadnice vrcholu háčku je 128. Kursorem přejedeme na souřadnice (50,128);

l (list): na obrazovce se vypíšou znaky obsažené v souboru C [soubor neobsahuje malá písmena];

s (store): ceiling gap = ? 0 width = ? 14 depth = ? 17 [původní výška (14) + háček (3)]; target = ? c [není obsaženo v souboru C]; C se uloží na místo c a nepřepíše žádný ze znaků souboru C;

u (use): cC - na obrazovce se vypíše C a C typem C a seznam všech znaků souboru C. Spodek písmena C je ovšem o 3 body níže (háček) než spodek písmena C. Proto je nutno všechny znaky souboru C posunout o 3 body níže [ceiling gap se pro ně zadá 3 a ne 0].

Autor: RNDr. Václav Černý, CSc.

 Sord+Amstrad 602. technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602.Zo Svazarmu pro potřeby vlastního aktivu, zodpovědný redaktor Petr Čihula (Sord), Dr.Štefan Rybár (Amstrad). Adresa redakce 602.ZO Svazarmu, Wintrova 3. 160 41. Praha 6. Telefon: 34-14-09. Povoleno ÚVTEI pod evidenčním číslem 87 006. Cena 7.50 Kčs dle ČCÚ č.1030/202/86.
 Náklad 550 výtisků
 Praha, říjen 1987