

SKALKA

C O M M O D O R E _ _ 1 2 8

PERSONAL COMPUTER

Úvod

OSOBNÍ POČÍTAČ COMMODORE 128

Všeobecné poučení

Obrazový signál 80-ti znakového modusu je proveditelný jen na výstupu RGBI, nikoliv na výstupu pro video nebo TV - tzn. - při zapojení televizoru nebo monitoru na výstupu TV nebo video může být pracováno pouze se 40-ti znakovým modusem. Opačně zase není použitelný 40 znakový modus na výstupu RGBI.
Systémová disketa, dodávaná společně s počítačem je oboustranně popsaná. Doporučujeme Vám, abyste si z každé strany vynotovali pracovní kopii a originál. disketu uložili na bezpečném místě.

1. ÚVOD

Seznamte se s novým počítačem C 128 - "příbuzným" světoznámého C64.

Nový počítač C128 Vám umožní volbu mezi třemi provozními způsoby tak, že výhoví ~~také~~ka všem Vašim přáním a požadavkům.

- MODUS C128

– k tomuto módu je použit nový Basic 7.0 - verze Basicu, rozšířená o mnoho funkcí a příkazů, která chrání paměť se 128 KByty.

- MODUS C64

– je absolutně kompatibilní s C64. Všechny programy /Software/ z C64 lze v tomto módu použít.

- MODUS CP/M

umožňuje nasazení všech Software, které jsou určeny pro provozní systémy CP/M 3.0 a CP/M 2.2

Přepínání mezi normálním provozem a jeho modifikacemi lze provést programovatelně.

Na základě kompatibility programového vybavení počítače k C64 a k provoznímu systému CP/M 3.0 je vložitelná takřka nepřehledná nabídka programů /Software/. Tím je Commodore 128 ve své třídě jedinečný.

1.1. Vybavení

- počítač
- síťová část
- HF - kabel
- příručka o uvedení do provozu/ příručka k obsluze
- informace / záruční list

Nezkoušejte propojovat kteroukoliv část vybavení s částí jinou a nevseouvezte zástrčku do zásuvky bez předchozího pročtení návodu!

Pokud chybí, nebo je poškozena některá z výše uvedených částí počítače, informujte o tom neprodleně v obchodě, ve kterém jste počítač zakoupili. Obal uschovujte pro případné zpětné vrácení počítače.

1.2. Zapojení Commodoru 128

Na pravé straně C 128 jsou následující přípojky a zdířky:

- 1 - vypínač ZAP / VYP
- 2 - Tlačítko RESET
- 3 - zdířka na připojení do proudu
- 4 - výstupy pro 2 joysticky

Na zadní straně se nachází přípojky pro zásuvkový modul, Floppy-disk, tiskárnu a monitor, stejně jako přípojka pro kazetový přehrávač a USER PORT.

- 5 - vstup pro zásuvkový modul - na tomto místě je možné vsouvat moduly progr. vybavení /Software/
- 6 - USER PORT - možnosti připojení pro modem,

- 7 - kazetový přehrávač - přípojka pro kazetovou paměť
- 8 - seriový port - přípojka pro disketting, tiskárnu apod.
- 9 - video - zdířka - zapojení pro 40 znakový monitor
- 10 - RGBI - zdířka - zapojení monitoru na 80 znaků
- 11 - Zdířka TV - zapojení televizoru

1. 3. Umístění počítače

C 128 můžete připojit na normální televizor /Č-B nebo barevný/

nebo na monitor /např. 1701, 1702 nebo 1902 - PAL /

Nezapínejte proud, pokud nejsou všechny části společně propojené!!

Připojení na televizor

C 128 se připojí s TV přijímačem přes 75 ohmový káblík. Televizor nejprve nalaďte na 36. kanál. Káblík má na jedné straně koaxiálovou CINCH - zástrčku na druhém konci normální zástrčku TV.

Připojení na monitor

Káblík, kterým je monitor již vybaven, zastrčte do zdířky na C128, označené VIDEO / viz návod pro monitor/

Zapojení do sítě

Při zapojování počítače do sítě zastrčte jednu stranu kabelu do zdířky k tomu určené / na pravé straně počítače/.

POZOR - vypínač musí být v poloze VYP /OFF/. Teprve nyní můžete zasuňout vědlici do ~~XXXXXX~~.zásuvky.

1. 4. Zapojení C128

Nejprve zapněte monitor. Pokud používáte televizor nezapomeňte jej nejprve naladit na 36. kanál. Poté zapněte počítač. Po zapnutí se rozsvítí v levém horním kraji červená provozní žárovka / POWER/

Obrazovka

Po zapojení monitoru nebo TV se objeví na obrazovce následující obraz:

COMMODORE BASIC V 7.0 122365 BYTES FREE
/C/ 1985 COMMODORE ELECTRONIC,LTD.
/C/ 1977 MICROSOFT CORP.
ALL RIGHTS RESERVED

READY

Kurzor:

Bod v levém rohu, který během úvodního textu bliká se nazývá kurzor. Kurzor ukazuje okamžitou záznamovou polohu na obrazovce. Pokud není obraz v pořádku, musíte opravit nastavení označených ladících knoflíků na TV nebo na monitoru / viz návod k monitoru/. Jestliže ještě stále obraz nenaskočil, vyzkoušejte, zdali jsou všechna propojení v pořádku.

1.5. Klávesnice

Klávesnice C 128 připomíná klávesnici obyčejného psacího stroje. Oproti psacímu stroji má však C 128 přece jenom některé zvláštní klávesy. přídavné klávesy leží v horní části a napravo od hlavní klávesnice. Přesnější informace najdete v kapitole 3.1. v příručce k obsluze.

1. 6. Z p u s o b y p r o v o z u

Existují 3 rozdílné způsoby provozu C 128

P O Z O R !

Při výměně módu musíte programy obsažené v paměti přenést do externího nosiče dat - nejsou automaticky jištěny!

M o d u s C 128

je modus, který je v provozu / je aktivní/ po zapnutí přístroje nebo po stlačení tlačítka RESET. Je to standartní modus s Basicem 7.0., který obsahuje mnoho nových funkcí a příkazů. Vkládat můžete až 25 řádků se 40 znaky.

Někdy je výhodné / např. při zpracování textu/ zdvojnásobit počet znaků na řádku. K jeho provedení jednoduše stisknete klávesu 40/80 DISPLAY a potom tlačítko RESET. Tím se přepne²¹ počítač na reprodukci s 80 znaky.

P O Z O R !

Výstup na obrazovku ze 40 znakových na 80 znakové údaje lze přepínat postupným stlačováním kláves ESC a X a naopak.

Jak jsme se již dříve zmínili, můžete modus C128 kdykoliv přeměnit na modus C 64 nebo CP/M. Modus C 128 lze však zpět vrátit pouze vypnutím přístroje nebo použitím tlačítka RESET.

M o d u s C 64

V tomto modu můžete použít všechny programy C64 /Software/ a programy, které jsou psány Basicem 2.0. U módů C64 lze mimo to připojit a použít všechny periferní přístroje počítače C64.

Existují 3 varianty, jak přepojit na modus C 64.

1. Udejte příkaz GO 64 a stiskněte klávesu RETURN
2. Vypněte počítač a do vstupu pro zásuvkový modul zastrčte modul programového vybavení /Software/ C 64. Potom počítač znova zapněte, automaticky se napojí na modus C 64.
3. Zapněte C 128 za současného stlačení / stálý stisk klávesy/ klávesy označené značkou Commodore.

Změna z modu C 64 na modus C 128 je možná pauze vypnutím přístroje nebo použitím tlačítka RESET.

P O Z O R !

Nikdy nelze přímo měnit z módu C 64 na modus CP/M.

M o d u s C P / M

CP/M 3.0. je provozní systém, který vyvinula firma Digital Research Corporation, a který má obsáhlé programové vybavení.

Existují 2 možnosti jak tohoto módu dosáhnout:

1. Do počítače vložte BOOT a poté stiskněte klávesu RETURN.
2. Do diskettingu vložte disketu CP/M a zapněte počítač. Dejte pozor na to, aby byl disketting zapnut dřív, než zapnete C128.

V módu CP/M je obraz nastaven na 40 nebo 80 znaků podle polohy klávesy 40/80 DISPLAY

Z módu CP/M na modus C 128 lze přejít pouze vypnutím přístroje, nebo stisknutím tlačítka RESET. Přitom musí být disketa vyjmuta z diskettingu.

1. 7. Vložení a spuštění programu

Co je to SOFTWARE?

Software je souhrn všech instrukcí, které Vašemu počítači sdělí, co má dělat - tedy jsou to provozní a aplikované programy. Mnohé programy lze zakoupit v obchodě, některé si můžete napsat sami. Programy, které lze zakoupit obdržíte ve 3 různých nosičích dat: disketa, kazeta, nebo zásuvkový modul

Disketa - může být použita jen v souvislosti s určitým disk-driveem

Kazeta: program z kazety lze přehrát pomocí kazetového přehrávače Commodore

Zásuvkový modul:

je umělé pouzdro, obsahující konstantní nastavenou hodnotu programu. Používá se zasunutím do zdírky označ. číslem 5 / zadní strana počítače/

Z následující tabulky si vyberete, který ze 3 nosičů dat použijete k vložení programu - CP/M program existuje však jen na disketě.

	C 128	C 64	CP/M
Disketa	x	x	x
Kazeta	x	x	■
zás. modul.	x	x	

vložení a spuštění programu se děje ve dvou krocích

1. Program začnete vkládat do počítače příkazem LOAD nebo DLOAD / viz kap. 4.6. nebo 5.2. v návodu k obsluze/
2. Udáte příkaz RUN

Program z diskety

S disketou je snadná manipulace. Nezkoušejte odstranit plastový obal, slouží jako ochrana a disketa tím může být poškozeny. Na C 128 lze připojit diskettingy 1541 nebo 1571.

Přípojny kabel pro disketting musí být zasunut na zadní straně počítače v 6 pólové zdířce. Nepokoušejte se jej zasouvat do 8-pólové zdířky pro video!

P O Z O R !

Disk-drive musí být vzdálen minimálně 70 cm od ostatních elektro-spotřebičů - mohly by způsobit rušení!

Vložení programu C 128

- Disketu opatrně vsuňte do diskettingu tak, aby nálepka byla otočena směrem nahoru a ochranný zářez záznamu byl na levé straně diskety.
- Páčku stiskněte opatrně směrem dolů
- Vložte:
DLOAD "Programname" /adresa, jméno programu/
• Za "název programu" dosaďte jméno programového návěstí, které jste zvolili. Poté stiskněte klávesu RETURN.

Pokud chcete vložit první program z diskety, můžete tak učinit zkrácenou formou

DLOAD "x"

Potom stiskněte klávesu RETURN - červená žárovka na diskettingu se rozsvítí a na obrazovce se objeví následující text:

Searching for Programname
LOADING

Pokud je program vložen, ohláší počítač : READY

Potom vložíte příkaz RUN a stisknete klávesu RETURN

Počítač provede program.

Některé programy provedou příkazy LOAD nebo RUN samostatně.

Bližší vysvětlení najdete v popisech přiložených k programům.

Pokud pracujete s verzí C 64, je nutné dát následující pokyn k vložení programu:

LOAD " Programname", 8

- jinak se úvodní proces od módu C 128 neliší.

Výše uvedené pokyny platí jak pro vlastní, tak i pro zakoupené programy. V návodu k obsluze jsou pokyny DSAVE/SAVE, stejně jako DLOAD/LOAD přesněji popsány. / body 4.6. a 5.2./

Předtím, než vložíte program do nové diskety, musíte jej nejprve zformovat, přičemž vkládáte následující příkaz:

HEADER " Name" , I Kennung /hesko/, DO

Za Name dosaďte 16 místné návští /jméno/ a za Kennung /identifikace, heslo/ jeden ze dvou libovolných znaků platných k identifikaci diskety.

DO je označení unašeče

Tento příkaz je blíže objasněn v návodu k obsluze v kapitolách 4.6. a 6.1.

Vložení a spuštění programu v módu CP/M

- dejte pozor na to, abyste měli zapnutý disketting, potom vložte disketu.
- udejte příkaz

BOOT

Indikace na obrazovce

A>

Vám ukáže, že modus CP/M je připraven na vkládání údajů z klávesnice.,

Vložení jednoho z popsaných CP/M programů se děje udáním programového jména s konečnou klávesou RETURN.

Program v seznamu disket musí být obsažen se jmenným rozšířením názvu COM.

Vložení programu ze zásuvkového modulu

Na co musíte dát pozor:

- předtím, než zasunete program zásuvkového modulu musíte vypnout počítač
- potom zasuňte zásuvkový modul do zadní strany počítače - nálepka přitom musí být na horní straně
- teprve potom zapněte počítač

P O Z O R !

Pokaždé, když zasouváte nebo vytahujete zásuvkový modul, nezapoměňte vypnout počítač! Jinak může dojít k poškození počítače i modulu.

Počítač nyní automaticky pracuje s programem, uloženým v zásuvkovém modulu. Potom následujte pokyny, které Vám dá počítač.

Bližší informace najdete v návodu, který je k zásuvkovému modulu připojen.

Pokud připojíte na C 128 kazetový přehrávač, můžete programy z kazety číst, ev. na kazetu nahrát, a to jak na módě C 128, tak i na módě C 64

Bližší podrobnosti najdete v popisu příkazů LOAD a SAVE v návodu k obsluze - kapitola 5.2.

Příručka k obsluze C128

Členění příručky

OBSAH:	čís. strany v orig.
1. Úvod	1 - 1
2. Všeobecné vlastnosti Basicu	
2.1. Druhy provozu Basicu	2 - 1
2.2. Řádkový formát a zásoba znaků v Basicu	2 - 2
2.3. Slovní zásoba	2 - 3
2.4. Konstanty v Basicu	2 - 5
2.5. Proměnné v Basicu	2 - 6
2.5.1. Názvy a typy proměnných	2 - 6
2.5.2. Proměnné v poli	2 - 7
2.6. Přesnost numerických hodnot	2 - 8
2.7. Numerické výrazy a operátory	2 - 9
2.7.1. Aritmetické operátory	2 - 9
2.7.2. Relační operátory	2 - 9
2.7.3. Logické operátory	2 - 11
2.7.4. Funkční operátory	2 - 13
2.7.5. Matematická hierarchie	2 - 13
2.8. Výrazy a operace s řetězcovými znaky	2 - 14
2.8.1. Řetězcové operátory	2 - 15
2.8.2. Funkce s řetězcovými argumenty	2 - 15
3. Vstupy a řízení programů v Basicu	
3.1. Klávesnice	3 - 1
3.2. Řádka programu	3 - 4
3.3. Nahrazení nebo změna programových řádků	3 - 6
3.3.1. Změna řádků s meditorem obrazovky	3 - 6
3.3.2. Změna řádků s chybou syntaxu	3 - 6
3.4. Vložení programu Basicu	3 - 7
3.5. Spuštění programů Basicu	3 - 7

4. Modus C128

4.1. Rozšířené funkce klávesnice v módu C128	4 - 1
4.2. Rozšířené funkce na obrazovce v módu C128	4 - 8
4.3. Basic 7.0. v módu C128	4 - 9
4.4. Struktura programu v jazyku Basic	4 - 10
4.5. Struktura a syntaxe ekvivalentů	4 - 11
4.6. Povely, příkazy, funkce a proměnné	4 - 12
4.7. Barvy, Sprites a grafické efekty	4 - 135
4.8. Zvuky a hudba s módem C128	4 - 152

↓ 2

5. Modus C 64

5.1. Program v jazyku Basic 2.0. v módu C 64	5 - 1
5.2. Povely, příkazy, funkce a proměnné	5 - 1
5.3. Barva a grafika v módu C 64	5 - 74
5.3.1. Sprite-grafika v módu C 64	5 - 84
5.4. Hudba v módu C 64	7 - 94

6. Provoz unašeče /Floppy disku/ s programem v Basicu

6.1. Formace disket	6 - 1
6.2. Vložení programů	6 - 3
6.3. Užití znaku JOKER v názvech	6 - 4
6.4. Spuštění programů	6 - 5
6.5. Seznamy disket	6 - 6
6.6. Otevření a uzavření dat	6 - 9
6.7. Záznamový a snímací ukazatel v relativních údajích	6 - 11
6.8. Vymazání údajů na disketě	6 - 13
6.9. Další funkce řízení	6 - 14
6.9.1. Zaplnění diskety, přezkoušení a uspořádání	6 - 14
6.9.2. Kopírování údajů	6 - 15
6.9.3. Uspořádání sekvencních dat do řetězců	6 - 15
6.9.4. Přejmenování údajů	6 - 16
6.9.5. Zdvojování diskety	6 - 17

7. Modus CP/M - viz příručka CP/M

8. Signalizace chyb

8.1. Signalizace chyb programu v Basicu	8 - 1
8.2. Signalizace chyb ve floppy disku	8 - 6

Dodatek A: Tabulky kódových znaků, řídící kódy

Dodatek B: Seřazení paměti

Dodatek C: Strojový jazyk - monitor

Dodatek D: Zvláštnosti klávesnice DIN.

Dodatek E: Rozvrh registru paměti u SID a VIC

Dodatek F: Tabulka hudebních not

Dodatek G: Zvláštnosti v módu C 64 /klávesy funkcí, grafika/

Dodatek H: Uspořádání nulové stránky /zero page/

Dodatek I: Zkratky Basicu

Dodatek J: Definice specificky použitelných vět/souvětí/ znaků

Dodatek K: Odvozené matematické funkce

Dodatek L: Uspořádání vidlic

Dodatek M: Přenos programů v Basicu 4 do Basicu 7

Všeobecné poučení k této příručce

Obrazový signál 80-ti znakového módu na obrazovce je možný pouze na výstupu RGBI, nikoliv na výstupu Video nebo TV. Při zapojení do zdiřky TV nabo viedo lze pracovat pouze se 40 znakovým módem. Opačně zase 40-ti znakový módus není realizovatelný při zapojení na výstupu RGBI!

Systémová disketa CP/M je oboustranně popsána. Doporučujeme Vám zhodnotit si jednu pracovní kopii a originální disketu uschovat na bezpečném místě.

1. ÚVOD

Nový počítač C 128 Vám dovolí volbu mezi třemi provozními způsoby tak, že vyhoví bezmála všem Vašim speciálním přáním a požadavkům.

• Modus C 128

k tomuto účelu je použit nový Basic 7.0, verze Basicu, rozšířená o mnoho funkcí, příkazů a povelů (instrukcí), která chrání paměť se 128 K- Byty.

• Modus C 64

je absolutně kompatibilní k C 64. Všechna programová vybavení počítače C 64 lze použít.

• Modus CP/M

umožňuje nasazení všech programových vybavení počítače (Software), které jsou určeny pro provozní systémy CP/M 3.0 a CP/M 2.2

Přepínání mezi provozem (normálním) a jeho modifikacemi lze provést programovatelně.

Na základě zaměnitelnosti programového vybavení počítače k C 64 a k provoznímu systému CP/M 3.0 je vložitelná téměř nepřehledná nabídka programů (Software). Tímto je Commodore 128 jedinečný ve své třídě.

2. Všeobecné vlastnosti jazyku Basic

- 2.1. Druhy provozu Basicu
- 2.2. Řádkový formát a zásoba znaků v Basicu
- 2.3. Slovní zásoba
- 2.4. Konstanty v Basicu
- 2.5. Proměnné v Basicu
- 2.6. Přesnost numerických hodnot
- 2.7. Numerické výrazy a operátory
- 2.8. Výrazy a operátory s řetězcovými znaky

2. VŠEOBECNÉ VLASTNOSTI JAZYKA BASIC

Tato kapitola obsahuje důležité informace o mnohých podstatných vlastnostech programového jazyka Basic. Ti, kteří nemají s programováním v Basicu na Commodoru zatím žádné zkušenosti by jí měli věnovat patřičnou pozornost. Tato kapitola ovšem obsahuje mnoho důležitého i pro odborníka.

2.1. DRUHY PROVOZU V BASICU

Program v Basicu má 2 způsoby provozu - příkazový a programový. Úroveň příkazů (Direktmodus) je aktuální ihned po zapnutí počítače, nebo hlášením chyby, ev. po řádném ukončení programu Basicu.

Na obrazovce je označena výstupem

READY

a pod ním blikajícím kurzorem

Na úrovni příkazů mohou být provedeny téměř všechny příkazy a funkce jazyka, které se udávají bez předchozího zadání čísla řádku. Stisk klávesy RETURN v každém případě uzavírá zadání a předává je počítači k provedení. Po provedení operace se nachází Basic opět na úrovni příkazů.

Př. 1 READY

PRINT 5 * 2

10

READY

Pokud se přece jen před udávaným příkazem předřadí číslo, přejímá interpret tento řádek jako řádek programu do hlavní paměti, a příkazy pod tímto číslem obsažené provádí teprve po vložení příkazu RUN nebo po odvolávce GOTO, vztahující se k tomu určitému řádku.

Programy mohou být vkládány tímto způsobem:

Př. 2 READY

10 PRINT " VÝSLEDEK JE "

20 PRINT 5*2

RUN

VÝSLEDEK JE 10

READY

V úrovni programu s počítač nachází v momentě, kdy uživatel odstartuje vložený program.

Kurzor se ztratí (vyjímku tvoří jen příkaz INPUT pro vkládání dat přes klávesnici) a objeví se opět až se přístroj po provedení programu vrátí opět do polohy příkazů(viz. př. 2)

2.2. Řádkový formát a zásoba znaků v Basicu

Program v Basicu se skládá z řádek programu, které musí mít následující formu:

nnnnn příkaz [:Příkaz.] [: REM komentář]

nnnnn - číslo řádku jako celé číslo mezi 0 a 63999 , které určí pořadí řádek programu v hlavní paměti a slouží jako vztažné (referenční) číslo pro větvení programu.

Příkaz: libovolný příkaz v Basicu nebo libovolná instrukce či proměnná v jazyku Basic (viz kap. 4.6. a 5.2.).

Víc takových příkazů na jednom řádku musí být odděleno dvojtečkou(:).

Komentář : volný text ke komentování programu, který následuje po instrukci REM

Poznámka: celková délka programové řádky nesmí překročit 160 znaků (v módu C 64 max. 80 znaků).

Vedle velkých a malých písmen latinské abecedy jako i čísel od 0 do 9, má Basic řadu zvláštních znaků, které mají zvláštní význam:

viz tab. následující strana.

Tabulka:

Znak	Význam
+	plusový nebo řetězcový symbol pro řetězcové výrazy
-	znaménko minus, nebo spojovací čárka
*	hvězdička nebo znaménko násobení
/	znaménko dělení nebo lomená čára
↑	umocňovací znaménko
=	rovnítko nebo znaménko přiřazovacího příkazu
<	znaménko menší než nebo levá strana závorky
>	znaménko větší než nebo pravá strana závorky
(levá závorka
)	pravá závorka
[levá hranatá závorka
]	pravá hranatá závorka
%	typové znaménko pro celé číslo nebo znaménko procenta
#	číselný znak
\$	typové znaménko pro řetězcové výrazy nebo označení USdolaru
.	desetinná tečka nebo tečka
,	čárka nebo tabulovací znak
:	rozdělovací znaménko pro příkazy nebo dvojtečka
;	středník nebo potlačení posunu řádku
"	začátek nebo konec řetězcového znaku nebo uvozovka
?	zkratka pro povel PRINT nebo otazník

Všechna tato znaménka mohou být normálními znaky součástí řetězcových výrazů.

Ostatní znaménka jsou obsažena v tabulce kódových znaků ASC II v příl.

2.3. Slovní zásoba

Slovní zásoba programového jazyka Basic je označení pro různé příkazy, povely, funkce, proměnné a logické operátory. Nemůže být zaměňována s proměnnými názvy /jmény/.

Celá zásoba slov je v dalším textu uspořádána podle abecedy.

Tabulka:

ABS	DLOAD	INPUT #	RCLR
AND	DO	INT	RDOT
APPEND	DOPEN		READ
ASC	DRAW	JOY	RECORD
ATN	DS		REM
AUTO	DSAVE	KEY	RENAME
	DS\$		RENUMBER
BACKUP	DVERIFY	LEFT\$	RESTORE
BANK		LEN	RESUME
BEGIN	EL	LET	RETURN
BEND	ELSE	LIST	RGR
BLOAD	END	LOAD	RIGHT\$
BOOT	ENVELOPE	LOCATE	RND
BOX	ER	LOG	RREG
	ERR\$	LOOP	RSPCOLOR
BSAVE	EXIT		RSPPOS
BUMP	EXP	MID\$	RSPRITE
		MONITOR	RUN
CATALOG	FAST	MOVSPR	RWINDOW
CHAR	FETCH		
CHR\$	FILTER	NEW	SAVE
CIRCLE	FNxx	NEXT	SCALE
CLOSE	FOR	NOT	SCNCLR
CLR	FRE		SCRATCH
CMD		ON	SGN
COLLECT	GET	OPEN	SIN
COLLISION	GETKEY	OR	SLEEP
COLOR	GO64		SLOW
CONCAT	GOSUB	PAINT	SOUND
CONT	GOTO	PEEK	SPC
COPY	GRAPHIC	PEN	SPCOLOR
	GSHAPE		SPRDEF
DATA		PLAY	SPRITE
DCLEAR	HEADER	POINTER	SPRSAV
DCLOSE	HELP	POKE	SQR
DEC	HEX\$	POS	SSHAPE
DEF		POT	ST
DELETE	IF	PRINT	STASH
DIM	INSTR	PRINT #	STEP
DIRECTORY	INPUT	PUDEF	STOP

Tabulka slovní zásoby - pokračování

SWAP	TI	UNTIL	WAIT
SYS	TIS	USING	WHILE
	TO	USR	WIDTH
TAB	TRAP		WINDOW
TAN	TRON	VAL	
TEMPO	TROFF	VERIFY	XOR
THEN		VOL	

2.4. KONSTANTY V BASICU

Konstanty jsou aktuální hodnoty, které používá interpret Basicu při provádění programu. Přitom rozlišuje mezi konstantami řetězcových znaků a číselnými konstantami.

Řetězcová konstanta

je sled až 255 libovolných /alfanumerických/ znaků, uzavřených v uvozovkách.

např.: " dům"

 " dům a kočka "

 " DM 12.80 "

Číselné konstanty

jsou kladná a záporná čísla, která mohou kromě znaménka + nebo - obsahovat pouze desetinnou tečku /nikoliv čárku/.

-Interpret Basicu rozlišuje 3 různé druhy číselných konstant

Celočíselné konstanty

jsou celá čísla v rozsahu mezi -32768 a +32767 bez desetinné tečky
např. 125

Konstanty s pevnou řádovou čárkou

jsou všechna kladná nebo záporná reálná čísla s jednou desetinnou tečkou

např. 15.12

Konstanty s pohyblivou řádovou tečkou /čárkou/

jsou všechna kladná nebo záporná čísla v exponenciálním znázornění /vědecké znázornění/. Skládají se z celé nebo z pevné řádové čárky Mmantisse/ s nebo bez určovacího znaménka, po kterém následuje písmeno E a celé číslo v rozsahu od -39 do +38 s nebo bez znaménka, které určí exponent k Basicu 10

např. 15E-3 má stejný význam jako 0.015

146.285E-6 - " - -"- 0.000146285

řádovou

Počítač akceptuje konstanty s pohyblivou čárkou v rozsahu od

od 1.7E+38 do 2.9E-39 a od -2.9E-39 do -1.7E+38

2.5. PROMĚNNÉ V BASICU

Proměnné jsou názvy, které představují hodnoty používané v programu Basicu. Počítač přitom rozlišuje mezi řetězcovými proměnnými a číselnými proměnnými

Řetězcové proměnné se mohou skládat až z 255 znaků. Počet těchto znaků určuje délku hodnot proměnných.

Číselné /numerické/ proměnné mají vždy jednu hodnotu, která sestává z jednoho čísla.

Hodnota jedné proměnné může být určena programátorem nebo výsledkem operací programu.

Je-li proměnná použita dříve, než jí byla přidělena hodnota, je její hodnota v případě číselné proměnné 0 a v případě řetězové proměnné prázdný řetězec o délce 0 /leer string/.

2.5.1. NÁZVY A TYPY PROMĚNNÝCH

Počítač Basicu akceptuje libovolně dlouhé proměnné, přesto vykládá jen 2 první znaky, tak jako poslední, který definuje typ /druh/ proměnné, a který musí být následován zvláštním označením \$ nebo %, pokud má být jiný typ definován jako pohyblivá řádová čárka.

Název proměnné musí jako první znak obsahovat písmeno abecedy a může se skládat z písmene a číslic v libovolné kombinaci.

Název proměnné nemůže být názvem žádného ze základních slov Basicu /víz kap. 2.4./

Příklady: SIN = 101zakázáno! /SIN-funkce/

FNS = "jméno" ..zakázáno! /uživatelská funkce DEFFN/

Název jedné proměnné neslouží jen k jejímu označení, ale také k definici jejího typu.

Řetězcové proměnné jsou definovány znaménkem \$ jako posledním znakem v názvu

např. CG = "COMMODORE"

Numerické proměnné celého čísla /typ integer/ jsou určeny znakem % jako posledním znakem v názvu:

DEN% = 30

Numerické proměnné s pohyblivou řádovou čárkou /typ real/
nemají žádný zvláštní znak na konci názvu
Daň=15.84

Podle druhu číselné proměnné je potřeba místa při
proměnné celého čísla 2 Byty
proměnné s pohyblivou řádovou čárkou 5 Bytů /10 míst/

2.5.2. PROMĚNNÉ V POLI

Pole je skupina nebo tabulka číselných nebo řetězcových hodnot, která je představována názvem proměnných. Každá z těchto hodnot je označena elementem pole a popsána indexem názvu proměnné. Tento index je celé číslo nebo celočíselný výraz. Jednomu názvu proměnné v poli je přiděleno přesně tolik indicií, kolik má pole dimenzí /rozměrů/.

Dimenzování pole, se kterým je spojeno současně také pojmenování, musí být deklarováno příkazem DIM /viz kap. 5.2./

Příklady:

DIM X /25/ dimenzuje se jednorozměrné pole s pohyblivou řádovou čárkou X se 26 elementy /0až 25/

DIM X\$ /2,6/ dimenzuje se dvojrozměrné řetězcové pole s 21 elementy /3*7/

Dimenzováním pole nabývají všechny články dimenzovaného pole hodnotu nula.

Index určuje polohu u článku v poli.

Max. počet článků v každém rozměru je 32767, je však možno použít max. 255 rozměrů. V každém případě ohraničuje použitelná paměť velikost jednoho pole.

U dvojrozměrných polí udává první index řádek a druhý index mezeru tabulky vytvořenou skrze takové pole.

Jednorozměrná pole až s 11 elementy nemusí být dimenzována, je-li v programu užit příkaz

A(6)= 12.3

bez předchozího dimenzování písmene A, provede počítač interně příkaz DIM A (10) a tím se sám dimenzuje.

Pro zadání názvu a typu platí ta samá pravidla jako pro jednoduché proměnné /viz kap. 2.5.1./

2.6. PŘESNOST NUMERICKÝCH HODNOT

Numerické hodnoty, o kterých jsme se právě zmínili, mohou být vloženy do paměti jako celá čísla, nebo jako reálná čísla.

Celočíselné hodnoty jsou uloženy ve dvou bytech hlavní paměti.

Reálné hodnoty jsou vloženy do paměti jako čísla s pohyblivou rádovou čárkou s 10-ti místnou přesností.

Konstanty, které nejsou celými čísly, mohou být psány následovně:

- až do 10 znaků např. 100.59
- znázornění exponentu s E např. 3.57E-05

2.7. NUMERICKÉ VÝRAZY A OPERÁTORY

Numerickým výrazem může být číselná konstanta nebo proměnná, nebo kombinace numerických konstant, proměnných a operátorů k výpočtu jednotlivých hodnot. Operátory přitom provádějí s hodnotami matematické nebo logické operace. Počítač s jazykem Basic má 4 různé kategorie operátorů:

- aritmetické operátory
- logické operátory
- funkční operátory

2.7.1. ARITMETICKÉ OPERÁTORY

Existuje 6 aritmetických operátorů.

Jsou to tyto, seřazené podle matematické hierarchie v číselném vyjádření:

operátor	operace	příklad
\uparrow	umocňování	A^5
-	negace	$- A$
$\times, /$	násobení, dělení	$A^5, A/5$
$+, -$	sčítání, odčítání	$A+5, A-5$

2.7.2. SROVNÁVACÍ OPERÁTOŘE

Srovnávací operátory slouží k porovnání druhé numerické nebo řetězcové hodnoty. Výsledek je buďto logicky "pravda" s hodnotou 1 nebo logicky "nepravda" s hodnotou 0 /nula/ a je používán zejména ve spojitosti s návodem IF /viz kap. 5.2./ k řízení průběhu programu.

V následující tabulce je seřazeno 6 srovnávacích operátorů, které basicový interpret ovládá:

operátor	srovnání	příklad
=	rovnost	A=B
<>, ><	nerovnost	A<>B, A><B
<	ménší než	A < B
>	větší než	A > B
<=, =<	ménší než nebo rovno	A<=B, A= < B
>=, =>	větší než nebo rovno	A>=B, A= > B

Pokud se v jednom výrazu objeví aritmetický i srovnávací operátor, jsou nejdříve zpracovány aritmetické operátory.

Příklady:

$A + B < (x - 4) / Z$

Tento výraz je "pravda" /následuje hodnota -1 / pokud je hodnota $A+B$ menší než hodnota $x-4$ děleno Z.

100 IF SIN(X) < 0 THEN 1000

pokud je sin x záporný, má srovnání "pravdivý" výsledek /-1/ a program se přesune na řádek 1000.

Při srovnávání řetězců se u obou řetězců podle znaků srovnává code CBM - ASCII.

Všechny znaky, ale také vedoucí a následná prázdná místa jsou srovnávána. Účinek je znatelný na následujících příkladech, které jsou všechny "pravdivé" a proto mají hodnotu -1

"AA" < "AB"
"Aa" < "AA"
"DUM" = "DUM"
"DUM" > "HANS"
"A!" < "A\\$"
"U" > "U"
X\\$ = "12": X\\$ < "13"

2.7.3. LOGICKÉ OPERÁTORY

Logické operátory slouží k testování mnohonásobných srovnání, k manipulaci s Bity nebo k provádění Booleových operací s numerickými hodnotami.

Logický operátor spojuje dvě operanda jako kombinaci hodnot "pravda" "nepravda" na úrovni bitů a produkuje jako výsledek hodnotu, která je interpretována buďto jako "pravda" /jiná než nula/ nebo "nepravda" /nula/

Počítač provádí operace s logickými operátory ve smíšeném výrazu po aritmetických a srovnávacích operacích.

Interpret zná 3 různé logické operátory, které jsou po řadě srovnávány tak, jak jsou zpracovány a znázorněny v tabulce pravdivosti:

Operátor	Operand 1	Operand 2	výsledek
NOT/logický komplement/	pravda	-	nepravda
	nepravda	-	pravda
AND/log. součin/	pravda	pravda	pravda
	pravda	nepravda	nepravda
	nepravda	pravda	nepravda
	nepravda	nepravda	nepravda
OR /log. součet/	pravda	pravda	pravda
	pravda	nepravda	pravda
	nepravda	pravda	pravda
	nepravda	nepravda	nepravda

Přesně tak, jak mohou přispívat srovnávací operátory svými výsledky k řízení, může tak dojít i spojením dvou nebo více srovnání logických operátorů, které vykazují opět hodnoty "pravda" nebo "nepravda" /viz instrukce IF v kap. 5.2./

Následující příklady řádků programu celou záležitost objasňují:

100 IF D < 200 AND F < 4 THEN 80
110 IF I > 10 OR K < 0 THEN 50
120 IF NOT P THEN 100

Logické operátory pracují uvnitř následovně:

Nejdříve se oba operandy přemění v celý, znaménkem označený binární komplement se 16 bitů čísel v rozsahu mezi -32768 a +32767.

Jsou-li operandy větší nebo menší, objeví se oznamení chyby

ILLEGAL QUANTUM ERROR /nepovolená hodnota/

Jsou-li oba operandy 0 nebo -1, provede logická operace právě 0 nebo -1

Operace se v každém případě provádí na způsob bitů, tzn. že každý bitový výsledek je určován oběma odpovídajícími bity v operandech. Přitom značí bit 1 "pravdu" a bit 0 "nepravdu". Tím je možné pomocí logických operátorů testovat vzor bitu paměťových řádků. Např. na propojovacím místě výstupu/vstupu může být ukryt Byte, aby se mohl testovat stav určitého Bitu /viz instrukce WAIT v kap. 5.2./. S operátorem OR může být také v určitém Bytu vyroben docela určitý vzor Bitu. Následující příklady objasní způsob práce logických operátorů:

63 AND 16 = 16	0000000000111111	63
	AND 000000000010000	16
	000000000010000 =	16
15 AND 14 = 14	0000000000001111	15
	AND 000000000001110	14
	000000000001110 =	14
-1 AND 8 = 8	1111111111111111	-1
	000000000001000	8
	000000000001000 =	8
4 OR 2 = 6	000000000000100	4
	OR 000000000000010	2
	0000000000000110 =	6
10 OR 10 = 10	000000000001010	10
	OR 000000000001010	10
	000000000001010 =	10
-1 OR -2 = -1	1111111111111111	-1
	OR 1111111111111110	-2
	1111111111111111 =	-1
NOT 1 = -2	0000000000000001	1
	NOT 1111111111111110 =	-2

Operátor NOT tvoří jednotkový logický doplněk operandu.

2.7.4. FUNKČNÍ OPERÁTORY

Funkční operátory /funkce Basicu/ se užívají k provádění určitých operací na operandu. Takové funkce, obsažené v interpretu Basicu jsou např. SQR /odmocňování/, nebo SIN /trigonometrický sinus/. Je také možné definovat vlastní funkce s pomocí příkazu DEF-FN. Podrobnosti k funkcím jsou popsány v kap. 4.6. a 5.2.

2.7.5. MATEMATICKÁ HIERARCHIE

Při zpracování numerických výrazů a operátorů dodržuje interpret Basicu následující matematickou hierarchii / 1 nejvyšší, 8 nejnižší priorita/.

1. Vyvolání funkcí
2. Umocňování / \uparrow /
3. Logická inverze / změna z plusu do minusu a naopak/
4. Násobení a dělení ($*$, $/$)
5. Sčítání a odčítání / $+$, $-$ /
6. Logický komplement /NOT/
7. Logický součin /AND/
8. Logický součet /OR/

Následné operátory toho samého stupně hierarchie jsou zpracovány zleva do prava.

Hierarchie může být vyrovnaná použitím svorek. Výrazy ve svorkách jsou zpracovávány nejdříve. U uzavřených svorkových výrazů je zpracován nejprve výraz vnitřní svorky.

Následující příklady Vám toto přiblíží:

$4 + 1 * 2$	rovná se 6
$(4 + 1)^2$	-" - 10
$100 * 4/2 - 1$	-" - 199
$100 * (4/2 - 1)$	-" - 100
$100 * (4/(2-1))$	-" - 400

2.8. ŘETĚZCOVÉ VÝRAZY A OPERACE S ŘETĚZCOVÝMI ZNAKY

Řetězcovým výrazem může být řetězcová konstanta nebo proměnná, nebo kombinace řetězcových konstant, proměnných a operátorů k vyrobení nového řetězce.

Přitom operátory kombinují řetězce k novým řetězcům.
Interpret Basicu zná 2 typy řetězcových operátorů:
• sdružený /sřetězený / operátor
• funkční operátor nebo funkce řetězcových znaků

Protože srovnání řetězců je v principu numericky srovnáním, je popsáno v kap. 2.7.2. u srovnávacích operátorů.

2.8.1. SDRUŽENÝ OPERÁTOR

Řetězce se mohou spojovat znaménkem plus/+, aby se mohly vytvořit řetězce nové.

10 A\$ = "DÉŠŤ" : B\$ = "DEŠTNÍK"

20 PRINT A\$ + B\$ + "Y"

RUN

DEŠTNÍKY

READY

2.8.2. ŘETĚZCOVÉ FUNKCE

Tak jako u numerických funkcí /viz kap. 2.7.4./ provádí řetězcová funkce na jednom nebo více operandech určené operace, které jako výsledek produkuji řetězec.

Tyto funkce obsažené v interpretu jsou např.:

CHR\$ /vytvoření řetězce z kodu ASCII/ nebo

STR\$ /vytvoření řetězcového ekvivalentu k numerickému výrazu/.

Přesto není možné definovat vlastní řetězcové funkce za pomocí příkazu DEF FN.

Detailedy k řetězcovým funkcím jsou podrobně popsány v kap. 4.6.a 5.2.

3. VKLÁDÁNÍ A ŘÍZENÍ PROGRAMŮ BASICU

- 3.1. Klávesnice
- 3.2. Vkládání řádek programu
- 3.3. Nahrazení nebo změna programových řádek
- 3.4. Vkládání programu Basicu do paměti
- 3.5. Vkládání programů z vnějších medií

3.1. KLÁVESNICE

Zobrazení 3.1. na následující straně ukazuje klávesnici C128. V módu C128 mohou být použity všechny klávesy, v módu C64 pouze klávesy šrafované. V tomto oddílu jsou popsány všechny klávesy které mohou být použity v obou módech.

Modifikace klávesnice

Klávesnice C128 má dvě modifikace provozu; jsou to:

- velká písmena / grafické symboly
- malá písmena / velká písmena

Po zapnutí počítače je aktivní modus "velká písmena / grafika. Abychom mohli měnit mezi oběma módy, zároveň se stiskne přepínací klávesa /SHIFT/ a klávesa C /Commodore/.

Klávesnice psacího stroje

Hlavní část klávesnice zaujímá klávesnice psacího stroje, která tak i funguje.

Vedle alfanumerických kláves je tu i řada obslužných kláves, jejichž funkce je částečně jiná, než u psacího stroje.

Přepínací klávesy

Přepínací klávesy /SHIFT/ slouží podle aktivního módu klávesnice ke vkládání velkých písmen nebo zvláštních znaků, které jsou zobrazeny na horní straně dvojitě obsazených kláves /např. <>/. S odpovídající znakovou klávesou musí být společně stisknuta buďto pravá nebo levá přepínací klávesa. Přepínání na velká písmena lze trvale zablokovat i pevně polohovanou klávesou SHIFT-LOCK.

Klávesa RETURN

odpovídá zpětné klávese u elektrického psacího stroje. Ukončuje zadání řádku programu.

Klávesa RESTORE

slouží společně s klávesou RUN/STOP k uvedení počítače do základního parametru - tj. tmavošedá obrazovka se světlezeleným okrajem /modus C128/, ev. modrá obrazovka /u módu C64/ stejně jako velká písmena/grafika - modus klávesnice.

Klávesa CONTROL

nacházející se nad klávesou RUN/STOP slouží k zadávání speciálních znaků, stejně jako k zapínání zvláštních akustických signálů funkcí obrazovky, nebo systémových funkcí. Používá se při současném stisku další klávesy. Nejvíce používanou funkcí klávesy CONTROL je nastavení barvy. K tomuto účelu se stiskne tlačka CONTROL s jednou z číselných kláves v horní řadě pole. Na čelní straně těchto kláves jsou označeny barvy /znak barvy v módu textu/.

Klávesa RUN/STOP

při stisku této klávesy bez klávesy SHIFT se okamžitě přeruší počítačem prováděné dané instrukce v Basicu, a počítač přejde do přímého módu. Při současném stisku s klávesou SHIFT se načte do paměti a dále spustí program z kazety /modus C64/ nebo z diskety /modus C128/

Klávesa C - COMMODORE

tato zvláštní klávesa má 3 funkce

1. Společně s klávesou SHIFT lze přepínat na již zmíněné módy klávesnice
2. Společně s číselnými klávesami v horní řadě klávesnice lze vybavit jednu z osmi barev z příslušných dvou souborů
3. Při současném stisku klávesy DIN mohou být zadány grafické znaky /viz dodatek D/

Klávesa CLR/HOME

s tlačením klávesy HOME se přesune kurzor do levého horního rohu obrazovky. Pokud se současně stiskne i klávesa SHIFT, vymaže se navíc i obrazovka.

Klávesa INST/DEL

bez současného stisku klávesy SHIFT, se vymažou znaky nalevo od kurzoru. Všechny znaky napravo od vymazaných znaků se posunou o jedno místo nalevo.

Při současném stisku s klávesou SHIFT se na místě kurzoru vloží mezera /vložený modus/. Všechny znaky napravo od této ~~klávesy~~ mezery, pokud jsou k dispozici, se posunou o jedno místo doprava. U tohoto vloženého módu se mimoto zruší přímý účinek ovládací klávesy kurzoru, a na místo toho se znázorní její ovládací kód ve formě inverzního znaku na obrazovce.

Klávesa CRSR /svislé šipky/

s klávesou SHIFT se posune kurzor v témže sloupci do řádku ležícího nad ním. Na nejvyšším řádku obrazovky nemá tato tlačka žádnou funkci. Bez klávesy SHIFT se kurzor posune v témže sloupci o řádku níž. V nejnižším řádku obrazovky způsobí tato klávesa rolování obrazovky o jeden řádek vzhůru.

Klávesa CRRR /vodorovné šipky/

bez klávesy SHIFT se kurzor posune o jedno místo doprava v tomtéž řádku. Pokud kurzor stojí na konci řádku, posune se na začátek řádku následujícího. V pravém dolním rohu obrazovky provede tato klávesa narolování obrazovky o jeden řádek.
Společně s klávesou SHIFT se kurzor posune v tomtéž řádku o jedno místo doleva. Pokud stojí kurzor na začátku řádku, posune se na konec předešlého řádku. Tato klávesa nemá žádnou funkci v levém horním rohu obrazovky.

Funkční klávesy

osm funkčních kláves F1 až F8, které jsou uspořádány v bloku nad oddělenou číselnou klávesnicí, jsou v módu C128 uspořádány v určité znakové posloupnosti /často používané příkazy a instrukce v Basicu/, které se při stisku příslušné klávesy objeví na obrazovce. Obsazení funkcí těchto kláves lze uživatelem kdykoliv zaměnit k vlastnímu účelu/viz kap. 4.1. instrukce KEY v kap. 4.6. a dodatek G/

3.2. ZADÁVÁNÍ PROGRAMOVÝCH ŘÁDKŮ

Interpret Basicu akceptuje každý textový řádek, který začíná číslem jako programový řádek, přičemž číslo představuje číslo řádku.

Programový řádek v Basicu tedy vždy začíná číslem, je ukončen klávesou RETURN a smí obsahovat max. 160 znaků včetně znakového kódu, přiděleného klávesou "RETURN". Pokud se zadá více znaků, jsou stiskem klávesy RETURN odříznuty, i když ještě zůstávají na obrazovce. Všechna klíčová slova Basicu a proměnné se musí zadávat při aktivním módu klávesnice "B" jen pomocí malých písmen.

Je-li při zadávání programového řádku použit otazník /?/ jako zkratka pro instrukci PRINT /viz kap. 3/, potom jej interpret při příkazu LIST přemění v klíčové slovo PRINT, přičemž je řádka delší než 80 znaků.

Při zadání programového řádku s číslem řádku, které je menší, než velikost až dosud zadávaných čísel řádků, vsune jej interpret na správné místo prozatím zadaných řádků. Řádky se tedy nemusí zadávat v číselné posloupnosti.

Pokud zadáte programový řádek číslem řádku, který již v daném programu existuje, přepíše se starý řádek novým.

Připojení programového řádku k programu, který rozsahem přesahuje kapacitu hlavní operační paměti, provede interpret chybové hlášení:

OUT OF MEMORY ERROR /hlavní paměť přeplněna/

přičemž řádek již nemůže být připojen.

Při zadání programového řádku se neprovádí syntaktická kontrola. Teprve po provedení řádku interpretem, se provede syntaktická kontrola s případným hlášením chyb /viz kap. 8.1./.

Při zadání lícého počtu uvozovek "/" se zapne "uvozovací mód", při kterém, jako při vkládacím módu /viz klávesa INST kap. 3.1./ se zobrazí kód již stlačené, ev. poté stlačené klávesy řídící klávesy kurzoru jako inverzního znaku na obrazovce. Tímto způsobem mohou být programovány ovládací funkce kurzoru za pomocí ovládacích řetězců ve spojení s instrukcí PRINT.

3.3. NAHRAZENÍ NEBO ZMĚNA PROGRAMOVÝCH ŘÁDKU

Jak již bylo uvedeno v kap. 3.2., jsou existující programové řádky nahrazovány, přičemž se zadá nový programový řádek s tímtéž číslem řádku. Přitom může být nový programový řádek delší nebo kratší, než řádek původní.

Programový řádek se vymaže tím, že se zadá jen číslo řádku a stiskne se klávesa RETURN.

Celý Basic-program v operační paměti lze vymazat příkazem "NEW". /viz kap. 5.2./.

3.3.1. ZMĚNA ŘÁDKU EDITOREM OBRAZOVKY

Každý viditelný řádek na obrazovce může být změněn pomocí editoru obrazovky, který je aktivní vždy, když se Basic nachází v úrovni příkazů /přímý modus/, nebo když program čeká na zadání z klávesnice.

Kurzor lze přesunout pomocí ovládacích kláves kurzoru na žádanou pozici v opravovaném řádku, poté se provede změna - také použitím klávesy INST/DEL.

Po stlačení klávesy RETURN je změna ukončena a celý opravený řádek uložen do programu.

Má-li být změněn řádek, který není viditelný na obrazovce, zobrazíme jej nejprve příkazem "LIST" /viz. kap. 5.2./ na obrazovce, a poté změníme tak, jak již bylo uvedeno.

Klávesa RETURN, která ukončuje změnu řádku, může být stlačena v jakékoli libovolné pozici kurzoru uvnitř opravovaného řádku. V každém případě přebírá interpret všechny programové řádky do programu.

3.3.2. ZMĚNA ŘÁDKŮ SE SYNTAKTICKÝMI CHYBAMI

Pokud program neobsahuje žádné programované zpracování chyb, a během zpracování programu zjistí interpret syntaktickou chybu, hlásí ji i se zadaným číslem řádku. Chybný řádek se musí zobrazit stiskutím klávesy LIST nebo klávesy HELP, a poté výše uvedeným způsobem být změněn.

3.3.2. Změna řádků se syntaktickými chybami - pokračování

Stisk klávesy RETURN zařadí opravený řádek do programu, který potom může být znova odstartován.

Při takovéto změně programu musíme respektovat následující opatření:

- všechny proměnné a pole se vymažou /nulová hodnota nebo délka/.
- program nemůže pokračovat příkazem "CONT" /viz kap. 5.2./

Mají-li být před změnou zkontrolovány obsahy určitých proměnných nebo polí, musí být interpret přesunut do úrovně příkazů /přímý modus/ stiskem klávesy RUN/STOP.

Potom lze provést kontrolu indikací příslušných proměnných nebo polí pomocí klávesy PRINT, změnit řádek a znova odstartovat program.

3.4. UKLÁDÁNÍ BASICOVÝCH PROGRAMŮ DO PAMĚTI

Nově vytvořený nebo opravený program musí být pro další zpracování uložen na kazetě nebo na disketě. Toto opatření se provádí příkazem SAVE nebo DSAVE. Oba příkazy jsou v kap. 5.2. a 4.6. podrobně popsány, takže je nyní nebudeme rozvádět.

3.5. VKLÁDÁNÍ PROGRAMU Z VNĚJŠÍCH MEDIÍ

Abychom mohli spustit Basic-program uložený na kazetě nebo na disketě, musí být tyto programy nejdříve vloženy do hlavní / operační / paměti počítače. Toto se děje v přímém módu počítače příkazy LOAD a DLOAD nebo RUN. Po vložení programu za pomoci LOAD nebo DLOAD se odstartuje program příkazem RUN.

Příkazy LOAD a DLOAD i RUN jsou podrobně popsány v kapitolách 5.2. a 4.6.

4. M O D U S C 128

- 4.1. Rozšířené funkce klávesnice v módu C 128
- 4.2. Rozšířené funkce obrazovky v módu C 128
- 4.3. BASIC 7.0. v módu C 128
- 4.4. Struktura Basicu
- 4.5. Struktura a syntaxe popisů /ekvivalentů/
- 4.6. Příkazy, instrukce, funkce a proměnné
- 4.7. Barvy, Sprites a grafické efekty
- 4.8. Zvuk a hudba v módu C 128

4. M O D U S C 128

C 128 se v tomto módu ohlásí po zapnutí počítače nebo po stisku klávesy RESET.

V tomto módu je uživateli k dispozici hlavní operační paměť se 128K Byty, volitelný 40ti nebo 80ti místný rozsah znázornění na obrazovce s 25 řádky, interpret Basicu s oproti C64 rozšířeným obsahem příkazů /Basic 7.0./, dle volby německou/DIN/ nebo americkou /ASCII/ klávesnici se speciálními obslužnými nebo programovacími funkčními klávesami jako disketing 1570/71 se značně rychlejším disketovým vybavením.

4.1. ROZŠÍŘENÉ FUNKCE KLÁVESNICE V MÓDU C 128

V kap. 3 je popsána a zobrazena klávesnice, která obsahuje řadu kláves, které jsou funkční jen v módu C 128.

Desítková klávesnice

Vpravo od klávesnice psacího stroje se nachází blok deseti číselných kláves, s desetinnou tečkou , s klávesami + a - a vstupní klávesou ENTER.

Tyto klávesy mají stejnou funkci jako odpovídající klávesy klávesnice psacího stroje. Klávesa ENTER přitom odpovídá klávese RETURN. Tento blok kláves ulohčuje vstup většího množství čísel a za pomoci vhodného programu umožnuje simulaci kapesního počítače.

Funkční klávesy

Nad desítkovou klávesnicí se nacházejí 4 funkční klávesy k celkem osmi různým programovatelným funkcím. Funkce f1,f3,f5 a f7 se provádějí stiskem odpovídající funkce a funkce f2,f4,f6 a f8 za dodatečného stisku klávesy SHIFT.

Funkce existují v simulovaných vstupech klávesnice a provádějí celé příkazy a příkazové řetězce.

Po zapnutí počítače jsou 8 funkčním klávesám přiřazeny následující příkazy:

f1 GRAPHIC	f2 DLOAD"
f3 DIRECTORY< RETURN >	f4 SCNCLR < RETURN >
f5 DSAVE"	f6 RUN < RETURN >
f7 LIST < RETURN >	f8 MONITOR < RETURN >

Stiskne-li se klávesa f1, objeví se na momentální pozici kurzoru příkazové slovo GRAPHIC, nyní se již pouze vloží potřebné parametry /viz kap. 4.6./

Při stisku klávesy f2 /f1 současně s SHIFT/, objeví se na místě kurzoru příkazové slovo DLOAD, poté je již jen potřeba vložit zvolené názvy dat s konečným uvedením znaku uvozovky.

Stiskne-li se klávesa f3, provede se příkaz DIRECTORY, který udá seznam programu diskety na připojeném disketové jednotce s adresou přístroje 8.

Po stisku f4 /f3 současně s SHIFT/ se příkazem SCNCLR vymaže obrazovka.

Stiskem klávesy f5 se na pozici kurzoru /aktuální pozici/ udá příkazové slovo DSAVE", nyní je třeba zadat pouze zvolené jméno dat s ukončujícím uvozovkou "/".

Stiskne-li se klávesa f6 /f5 současně s SHIFT/, provede se příkaz RUN, a odstartuje se Basic-program, nacházející se právě v hlavní paměti.

Stiskem f7 se provede příkaz LIST a Basic-program, který se nachází v hlavní paměti, se listuje na obrazovce.

Stiskem klávesy f8 /f7 současně s SHIFT/, vyvolá se příkazem MONITOR monitor strojového jazyka.

Předem nastavené obsazení funkčních kláves lze kdykoliv změnit nebo pouze indikovat na každé libovolné klávese příkazem KEY /viz kap. 4.6./

Řídící klávesy kurzoru:

Čtyři klávesy se šipkovými symboly, které se nacházejí vedle funkčních kláves, slouží k jednoduchému řízení pohybu kurzoru na obrazovce. Jinak než u C 64 se zde nachází pro každý směr kurzoru zvláštní klávesa. Směr je určován šipkou na té které klávese.

Klávesa HOME

Stiskne-li se dvakrát po sobě klávesa HOME, zruší se definovaný obraz na obrazovce /viz WINDOW-instrukce v kap. 4.6. jako i sekvence kláves ESC T a ESC B v tomto oddíle/, a umístí se znět na celkovou obrazovku.

Klávesa NO SCROLL

Začne-li se během indikace dat na obrazovce tato rolovat směrem nahoru, lze toto rolování zrušit stiskem klávesy NO-SCROLL, která se nachází v pravé části bloku kláves nalevo od řídících kláves kurzoru. Funkce této klávesy se podobá funkci klávesy SHIFT LOCK. Jednoduchý stisk klávesy zablokuje funkci /rolování obrazovky zastaveno/, opětné stisknutí klávesy funkci odblokuje /rolování obrazovky povoleno/.

Klávesa 40/80 DISPLAY

Pomocí této klávesy lze měnit mezi oběma rozsahy indikace módu C 128./stisknuto: 80ti znaková indikace/-předpokládá se však i zapnutí 80ti znakového monitoru, např. Commodore 1902/. Tato klávesa se při stisku automaticky zablokuje a při novém stisknutí se znova uvolní. Přepínání se musí provádět v ž d y p ř e d zapnutím nebo vrácením počítače /klávesa RESET/

Klávesa LINE FEED

Stisk této klávesy způsobí posunutí kurzoru ve stejném sloupci do následujícího řádku.

Klávesa HELP

Zjistí-li interpret během zpracovávání programového řádku chybu, ohlásí toto indikací chyby /viz kap. 8.1./. Po stisku klávesy HELP se indikuje chybný řádek, který se v místě chyby znázorní podle módu obrazovky buďto inverzně/40ti znaková obr/ nebo podtrženě/80ti znaková obrazovka/. Tímto způsobem je korektura programu velmi ulehčena.

Klávesa ASCII/DIN

Toto je samostatně blokovatelná klávesa, kterou lze přepínat mezi německou-DIN a americkou ASCII klávesnicí.

Při stisku a tím i zablokování klávesy je aktivní německá klávesnice DIN..

Klávesa ALT

Touto klávesou lze volit alternativní, samodefinovatelnou tabulku klávesnice, ve které může být téměř všem klávesám klávesnice psacího stroje přiřazen jiný kód. Je to důležité v případě, že texty mají být psány v jiném jazyce. Definice takovýchto tabulek se děje ukládáním určitých hodnot do paměti počítače na zvláštních místech v paměti. /viz dodatek J/

Klávesa TAB

Stisk klávesy TAB současně s klávesou SHIFT postaví na aktuální pozici kurzoru stop-tabulátor, který stiskem klávesy TAB znova naskočí. Přednastaveny jsou tabulátory-stop všech 8 sloupců /mezer/.

Klávesa ESC

umožňuje společně s alfabetickými klávesami A až Z celou řadu zvláštních funkcí klávesnice a obrazovky, které jsou dále podrobněji popsány. K provedení zvolené funkce je třeba nejprve stisknout klávesu ESC a poté ji opět pustit. Nakonec se stiskne zvolená alfabetická klávesa.

Klávesa Funkce

- A Automatický připojený modus se zapne. Znak lze vkládat tak dlouho, dokud se nestiskne ESC-C.
- B Aktuální pozice kurzoru definuje pravý dolní roh obrazu na obrazovce.
- C Vložený /připojený/ modus se vypne.
- D Aktuální řádek na obrazovce se vymaže a obrazovka pod tímto řádkem se naroluje o jeden řádek nahoru.
- E Uvede modus kurzoru z blikací na konstantní indikaci.
- F Uvede modus kurzoru z konstantní indikace na blikací.
- G Umožní akustické znaky skrze CTRL-G
- H Zruší akustické zvuky skrze CTRL-G.
- I Uvede nad aktuální řádek na obrazovce řádek nový.
- J Umístí cursor na začátek aktuálního řádku.
- K Umístí cursor za poslední znak na aktuálním řádku.
- L Umožňuje rolování obrazovky.
- M Ruší rolování obrazovky.
- N Přepne obrazovku z inverzního zobrazení na normální / pouze 80ti znaková obrazovka/
- O Způsobí vypnutí všech módů /blikání, podržování, revers/
- P Aktuální řádek se vymaže od začátku řádku až do pozice kurzoru.
- Q Aktuální řádek se vymaže od pozice kurzoru až do konce řádku.
- R Přepíná obrazovku z normálního zobrazení na zobrazení inverzní /pouze 80ti znaková obrazovka/.
- S Zapne zobrazení bloku pro cursor.
- T Aktuální pozice kurzoru definuje levý horní roh obrazu.
- U Zapíná čárové zobrazení kurzoru /pouze 80ti znaková obr./

Klávesa Funkce

V	Roluje obrazovku o jeden řádek směrem nahoru.
W	Roluje obrazovku o jeden řádek směrem dolů.
X	Přepíná ze 40ti znakového na 80ti znakový monitor a naopak.
Y	Zapíná předem nastavené stopky tabulátoru /všech 8 sloupců
Z	Maže všechny nastavené stopky tabulátoru
(a)	Maže obrazovku od aktuální pozice kurzoru.

Klávesa CTRL

Dodatečně k funkcím, které mohou být v módu C 64 vymazány speciálními sequencemi CTRL /např.volba barvy/, umožňuje modus C128 řadu dalších funkcí, které lze při stisku jedné z následujících kláves ve spojení se stiskem klávesy CTRL vyrušit.

Klávesa	CHR\$ kód	Funkce
B	2	Zapíná modus podtrhování pro všechny zobrazené znaky/80ti znaková obrazovka/
G	7	Ruší akustický zvonkový znak.
I	9	Vypíná skok tabulátoru až po další TAB-stopku
J	10	Provede posun řádku
K	11	Zakazuje přepínání mezi velkými a malými písmeny a mezi velkými a grafikou.
L	12	Umožní přepínání velká/malá a velká/grafika
O	15	Zapíná blikací zobrazení pro všechny zobrazené znaky /pouze 80 znak. obrazovka/
X	24	Umísťuje nebo maže stop-tabulátor.
[27	kód ESC

POZOR

Funkce ESC a CTRL mohou být provedeny v programu také pomocí instrukce PRINT.

Např: PRINT CHR\$(27)+"M" zakazuje rolování obrazovky
PRINT CHR\$(7) vypíná akustický zvonkový znak

Při aktivní klávesnici DIN platí pro používání kláves společně s klávesou CTRL obsazení klávesnice ASCII.

Zvláštnosti klávesnice DIN/ASCII

Uživatelé, kteří chtějí nastavit čistě americkou klávesnici, mohou příkazem

POKE 2757,129 /pouze v 80-znakovém módu/

vypnout provozním systémem jištění přepínání klávesnice. Toto opatření lze pojmut jen při aktivní klávesnici ASCII /volná klávesa ASCII/DIN/ a vrátit zpět se může jen po stisknutí klávesy RESET.

Další zvláštnost klávesnice DIN určuje klávesa s oběma akcenty /přízvuky/ v nejvyšší řadě klávesnice pracovního stroje. Po stisku této klávesy se ukáže akcent, kurzor však zůstane stát na této pozici. Pokud se dále stiskne alfabetická klávesa, zkouší klávesnicový program ~~na~~ provozního systému, je-li tato znaková kombinace povolena. Není-li povolena, objeví se pouze písmeno, jinak akcentované písmeno, a kurzor se posune o jedno místo do prava.

Abyste porozuměli každé aktivní klávesnici, jsou na některých klávesách, které jsou u obou volitelných klávesnic rozdílně obsazeny, poznačena všechna obsazení.

4.2. ROZŠÍŘENÉ FUNKCE OBRAZOVKY V MÓDU C 128

V módu C 128 může uživatel volit mezi 40ti nebo 80ti znakovou indikací. Oba indikační módy lze použít v jednom programu.

40ti znaková indikace je také ta, která platí pro modus C 64. Mimo to slouží ke znázornění všech rozmanitých grafických možností, které modus C 128 nabízí /viz kap. 4.7./.

80ti znaková indikace umožňuje provedení profesionálních programů jako je zpracování textu, administrativa databanky apod.

V tomto znázornění je grafika možná pouze ve spojení s příkazy PEEK a POKE nebo s programem strojového jazyka.

Použití grafických zvláštních znaků z Commodore - znakové věty je možné stejně jako barvy pozadí nebo popředí.

40/80 DISPLAY - klávesa

Tento klávesou lze přepínat mezi 40ti a 80ti znakovou indikací. Klávesu je možné v kterémkoliv momentě stisknout. Přepínání však probíhá jen tehdy, když:

- je počítač vypnut a opět zapnut
- se stiskne klávesa RESET vedle síťového vypínače
- se stiskne klávesa RUN/STOP s klávesou RESTORE

Tato klávesa se, stejně jako klávesa SHIFT, stisknem sama zablokuje, odblokovat ji lze opětným stisknutím.

Ve stlačené poloze je zapnuta 80ti znaková indikace, naopak při od blokování se zapíná indikace 40ti znaková.

VIDEO-PŘÍPOJKY A MONITORY

Jak jsme se již zmínili v úvodu, má počítač C 128 na zadní straně 2 různé přípojky videa /VIDEO a RGBI/. Na přípojku VIDEO se zapíná 40ti znakový monitor /např. Commodore 1701/. Na přípojku RGBI se připojuje 80ti znakový monitor /monitor RGBI/

Obě přípojky najednou použijeme, pokud připojíme dvojitý monitor jako je Commodore 1901. Tento monitor pracuje jak s 40ti znakovým, tak i s 80ti znakovým módem a umožňuje programované přepínání mezi oběma módy.

Při zapojení dvou různých monitorů lze současně znázorňovat jak grafická, tak i textová zadání.

PŘEPÍNÁNÍ OBRAZOVKY

Jednou z největších předností módus C 128, je možnost práce s třemi na sobě nezávislými obrazovkami /80ti znaková obrazovka a rozdělená obrazovka na grafické a textové zpracování/. Výstup na obrazovku je možný na jedné obrazovce bez změny údajů indikovaných na obou dalších obrazovkách, také lze pře-

pínat mezi všemi třemi obrazovkami bez jakékoliv ztrát_v dat. Mimo to lze přepínat mezi 40 a 80ti znakovou indikací bez předchozího výmazu údajů z obrazovky. K přepnutí dojde následným stiskem kláves ESC a X nebo PRINT CHR\$(27) +"X".

K programovanému přepínání obrazovky slouží instrukce GRAPHIC /viz kap 4.6/. Touto instrukcí lze indikovat v programech 80ti znakového módu grafické znázornění na 40ti znakové obrazovce.

Další informace viz kapitola 4.6. a 4.7.

4.3. BASIC 7.0 v MÓDU C 128

Na základě Basicu 2.0 /Commodore Standard/, který je podrobně popsán v kap. 5.2., nabízí počítač uživateli Basic 7.0, rozšířený o mnoho dalších příkazů, instrukcí, funkcí a systémových proměnných k užití mnohostranných možností.

Tento oddíl není návodom k naučení se všeobecných programových technik, nebo programového jazyka Basic, ale příručka pro uživatele k obeznámení se všeobecnými koncepty programování. Na následujících stranách najdete důležitá fakta k sestavení efektivního programu Basic. Začátečníkům doporučujeme prostudování lekcí vydaných kursů Basicu. V módu C +28 je možné použít celý rozsah příkazů jak Basicu 2.0 /viz kap. 5/ tak i Basicu 7.0, jak je dále uvedeno v kap. 5.2. Dříve, než popíšeme jednotlivé instrukce, věnujme trochu pozornosti struktuře programového jazyka Basic a syntaxi.

4.4. STRUKTURA BASICU

Jazykový rozsah Basicu-Commodore se člení na:
příkazy, instrukce, funkce a proměnné.

PŘÍKAZY se používají k zpracovávání programů, k listování, k změnám v programu, k výmazu, ukládání do paměti nebo ke nahrávání. Proto se příkazy zadávají většinou v úrovni příkazů /přímý modus/. Mohou však být i částí programu

INSTRUKCE řídí průběh programu. Jsou vlastním obsahem programů, mohou však být zadávány také v přímém módu. Instrukce se člení do tří skupin:

- instrukce vstupu/výstupu pro provoz dat mezi počítačem na jedné straně, stejně jako klávesnicí, obrazovkou, disketovou jednotkou, tiskárnou nebo spojovacím místem dálkového přenosu dat na straně druhé.
- Deklarační instrukce k definování údajů, vlastních programových funkcí, segmentů paměti nebo okolí programu.
- Řídící instrukce k určování průběhu programu jako ladění, přeskoky, vyvolávání podprogramu a přerušení programu.

FUNKCE produkují jako výsledek určité hodnoty k dalšímu zpracování programu a dělí se na:

- numerické funkce, které jako výsledek předávají celočíselné hodnoty, nebo hodnoty s pohyblivou řádovou čárkou
- řetězcové funkce, které jako výsledek předávají řetězce.

PROMĚNNÉ jazyka Basic nemohou být programátorem zaměňovány s volně definovanými proměnnými programu. Tyto systémové proměnné mají pevné, Basicem předurčené názvy a funkce. Jim se při vyvolání interpretem přiřazuje hodnota, která popisuje určité stavy funkcí počítače nebo počítačového systému v určitém momentu. Existují numerické a řetězcové systémové proměnné.

4.5. STRUKTURA A SYNTAXE POPISOVÁNÍ

V kap. 4.6. jsou detailně včetně příkladů popsány všechny příkazy, instrukce, funkce a proměnné, které interpret Basicu ovládá. Jednotlivé popisy jsou alfabeticky seřazeny podle

názvů a členěny podle následujícího schematu:

Formát: Zde je uveden exaktní způsob psaní příkazu, instrukce funkce nebo proměnné. Přitom platí následující pravidla syntaxe:

1. Všechna slova, psaná velkými písmeny, jsou klíčovými slovy Basicu, jejichž způsob psaní, jak již bylo uvedeno, je závazný.
2. Všechna zadání, psaná ležatým/kurzívním/ písmem-velkým či malým, znázorňují parametry, které musí být uživatelem zadány.
3. Zadání v hranatých závorkách jsou volitelná.
4. Řada teček /.../ za zadáním, znamená, že toto zadání může být, pokud je potřeba, dále specifikován.
5. Všechny uvedené zvláštní znaky, kromě hranaté závorky, tedy , ; - = () ' " , musí být zadány tam, kde stojí.
6. Při zadání parametrů mají použité zkratky následující význam:

x,y,z	libovolné numerické výrazy
i,j,k,m,n	libovolné celočíselné výrazy
x\$, y\$	libovolné řetězcové výrazy
v,v\$	numerické nebo řetězcové proměnné

Účel: Zde je detailně popsán účinek právě platného příkazu nebo instrukce, funkce nebo proměnné a jsou objasněny parametry a jejich definiční rozsahy.

Poznámky: Zde jsou objasněny zvláštní vlastnosti jednotlivých příkazů, instrukcí, funkcí a proměnných.

Příklad: Příklady uváděné u každého popisu znázorňují, jak lze použít jednotlivé příkazy, instrukce, funkce a proměnné a poukazují také na jejich zvláštnosti.

4.6. POPISOVÁNÍ JEDNOTLIVÝCH PŘÍKAZŮ, FUNKCÍ, INSTR. A PROMĚNNÝC

K usnadnění vyhledávání určitých příkazů, instrukcí, funkcí nebo proměnných basicu nejsou následující popisy členěny podle struktury jazyka Basic, uvedené v kap. 4.4., nýbrž všechny elementy jazyka jsou shrnutы a alfabeticky uspořádány.

INSTRUCE APPEND

Formát: APPEND # log Filnr, Dateiname [,D disket.jednotka]
[,U přístrojová adresa]
/Filnr=číslo souboru, Dateiname=jméno data/

Účel: Otevří existující sekvenční data typu SEQ nebo USR jako výstupní data a umíšťuje psací ukazatel na konec data, aby mohla být další data připojena/viz kap.6.6./

Log.číslo souboru -log Filnr

Celočíselná hodnota mezi 1 a 255, která se přiřadí otevřeným údajům jako poznávací číslo. U hodnot nižších, než 128 je u dat vydaných instrukcí #PRINT# vydán chod zpětného vozíku. U hodnot přes 127 se dodatečně vydá kód posuvu řádků.

Dateiname

"filename"

Řetězcová konstanta v uvozovkách (") nebo řetězcová proměnná o max. délce 16 znaků pro jména otevřených dat.

Disketová jednotka

Celočíselná hodnota # 0 nebo 1. Přenastavena je 0.

Přístrojová adresa

Celočíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojeného floppy-disku. Přenastavena je hodnota 8

Poznámky:

Použijí-li se pro jméno data, disketovou jednotku a přístrojovou adresu proměnné, pak musí být tyto umístěny v závorkách.

Příklad: APPEND #2, "TEXT"

Na floppy disku s přístr. adr. 8 se otevřou data TEXT pod log. číslem souboru 2 a psací ukazatel se umístí nakonec údaje tak, aby mohla být připojena další data.

Příklad: APPEND #3,(DAZ),D(VAL(ALG)) ON U9

Jméno data a disket. jednotka se předávají jako proměnné.

PŘÍKAZ A U T O

Formát: AUTO [Schrittw]

Schrittw=délka kroku.

Účel: Během zadávání programových řádků Basic, se na začátku řádku ukáže nové číslo řádku jen tehdy, je-li na konci předešlého řádku stisknuta klávesa RETURN.

Po zadání příkazu AUTO musí být první programový řádek zadan s číslem řádku. Dalším řádkem probíhá číslování řádků automaticky.

Poznámky:

Schrittw - délka kroku

udává délku kroku mezi čísly řádků. Použije-li se příkaz AUTO bez délky kroku, přeruší se automatické číslování řádků.

Příklady: AUTO 10

Čísla řádků se vypisují automaticky s délkou kroku 10.

AUTO

Automatické číslování řádků se během zadávání programu odpojí.

PŘÍKAZ B A C K U P

Formát: BACKUP DQuelle TO DZiel [,UGerät]

Účel: Duplikuje disketu na dvojitém floppy-disku.

Quelle - zdroj

Hodnota 0 nebo 1 pro disketovou jednotku, ve které je vložená duplikovaná disketa.

Ziel - cíl

Hodnota 1 nebo 0 pro disketovou jednotku, ve které je uložena disketa, na kterou se bude kopírovat.

Gerät - přístroj

Celočíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojené dvojité disketové jednotky. Přednastavena je 8.

Příkaz BACKUP - pokračování

Poznámky: Provedení příkazu BACKUP jemožné jen u floppy-disku se dvěma disketovými jednotkami.

Duplikát je přesným obrazem původní diskety.

Příkaz BACKUP obsahuje příkaz HEADER (viz dále), kterým je disketa formatována před začátkem duplikovačního procesu.

Před začátkem duplikovačního procesu se dotazem

ARE YOU SURE?

vyžaduje od uživatele potvrzení (odpověď y jako yes) že ~~aby~~ příkazem BACKUP se případně dříve existující data na cílové disketě zničí.

Příklad 1: BACKUP D0 TO D1

Duplikace diskety v disketové jednotce 0 na disketu v disketové jednotce 1.

Příklad 2: BACKUP D1 TO D0 ON U9

Kopírování z disketové jednotky 1 na disketovou jednotku 0 (nula) v přístroji s adresou 9.

PŘÍKAZ BANK

Formát: BANK n

Účel: Definuje paměťovou banku s 64kByty pro následující instrukce PEEK-, POKE-, SYS-, nebo WAIT. n musí mít hodnotu mezi 0 a 15.

Poznámky:

Paměť C 128 je rozdělena do 15 paměťových desek /bank/, vždy s 64kByty. Přitom jsou však k dispozici pouze banky č. 0, 1, a 15 (viz hierarchie paměti v dodatku B).

Příklad: 10 REM HODNOTA 20 NA ADRESE 1024 V BANCE

15 REM 1 ULOZIT

20 BANK 1:POKE 1024, 20

INSTRUKCE BEGIN a BEND

Formát: IF ... THEN BEGIN

•
• INSTRUKCE
•
BEND

Účel: BEGIN a BEND uzavírají libovolný dlouhý blok instrukcí Basicu, který může mít i více řádků.

Poznámky:

BEGIN se používá vždy ve spojení s instrukcí IF.

Předností oproti instrukci IF THEN ELSE je ta, že instrukce následující po THEN nemusí být omezena na jeden programový řádek.

Všechna ostatní pravidla pro instrukci IF...THEN...ELSE však zůstávají v platnosti./viz příklad/

Příklad 1: 100 IF X=1 THEN BEGIN:A=5
 110 B=6:C=7
 120 PRINT A+B+C:BEND:PRINT"AHA!"
 130...

AHA! se vytiskne pouze tehdy, když X=1 je logická pravda. V opačném případě by program pokračoval řádkem 130.

Příklad 2: 10 INPUT"CISLO=";Z
 20 IF Z> 0 THEN BEGIN
 30 PRINT"CISLO JE> A"
 40 BEND:ELSE GOTO 60
 50 END
 60 PRINT"CISLO JE< A"
 70 GOTO 10

Zadá-li se kladné číslo, ukončí se program v řádku 50, jinak se program bude opakovat od řádku 10.

PŘÍKAZ B L O A D

Formát: BLOAD Dateiname [Disketová jednotka] [,Upřístroj]
[,ON B Bank] [,P Adr]

Účel: Do hlavní paměti seuloží data s binárně kódovaným obsahem /např. strojový jazyk, binární data/ viz kap. 6.4.

Dateiname -jméno data v uvozovkách "
Znaková řetězcová konstanta, ~~která~~ nebo řetězcová proměnná o délce až 16 znaků, která udává jméno ukládaných binárních dat.

Disketová jednotka

Hodnota 0 nebo 1 pro disketovou jednotku, ze které má být nahráváno. Přednastavena je zde 0.

Přístroj

Celočíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojeného floppy-disku.

Banka

Celočíselná hodnota mezi 0 a 15 pro paměťovou banku, do které se má nahrávat. Přednastavena je zde 0.

Adr Celočíselná hodnota mezi 0 a 65535, která udává adresu ve zvolené bance, ze které se má nahrávat. Přednastavena je zde adresa, která se skládá z prvních dvou bytů binárního data.

Poznámky: Použijí-li se pro jméno data, disketovou jednotku přístroj, banku a Adr proměnné, pak musí být tyto uváděny v závorkách.

Příklad: BLOAD "BIN",ON B1,P1024
Údaj BIN se nahraje z floppy-disku s adresou přístroje 8 do banky 1 v hlavní paměti od adresy 1024.

Příklad: BLOAD(DG), D(LW),U(GA),ON B(B),P(AD)

V tomto případě byly pro všechny parametry použity proměnné.

PŘÍKAZ B O O T

Formát: BOOT [Dateiname [,D disketová jednotka] , [Upřístroj]

Účel: Ukládá binární soubor dat způsobilý přenosu z diskety do hlavní paměti a startuje ho od jeho startovací adresy

Jméno data /dateiname/

Řetězcová konstanta v uvozovkách "/" nebo řetězcová proměnná o délce až 16 znaků, které udávají jméno nahrávaného binárního data.

Disketová jednotka

Hodnota 0 nebo 1 pro disketovou jednotku, ze které má být nahráváno. Přednastavena je zde 0.

Přístroj

Celočíselná hodnota mezi 4 a 15 pro adresu přístroje připojeného floppy-disku.

Poznámky: Použijí-li se proměnné jako parametr, musí být zadávány v závorkách.

Užije-li se BOOT bez parametru, odebere provozní systém sektorem 0 ve stopě i diskety ve floppy-disku č.8 informaci, co má být uloženo a odstartováno. Tato informace může být na disketu zanesena pomocí přímo zasahujících příkazů. Blížší informace viz příručka Floppy-disk.

Příklad: BOOT "HILFE" Hilfe=pomoc

Z disketové jednotky s přístrojovou adresou 8 se nahraje binární soubor dat, způsobilý přenosu /HILFE/ a jeho startovací adresou se odstartuje.

INSTRUKCE B O X

Formát: BOX [Farbquelle],x1,y1[,,[x2,y2]]
[,,[Winkel]] [,,[Ausmal]]

Účel: Na libovolné pozici obrazovky se zobrazí a podle volby vymaluje pravoúhelník.

Farbquelle - zdroj barvy

Celočíselná hodnota mezi 0 a 3, která definuje zdroj barvy, ve které má být zobrazováno.

To znamená:

- 0 pozadí
- 1 popředí
- 2 doplnková barva 1
- 3 doplnková barva 2

x1,y1,x2,y2

jsou souřadnice levého horního, popř. pravého dolního rohu pravoúhelníku. Souřadnice x je od 0 do 320, souřadnice y od 0 do 199.

Přednastaven je tu bod na obrazovce vlevo nahoře, stejně jako aktuální pozice grafického kurzoru/Pixel-kurzor/.

Winkel - úhel

Úhel ve stupni, o který se má pravoúhelník otočit okolo svého středu ve směru hodinových ručiček.
Přednastavena je hodnota 0.

Ausmal - vymalování

Celočíselná hodnota, která určuje, jestli má být /1/ pravoúhelník vymalován ve zvolené barvě, nebo ne/2/.

Poznámky: Po nakreslení pravoúhelníku je bod x2,y2 aktuální pozici grafického kurzoru.

Příklady: BOX 1,10,10,60
Kreslí obrys pravoúhelníku v aktuální barvě popředí.

 BOX,10,10,60,60,45,1
Kreslí vyplněný kosočtverec v aktuální barvě popředí.

 BOX,30,90,45,1
Kreslí vyplněné mnohoúhelník, pootočený o 45 stupňů.

PŘÍKAZ BSAVE

Formát: BSAVE Dateiname [,DLaufwerk][,U Gerät
[,ON B Bank][P Adr1][TO P Adr2]

Účel: Rozsah hlavní paměti /např. strojový jazyk, binární data/ se zanese na disketu jako binární údaj.

Dateiname - jméno data

Řetězcová konstanta v uvozovkách "/" nebo řetězcová proměnná o délce až 16 znaků, která udává jméno ukládaného binárního data.

Laufwerk - disketová jednotka

Hodnota 0 nebo 1 pro disketovou jednotku, za které se má nahrávat. Přednastavena je zde 0.

Gerät - přístroj

Celočíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojeného floppy-disku.

Bank - banka

Celočíselná hodnota mezi 0 a 15 pro paměťovou banku do které se má nahrávat. Přednastavena je zde 0.

Adr1,Adr2

Dvě celočíselné hodnoty mezi 0 a 65535, které udávají počáteční a konečnou adresu rozsahu paměti ve zvolené bance, který má být uložen.

Poznámky: Použijí-li se pro jméno data, disketovou jednotku, přístroj, banku a/nebo Adr1 a Adr2 proměnné, pak musí být tyto zadány v závorkách.

Příklad 1: BSAVE "BIN",ON B2,P1024 TO P2047

Rozsah hlavní paměti od adresy 1024 do 2047 včetně /1kB/ v bance 2 se uloží jako údaj BIN na disketu ve floppy-disku s přístrojovou adresou 8.

Příklad 2: BSAVE (DS), D(LW), U(GA), ON B(B)
, P(A1) TO P(A2)

V tomto případě se pro všechny parametry použijí proměnné.

FUNKCE B U M P

Formát: v=BUMP(n)

Účel: Poskytuje číslo Sprite /1až 8/, který od posledního dotazu BUMP kolidoval /srazil se/ buďto s jiným Spritem /n=1/ nebo se zobrazenými daty /n=2/.

Poznámky: Použije-li se funkce BUMP, nemusí být přerušení programu pro kolize Spritu aktivováno instrukcí COLLISION /viz dále/.

Při vícenásobných dotazech by měly být napřed Sprite-atributy s pomocí funkce RSPRITE dotázány, aby se mohlo zjistit, který Sprite s kterým kolidoval.

BUMP /n/ se po dotazu nastaví na 0.

Příklad: 100 ON BUMP (2) GOTO 120,130,140,150

Kolidoval-li Sprite č.1 s indikovanými daty, rozvětví se program na řádek 120, při čísle 2 na 130 atd.

PŘÍKAZ C A T A L O G

Příkaz CATALOG je identický s příkazem DIRECTORY /viz dále/

INSTRUKCE C H A R

Formát: CHAR [zdroj barvy],x,y [znak. řetězec]
[inverze]

Účel:

Indikuje znakový řetězec s volitelnou pozicí na obrazovce.

Barevný zdroj - Celočíselná hodnota mezi 0 a 3, která definuje barvu, ve které má být text zobrazen.

To znamená: 0 barva pozadí

1 barva popředí /znaky/

2 doplňková barva 1

3 doplňková barva 2

x,y - pozice sloupců /x/ a řádků /y/, od které má být text indikován. Sloupce 0 až 39 event. 79, řádky 0 až 24.

Znakový řetězec - zobrazovaný text v uvozovkách. ("")

Inverze - parametr pro inverzní zobrazení textu. Má-li hodnotu 0, je text normální, je-li hodnota 1, znázorní se text inverzně.

Poznámky: V textovém módu pracuje instrukce CHAR analogicky k instrukci PRINT /viz kap.3/, tzn. provedou se také ev. řídící kódy kurzoru ve znakovém řetězci. V grafickém módu se takové řídící kódy n e p r o v e d o u . K indikaci znaků ve vícebarevném módu /viz kap.7.7/ v doplňkové barvě 2 je nutné nastavit barevný zdroj na 0 a inverzi na 1. K indikaci znaků ve vícebarevném módu v doplňkové barvě 1 musí být nastaven barevný zdroj na 1 a inverze na 0.

Příklad: 100 CHAR 1,38,12,"DOBRE RANO"

Text DOBRE RANO se zobrazí v aktuální barvě popředí ve středu obrazovky.

INSTRUKCE CIRCLE

Formát: CIRCLE [zdroj barvy],[x,y],xr[,yr]
[,start][,[konec][,[úhel]
[,segmentový úhel]

Účel: Na obrazovce se bude indikovat elipsa, kružnice, nebo libovolný trouhelník se středem x,y.

Zdroj barvy: Celočíselná hodnota mezi 0 a 3, která definuje barvu. Přitom je:

- 0 barva pozadí
- 1 barva popředí - znaků
- 2 doplňková barva 1
- 3 doplňková barva 2

x,y Souřadnice středového bodu elipsy v měřítku /viz instrukce SCALE/. Přednastavena je aktuální pozice grafického kurzoru /Pixel-kurzor/.

xr Poloměr hlavní osy / v měřítku/

yr Poloměr vedlejší osy / v měřítku/. Přednastaven je xr.

Start, konec Numerické výrazy, jejichž hodnoty představují údaje o úhlu a udávají, kde má kresba obrazu začínat a končit. Úhly se počítají oproti matematické konvenci od 0 ve směru hodinových ručiček. Přednastaveno je pro start 0 a pro konec 360.

Úhel Numerický výraz, jehož hodnota udává úhel ve stupní, o který má být obraz ve směru hodinových ručiček pootočen. Přednastavena je zde hodnota 0.

Segmentový úhel Numerický výraz, jehož hodnota udává úhel, o který má být každý segment obrazu vzhledem k předcházejícímu přemístění. Přednastavena je zde hodnota 2.

Poznámky: Po nakreslení obrazu je aktuální pozici grafického kurzoru konečný bod úhlu KONEC /END/.

Instrukce CIRCLE - Poznámky - pokračování

Každé pootočení obrazu probíhá záhadně kolem středového bodu x,y. Rovnají-li se xr a yr, pak se vykreslí kružnice.

Parametr segmentový úhel určuje počet stran /hranatost/ obrazu. Menší hodnoty vytvářejí kulatější kontury.

Příklady: CIRCLE, 160, 100, 65, 10

Vykreslí elipsu v aktuální barvě popředí.

CIRCLE, 160,100,65,50

Kreslí kruh.

CIRCLE, 60,40,20,18,,,45

Vykreslí osmiúhelník

CIRCLE, 260,40,20,,,90

Vykreslí se kosočtverec

CIRCLE,60,140,20,18,,,120

Kreslí trojúhelník.

PŘÍKAZ COLLECT

Formát: COLLECT [Ddisket.jednotka] [ON U přístrojová adresa]

Účel: Maže všechna neuzavřená data a uvolníme takovým datům přidělené paměťové místo na disketu ve specifikované disketové jednotce/viz kap. 6.9.1/.

Disketová jednotka Celočíselná hodnota 0 nebo 1. Přednastavena je zde 0.

Přístrojová adresa Celočíselná hodnota mezi 8 a 15. Přednastavena je zde 8.

Příklady: COLLECT

COLLECT D1 ON U10

INSTRUKCE C O L L I S I O N

Formát: COLLISION Typ, [číslo řádku]

Účel:

Inaktivuje nebo aktivuje přerušení programu pro kolize spritů a definuje v posledním případě programová řádek pro větvění programu při kolizích.

Typ Celočíselná hodnota mezi 1 a 3, která definuje jev, který má vést k přerušení. Platí:

- 1 Kolize Sprite-Sprite
- 2 Kolize Sprite-aktivovaná data
- 3 aktivace světelné tužky

Číslo řádku Platné číslo programového řádku, ke kterému se má při přerušení větvit.

Poznámky:

Dojde-li k udanému jevu, provede interpret ještě okamžitě interpretovanou instrukci do konce a rozvětví pak s GOSUB k udanému číslu řádku. Probíhající podprogram musí být ukončen instrukcí RETURN. Nakonec pokračuje program následující instrukcí, navazující na instrukci přerušenou.

Nepoužije-li se číslo řádku inaktivuje se přerušení programu pro specifikovaný jev.

Současně může být aktivováno více různých jevů přerušení, v určitém momentě však lze zpracovat pouze jedno přerušení. Jakkoliv třídění přerušení tedy není možné.

Je třeba brát ohled na to, že příčina programového přerušení může účinkovat delší dobu, ačkoliv právě existuje jiná situace, nebo se mezitím přerušení inaktivuje.

Ztratí-li se Sprite z viditelného dosahu obrazovky, nelze již přerušení vybavit.

Instukce COLLISION - pokračování

K určení Spritu, který od posledního dotazu způsobil kolizi, lze použít funkci BUMP (viz tam).

Příklad 100 COLLISION2,500

Program se na řádku 500 rozvětví k podprogramu, jakmile dojde ke kolizi mezi Spritem a indikovaným údajem.

INSTRUKCE COLOR

Formát : COLOR barevný zdroj, barevný kód

Účel: Definuje pro každý možný barevný zdroj jednu ze 16 možných barev.

Barevný zdroj K dispozici jsou následující kódy barevných zdrojů:

- 0 pozadí (40-ti znaková obrazovka)
- 1 grafické popředí
- 2 grafický vícebarevný modus 1
- 3 grafický vícebarevný modus 2
- 4 okraj (rámeček)
- 5 barva textu
- 6 pozadí (80-ti znaková indikace)

Barevný kód K dispozici jsou následující barevné kódy:

1	černá	9	světlehnědá
2	bílá	10	hnědá
3	červená	11	růžová
4	zelená	12	tmavě šedá
5	fialová	13	šedá
6	tmavá zelená	14	světle zelená
7	modrá	15	světle modrá
8	žlutá	16	světle šedá

Instrukce COLOR - pokračování

Poznámky: Podání barev může vyjít u různých obrazovek rozdílně.

Na černobílé obrazovce se objeví barvy jako různé odstíny šedé barvy.

Příklad: COLOR 0,2:COLOR1,3:COLOR 4,1

Pozadí bude bílé, popředí(text) červené, rámeček černý.

PŘÍKAZ C O N C A T

Formát: CONCAT D [zdrojová disketová jednotka,] zdroj.údaj TO D [cílová disketová jednotka,] cílový údaj
[ON Upřístrojová adresa]

Účel: Připojuje udaný sekvenční údaj na konec jiného udaného sekvenčního data na disketu v udané disketové jednotce.
(viz kap.. 6.9.3.)

Zdrojová disket. jednotka, cílová disket. jednotka

Celočíselné hodnoty 0 nebo 1. Přednastavena jezdí hodnota 0.

Zdrojová data, cílová data

Řetězcové konstanty v uvozovkách ("") nebo řetězcové proměnné o délce až 16 znaků, které popisují sekvenční data, určená k připojení nebo prodloužení.

Adresa přístroje

Celočíselná hodnota mezi 4 a 15, která udává přístrojovou adresu floppy-disku. Přednastavena je zde hodnota 8.

Příklady: CONCAT"DATEN1"TO"DATEN2"

Sekvenční údaj DATEN1 se připojí na sekv. údaj DATEN2 na disketové jednotce 0, floppy-disku s přístrojovou adresou 8.

Příkaz CONCAT - pokračování, příklady

CONCAT D(QL), (D\$1) TO D(ZL),(D2\$)

V tomto případě se pro zadání disketové jednotky a údajů použily proměnné.

PŘÍKAZ C O P Y

Formát: COPY D[zdroj.disket.jednotka,] zdroj.údaj TO
D[cílová disketová jedn.,] cílový údaj
[ON U adr.přísrtoje]

Účel: Kopíruje zadaný zdrojový údaj pod tím samým nebo novým jménem na disketu v ustanovené disketové jednotce též floppy-diskové jednotky.

Zdrojová/cílová disketová jednotka

Celočíselné hodnoty 0 nebo 1. Přednastavena je zde 0.

Zdrojová/cílová data

Znakové řetězcové konstanty v uvozovkách "/" nebo formené o délce až 16 znaků, které označují kopírující a kopírovaná data.

Přístrojová adresa

Celočíselná hodnota mezi 4 a 15, která udává přístrojovou adresu floppy-disku. Přednastavena je hodnota 8.

Poznámky: Při zadání zdrojového data lze k okopírování většího množství dat při jednom průběhu /průchodu/ použít Joker-znaky * a ? .

Příklady: COPY "DATEN" TO "ZIEL"

Údaj DATEN na disketu v disketové jednotce 0 floppy-disku s přístrojovou adresou 8 se okopíruje /duplikuje/ pod jménem ZIEL.

Příkaz COPY pokračování

Příklady: COPY"QUELLE" TO D1,"*

Údaj QUELLE se okopíruje z disketové jednotky 0 na disketovou jednotku 1 pod stejným jménem.

COPY D1,"ADR*" "TO" "*" ON U9

Na floppy-disku s přístrojovou adresou 9 se okopírují všechna data, obsažená v disketové jednotce 1, jejichž jména začínají na ADR, na disketovou jednotku 0 s uchováním si svých jmen.

INSTRUKCE D C L E A R

Formát: DCLEAR D disketová jednotka [ON Upřístr.adresa]

Účel: Všechny otevřené kanály floppy-disku se pro udanou disketovou jednotku uzavřou / data však nikoliv/.

Disketová jednotka: Celočíselná hodnota 0 nebo 1

Přístrojová adresa: Celočíselná hodnota mezi 4 a 15, která udává přístrojovou adresu floppy-disku. Přednastavena je zde hodnota 8.

Poznámky: Otevře-li se kanál např. instrukcí CMD /viz kap. 5.2./, zůstane tento kanál aktivní do té doby, než bude následovat výstup s PRINT #, nebo než bude zadána instrukce DCLEAR.

POZOR!!

DCLEAR neukončuje žádná otevřená data. Toto musí proběhnout před použitím instrukce DCLEAR, pokud byla data vydána instrukcí CMD.

Příklad: DCLEAR

Všechny otevřené kanály disketové jednotky 0 na přístroji 8 se uzavřou.

INSTRUKCE D C L O S E

Formát: DCLOSE [#log. číslo data] [ON Upřístrojová adr.]

Účel:

Ukončuje specifikovaná nebo všechna otevřená data na připojeném floppy-disku /viz kap. 6.6./

log číslo data Celočíselná hodnota mezi 1 a 255, pod kterým byla otevřena data na floppy-disku instrukcí OPEN /viz kap. 5.2./ nebo DOPEN /viz tato kap./

Přístrojová adresa

Celočíselná hodnota mezi 4 a 15, která udává přístrojovou adresu floppy-disku. Přednastavena je hodnota 8.

Příklady: DCLOSE

Všechna data otevřená pro floppy-disk s přístrojovou adresou 8 se uzavřou.

DCLOSE #5 ON U9

Údaj s logickým číslem data 5 na floppy-disku s přístrojovou adresou 9 se uzavře.

PŘÍKAZ D E L E T E

Formát: DELETE [číslo řádku 1] [-číslo řádku2]

Účel: Udaný rozsah čísel řádků se vymaže v programu, nacházejícím se v hlavní paměti.

Číslo řádku 1 : Číslo prvního z mazaných řádků.

Číslo řádku 2 : Číslo posledního z mazaných řádků.

Poznámky: Příkaz DELETE je použitelný jen v přímém módu.

Příklady: DELETE 10

Řádek 10 se vymaže

DELETE 10-200

Řádky 10-200 budou smazány.

Příkaz DELETE - pokračování

DELETE-200

Všechny řádky programu od začátku až po řádek 200 včetně budou smazány.

DELETE200-

Všechny řádky počínaje řádkem 200 až do konce programu budou smazány.

FUNKCE D E C

Formát: v\$=DEC (hex znakový řetězec)

Účel: Vydává decimální hodnotu hodnoty čísla v hexdecimálním vyjádření.

Hex znakový řetězec

Znakový řetězec v uvozovkách "/", který představuje hexdecimální hodnotu v rozsahu mezi 0 a FFFF.

Příklad: PRINT DEC("FFFF")

65535

READY.

PŘÍKAZ D I R E C T O R Y

Formát: DIRECTORY [D disketová jednotka][,Upřístr.adresa]
[,jméno data]

Účel: Indikuje seznam diskety na specifikované disketové jednotce, nebo jeho části na obrazovce. /viz kap. 6.5./

Disketová jednotka Celočíselná hodnota 0 nebo 1. Přednastavena je zde hodnota 0.

Přístrojová adresa Celočíselná hodnota mezi 4 a 15, která udává přístrojovou adresu floppy-disku. Přednastavena je zde hodnota 8.

Příkaz DIRECTORY - pokračování

Jméno data Způsob volby jména data jako znakové řetězcové konstanty v uvozovkách "/", pro který má být indikován záznam seznamu. K indikaci jmen se stejnými částmi jména lze použít Joker-znaky * a ?. /viz kap. 6.3./

Poznámky: Indikace delšího seznamu vede k rolování obrazovky. Pomocí klávesy NO SCROLL /viz kap. 5.2./ lze indikaci pozastavit, klávesou Commodore ji lze zpomalit.

Příklady: DIRECTORY

Indikují se všechny záznamy seznamu z diskety v disketové jednotce 0 floppy-disku s přístrojovou adresou 8.

DIRECTORY D1,U9"DATEN"

Je-li k dispozici registr seznamu dat "DATEN" na disketě v disketové jednotce 1, floppy-disku s přístrojovou adresou 9, bude indikován.

DIRECTORY "AB *"

Registry seznamu pro všechna data, jejichž jména začínají na AB, se budou indikovat na disketě v disketové jednotce 0, floppy-disku s přístrojovou adresou 8.

DIRECTORY"DATEI???.ARB"

Registry seznamu všech dat, jejichž jména obsahují mezi znaky DATEI a .ARB tři libovolné znaky, se budou indikovat na disketě v disketové jednotce 0 floppy-disku s přístrojovou adresou 8.

POZOR!!

K vydání seznamu diskety např. z disketové jednotky 0 na floppy-disku s přístrojovou adresou 8, na tiskárně s přístrojovou adresou 4, musí být zadán následující sled instrukcí v přímém módu:

```
LOAD"SØ",8
OPEN4,4:CMD4:LIST
PRINT#4:CLOSE4
```

PŘÍKAZ DLOAD

Formát: DLOAD jménn data [,Ddisket. jednotka]
[,U přístrojová adresa]

Účel: Ukládá program Basicu z ustané diskety do hlavní paměti počítače /viz kap.6.6/

Jméno data: Znaková řetězcová konstanta v uvozovkách "" nebo-promenná v závorkách () o délce až 16 znaků, které označují jméno programových dat.

Disketová jednotka : Celocíselná hodnota 0 nebo 1. Přednastavena je zde hodnota 0.

Přístrojová adresa: Celocíselná hodnota mezi 4 a 15, která popisuje přístrojovou adresu připojeného floppy-disku. Přednastavena je zde hodnota 8.

Poznámky: Uvnitř Basic programu slouží příkaz DLOAD k vyvolání a startu jiného Basic-programu, který vyvolávající program překryje, Všechny ~~xprogramy~~ promenné vyvolávacího programu zůstanou zachovány.

Nebude-li specifikovaný program nalezen, indikuje se chybové hlášení:

FILE NOT FOUND ERROR

INSTRUKCE DO

Formáty: DO [UNTIL logický výraz]

- instrukce
- ... [EXIT]
LOOP [UNTIL log. výraz]
- DO [WHILE log. výraz]
- instrukce
- ... [EXIT]
LOOP [WHILE log. výraz]

Instrukce DO - pokračování

Účel: Definuje a řídí průběh programových ~~smyček~~ smyček.

Poznámky: Není-li zadán dodatek UNTIL nebo WHILE ve spojitosti s logickými výrazv, budou prováděny instrukce smvčky nekonečně dlouho.

Vloží-li se do instrukční smvčky dodatek EXIT, bude smyčka na tomto místě opuštěna a program bude pokračovat instrukcí následující za LOOP.

Programové smyčky DO...LOOP lze třídit podle těch samých pravidel jako smyčky FOR...NEXT /viz kap. 5.2./.

Použije-li se dodatek UNTIL, provádí se smyčka tak dlouho, až logický výraz udaný po UNTIL bude logická pravda (-1)

Použije-li se dodatek WHILE, provádí se smyčka tak dlouho, až logický výraz udaný po WHILE bude nepravda (0)..

Příklad 1: 100X=10

```
110 DO UNTIL X=0 OR X=1  
120 PRINT "X ještě není 0 nebo 1"  
130 X=X-1  
140 LOOP  
150 ...
```

Smyčka proběhne devětkrát, potom je X=1 pravda a program pokračuje řádkem 150.

Příklad 2: 100 DO WHILE A\$="" : GET A\$: LOOP

Provádění programu bude pozastaveno tak dlouho, dokud nebude stlačena nějaká klávesa.

INSTRUKCE DOPEN

Formát: DOPEN # log. číslo souboru, jméno data [L délka]
[,Ddisketová jednotka][,Upřístr.adr][,w]

Účel: Otevří sekvenční nebo relativní data na disketě pro vstup nebo výstup dat /viz kap. 6.6./

Logické číslo souboru

Celočíselná hodnota mezi 1 a 255, které bude přiřazeno otevřenému údaji jako poznávací číslo. U hodnot menších než 128, se po každé instrukci PRINT vydá kód pro zpětný návrat vozíku. U hodnot přes 127 se dodatečně vydá kód posunu řádků.

Jméno data Znaková řetězcová konstanta v uvozovkách " nebo - proměnná o délce až 16 znaků, která označuje jméno otevíraných dat.

Délka Celočíselná hodnota mezi 1 a 254 pro délku logických záznamů relativních dat.

Disketová jednotka Celočíselná hodnota 0 nebo 1. Přednastavena je zde hodnota 0.

Přístrojová adresa Celočíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojeného floppy-disku. Přednastavena je zde hodnota 8.

W Zadá-li se tento parametr, otevře se sekvenční údaj k zápisu, v opačném případě ke čtení.

Poznámky: Předají-li se parametry jako proměnné, potom musí být tvto umístěny v závorkách.

L délka a W nesmí být zadány současně v jedné instrukci DOPEN.

Otevřou-li se dvě data pod tímto logickým číslem data, indikuje se chybové hlášení FILE ALREADY OPEN.

Příklady: DOPEN #2,"DATEN"

Sekvenční údaj DATEN na disketové jednotce 0, floppy-disku s přístrojovou adresou 8 se otevře ke čtení.

DOPEN #3,"ADRESY",D1,U9,W

Sekvenční údaj ADRESY na disketové jednotce 1, floppy-disku s přístrojovou adresou 9 se otevře k zápisu.

DOPEN #5, "Klienti",L112

Relativní údaj "klienti" s logickou délkou záznamu o 112 Bytech se otevře na disketové jednotce 0 floppy-disku s přístrojovou adresou 8.

INSTRUKCE D R A W

Formát: DRAW [zdroj barvy], [x₁,y₁] TO x₂,y₂ ...

Účel: Spojuje čarou libovolný počet specifikovaných souřadnic.

Zdroj barvy: Celočíselná hodnota mezi 0 a 3, která definuje barvu, ve které má /mají/ být čára /-y/ kreslena /-y/.

Přednastavena je barva popředí /1/.

Znamenají:

- 0 barva pozadí
- 1 barva popředí /znaky/
- 2 doplňková barva 1
- 3 doplňková barva 2

x₁, y₁ souřadnice počátečního bodu čáry

x₂, y₂ souřadnice koncového bodu čáry

Poznámky:

S předem umístěným parametrem TO může být specifikováno kolik souřadnic, kolik je místa v jednom programovém řádku.

Koncový bod předcházející čáry je pak současně i počátečním bodem čáry následující.

Jsou-li počáteční a koncová souřadnice stejné, zobrazí se bod.

Nezadá-li se počáteční souřadnice, platí aktuální pozice grafického kurzoru /Pixel-kursor/ jako počáteční bod.

Příklady: DRAW 1,100,50

Nakreslí bod, protože není zadáná koncová souřadnice.

100 DRAW,10,10 TO 100,60
110 DRAW TO 25,30

Nakreslí úhel.

DRAW ,10,10 TO 10,60 TO 100,60 TO 10,10

Nakreslí obrazec.

PŘÍKAZ D S A V E

Formát: DSAVE jméno data [,D disketová jednotka]
[,U adresa přístroje]

PŘÍKAZ DSAVE - pokračování

Formát: DSAVE jméno data [, D disketová jednotka]
[, U přístrojová adresa]

Účel: Ukládá Basic- program z hlavní paměti na disketu v
udané disketové jednotce /viz kap. 6.2./

Jméno data Znaková řetězcová konstanta v uvozovkách " nebo
- proměnná v závorkách () o délce až 16 znaků, která označuje
jméno programového data.

Disketová jednotka Celocíselná hodnota 0 nebo 1. Přednastavena je zde hodnota 0.

Přístrojová adresa Celocíselná hodnota mezi 4 a 15, která označuje adresu přístroje připojeného floppy-disku. Přednastavena je zde hodnota 8.

Poznámky:

Budou-li parametry předány jako proměnné, pak musí být umístěny v závorkách.

Po provedení příkazu DSAVE se interpret v každém případě vrací zpět do přímého módu.

K uložení programu na kazetu je třeba použít instrukci SAVE /viz kap. 5.2./

Existuje-li již zadaný údaj na kazetě, pak dojde k chybovému hlášení FILE EXISTS, které lze indikovat s PRINT DS\$.

Příklady: DSAVE "PROG1"

Program, nacházející se v hlavní paměti se uloží na disketu v disketové jednotce 0, přístroj 8 pod jménem PROG1.

DSAVE (AS) ,D(LW) ,U(GA)

Zde jsou parametry předávány jako proměnné.

SYSTÉMOVÉ PROMĚNNÉ DS- a DS\$

Formát: v=DS
v\$=DS\$

Účel: Vydává chybový stav jako kód/DS/ a jako kompletní stavové hlášení /DS\$/ naposledy volaného floppy-disku.

Poznámky:

Stavový kód 0 znamená - žádná chyba.

Stavové hlášení je řetězec, a sestává ze 4 částí:

- stavový kód
- stavový text
- číslo stopy, ve které se chyba vyskytuje
- číslo sektoru, ve kterém se chyba vyskytuje

Chybovým dotazem se chybový stav uvede zpět.

Protože DS a DS\$ jsou systémové proměnné, nesmějí být použity v programu jako specificky uživatelské proměnné. V opačném případě se bude indikovat chyba SYNTAX ERROR.

Příklad: PRINT DS\$:PRINT DS
32,SYNTAX ERROR,00,00
32
READY

Naposledy volaný floppy-diskový přístroj ukáže syntaxní chybu, protože předávaný příkazový znakový řetězec byl příliš dlouhý.

PŘÍKAZ DVERIFY

Formát: DVERIFY jméno data [,D disketová jednotka]
[,Upřístrojová adresa]

Účel:

Porovnává bytovým způsobem program v operační/hlavní/ paměti s programovým datem na zadané disketě.

Jméno data Znaková řetězcová konstanta v uvozovkách "/", nebo -proměnná v ~~maximálně~~ závorkách o délce až 16 znaků, které označují jméno programového data.

Disketová jednotka Celočíselná hodnota 0 nebo 1. Přednastavena je zde hodnota 0.

Přístrojová adresa Celočíselná hodnota mezi 4 a 15, která označuje přístrojovou adresu připojeného floppy-disku. Přednastavena je zde hodnota 8.

Poznámky:

Bude-li rezervován, nebo opět uvolněn grafický rozsah po provedení příkazu SAVE nebo DSAVE, vydají příkazy VERIFY nebo DVERIFY chybové hlášení VERIFY ERROR, ačkoliv je program v pořádku. Dojde k tomu tím, že rezervováním, nebo uvolněním grafického rozdahu se Basic-program přesune ze své původní paměťové pozice, čímž se změní adresový ukazatel uvnitř Basicových řádků a není dána bytová shoda /přizpůsobení s uloženým programem.

Příklady: DVERIFY"PROG1"

Program PROG1 na disketové jednotce 0 ve floppy-disku 8 se bytově porovná s rezidentním programem hlavní paměti.

DVERIFY"MULT",D1,1

Program MULT na disketové jednotce 1, floppy-disku 8 se porovná s programem stávajového jazyka umístěným v hlavní paměti, který nezačíná na začátku programové paměti Basicu.

SYSTÉMOVÁ PROMĚNNÁ E L

Formát: v=EL

Účel: Vydává číslo programového řádku, ve kterém se vyskytla chyba.

Poznámky:

Protože je EL systémovou proměnnou, nesmí být použita v programu jako uživatelsky specifikovaný proměnná. Jinak se indikuje chyba SYNTAX ERROR.

Příklady: 800 IF EL=100 THEN 1000

Je-li EL použito v IF-dotazu, musí stát srovnávací číslo napravo od proměnné EL, aby bylo při případném přečislování respektováno.

INSTRUKCE ENVELOPE

Formát: ENVELOPE n [, [An][, [Ab][, [Ha]
[,[Au][, [Wf][, Ib]]]]]]]

Účel: Definuje jednu z 10 možných obalových křivek pro hudební nástroje.

n číslo obalové křivky /0 až 9/

An doba úhozu / 0 až 15/

Ab doba opadnutí /slábnutí/ / 0 až 15/

Ha doba udržení / 0 až 15/

Au doba doznívání / 0 až 15/

Wf tvar vln:

0 trojúhelníkový

1 pilovitý

2 obdélníkový

3 šumění

4 kruhová modulace

Ib šířka impulsu / 0 až 4095/

Poznámky:

Nespecifikovaný parametr si uchová svoji momentální hodnotu.

Šířka impulsu je definovatelná jen v souvislosti s obdélníkovým tvarem vln /Wf=2/ a stanoví se ze vzorce:

šířka pulsu = Ib/40.95% určuje, že Ib = 2048 a vytváří obdélníkovou vlnu.

Po zapnutí počítače jsou pro parametry přenastaveny následující hodnoty:

n	An	Ab	Ha	Au	Wf	Ib	nástroj
0	0	9	0	0	2	1536	klavír
1	12	0	12	0	1		akordeon
2	0	0	15	0	0		cirkusové varhany
3	0	5	5	0	3		buben
4	9	4	4	0	0		flétna
5	0	9	2	1	1		kytara
6	0	9	0	0	2	512	cemballo
7	0	9	9	0	2	2048	varhany
8	8	9	4	1	2	512	trumpeta
9	0	9	0	0	0		xylofon

Další podrobnosti k instrukci ENVELOPE viz oddíl Vytváření hudby s C 128" kap. 4.8. /orig. str.4-156/

SYSTÉMOVÁ PROMĚNNÁ E R

Formát: v=ER

Účel: Poskytuje kód chyby naposledy diagnostikované interpretem.

V kap. 8.1. jsou uvedena chybová hlášení společně s jim přiřazenými chybovými kódy.

Poznámky:

Protože je ER systémovou proměnnou, nesmí být použita v programu jako uživatelská specifikovaná proměnná. Jinak se indikuje chyba SYNTAX ERROR.

Příklady: 100 IF ER=30 THEN 1000

Stlačí-li se klávesa STOP, přeskočí program na řádek 1000.

FUNKCE E R R \$

Formát: v\$=ERR\$(n)

Účel: Vydává text chyby specifikované hodnotou n.

n je celočíselná hodnota mezi 1 a 127, která udává kód chyby /viz kap. 8.1./

Příklad: 10 TRAP 1000

• basicové instrukce

1000 REM ANALÝZE CHYBOVÝCH HLÁŠENÍ
1001 REM PŘI SYNTAX-ERROR PROGRAM STOP
1010 IF ER=11 THEN PRINT EL,ERR\$:STOP
1011 REM POKUD NENÍ SYNTAX ERROR, BUDE
1012 REM HLÁŠENÍ ZKONTROLOVÁNO A VYTISKNUTO
1020 IF ER=9 THEN PRINT EL,ERR\$(ER):
RESUME 100
1030 IF ER=30 THEN RESUME 150
1040 RESUME 975

Ostatní instrukce nutné ke zpracování chyb jsou popsány u RESUME a TRAP.

PŘÍKAZ F A S T

Formát: FAST

Účel: Přepíná procesor z režimu v 1 Mhz frekvenci do režimu ve 2Mhz frekvenci. Počítač je tím dvakrát rychlejší.

Poznámky: Příkazem FAST se 40ti znakové obrazovka dezaktivuje, tzn. výstup textu nebo grafiky je viditelný pouze na 80ti znakové obrazovce.

Používáte-li výhradně televizní přijímač nebo monitor se 40ti znakovou obrazovkou, přepněte prosím, před zadáním textu nebo grafiky počítač zpět do 1 Mhz režimu příkazem SLOW /viz tam/.

PŘÍKAZ F E T C H

Formát: FETCH Byte, Intadr, Extadr, Bank

Účel: Převádí volitelný počet bytů z rozšířené paměťové banky do pracovní paměti Basicu.

Byte Celocíselná hodnota mezi 0 a 65535, která udává počet přenášených bytů.

Intadr: Celocíselná hodnota mezi 0 a 65535, která udává adresu v pracovní paměti Basicu, od které mají být Byty ukládány.

Extadr: Celocíselná hodnota mezi 0 a 65535, která udává adresu v paměťové bance /datová základna/, od které mají být Byty brány.

Bank: Celocíselná hodnota mezi 0 a 7, která udává datovou základnu; ze které mají být Byty čerpány.

Poznámky:

Tento příkaz být prováděn jen tehdy, je-li na počítač připojen Ram-Disk. Ram-disk dekóduje odpovídající příkaz /příslušný příkaz/ a provádí jej.

Parametr Intadr se vztahuje na banku, která byla nastavena příkazem BANK. Přednastavena je banka 15.

Příkaz FETCH - pokračování

Poznámky:

Mají-li se např. uložit data z banky 0 do Ram-disku, musí být před příkazem STASH zadán příkaz 'BANK 0'.

Příklad: FETCH 1000,52000,2000,7

Přenáší 1000 Bytů z paměťové základny 7 od adresy 2000 do pracovní paměti Basicu od adresy 52000.

POZOR!

K provádění tohoto příkazu potřebujete nástrčný modul Ram-Disk.

INSTRUKCE FILTER

Formát: FILTER [Freq][,[Tp][,[Bp][,[Hp]
[,,[Res]]]]]

Účel: Definuje parametry pro zvukový filtr.

Freq Celočíselná hodnota mezi 0 a 2047, která určuje mezní kmitočet filtru.

Tp,Bp,Hp Celočíselná hodnota mezi 0 a 1, která zapíná nebo vypíná dolní, pásmovou nebo horní propust filtru. /Zap=1, vyp.=0

Res: Celočíselná hodnota mezi 0 a 15, která určuje rezonanci.

Poznámky: Nejsou-li zadány parametry, zůstává v platnosti aktuální hodnota.

Skutečná mezní frekvence filtru se vypočítá z:

$$\text{MEZNÍ FREQ} = \text{Freq}^* 5.8 + 30.$$

Tatoxx hodnota je však jen opěrnou hodnotou a může se nepatrně lišit v závislosti na počítači.

Výstupy filtru jsou aditivní. Může tak být docíleno společnou volbou z hluboko- a vysokopásmového filtru, zásekových nebo pásmových závorových efektů.

K docílení slyšitelných filtrových efektů se musí zvolit minimálně jeden filtr a provést nejméně jedna tónová frekvence přes filtr.

FUNKCE FRE

Formát: v=FRE(Bank)

Význam: Vydává počet ještě nepoužitých míst v paměti v programové nebo pracovní paměti Basicu.

Bank: Celočíselná hodnota 0 nebo 1, která označuje datovou základnu /paměťová banka/, jejíž nevyužitá paměťová místa mají být dotazována.

Poznámky: Jestliže se vyvolá FRE(1), uspořádá se organizátorsky paměťový rozsah znakových řetězců v datové základně 1, dříve, než bude předán počet volných paměťových míst. Tento paměťový rozsah bude interpretem dynamicky veden, tzn. více-násobné přidělení znakových řetězců k jedné a té samé proměnné povede ke stále novému ukládání těchto řetězců v paměti.

INSTRUKCE GETKEY

Formát: GETKEY seznam řetězcových proměnných

Účel: Instrukce GETKEY se podobá instrukci GET /viz kap.5.2/. Čeká však tak dlouho, až je na klávesnici stlačena klávesa a stlačené klávese přiděluje ASCII kód jako 1-Bytový řetězec specifikované proměnné. Je-li zadáno více řetězcových proměnných, rozdělených čárkou, je potřeba uvést v činnost i přiměřeně více kláves.

Seznam řetězcových proměnných: Je jedna nebo více řetězcových proměnných, oddělených od sebe čárkou.

Poznámky: Instrukce GETKEY může být použita jen v programovém módu. Jinak se indikuje chybové hlášení ILLEGAL DIRECT.

Příklady: 100 GETKEY A\$,B\$,C\$
ASCII-kody tří stlačených kláves se nvní přidělí proměnným A\$,B\$,C\$ jako 1-Bytové řetězce.

```
100 GETKEY A$  
110 IF A$ < "A" OR A$ > "B" THEN 100  
120 ON ASC( A$ )-64 GOTO 1000,2000
```

Bude-li stlačena klávesa A nebo B , rozvětví se program na řádky 1000 nebo 2000.

PŘÍKAZ G O 64

Formát: GO64

Účel: Počítač se přepíná z módu C128 do módu C64.

Poznámky:

Interpret požaduje indikaci

ARE YOU SURE?

od uživatele potvrzení k příkazu. Zadá-li se Y /YES/, vymaze se právě existující program v paměti a počítač se přepne do módu C64. Při zadání N /NO/ zůstane aktivní modus C128 a na stávajícím programu nedojde k žádným změnám.

Tento příkaz může být zadáván jak v přímém tak i v programovém módu.

INSTRUKCE G R A P H I C

Formáty: GRAPHIC Modus [,[výmaz], text.řádky]
GRAPHIC CLR

Účel: Aktivuje grafický modus a rezervuje grafickou paměť /bit mapped/ ev. uvolňuje opět grafickou paměť.

Účel:

Modus Celočíselná hodnota mezi 0 a 5. Jsou volitelné následující módy:

- 0 Text se 40ti znaky/řádek
- 1 Vysokorozlišitelná /jemná/ grafika /320x200 bodů/
- 2 Vysokorozlišitelná /jemná/ grafika, rozdělená obrazovka pro grafiku a text.
- 3 Vícebarevná grafika /160x200 bodů/
- 4 Vícebarevná grafika, rozdělená obrazovka pro grafiku a text.
- 5 Text s 80ti znaky/řádek

Výmaz: Celočíselná hodnota 0 nebo 1, která určuje, má-li být obrazovka při vyvolání grafického módu 1 až 4 vymazána /1/, nebo ne/0/.

Text řádky : Celočíselná hodnota mezi 0 a 24, která určí při módech 2 a 4 řádek, u kterého začíná část textové obrazovky. Přednastavena je zde hodnota 19.

INSTRUKCE - GRAPHIC - pokračování

GRAPHIC CLR uvolňuje instrukcí GRAPHIC-Modus rezervovanou paměť.

Poznámky:

V kap. 4.7. je podrobně popsána jemná i vícebarevná grafika. Při módu 1 až 4 je rezervován 9KByte rozsah paměti na začátku programové paměti BASIC jako grafická paměť.

Začátek paměti programu BASIC se tím přesune směrem nahoru. Tento rozsah zůstane rezervován i při zapnutí textového módu /1nebo5/ a uvolní se teprve zadáním instrukce GRAPHIC CLR.

Příklady: GRAPHIC 1

Přepíná na jemnou grafiku.

GRAPHIC 4,1,15

Zapíná na vícebarevnou grafiku s rozdelenou obrazovkou a začátek textové obrazovky umísťuje na řádek 16./Řádky začínají od 0/.

GRAPHIC CLR

Rezervovaná grafická paměť se opět uvolní.

INSTRUKCE GSHAPE

Formát: GSHAPE řetězec[, [x,y[, Modus]]]

Účel: Přenáší hodnotu zadaného řetězce jako binární obrazovou informaci na grafickou obrazovku /viz instrukce MOVSPR/.

Řetězec: Řetězcová proměnná nebo konstanta, která obsahuje binárně zobrazenou obrazovou informaci/viz instr. SSHAPE/.

x,y : Měřítkem nastavené obrazovkové souřadnice levého horního rohu zobrazeného obrazu.

Modus: Celocíselná hodnota mezi 0 a 4, která určuje druh zobrazení. Proveditelné jsou následující módy:

0 Data z řetězce se zobrazí přesně tak, jak byla uložena do paměti s SSHAPE.

1 Obraz bude invertní.

INSTRUKCE - GSHAPE - pokračování

- 2 Obraz , který je na obrazovce k dispozici, se překrýve obrazovou informací z řetězce /logické NEBO/.
- 3 Obrazové body se přenesou jen tehdy, existují-li pod přenašenými body obrazové body na obrazovce /logické A/.
- 4 V existujícím obrazu se budou invertovat obrazové body tam, kde v přenašeném obrazu existují. Tím lze posunout objekt přes obrazovku bez výmazu pozadí.

PŘÍKAZ H E A D E R

Formát: HEADER jména, D disket.jednotka, I identifikace] [, Upřístroj]

Účel: Formuje disketu v zadané disketové jednotce zvoleného Floppy-disku /nově/ nebo maže seznam /viz kap. 6.1./

Jméno: Řetězcová konstanta v uvozovkách "/" nebo řetězcová proměnná o max. délce 16 znaků, které popisují jméno diskety.

Disketová jednotka: Celocíselná hodnota 0 nebo 1 pro označení disketové jednotky. Přednastavena je zde hodnota 0.

Identifikace: Dva libovolné znaky pro identifikaci diskety. Tato identifikace se zapíše při každé formátování do každého sektoru. Každé disketě by měla být bezpodmínečně zadána vlastní identifikace.

POZOR!

Vzhledem ke zvláštnímu způsobu, ve kterém interpretuje interpret floppy-diskové příkazy, nesmí být pro parametr Identifikace použito klíčových slov Basicu jako ON,FN,TO atd. nebo zkratek klíčových slov jako aB,c0 apod. Jinak se indikuje chybové hlášení SYNTAX ERROR.

Přístroj: Celocíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojeného floppy-disku. Přednastavena je 8.

Příkaz HEADER - pokračování

Poznámky: Vzhledem k tomu, že se při formování zničí všechna příp. již dříve uložená data na disketu, je po uživateli otázkou ARE YOU SURE? žádáno potvrzení.

Odpověď Y /yes/ provede příkaz HEADER, každá jiná odpověď ne.

Má-li být již formovaná disketa pouze smazána, vynechá se v příkazu HEADER parametr I identifikace. Disketa pak obdrží jen nové jméno, ponechá si však starou identifikaci. Použijí-li se jako parametr proměnné, pak musí být umístěny v závorkách.

Příklady: HEADER"DATA",DØ, I DA

Disketa v disketové jednotce 0, floppy-disku s přístrojovou adresou 8 se nově zformuje identifikací DA a obdrží jméno DATA.

HEADER"PROGRAMY", I P1, D1,U9

Disketa v disketové jednotce 1 floppy-disku s přístrojovou adresou 9 se nově formátuje identifikací P1 a obdrží jméno PROGRAMY.

HEADER(N_S), I(K_S), D(L), U(G)

Jsou tu pro všechny parametry použity proměnné.

HEADER "KMENOVÁ DATA",DØ

Disketa v disketové jednotce 0, floppy-disku s přístrojovou adresou ~~8~~ 9, se jen vymaže, obdrží jméno KMENOVÁ DATA a podrží si svoji starou identifikaci.

PŘÍKAZ H E L P

Formát: HELP

Účel: Po chybovém hlášení interpretu indikuje chybný programový řádek na obrazovce a zobrazuje chybou část při 40ti znakové obrazovce inverzně, při 80ti znakové obrazovce podtrženě / viz klávesa HELP v kap. 4.1./

Poznámky: Příkaz HELP/nebo stisk klávesy HELP/ nemá žádný účinek u chyb, které byly diagnostikovány instrukcemi, zadánymi v přímém módu.

FUNKCE H E X \$

Formát: .v\$=HEXS(n)

Účel:

Vydává znakový řetězec, který obsahuje hexdecimální vyjádření hodnoty numerického výrazu n.

n Libovolný numerický výraz, jehož celočíselná hodnota se bude měnit a musí ležet v intervalu od 0 do 65535.

Příklad: 100X=23:Y=1456

110 PRINT X; HEXS(X), Y, HEXS(Y)

RUN

23 17 1456 5BØ

READY

INSTRUKCE I F

Formát: IF výraz THEN instrukce [:ELSE instr.]

Účel: Dovoluje větvení v různých částech programu v závislosti na logickém obsahu pravdy numerického výrazu. Numerický výraz je logicky "pravdivý" tehdy, když je jeho hodnota různá od nuly, a logicky "nepravdivý", je-li jeho hodnota rovna nule.

Výraz: Každý libovolný numerický výraz

Instr.: Jediná instrukce, nebo sled instrukcí, oddělených od sebe dvojtečkou, nebo prostě číslo řádku.

Poznámky:

Je-li numerický výraz logicky "pravdivý" bude/budou/ instrukce za THEN provedena /prováděny/. Je-li oproti tomu logicky "nepravdivý", bude/budou/ instrukce za THEN ignorována /-y/ a instrukce za ELSE, je-li /jsou-li/ k dispozici, nebo další programový řádek, budou provedeny.

U testu rovnosti musí být přihlédnuto k nepřesnosti na základě interního zobrazení pohyblivé řádové čárky. V kritických případech by měl být proveden test o rozsahu, uvnitř jehož mezi přesnost kolísá.

Instrukce IF - pokračování

Např.: IF ABS (X-2.5) < 1.06E-6 THEN...

dává výsledek "pravdivý", když je X=2.5 s relativníchybou menší než 1.06E-6.

Instrukce IF ... THEN ... : ELSE musí být umístěna v jednom programovém řádku. Ani THEN, ani ELSE nesmějí být umístěny v separátních řádcích, ledaže by byla do instrukce IF vložena instrukce BEGIN. /viz př. 2 v instrukci BEGIN/.

Je-li počet údajů THEN a ELSE rozdílný, pak se ELSE vztahuje vždy k poslednímu předešlému THEN.

Příklad 1: 100 NMSG\$=" HODNOTA MUSÍ BYT Kladna"

```
110 PMSG$ ="ODMOCNINA JE"  
120 INPUT"PROSIM ZADEJ CISLO"; N  
130 IF N<0 THEN PRINT NMSG$:GOTO 120  
:ELSE PRINT PMSG$;SQR(N)  
140 INPUT"JESTE JEDNO CISLO ( J/N )"; Z$  
150 IF Z$="J" THEN 120:ELSE END
```

Odmocnina zadaného čísla bude vypočtena a vytiskána jen tehdy, bude-li číslo kladné.

Příklad 2: 100 IF X>Y THEN PRINT "VETSI"
:ELSE IF Y>X THEN PRINT "MENSI"
:ELSE PRINT "ROVNO"

Toto je příklad pro vnořené /roztříděné/ instrukce IF.

Příklad 3: 100 IF A=B THEN IF B=C THEN PRINT "A=C"
:ELSE PRINT "A<>C"

Nerovná-li se A, B, potom již druhé srovnání nebude provedeno.

FUNKCE I N S T R

Formát: v=INSTR(xs,y\$,[,n])

Účel:

Hledá a vydává pozici prvního výskytu části řetězce y\$ v xs.

xs Libovolný řetězcový výraz, jehož hodnota vyjadřuje prohledávaný řetězec.

y\$ Libovolný řetězcový výraz, jehož hodnota vyjadřuje hledanou část řetězce.

Funkce INSTR - pokračování

n: Číselný výraz, jehož celočíselná hodnota musí ležet mezi 0 a 255 a která udává startovací pozici v prohledávaném řetězci. Přednastavena je zde hodnota 1.

Poznámky:

Hodnotu 0 vydává funkce INSTR v následujících případech:

- n je větší než LEN (x\$)
- x\$ je prázdný
- y\$ není obsažen v x\$

Příklad: PRINT INSTR ("COMMODORE", "R")

8

READY

FUNKCE J O Y

Formát: v=JOY (n)

Účel:

Vydává hodnotu pro pozici jakož i pro stav odpalovacího tlačítka volitelného joysticku /hrací panel/.

n je celočíselná hodnota 1 nebo 2, která označuje žádaný joystick.

Hodnota předávaná funkcí JOY má následující význam:

- 0 střední poloha
- + dopředu
- 2 po diagonále vpravo dopředu
- 3 napravo
- 4 po diagonále vpravo vzadu
- 5 dozadu
- 6 po diagonále vlevo vzadu
- 7 doleva
- 8 po diagonále vlevo dopředu

Hodnoty od 128 do 136 znamenají, že bylo dodatečně stlačeno odpalovací tlačítko.

Příklad: PRINT JOY (2)
135
READY

Joystick 2 je při stlačeném odpalovacím tlačítku v poloze vlevo.

PŘÍKAZ KEY

Formát: KEY [n, x\$]

Účel: Nově obsazuje funkční klávesy nebo indikuje jejich obsazení.

n: Libovolný numerický výraz, jehož celočíselná hodnota musí ležet mezi 1 a 8 a který označuje funkční klávesu.

x\$ Libovolný řetězcový výraz, jehož hodnota bude přiřazena zvolené funkční klávese a při uvedení této klávesy do činnosti se bude indikovat na obrazovce.

Poznámky:

Zadá-li se příkaz KEY bez parametru, indikují se na obrazovce okamžitá obdazení funkčních kláves.

Délka všech osmi obsazení dohromady nesmí překročit 256 znaků.

Příklad: KEY 8,"GRAPHIC 0" +CHR\$(13)+"LIST"+CHR\$(13)

Bude-li stlačena v přímém módu funkční klávesa f8 /čtvrtá klávesa současně s SHIFT/, zapne se textový modus a program nacházející se v hlavní paměti bude listován.

INSTRUKCE LOCATE

Formát: LOCATE x,y

Účel: Polohuje grafický kurzor /Pixel-kurzor/ na obrazovce.

x,y souřadnice v měřítku /skálované s./ /viz instr. SCALE/. Pro x /horizontální souřadnice/ a pro y /vertikální souřadnice/ mohou být dle stupňování zadávány hodnoty mezi 0 a 1023.

Poznámky:

Na rozdíl od normálního textového kurzoru není grafický kurzor viditelný.

Aktuální pozice grafického kurzoru je výchozí pozicí pro všechny grafické znakové instrukce.

INSTRUKCE LOCATE - pokračování

Poznámky: Po provedení grafické instrukce znaků je poslední zakreslený obrazový bod novou aktuální pozicí grafického kurzoru.

Aktuální pozice grafického kurzoru může být nalezena za pomocí funkce RDOT. /viz tam/.

Příklad: LOCATE 160,100

Grafický korzor se umístí do středu obrazovky /vysoká rozlišovací schopnost, bez stupnice/.

PŘÍKAZ M O N I T O R

Formát: MONITOR

Účel: Monitor strojového jazyka se vyvolá.

Poznámky: Funkce a obsluha monitoru jsou podrobně popsány v dodatku C.

INSTRUKCE M O V S P R

Formát: MOVSPR n,x,y
MOVSPR n,+/-x1,+/-y1
MOVSPR n, úhel # rychlosť

Účel: Uvede do pohybu Sprite se specifikovanou rychlostí nebo umístí Sprite na určité místo na obrazovce.

n Je celočíselná hodnota mezi 0 a 7, která udává číslo spritu.

x,y absolutní souřadnice se stupnicí, na kterých má být sprite umístěn.

+/-x1,+/-y1 Relativní souřadnice se stupnicí k poslednímu polohování Sprite, na kterých má být Sprite umístěn.

úhel Celočíselná hodnota mezi 0 a 360, která udává, ve jakém úhlu se má sprite pohybovat ze své původní pozice.

Instrukce MOVSPR - pokračování

Účel:

rychlosť Celočíselná hodnota mezi 0 a 15, která udává rychlosť, jakou se má sprite pohybovat.

Poznámky: Použití instrukce MOVSPR je podrobně popsáno v mnoha příkladech v kap. 4.7.

INSTRUKCE PAINT

Formát: PAINT [zdroj barvy][, [x,y]] [,modus]]

Účel: Vymalovává rozsah obrazovky ve volitelné barvě.

Zdroj barvy: Celočíselná hodnota mezi 0 a 3, která definuje barvu, jakou má být plocha obrazovky vymalována. Přednastavena je zde barva popředí/1/.

Význam:

0	barva pozadí
1	barvá popředí /znaků/
2	doplňková barva 1
3	doplňková barva 2

x,y Souřadnice /na stupnici/ startovacího bodu malování. Přednastavena je zde aktuální pozice grafického kurzoru.

Modus Celočíselná hodnota 0 nebo 1, která určuje, jestli vymalovávaný rozsah obklopen volitelnou barvou /0/ nebo jinou barvou než je barva pozadí /1/.

Poznámky: Instrukce PAINT využívá plňovacího plochu okolo zadávaného bodu tak dlouho, až je dosaženo okraje/rámečku/ v zadáné barvě /Modus 0/ nebo v barvě jiné, než je barva pozadí.

Konečná pozice grafického kurzoru je zadána dvojicí souřadnic x,y.

Má-li počáteční bod již zadanou barvu v módu 0 ev. nějakou jinou barvu v módu 1, než je barva pozadí, nemá příkaz PAINT žádný účinek.

Příklad: 10 CIRCLE,160,100,65,50
20 PAINT , 160,100

Kruh kreslený na řádku 10 bude vymalován v barvě popředí /přednastavení/.

FUNKCE P E N

Formát: v=PEN(n)

Účel: Poskytuje stav nebo obrazovkové souřadnice světelné tužky.

- Celocíselná hodnota mezi 0 a 4, která má následující účinek:
- 0 x-ová souřadnice polohy světelné tužky na grafické obrazovce /60 až 320/.
 - 1 y-ová souřadnice polohy světelné tužky na grafické obrazovce /50 až 200/.
 - 2 poloha znakového sloupce světelné tužky při 80ti znakové obrazovce.
 - 3 poloha znakového řádku světelné tužky při 80ti znakové obrazovce.
 - 4 Předá se 1 v případě, že od posledního dotazu byla světelná tužka aktivována, jinak 0.

Poznámky: Tak jako souřadnice Spritu, tak i souřadnice světelné tužky nejsou souřadnicemi v měřítku, nýbrž skutečné souřadnice uvnitř orámovaného rozsahu obrazovky.

Leží-li poloha světelné tužky mimo platný rozsah obrazovky, jsou předávané hodnoty souřadnic rovny 0.

Pro funkci PEN není třeba aktivovat instrukci COLLISION.

Všeobecně je pro stimulaci světelné tužky žádoucí bílé pozadí obrazovky.

Hodnoty předávané funkcí PEN mohou být na různých obrazovkách odlišné.

Příklad: 100 DO UNTIL PEN(4):LOOP
110 PRINT PEN(2);PEN(3)

Jakmile se světelná tužka zaktivuje, vytiskne se sloupcová a řádková poloha znaku na obrazovce, u kterého byla tužka nasazena.

FUNKCE π /pí/

Formát: $v = \pi$

Účel: Předává hodnotu Ludolfova čísla /3.14159265/.

Příklad: PRINT COS (π)

-1

READY.

INSTRUKCE P L A Y

Formát: PLAY znakový řetězec

Účel: Hraje sled tónů, definovaných pomocí znakového řetězce.

Řetězec Znakový řetězec v uvozovkách, který se skládá z příkazů tvorby a odehrání tónů.

On Určuje jednu ze 7 oktáv /n=0 až n=6/.

Tn Určuje jednu z 10 následujících zvukových obalových křívek:

- 0 klavír /přednastaveno/
- 1 akordeon
- 2 cirkusové varhany
- 3 buben
- 4 flétna
- 5 kytara
- 6 cembalo
- 7 varhanv
- 8 trumpeteta
- 9 xylofon

Un Stanoví hlasitost /n od 0 do 9/.

Vn Stanoví jeden z 3 možných hlasů.

Xn Zapíná /n=1/ nebo vypíná /n=0/ instrukcí FILTER /viz tam/ zvolený filtr.

N Jedna z not A B C D E E G.

#N Nota N bude zahrána o půltón výše

SN Nota bude zahrána o půltónníže.

W Následující nota bude zahrána jako celá nota.

H Následující nota bude zahrána jako poloviční nota

Q Následující nota bude zahrána jako čtvrtková nota

I Následující nota bude zahrána jako osminová nota.

Instrukce PLAY - pokračování

- S** Následující nota bude zahrána jako šestnáctinová nota.
- M** Následující nota bude zahrána jako tečková nota /polovina její hodnoty/.
- R** Stanoví pauzu.
- U** Čeká až právě hraná hudba dohraje.

FUNKCE P O I N T E R

Formát: v=POINTER (proměnná)

Účel: Poskytuje adresu hlavní paměti prvního Byte ukazatele proměnných v paměťové bance 1, ve které ukládá interpret Basicu všechny proměnné, pole a znakové řetězce.

Proměnná Libovolná proměnná nebo libovolný element pole proměnných, kterým již musela být přidělena hodnota.

Poznámky: Předaná adresa v rozsahu 0 až 65535 znázorňuje offset v paměťové bance 1.

Použije-li se funkce POINTER pro element pole proměnných, měly by být všechny jednoduché proměnné předem definovány, aby se při každé nové definici jednoduché proměnné neměnily adresy elementů pole.

POINTER je užitečný k předávání adres počátečních pro popis proměnných v podprogramech strojového jazyka.

Příklad: 100 BX% = 1024
110 VP=POINTER (BX%)
120 BANK 1:AX=256 * PEEK(VP) + PEEK(VP+1)

Data, která jsou uložena do celočíselné proměnné BX%, se přečtou v řádku 120.

FUNKCE P O T

Formát: v=POT(n)

Účel: Poskytuje polohu jednoho ze 4 připojitelných natáčivých transformátorů /paddles/.

n je celočíselná hodnota mezi 1 a 4, která definiuje natáčivý transformátor.

1	natáčivý transformátor 1
2	" - " - 2
3	" - " - 3
4	" - " - 4

Poznámky: Hodnota předaná funkcí POT může ležet mezi 0 a 255. Hodnota zvýšená o 256 znamená, že mimo to bylo stlačeno i odpalovací tlačítko.

INSTRUKCE P R I N T U S I N G

Formát: PRINT [číslo souboru.] USING v\$;
seznam výrazů [;]

Účel: Dává na obrazovce nebo na výstupním přístroji hodnoty ze seznamu výrazů již zformátované.

Číslo souboru Číslo, pod kterým se otevřou data pro výstupní přístroj. U čísel mezi 1 a 127 se vydávaným datům připojí kód zpětného chodu vozíku /CHR\$/13//. U čísel mezi 128 a 255 navíc ještě kód posuvu řádků /CHR\$ /10//.

v\$ Řetězcová proměnná, nebo řetězcová konstanta, která definuje formát tisku / viz příklady/.

Seznam výrazů: Libovolný seznam numerických a/nebo řetězcových výrazů, jejichž hodnoty mají být vydávány. Jednotlivé výrazy musí být od sebe odděleny čárkou, středníkem nebo jením či více prázdnými místy. Formátování výstupu probíhá

Instrukce PRINT USING - pokračování

řídícími znaky, jejichž účinek je v následujícím popsán odděleně pro řetězcové a námerické údaje podle příkladů. Zde však čárka ~~px~~ za výrazem nevede k tabelaci na začátek dalšího, 10 znaků dlouhého tiskového pole.

Řetězcová tisková pole

Každý číselný znak ve formátovém poli je jako znak v tiskovém poli. Každé formátové pole musí obsahovat minimálně jeden číselný znak. Má-li hodnota jednoho řetězcového výrazu více znaků než tiskových míst definovaných #, řetězec se odřízne, v opačném případě se doplní prázdnými znaky. Data se vytisknou v levé rovině pole.

- = Normálně indikovaná data v levé rovině tiskového pole se zobrazí centrováně.
- > Data se budou indikovat v pravé rovině tiskového pole.

Numerická tisková pole

Má-li hodnota jednoho numerického výrazu více tiskových míst než pozič definovaných #, vyplní se tiskové pole hvězdičkami, má-li méně tiskových míst doplní se vlevo prázdná místa, aby mohla být data vytisknuta v pravé rovině pole.

+ a - Znaky plus nebo minus mohou stát buďto na prvním nebo na posledním místě formátovaného řetězce. Znaménko výstupní hodnoty se pak vytiskne na definovaném místě.

- Určuje polohu desetinné tečky v čísle s desetinnými místy. V každém formátovém řetězci se smí vyskytnout jen jedna desetinná tečka.
- Určuje pozici, na které má být v tiskovém poli vydána čárka. Čárce musí předcházet alespoň jeden symbol # ve formátovém znakovém řetězci. Čárka není desetinnou čárkou, slouží jen k lepší čitelnosti velkých čísel.

Instrukce PRINT USING - pokračování

$\$$ Znak dolžku se vytiskne vždy před první platnou číslicí výstupního čísla, je-li před $\$$ - znakem ve formátovaném řetězci symbol $\#$.

↑↑↑ Končí-li formátový znakový řetězec čtyřmi ↑-symboly, po nichž ještě následuje + nebo -, vytiskne se dané číslo v exponenciálním tvaru.

Následující tabulka obsahuje řadu příkladů k použití instrukce PRINT USING:

Formát	Výraz	Výsledek	Poznámky
+ # #	1	+1	Prázdne místo mezi znaménkem a číslem
# - # # +	-.01	0.01-	Hlavní nula připojena
- . # #	-.1	-.10	Hlavní nula přepsaná minusem
# # . # -	1	1.0	Připojena následující nula
+ # # +	1	SYNTAX ERROR	Dva plusové symboly
+ # # . # -	1	SYNTAX ERROR	Symboly plus a minus s
# # # #	-100.5	-101	Zaokrouhleno na 4 místa
# # # #	-1000	* * * *	Přetečení, pole jen pro 4 znaky
# . # #	-4E-03	-.00	Zaokrouhleno na 4 místa
# # .	10	10.	Připojena desetin.tečka
# . # .	1	SYNTAX ERROR	Dvě destinné tečky
# # # , # #	100	1,00	Žádná desetin. čárka, /jak v češtině/
# #, # #	10.4	10	Zaokrouhleno, čárka/tečka/potlačena
#, # # . # #	10000.009	1,000.01	Zaokrouhleno
# # . # #	-1	-1	Čárka potlačena
# #, # #	-10	-10	Čárka přepsaná minusem
# #, # #	-100.9	-1,01	Zaokrouhleno
# # = > . >	1000	1000.0	= a > b d o u , interpretovány jako # poněvadž stojí v číselném poli

Instrukce PRINT USING - pokračování

Formát	Výraz	Výsledek	Poznámky
+ >==,	1	+ >= =,1	Před čárkou musí stát min. jeden # jinak budou >, = a čárka interpretovány jako tisknutelné znaky
+ > =#, #	1	+1	>a = budou interpretovány jako #, protože stojí v číselném poli
# \$##	1	\$1	Vedoucí \$ znak
# ## \$	-1	-\$1	Znaménko stojí před \$
# ## \$##	-1	-\$1	Znaménko stojí před \$
# ## \$-	-1	\$1-	Znaménko stojí na konci
+ \$## #	-1	+ \$-1	Před \$ musí stát min. jeden # jinak bude + interpretováno jako tisknutelný znak
+ # \$##	-1	-\$1	
# .# \$ #	1	* * * *	Přetečení, protože není místo pro \$
+ #. # ↑↑↑↑	1	+1.0E+00	Výraz v exponenciálním vyjádření
# ↑↑↑↑ +	-340	3E+02-	Znaménko za číslem
# ↑↑↑↑	1.5E+11	2E+11	Zaokrouhleno
# ↑↑↑↑	-1.5E+11	* * * *	Přetečení, protože není místo pro znaménko
# #. # ↑↑↑↑	100000	10.00E+04	
# #. #*↑↑↑↑	-100000	-1.00E+05	
#*↑↑↑	1	SYNTAX ERROR	Jen 3 šipky
# # ↑↑↑↑↑	34	34E+001	Pátá šipka se interpretuje jako tisknutelný znak
# # . #	CBM	CBM	Text vytištěn v xx levé rovině
# # # > #	CBM	CBM	V pravé rovině v 5ti znakovém tiskovém poli

Instrukce PRINT USING - pokračování

Formát	Výraz	Výsledek	Poznámky
# # # # #	CBM	CBM	Vlevo v 5ti znakovém tiskovém poli
= # # # #	CBM	CBM	Centrováně v 5ti znakovém tiskovém poli
, # # = +	CBM	CBM	Jako výše, protože ostatní znaky budou interpretovány jako tiskové znaky.

INSTRUKCE P U D E F

Formát: PUDEF v\$

Účel: Umožňuje předefinování až čtvrt různých řídících znaků v instrukci PRINT USING.

v\$ Znaková řetězcová konstanta nebo proměnná v uvozovkách, jejichž hodnota může mít délku až 4 znaky. Postavení těchto znaků ve znakovém řetězci je určitým řídícím znakem instrukce PRINT USING. Platí:

Pozice 1 Vyplňovací znak. Přednastaven je tu prázdný znak.

Pozice 2 Čárka. Přednastavena je zde rovněž čárka.

Pozice 3 Desetinná tečka. Přednastavena je zde rovněž desetinná tečka.

Pozice 4 Dolarový znak. Přednastaven je zde rovněž dolarový znak.

Poznámky: PUDEF mění znaky jen tehdy, když je k výstupu použita instrukce PRINT USING. U instr. PRINT nemá PUDEF žádný účinek.

PUDEF ovlivňuje pouze samotný výstup, nikoliv znakový řetězec formátu z PRINT USING.

Instrukce PUDEF - pokračování

Příklad 1: PUDEF"? "

Prázdná místa se při výpisu /výtisku/ nahradí otazníkem.

Příklad 2:

```
10 PUDEF".,"  
20 PRINT USING "#.#;###.##";-1234.567  
RUN  
-1.234,57  
READY
```

Čárka na tisícové pozici a desetinná tečka /americký způsob zápisu/ se přemění do německého způsobu zápisu.

FUNKCE RCLR

Formát: v=RCLR(n)

Účel: Poskytuje zadánému barevnému zdroji přiřazený aktuální barevný kód v rozsahu od 1 do 16 /viz.instr. COLOR/.

n je celočíselná hodnota mezi 0 a 6, která udává zdroj barevn. Jsou proveditelné následující barevné zdroje:

- 0 Barva pozadí /40ti znaková indikace/
- 1 Grafické popředí
- 2 Vícebarevný modus 1
- 3 Vícebarevný modus 2
- 4 Okraj, rámeček /40ti znaková indikace/
- 5 Text /80ti znaková indikace/
- 6 Pozadí /80ti znaková indikace/

Příklad: A=RCLR(4)

Aktuální barevný kód pro rámeček obrazovky bude přiřazen proměnné A.

FUNKCE R D O T

Formát: v=RDOT(n)

Účel: Vydává aktuální pozici grafického kurzoru /Pixel Kurzor/, na obrazovce nebo kód barevného zdroje platný pro aktuální pozici.

n je celočíselná hodnota mezi 0 a 3 s následujícím významem:

0 x-ová pozice

1 y-ová pozice

2 kód barevného zdroje při aktuální pozici

Příklad:

10 GRAPHIC 1,1

20 CIRCLE 1,160,100,80,80

30 DRAW TO RDOT(0),180

Kreslí kruh se svislým průměrem.

INSTRUKCE R E C O R D

Formát: RECORD #číslo souboru, číslo věty [, čís. Byte]

Účel: Polohuje psací/čtecí ukazatel na libovolnou pozici uvnitř relativního data diskety /viz kap. 6.7/.

Číslo souboru Logické číslo v rozsahu mezi 1 a 255, pod kterým byl otevřen relativní údaj diskety.

Číslo věty: Celočíselná hodnota mezi 1 a 65535, která popisuje údajovou větu v disketovém datu, do kterého má být zasáhnuť.

Číslo Byte: Celočíselná hodnota mezi 1 a 254, která popisuje Byte ve zvolené údajové větě relativního data, na kterou má být umístěn psací/čtecí ukazatel. Přednastavena je zde 1.

Poznámky: Použijí-li se pro parametry proměnné, potom musí být umístěny v závorkách.

Polohování do neexistující věty vede k chybovému hlášení RECORD NOT PRESENT.

Instrukce RECORD - pokračování

Příklad: RECORD #2,452,23

Čtecí/psací /univerzální/ ukazatel relativního data, otevřeného pod logickým číslem data 2, bude umístěn na 23. Byte v 452.větě.

PŘÍKAZ RENAME

Formát: RENAME staré jméno TO nové jméno [,D disket.jedn.]
 [,Upřístrojová adresa]

Účel: Přiděluje disketovému datu ve zvolené disketové jednotce nové jméno. /viz kap. 6.9.4./

Staré jméno, nové jméno: Znaková řetězcová konstanta v uvozovkách nebo znaková řetězcová proměnná o délce až 16 znaků, která popisuje staré ev. nové jméno data na disketě.

Disketová jednotka Celočíselná hodnota 0 nebo 1, která označuje disketovou jednotku. Přednastavena je zde hodnota 0.

Přístrojová adresa: Celočíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojeného floppy-disku. Přednastavena je zde hodnota 8.

Poznámky: Použijí-li se pro parametry proměnné, potom musí být zadány v závorkách.

Příklady: RENAME"ADRESY" TO "NOVE ADRESY"

Údaj ADRESY na disketě v disketové jednotce 0 floppy-disku obdrží jméno NOVE ADRESY.

RENAME (AD\$) TO (ND\$), D(LW), U (GA)

Zde jsou pro parametry použity proměnné.

PŘÍKAZ R E N U M B E R

Formát: RENUMBER[Nčís[, [délka kroku][, A čís]]]

Účel: Přečíslovává úplně nebo jen částečně program Basicu nacházející se v hlavní paměti.

N číslo Číslo řádku, kterým má přečíslovaná část programu začít. Přednastavena je zde 10.

Délka kroku: Celočíselná hodnota, o kterou se má při novém číslování zvětšit každé číslo řádku oproti předcházejícímu. Přednastavena je zde 10.

A číslo Číslo řádku v aktuálním programu, od kterého má být přečíslování provedeno. Přednastavena je zde 0.

Poznámky: Tento příkaz lze zadávat pouze v přímém módu.

Všechny řádkové reference v instrukcích

GOTO ON GOSUB...

GOSUB RESTORE

THEN RESUME

ELSE EL

ON GOTO ...

se přizpůsobí novým číslováním instrukcí RENUMBER.

RENUMBER nemůže změnit sled programových řádků.

Příklady: RENUMBER

Celý program v hlavní paměti se přečísluje v krocích po 10.

První řádek obdrží číslo 10.

RENUMBER 100,15

Celý program v hlavní paměti se přečísluje v krocích po 15. První řádek obdrží číslo 100.

RENUMBER 2500,50,1000

Program nacházející se v hlavní paměti se nově očísluje od řádku 1000 v krocích po 50. Řádek 1000 obdrží číslo 2500.

INSTRUKCE - RESTORE

Formát: RESTORE [číslo řádku]

Účel: Umístuje čtecí ukazatel instrukce READ /viz kap.3/ na první údajový prvek /Data-element/ prvního nebo zvoleného údajového řádku k opakovanému čtení údajových prvků.

Číslo řádku Platné číslo programového řádku, u kterého musí být umístěna instrukce data.

Příklad:

```
100 DATA 1,2,3,4,5,6
110 READ U,V,W:RESTORE 100
120 READ X,Y,Z
130 PRINT U;V;W;X;Y;Z
RUN
1 2 3 1 2 3
READY
```

INSTRUKCE RESUME

Formát: RESUME
RESUME NEXT
RESUME číslo řádku

Účel: Pokračuje v programu po provedení programu na odstranění chyb /viz instrukce TRAP/ na definovaném místě.

RESUME Přerušený program bude pokračovat instrukcí, která chybu způsobila.

RESUME NEXT Přerušený program bude pokračovat instrukcí, která následuje po instrukci, která chybu způsobila.

RESUME číslo řádku : Přerušený program bude pokračovat řádkem definovaným číslem řádku.

Příklad:

```
100 TRAP 500
.
.
.
500 IF ER=11 THEN RESUME 250
```

Při hlášení SYNTAX ERROR má program pokračovat řádkem 250.

FUNKCE R G R

Formát: v=RGR(n)

Účel: Vydává aktuálně nastavený grafický modus /0 až 5;
viz instrukce GRAPHIC/

n Symbolický argument, jehož hodnota může být libovolná,
ale musí být zadána.

INSTRUKCE RREG

Formát: RREG [APROM][,[XPROM][,[YPROM]
[,[SPROM]]]

Účel: Přiřazuje specifikovaným proměnným aktuální obsahy
registru procesoru A,X,Y a SR /stavový registr/ při zakon-
čení podprogramu strojového jazyka vyvolaného instrukcí SYS
/viz v kap. 5.2./

APROM, XPROM, YPROM, SPROM jsou libovolné numerické proměnné
nebo elementy pole.

Poznámky:

Tento instrukci mohou být, po ukončení podprogramu strojového
jazyka, přejat v obsahy registrů procesoru do programu Basicu.

FUNKCE R S P C O L O R

Formát: v=RESPCOLOR(n)

Účel: Vydává kód aktuální doplňkové barvy pro Sprite.

n je celočíselná hodnota 1 nebo 2 s následujícím významem:
1 Poskytuje kód barvy mezi 1 a 16 pro doplňk. barvu Spritu 1,
2 Poskytuje kód barvy mezi 1 a 16 pro doplňk. barvu Spritu 2.

FUNKCE R S P P O S

Formát: v=RSPPOS(n,m)

Účel: Poskytuje pro určitý Sprite pozici nebo aktuální rychlosť.

n je celočíselná hodnota mezi 1 a 8, která definuje prohledávaný Sprite.

m je celočíselná hodnota mezi 0 a 2 s následujícím významem:

0 poskytuje aktuální souřadnici x

1 poskytuje aktuální souřadnici y

2 poskytuje aktuální rychlosť v rozsahu mezi 0 a 15

FUNKCE R S P R I T E

Formát: v=RSPRITE(n,m)

Účel: Poskytuje určitému Spritu aktuální atributy.

n je celočíselná hodnota mezi 1 a 8, která definuje prohledávaný Sprite.

m je celočíselná hodnota mezi 0 a 5 s následujícím významem:

0 Poskytuje 1, je-li Sprite aktivován a 0, je-li dezaktivován.

1 Poskytuje barvu Sprite v rozsahu mezi 1 a 16.

2 Poskytuje 0, má-li Sprite prioritu nad pozadím, v opačném případě 1.

3 Poskytuje 1, je-li Sprite roztažen ve směru x, v opačném případě 0.

4 Poskytuje 1, je-li Sprite roztažen ve směru y, v opačném případě 0.

5 Poskytuje 1, je-li pro Sprite aktivní vícebarevný modus, v opačném případě 0.

PŘÍKAZ R U N

Formát: RUN [číslo řádku]

RUN jméno data [Ddisket. jednotka] [,Upřístr.adresa]

Význam: Bude spušten program Basicu , nacházející se buď v hlavní paměti nebo který bude nahrán z disku.

Číslo řádku Platné číslo programového řádku, u kterého má být program spuštěn.

Jméno data Znaková řetězcová konstanta v uvozovkách, nebo znaková řetězcová proměnná v závorkách, které označují jméno programového data.

Disketová jednotka Celocíselná hodnota 0 nebo 1. Přednastavena je zde 0.

Přístrojová adresa Celocíselná hodnota mezi 4 a 15, která označuje adresu přístroje připojeného floppy-disku. Přednastavena je zde hodnota 8.

Poznámky: Nezadá-li se číslo řádku, odstartuje se program nejnižším číslem řádku,

Příkazem RUN se před spuštěním programu smažou všechny příp. definované proměnné.

Specifikuje-li se v příkazu RUN údaj, potom se před zaváděním programu vymaže programová paměť. Mimo to se všechna příp. otevřená data uzavřou.

Příklad 1: 100 FOR I=1 TO 2:PRINT I↑ 2;:NEXT
 200 PRINT "HALLO!"
 RUN
 1 4HALLO!
 READY.
 RUN 200
 HALLO!
 READY

Příklad 2: RUN "PROG1",D0,U9

Program PROG1 se uloží z diskety v disketové jednotce 0 floppy-disku s přístrojovou adresou 9 a spustí se.

FUNKCE R W I N D O W

Formát: v=RWINDOW(n)

Účel: Poskytuje parametry pro aktuální obrazové okno /viz příkaz WINDOW/.

n je celočíselná hodnota mezi 0 a 2 s následujícím významem:

0 Poskytuje číslo posledního řádku v aktuálním obrazu /0-24/.

1 Poskytuje číslo posledního sloupce v aktuálním obrazu /0-79/

2 Poskytuje 40 nebo 80 podle nastavené šířky obrazu.

INSTRUKCE S C A L E

Formát: SCALE [n][,xmax,ymax]

Účel: Zapíná nebo vypíná nastavení měřítka v grafickém módu a mění měřítko obrazovkových souřadnic.

n je celočíselná hodnota 0 nebo 1 s následujícím významem:

0 vypíná nastavení měřítka. Grafické obrazovkové souřadnice x/y jsou 0-159/0-199 ve vícebarevném módu a 0-319/0-199 ve vysokorozlišitelném módu.

1 zapíná nastavení měřítka. Grafické obrazovkové souřadnice jsou nastaveny tak, jak je níže uvedeno.

xmax,ymax Největší možná x-ová ev. y-ová souřadnice při zapojeném nastavení měřítka. xmax musí ležet mezi 160 ev. 320 a 1023, ymax musí ležet mezi 200 a 1023. Zapojené nastavení měřítka má přednastavenou hodnotu 1023 pro obě souřadnice.

Poznámky: S instrukcí SCALE lze v grafickém módu zobrazit objekty na obrazovce v jiných měřítcích.

Příklad: SCALE 1

Obrazovkové nastavení měřítka se zapojí a ev. zobrazované objekty budou změněny, protože max. souřadnice v x- a y-ovém směru jsou nyní 1023.

INSTRUKCE SCNCLR

Formát: SCNCLR [n]

Účel: Maže udanou obrazovku.

- n celočíselná hodnota s následujícím významem:
- 0 40-ti znaková textová obrazovka se vymaže
 - 1 Vysokorozlišitelná grafická obrazovka se vymaže
 - 2 Vysokorozlišitelná rozdělená obrazovka /textové a grafické zobrazení/ se smaže.
 - 3 Smaže se vícebarevná grafická obrazovka
 - 4 Smaže se vícebarevná rozdělená grafická obrazovka /textové a grafické znázornění/
 - 5 Vymaže se 80-ti znaková textová obrazovka.

Při aktivované grafické obrazovce /viz instrukce GRAPHIC/ může být instrukce SCNSLR zadána bez parametru.

PŘÍKAZ SCRATCH

Formát: SCRATCH jméno data [,Ddisketová jednotka]
[,Upřístrojová adresa]

Účel: Maže datový záznam pro zadaná data ze seznamu diskety a tím i data z diskety /viz kap. 6.8./

Jméno data Znaková řetězcová konstanta v uvozovkách nebo znaková řetězcová proměnná o délce až 16 znaků, která označuje mažaná data.

Disketová jednotka Celočíselná hodnota 0 nebo 1, která popisuje disketovou jednotku. Přednastavena je zde hodnota 0.

Přístrojová adresa Celočíselná hodnota mezi 4 a 15 pro přístrojovou adresu připojeného floppy-disku. Přednastavena je zde hodnota 8.

Poznámky:

Příkaz SCRATCH vyžaduje od uživatele otázkou ARE YOU SURE? potvrzení k provedení příkazu /odpověď Y jako YES/. Při každé jiné odpovědi bude příkaz ignorován.

Příkaz SCRATCH - dokračování

Poznámky:

Použijí-li se ve jménu data Joker-znaky * a ? /viz kap.6.3/ může tak být jedním příkazem smazáno více dat.

Zadí-li se příkaz scratch v přímém módu, pak hlásí úspěšný výmaz:

01, FILES SCRATCHED,nn,00

přičemž nn udává počet vymazaných dat.

Příklady: SCRATCH"DATEN"

Údaj DATEN se vymaže na disketě v disketové jednotce 0 z floppy-disku s přístrojovou adresou 8.

 SCRATCH"*",U9

Všechna data z diskety v disketové jednotce 0, floppy-disku s přístrojovou adresou č.9 se vymažou.

 SCRATCH"????.PRG",D1

Všechna data, jejichž 8 znaků dlouhá jména končí na .PRG se smažou z diskety v disketové jednotce 1 / u dvojitě disketové jednotky/z floppy-disku s přístrojovou adresou 8.

 SCRATCH(AZ),D(LW),U(GA)

Zde byly pro všechny parametry použity proměnné.

PŘÍKAZ SLEEP

Formát: SLEEP n

Účel: Zastavuje provádění programu na určitou dobu.

n je celočíselná hodnota mezi 1 a 65535, která definuje čekací dobu v sekundách.

Poznámky:

Čas stanovený příkazem SLEEP nelze ovlivnit instrukcí FAST / viz tam/.

PŘÍKAZ SLOW

Formát: SLOW

Účel: Přepíná procesor z režimu ve 2 MHz taktu do režimu v 1MHz taktu. Počítač se tím o polovinu zdomalí.

Poznámky: Příkazem SLOW se zaktivuje 40-ti znaková obrazovková indikace /viz příkaz FAST/.

INSTRUKCE SOUND

Formát: SOUND hlas,frekvence,trytí [,,[směr]
[,,[maxfreq],[[stupen],[vlna]
[,,[šířka impulsu]]]]]

Účel: Vytváří tónové a zvukové efekty volitelné frekvence a trvání a vydává je přes reproduktory monitoru nebo televizního přijímače.

Hlas Celočíselná hodnota mezi 1 a 3 s následujícím významem:

- 1 Hlas 1 /tón/
- 2 Hlas 2 /Tón/
- 3 Hlas 3 /tón/

Frekvence Celočíselná hodnota mezi 0 a 65535, která udá frq.

Trvání Celočíselná hodnota mezi 0 a 32767, která udává dobu trvání zvukového efektu v časových taktech počítače /1/60 sek./ Hodnota 0 přeruší s okamžitým účinkem všechny příp. běžící tónové nebo zvukové efekty.

Směr Celočíselná hodnota mezi 0 a 2 s následujícím významem:

- 0 narůstající stupeň zvuku
- 1 klesající stupeň zvuku
- 2 oscilující stupeň zvuku

Přednastavena je zde hodnota 0.

Maxfrekv. Celočíselná hodnota mezi 0 a 65535 pro maximální frekvenci narůstajících zvukových efektů. Přednastavena je zde 0.

Instrukce SOUND - pokračování

Stupeň Celočíselná hodnota mezi 0 a 32767, která udává dobu trvání stupňovaných zvukových efektů.

Vlna Celočíselná hodnota mezi 0 a 3 která udává tvar vlny.

- 0 trojúhelník
- 1 pilovitý tvar
- 2 obdélník
- 3 šum

Přednastavena je zde hodnota 2.

Šířka impulsu Celočíselná hodnota mezi 0 a 4095. Přednastavena je zde hodnota 2048 / 50% střída/.

Poznámky: Zadá-li se instrukce SOUND s určitým hlasem, pokud je ještě aktivní předtím zadaná instrukce SOUND s tím samým hlasem, provede se nejprve první hlas až do konce dříve, než začne druhý.

Příklad: SOUND 2,800,3600

Hraje notu hlasu 2 při hodnotě frekvence 800 po iccu 1 minutv.

INSTRUKCE S P R C O L O R

Formát: SPRCOLOR[doplňk.barva1][,doplňk.barva2]

Účel: Určuje doplňkovou barvu ve vícebarevném módu pro všechny Sprite.

Doplňková barva 1 Celočíselná hodnota mezi 1 a 16, která určuje doplňkovou barvu 1 pro všechny Sprite ve vícebarevném módu.

Doplňková barva 2 Celočíselná hodnota mezi 1 a 16, která udává doplňkovou barvu 2 pro všechny Sprite ve vícebarevném módu.

PŘÍKAZ S P R D E F

Formát: SPRDEF

Účel: Zapíná Sprite-editor

PŘÍKAZ SPRDEF - pokračování

Poznámky: Tento příkaz je povolen jak v přímém, tak i v programovém módu. Sprite - editor rozeznává následující příkazy:

1 až 8 Volí číslo Sprite

A Zapíná nebo vypíná automatické opakování pro klávesy znaků 1,2,3 a 4.

Klávesy CRSR Řídí kurzor přes Sprite-rastr.

Klávesa RETURN Umísťuje kurzor na další řádek Sprite-rastru.

SHIFT-RETURN Ukládá navržený Sprite. Stiskne-li se bezprostředně potom klávesa RETURN, ukončí se Sprite - editor a interpret se vrátí do přímého módu.

Klávesa HOME Umísťuje kurzor do levého horního rohu Sprite-rastru.

CLR -klávesa Maže celý Sprite-rastr.

1 až 4 Ve vysokorozlišitelném grafickém módu kreslí 1 obrazový bod - Sprite v aktuální barvě pozadí a 2 oborazový bod - Sprite v aktuální barvě popředí na pozici kurzoru. Ve vícebarevném módu mají klávesy 1 a 2 tu samou funkci jako ve vysokorozlišitelném grafickém módu. Klávesa 3 kreslí obrazový bod - Sprite v doplňkové barvě 1 a klávesa 4 v doplňkové barvě 2 vícebarevného módu. V tomto případě budou stále kresleny dvějité obrazové body, poněvadž ve vícebarevném módu je k dispozici v x-ovém směru jen poloviční obrazová rozlišitelnost. Každý nakreslený obrazový bod v rastru se bude opakovat vpravo v originálním zobrazení aktuální obrazovkové rozlišitelnosti, abychom měli kontrolu nad skutečnou velikostí navrhovaného Sprite.

CTRL 1 až CTRL 8 Volí pro Sprite jednu z osmi barev popředí.

C 1 až C 8 /klávesa Commodore/ Volí jednu z osmi barev popředí 9 až 16.

Klávesa STOP Maže všechny zadané Sprity. Stiskne-li se bezprostředně po klávese STOP klávesa RETURN, ukončí se Sprite - editor a zaktivuje se opět přímý modus.

Příkaz SPRDEF - vokračování

X roztahuje Sprite ve směru x

Y roztahuje Sprite ve směru y

M volí pro Sprite vícebarevný modus 1 nebo 2.

Poznámky: Vyvolá-li se Sprite-editor, nainstaluje se v levém horní části obrazovky rastr o 21x24 znakových pozicích. K nakreslení obrazového bodu se nařídí Sprite-kurzor /znak plus/ do zvolené polohy ve Sprite-rastru a potom se podle nastaveného grafického módu stisknou klávesy 1,2,3 nebo 4. Chcete-li uložit na disketu Vámi navržené Sprity, postupujte v přímém módu následovně:

BSAVE"jméno",B0,P3584 TO P4095

K nahrání takovýchto dat zadejte pouze:

BLOAD"jméno".

S pomocí monitoru musíte k uložení na kazetu nebo na disketu zadat:

S"jméno",přístroj,0E00,0FFF

a k nahrání

L"jméno",přístroj

Za jméno se zadá jméno data a za přístroj přístrojová adresa v hexdecimálním tvaru.

INSTRUKCE S P R I T E

Formát: SPRITE n[,,[akt][,[barva][,[Pri]
[,[xroztažení][,[yroztažení]
[[modus]]]]]]]

Účel: Stanoví atributy pro jeden z osmi Sprite.

n je celočíselná hodnota mezi 1 a 8 která udá číslo Sprite.

akt Celočíselná hodnota 0 nebo 1, která udává, je-li Sprite aktivován /1/ nebo dezaktivován /0/.

barva Celočíselná hodnota mezi 1 a 16 pro barvu Sprite.

Přiřazení barevných kódů a barev je popsáno v instrukci COLOR.

Instrukce SPRITE - pokračování

Pri Celočíselná hodnota 0 nebo 1, která určuje, zda má Sprite při pohybu mezi objekty, které jsou k dispozici, probíhat před /0/ nebo za /1/ těmito objekty.

Xroztažení Celočíselná hodnota 0 nebo 1, která udává je-li Sprite roztažen /1/ v x-ovém směru, nebo ne /0/.

yroztažení Celočíselná hodnota 0 nebo 1, která udává, je-li Sprite roztažen /1/ v y-ovém směru, nebo ne /0/.

Modus Celočíselná hodnota 0 nebo 1, která udává, má-li být Sprite indikován ve vysokorozlišitelném /0/ nebo ve vícebarevném /1/ módu.

Poznámky: Nezadané parametry značí, že popř. dříve stanovené ~~parametry~~ atributy Sprite zůstanou zachovány.

Vysokorozlišitelný nebo vícebarevný Sprite - modus nesmí být zároveň nován s vysokorozlišitelným ev. vícebarevným grafickým modelem. Vysokorozlišitelné sprity mohou být použity ve vícebarevném grafickém módu a naopak.

Ko zjištění atributu pro určitý nastavený sprite lze použít funkci RESPRITE /viz tam/.

Příklad: SPRITE 4,1,6,1,,,1

Zaktivuje se Sprite č.4. Bude se indikovat ve vícebarevném módu v barvě zelené a má prioritu nad indikovanými obrazovkovými daty.

INSTRUKCE S P R S A V

Formát: SPRSAV n,v\$
SPRSAV v\$,n

Účel: Ukládá určitý Sprite jako bodový rastr do znakové řetězcové proměnné nebo naopak.

n Celočíselná hodnota mezi 1 a 8 pro číslo Sprite.

Instrukce SPRSAV - pokračování

v\$ Libovolná znaková řetězcová proměnná, která přijímá nebo obsahuje bodový rastr jako binární informaci.

Příklady: SPRSAV 1,A\$

Přenáší bodový rastr ze Sprite 1 jako binární informaci do znakové řetězcové proměnné A\$.

SPRSAV B\$,4

Přenáší hodnotu znakové řetězcové proměnné B jako binární bodový rastr do Sprite č.4.

INSTRUKCE SSHAPE

Formát: SSHAPE znakový řetězec,x1,y1[x2,y2]

Účel: Přenáší obsah zadaného rozsahu na grafické obrazovce jako binární obrazovou informaci do řetězcové proměnné.

Znakový řetězec Řetězcová proměnná, která přijímá zvolený rozsah obrazovky jako binární obrazovou informaci.

x1, y1 Obrazovkové souřadnice s nastaveným měřítkem levého horního rohu znázorňovaného obrazu.

x2,y2 Obrazovkové souřadnice s nastaveným měřítkem v pravém dolním rohu znázorňovaného obrazu. Přednastavena je zde aktuální pozice grafického kurzoru./Pixel-kurzor/

Poznámky: Vzhledem k tomu, že znakové řetězce Basicu jsou ohraničeny maximální délkou do 255 znaků, je také velikost ukládaného grafického kózsahu omezena.

Podle aktivovaného grafického módu /vyšokorozlišitelný nebo vícebarevný/ lze následovně spočítat délku znakuvého řetězce:

$$L(m) = \text{INT}((\text{ABS}(x_1 - x_2) + 1)/4 + .99)^*$$
$$(\text{ABS}(y_1 - y_2) + 1) + 4$$

$$L(h) = \text{INT}((\text{ABS}(x_1 + x_2) + 1)/8 + .99)^*$$
$$(\text{ABS}(y_1 - y_2) + 1) + 4$$

PŘÍKAZ S T A S H

Formát: STASH Byte, Intadr, Extadr, Banka

Účel: Přenáší volitelný počet bytů z pracovní paměti Basicu do rozšířené paměťové banky.

Byte Celočíselná hodnota mezi 0 a 65535, která udává počet přenášených Byte.

Intadr Celočíselná hodnota mezi 0 a 65535, která udává adresu v pracovní paměti Basicu, od které mají být Byty v paměti přeloženy.

Extadr Celočíselná hodnota mezi 0 a 65535, která udává adresu v datové základně, od které mají být Byty ukládány.

Banka Celočíselná hodnota mezi 0 a 7, která udává datovou základnu, ve které mají být Byty ukládány.

Poznámky:

Tento příkaz může být proveden jen tehdy, je-li je počítací připojen Ram-disk. Ram-disk dekóduje odpovídající příkaz a provádí jej.

Parametr Intadr se vztahuje na banku, která byla nastavena příkazem BANK. Před nastavenou je Banka 15.

Mají-li být uložena např. data z banky 0 do Ram-disku, je třeba udat před příkazem STASH "BANK 0".

Příklad: STASH 1000,52000,2000,7

Přenáší 1000 Byte od adresy 52000 z pracovní paměti Basicu do datové základny 7 od adresy 2000.

POZOR!!

K provádění tohoto příkazu potřebujete nástrčný modul Ram-disk.

PŘÍKAZ SWAP

Formát: SWAP Byte, Intadr, Extadr, Banka

Účel: Mění volitelný počet Bytů mezi pracovní pamětí Basicu a rozšířenou paměťovou bankou.

Byte Celočíselná hodnota mezi 0 a 65535, která udává počet vyměňovaných Byte.

Intadr Celočíselná hodnota mezi 0 a 65535, která udává adresu v pracovní paměti Basicu, od které mají být Byte vyměňovány.

Extadr Celočíselná hodnota mezi 0 a 65535, která udává adresu v datové základně /paměťová banka/, od které mají být Byte vyměňovány.

Banka Celočíselná hodnota mezi 0 a 7, která udává datovou základnu, se kterou mají být Byte vyměňovány.

Poznámky: Tento příkaz může být proveden jen tehdy, je-li na počítač připojen Ram-disk. Ram-disk dekóduje příslušný příkaz a provádí jej.

Parametr Intadr se vztahuje na banku, která byla nastavena příkazem BANK. Přednastavena je banka 15.

Mají-li být uložena např. data z banky 0 do Ram-disku, je nutné před udat před příkazem STASH, 'BANK 0'.

Příklad: SWAP 1000,52000,2000,7

Vyměňuje 1000 Bytů od adresy 52000 v pracovní paměti Basicu se stejně velkým rozsahem v datové základně 7 od adresy 2000.

P O Z O R !!!

K provádění tohoto příkazu, potřebujete nástrčný modul Ram - disk.

INSTRUKCE T E M P O

Formát: TEMPO n

Účel: Stanoví tempo hry hudebních not.

n Celočíselná hodnota mezi 0 a 255, která udává relativní dobu trvání, po kterou mají být hráné hudební noty podrženy. Skutečná doba trvání se vypočte podle:

Doba trvání = $19.22/n$ x a je udaná v sekundách.

Přednastavena je hodnota 8. Při hodnotě 0 se nota trvale podrží.

INSTRUKCE T R A P

Formát: TRAP [číslo řádku]

Účel: Zaktivuje se chybové přerušení a program odskočí k zadanému číslu řádku v momentě, kdy chce interpret vydat chybové hlášení.

Číslo řádku Platné číslo programového řádku, kterým začíná standardní program na zpracování chyb. Nezadá-li se číslo řádku, dezaktivuje se tím chybové přerušení.

Poznámky: Při aktivovaném chybovém přerušení se rozvětví program při všech podmírkách chyb včetně stisku klávesy STOP.

Při výskytu chyby se umístí chybová značka, program se rozvětví k zadanému číslu řádku a tam provede zpracování chyby.

Číslo chybného řádku lze zjistit pomocí EL-systémové proměnné /viz tam/, kód chyby pomocí ER-systémové proměnné a text chybového hlášení pomocí funkce ERR\$ (ER).

S pomocí instrukce RESUME /viz tam/ hlavní program pokračuje.

POZOR!!!

Chyba uvnitř standardního programu na zpracování chyb nemůže být zachycena instrukcí TRAP.

PŘÍKAZY T R O N a T R O F F

Formát: TRON
TROFF

Účel: Zapojuje /TRON/ nebo odpojuje /TROFF/ sledování průběhu programu.

Poznámky: Při zapnutém sledování průběhu programu se na obrazovce indikují čísla řádků v hranatých závorkách v pořadí jejich zpracování interpretem.

Příkazy TRON a TROFF slouží ke krokovému testování programu Basicu.

INSTRUKCE V O L

Formát: VOL n

Účel: Stanovuje hlasitost tónových a zvukových efektů, vytvářených instrukcemi SOUND a PLAY /viz tam/.

n Celočíselná hodnota mezi 0 a 15, která určuje hlasitost.
0 znamená vypnutí tónových a zvukových výstupů.

Poznámky: Instrukce VOL platí pro všechny volitelné hlasby /viz instrukce SOUND a PLAY/

INSTRUKCE W I D T H

Formát: WIDTH n

zrušení

Účel: Stanovuje tloušťku čáry pro všechny grafické instrukce ~~pro kresbu~~ /BOX, CIRCLE, DRAW atd./

n Celočíselná hodnota 1 pro jednoduchou a 2 pro dvojitou tloušťku čáry.

INSTRUKCE W I N D O W

Formát: WINDOW xlo,ylo,xru,yru[,mazání]

Účel: Nastavuje obrazové okno pro textový výstup a dle volby jej vymazává.

INSTRUKCE WINDOW - pokračování

Účel:

Formát:

xlo,ylo Číslo sloupce a řádku levého horního rohu textového okna.

xru,yru Číslo sloupce a řádku pravého dolního rohu textového okna.

mazání Celočíselná hodnota 0 nebo 1, která udává, má-li /1/ být po definici obrazové okno smazáno, nebo ne/0/.

Přednastavena je 0.

FUNKCE X O R

Formát: v=XOR(n1,n2)

Účel: Poskytuje celočíselnou hodnotu v rozsahu mezi 0 a 65535, která plyně z exclusiv-NEBO-spojení dvou celých 16-ti bitových hodnot bez znaménka.

n1,n2 Dvě celočíselné hodnoty v rozsahu mezi 0 a 65535, které se interně přemění v 16-ti bitové číslo dříve, něž spoju vydou logicky spojeny.

Příklady: PRINT XOR(124,12);XOR(0,12)

112 12

READY.

4.7 BARVY, SPRITES a GRAFICKÉ EFEKTY

C128 obsahuje 14 nových příkazů, které Vám ulehčí vytváření grafik.

BOX	CHAR
CIRCLE	DRAW
COLCR	LOCATE
GRAPHIC	PAINT
MOVSPR	GSHAPE
SCALE	SSHAPE
SPRSAV	SPRITE

Přesnější informacek těmto příkazům najdete v kap. 4.6.-příkazy.

GRAFICKÉ ZVLÁŠTNOSTI

Vynikající známkou C 128 je jeho grafická programovatelnost. Může vytvářet 16 různých barev /viz tab.dole/. Mimo to obsahuje ještě 5 různých grafických modifikací a může řídit 8 různých pohyblivých objektů, zvaných Sprites.

Proveditelné jsou následující barevné kódy:

1 černá	9 světlehnědá
2 bílá	10 hnědá
3 červená	11 růžová
4 zelená	12 tmavěšedá
5 fialová	13 šedá
6 tmavě zelená	14 světle zelená
7 modrá	15 světle modrá
8 žlutá	16 světle šedá

Volitelné jsou následující grafické modifikace:

- 0 text se 40 znaky na řádku
- 1 vysokorozlišitelná grafika /320x200 bodů/
- 2 vysokorozlišitelná grafika, rozdělená obrazovka na grafiku a text
- 3 vícebarevná grafika /160x200 bodů/
- 4 vícebarevná grafika, rozdělená obrazovka na grafiku a text
- 5 text s 80-ži znaky na řádek

VYSOKOROZLIŠITELNÁ a VÍCEBAREVNÁ GRAFIKA

Jak jsme se již dříve zmínili, patří ke zvláštnostem C 128 dva různé grafické provozní módy, totiž provoz s vysokorozlišitelnou a vícebarevnou grafikou, které lze volit instrukcí GRAPHIC.

U obou způsobů provozu / jako také u smíšeného provozu s rozdelenou obrazovkou/ je rezervován rozsah paměti o celkem 9 ev. 10 KBytech. Z tohoto rozsahu je pro grafickou obrazovku obsazeno 8000 Bitů.

Protože každý Byte sestává z 8 Bitů, má tedy grafika k dispozici dohromady 64000 Bitů. Každému z těchto Bitů je přiřazen jeden obrazový bod grafiky / $320 \times 200 = 64000$ obrazových bodů/. Proto se také hovoří o Bit-mapped - paměti.

Každý Bit může přijmout dva stavy, 0 a 1. 0 znamená vyp., a 1 znamená zap.

"Základní kámen" /VIC=Video Interface Controller/, který nyní řídí všechny procesy na obrazovce, stejně jako reprodukci barev, dává k dispozici barvu pro popředí i pro pozadí, které jsou volitelné ze stupnice 16 různých barev.

Vzhledem k tomu, že u vysokorozlišitelné grafiky je každý obrazový bod zobrazen jedním Bitem, může přijmout také jen dvě různé barvy - a to barvu pozadí je-li bit 0/vyp/ nebo barvu popředí, je-li Bit 1 /zap./. U vysoké rozlišitelnosti jsou tedy možné pouze 2 barvy.

U vícebarevné grafiky je rozlišitelnost redukována na $160 \times 200 = 32000$ obrazových bodů. Tím jsou v paměti o 64000 obrazových bodech pro každý obrazový bod k dispozici 2 Bity, které dovolují společně čtyři různé stavy:

00 01 10 11

Tím může nyní jeden obrazový bod vedle barvy pozadí a barvy popředí přijmout ještě dvě další barvy /doplňkové barvy 1 a 2/. U vícebarevné grafiky lze tedy pracovat dohromady se 4 barvami.

Instrukcí COLOR, jejíž účinek je dále ještě blíže objasněn, se určí, které barvy budou použity. Touto instrukcí se přidělí každému ze 4 barevných zdrojů /pozadí, popředí, dodatek/ barvy 1 a 2/existujících v grafickém provozu, jedna z 16 možných barev. Tyto zdroje barev a tím i jim přiřazené barvy jsou opět parametry grafických instrukcí znakových EOK, CHAR, CIRCLE, DRAW a PAINT.

VYPRACOVÁNÍ GRAFICKÝCH PROGRAMŮ V MÓDU C 128

S výše popsanými základy již nemůže být tak těžké sestavit vlastní grafické programy pro C 128.

Budete-li v tomto oddíle postupovat krok po kroku, jak je uvedeno, můžete na konci sám sepsat grafický program.

1. Nejdříve musíte zvolit barvy pro popředí a pozadí, stejně jako pro rámeček. K tomu slouží následující příkaz:

COLOR zdroj barvy, barevný kód

Zdroj barvy označuje tu část obrazovky, kterou chcete barevně změnit. Barevný kód je číslo, kterým určíte barvu zvoleného barevného zdroje /viz tab. dole/.

Kód	Zdroj
0	Pozadí /40-ti znaková indikace/
1	Popředí
2	Vícebarevný modus 1
3	Vícebarevný modus 2
4	Rámeček
5	Barva textu
6	Pozadí /80-ti znaková obrazovka/

Kód	Barva	Kód	Barva
1	černá	9	světle hnědá
2	bílá	10	hnědá
3	červená	11	růžová
4	zelená	12	tmavě šedá
5	fialová	13	šedá
6	tmavě zelená	14	světle zelená
7	modrá	15	světle modrá
8	žlutá	16	světle šedá

Pro příklad s černým pozadím, červeným popředím a černým rámečkem vyzkoušejte následující program:

```
10 COLOR 0,1  
20 COLOR 1,3  
30 COLOR 4,1
```

2. Další krok určuje grafický modus.

GRAPHIC Modus[, [vymaz], [textový řádek]]

Modus musí být číslo mezi 0 a 5. Tabulka dole ukazuje, která čísla odpovídají právě platnému grafickému módu.

Výmaz je celočíselná hodnota 0 nebo 1, která při vyvolání grafického módu 1-4 určuje, má-li být obrazovka vymazána/1/ nebo ne /0/.

Textový rádek je celočíselná hodnota mezi 0 a 24 /platí jen pro modifikace 2 a 4/, která určuje řádek, u kterého má začínat textová část obrazovky. Přednastavena je hodnota 19.

Modus	Význam
0	text 40 znaků/řádek
1	vysokorozlišitelná grafika /320x200 bodů/
2	vysokorozlišitelná grafika /rozdělená obrazovka na grafiku a text/
3	vícebarevná grafika /160x200 bodů/
4	vícebarevná grafika /rozdělená obrazovka na grafiku a text/
5	text 80 znaků/řádek

Následující příklad pracuje s vysokorozlišitelnou grafikou.

Zadejte následující programový řádek:

40 GRAPHIC 1,1

Druhé číslo 1 v grafickém příkazu GRAPHIC maže obrazovku. Nechcete-li, aby se obrazovka vymazala, zadejte na druhém místě 0 /nulu/.

POZOR!

Pracujte pouze s jednou obrazovkou. Pokud zůstanete "viset" kvůli např. SYNTAX ERROR ve vysokorozlišitelné grafice, stiskněte funkční klávesu f1 /GRAPHIC/, potom 0 a poté RETURN. Pracujete-li se 40-ti znakovou obrazovkou a máte-li aktivovanou grafickou obrazovku, způsobí to automatický návrat do textového módu.

3. Nyní můžete začít se zadáváním Vašich grafických příkazů.

Nejdříve musíte umístit grafický kurzor /Pixel-kurzor/ na to místo obrazovky, na kterém chcete začít s kresbou. K tomu slouží instrukce LOCATE.

Zadejte tedy následující programový řádek:

50 LOCATE 150,130

Tato instrukce umístí grafický kurzor na obrazovce./oproti textovému kurzoru není grafický kurzor viditelný/.

150 je x-ová souřadnice /horizontální/ a 130 je y-ová souřadnice /vertikální/ na obrazovce. x - ová souřadnice sahá od 0 do 319 a y-ová od 0 do 199 /v obrazových bodech/.

Aktuální pozici grafického kurzoru zjistíte pomocí funkce RDOT /viz fce v kap. 4.6/

4. Nyní nakresleme kružnici

Zadejte následující programový řádek:

60 CIRCLE 1,150,130,40,40

První číslo /1/ znamená, že bude kružnice kreslena v barvě popředí. 0 zde způsobí, že bude kružnice kreslena v barvě pozadí, tedy vymaže se.

Příkaz způsobí dále, že bude kružnice kreslena ve středu obrazovky. Instrukce CIRCLE zná 9 různých parametrů, tím můžete kreslit nejrozličnější kruhy a geometrické obrazce.

Co značí ostatní čísla, zjistíte v popisu tohoto příkazu v kap. 4.6/

5. Vyzkoušejte si teď nakreslit ~~nejdálejší~~.pravouhlý rovnoběžník

Zadejte:

80 BOX 1,20,100,80,160,,1

Tato instrukce způsobí, že se vlevo od kružnice nakreslí čtverec. Instrukce BOX /viz v kap.4.6/ obsahuje sedm parametrů, které Vám umožní kreslit různé pravoúhlé rovnoběžníky.

První hodnota může být 0 nebo 1. Je-li 1, nakreslí se čtverec v barvě popředí, při 0 v barvě pozadí, tedy vymaže se.

2.a3. hodnota jsou x- a y-ové souřadnice levého horního rohu.

4.a5. hodnota jsou x- a y-ové souřadnice pravého dolního rohu.

Poslední hodnota může být 0 nebo 1. 0 znamená, že se nakreslí jen orámování. 1 pravoúhlý rovnoběžník vyplní.

Nyní změňte barvu popředí a nakreslete další čtverec s níže uvedenými souřadnicemi.

Napište tedy následující programové řádky:

90 COLOR 1,9:REM ZMENA BARVY V POPREDI

100 BOX 1,220,100,280,160

6. Pomocí instrukce DRAW /víz kap. 4.6/ lze kreslit na obrazovce také jednotlivé čáry /linie/. Následující instrukce potáhne čáru pod již dříve nakreslenými obrazci:

120 DRAW 1,20,180 TO 280,180

Teď jen ve zkratce, co jednotlivá čísla znamenají. První hodnota 1 znamená barvu popředí. 20 a 180 jsou x- a y-ové souřadnice pro výchozí bod. Hodnoty 280 a 180 jsou x- a y-ové souřadnice koncového bodu. Vyzkoušejte sami některé z instrukcí DRAW.

Instrukce DRAW se dá také měnit tak, že kreslené čáry mění svůj směr.

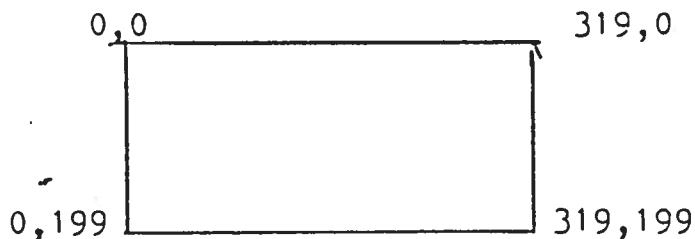
Vyzkoušejte následující programový řádek:

130 DRAW 1,10,20 TO 300,20 TO 150,80 TO 10,20

Uvidíte, že počítač teď nakreslí trojúhelník. Čtyři dvojice hodnot jsou právě platnými souřadnicemi tří stran. Je tedy možné kreslit různé matematické obrazce.

Snad Vás již napadlo, že při $x=0$ a $y=0$, se grafický kurzor nachází v levém horním konci počítače /HOME-pozice/.

Nákres Vám ukazuje uspořádání x- a y-ových souřadnic na obrazovce:



Existuje také možnost kresby jednotlivých bodů na obrazovce.

Zadejte následující programový řádek:

150 DRAW 1,150,190

Tento instrukcí se pod kružnicí nakreslí bod.

7. Instrukcí DRAW můžete také zvýraznit orámováním rozsahu obrazovky.

Pokud byste přece jen chtěli vymalovat plochu barvou, použijte instrukci PAINT /viz tam/.

Zadejte např.:

160 PAINT 1,150,97
maluje

Tento příkaz vykreslí kružnici, kterou jste kreslili v programovém řádku 60.

Nyní dva příkazy, které vymalují nahore nakreslený trojúhelník a čtverec napravo od kružnice:

180 PAINT 1,50,25

200 PAINT 1,225,125

Upozornění:

Při použití instrukce PAINT musíte zvolit výchozí bod uvnitř vymalovávaného obrazce, jinak se obrazec nevymaluje.

8. Také je možné indikovat na obrazovce text v grafickém módu.

Zadejte např.:

210 CHAR 1,10,1,"GRAFICKY PRIKLAD"

U tohoto příkazu se indikuje text od sloupce 10 v řádku 1.

9. Měřítka, ve kterém kreslí grafická instrukce na obrazovce, lze měnit pomocí instrukce SCALE.

Tím můžete na obrazovce současně zobrazit více obrazců.

Následně bude objasněna další možnost instrukce SCALE.

U vysokorozlišitelné grafiky je možných 320x200 bodů, ve vícebarevném módu však jen 160x200. Nevýhoda menší rozlišitelnosti se zmírní 2 doplnkovými barvami. Nyní máte tedy pro každý obrazový bod k dispozici 4 barvy.

SOUŘADNICE - NASTAVENÍ MĚŘÍTKA

Instrukce SCALE roztahuje grafický rozsah souřadnic v x-a y-ovém směru na 1024, nezávisle na tom, jestli se pracuje ve vysokorozlišitelném nebo vícebarevném módu.

SCALE 1 mění měřítko souřadnic /x/y od 0 do 1023/

SCALE 0 zpětně nastavuje měřítko souřadnic na přednastavené hodnoty /x/y od 0 do 319/199/.

Doposud popsané a zadané příkazové řádky tvoří program.

K výměně barev každému x, tímto programem, vytvořenému obrazci se vloží ještě řádky 70,110,140,170,190 a 210 s instrukcemi COLOR./viz seznam programu dole/.

Zadejte program a spusťte jej zadáním příkazu RUN /nezapomeněte na klávesu RETURN/.

Poté co jste si všechny příklady odzkoušeli, doporučujeme Vám nahrát si je na disketu nebo na kazetu, abyste je mohli později opět použít. Další informace o použití popsaných instrukcí Basicu najdete v kap. 4,6.

Příklad programu:

```
10 COLOR 0,1
20 COLOR †,3
30 COLOR 4,1
40 GRAPHIC 1,1
50 LOCATE 150,130
60 CIRCLE 1,150,130,40
70 COLOR 1,7:REM BARVA MODRA 1.CTVEREC
80 BOX 1,20,100,80,160,,1
90 COLOR 1,9: REM BARVA RAMECKU SVĚTLÉ MODRA2.CTVEREC
100 BOX,220,100,280,160
110 COLOR 1,10:REM BARVA HNEDA PRO CARY
120 DRAW 1,20,180 TO 280,180
130 DRAW 1,10,20 TO 300,20 TO 150,80 TO 10,20
140 COLOR 1,6:REM BARVA ZELENA PRO BOD
150 DRAW 1,150,190:REM KRESLI BOD POD KRUHEM
160 PAINT 1,150,97:REM VYPDNÍ KRUZNÍCI
170 COLOR 1,3:REM CERVENA BARVA PRO TROJUHelníK
180 PAINT 1,50,25:REM VYPDNÍ TROJUHelníK
190 COLOR 1,2:REM BILA BARVA PRO 2.CTVEREC
200 PAINT 1,225,125:REM VYPLNI 2.CTVEREC
210 COLOR 1,14:REM PISE TEXT BLEDE SEDY
220 CHAR 1,10,1,"GRAFICKY PRIKLAD"
```

SPRITES: PROGRAMOVATELNÉ A POHYBLIVÉ OBJEKTY

Váš C 128 Vám dále nabízí možnost, pohybovat samostatně navrhnutými obrazci po obrazovce. Lze to provést pomocí instrukce SPRITE.

K vytvoření Sprites nepotřebujete milimetrový papír, pokud pracujete se zabudovaným Sprite - editorem /viz příkaz SPRDEF v kap. 4.6/

Dále popíšeme, jak můžete Sprite navrhovat a používat.

Pokud jste již někdy pracovali s C 64, znáte již základní pojmy Sprite-grafiky. Sprite - grafika dovoluje počítači pohyb obrazců všech tvarů po obrazovce. Sprity mohou být kresleny v 16 různých barvách a každý Sprite může být zobrazen barevně. Návrh a použití Sprite dovolují následující instrukce:

SPRDEF	SSH!PE
GSHAPE	SPRSAV
SPRITE	MOVSPR
SPRCOLOR	COLLISION

NÁVRH SPRITE

Přetím, než začnete po obrazovce pohybovat, musíte si nejprve ujasnit, jak má Vámi zvolený Sprite vypadat /např. raketu nebo závodní vůz/

V módu C 128 můžete vytvořit Sprite 3 různými způsoby:

1. Instrukcemi SSHAPE, SPRSAV a SPRITE
2. Příkazem SPRDEF
3. Instrukcemi POKE jako u C 64 /viz kap. 5.3.1/.

POUŽITÍ INSTRUKcí SSHAPE, SPRSAV A SPRITE

Použití těchto instrukcí umožňuje vytvářet Sprites jako část grafiky rychlým a jednoduchým způsobem. Jednotlivé instrukce jsou popsány v kap. 4.6.

Zde popis jednoho z postupů návrhu Sprite:

- Nakreslete libovolný obraz za pomocí instrukcí DRAW, CIRCLE, BOX a PAINT z předešlé kapitoly. Vnější hranice této kresby se určí bodovou maticí 24x21 ve vysokorozlišitelném- nebo bodovou maticí 12x21 ve vícebarevném módu.
- K uložení tohoto obrazu do řetězcové proměnné použijte instrukce SSHAPE.

- Přeneste Váš návrh za pomocí instrukce SPRSAV do Spritegrafiky.
- Zaktivujte Sprite pomocí instrukce Sprite, vymalujte jej, zvolte barevný modus a Sprite zvětšete.
- Pohybujte /uveďte do pohybu/ Sprite pomocí instruce MOVSPR.

2. PŘÍKAZ SPRDEF

Příkaz SPRDEF je nejrychlejší a nejjednodušší způsob k vytváření Sprite. Příkaz SPRDEF zapíná Sprite - editor. Přesnější informace najdete v popisech příkazů v kap. 4,6. V dalším ^{vlastní} příkladě uvidíte, jak můžete navrhnout Sprite jako část grafiky.

Tento příklad pracuje s vysokorozlišitelnou grafikou. Pro přepínání do grafického módu při černém pozadí, zadejte následující programové řádky:

```
5 COLOR 0,1:REM POZADI CERNE  
10 GRAPHIC 1,1:REM VYSOKOROZLISITELNA GRAFIKA
```

Následující krátký program kreslí závodní vůz na levém horním okraji obrazu:

```
15 BOX 1,2,2,45,45:REM KRESLIPRAVOUHLY ROVNORBENIK OKOLO AUTA  
20 DRAW 1,17,10TO28,10TO26,30TO19,30TO  
17,10:REM KAROSERIE  
22 DRAW 1,11,10TO15,10TO15,18TO11,18TO  
11,10:REM LEVE HORNI KOLO  
24 DRAW 1,30,10TO34,10TO34,18TO30,18TO  
30,10:REM PRAVE HORNI KOLO  
26 DRAW 1,11,20TO15,20TO15,28TO11,28TO  
11,20:REM LEVE DOLNI KOLO  
28 DRAW 1,30,20TO34,20TO34,28TO30,28TO  
30,20:REM PRAVE DOLNI KOLO  
30 DRAW 1,26,28TO19,28:REM CHLADIC  
32 BOX 1,20,14,26,18,90,1:REM PREDNI SKLO  
35 BOX 1,150,35,195,40,90,1:REM ULICE  
37 BOX 1,150,135,195,140,90,1:REM ULICE  
40 BOX 1,150,215,195,220,90,1:REM ULICE  
42 DRAW 1,50,180TO300,180:DRAW1,50,180TO  
50,190:REM CILOVA CARA  
43 DRAW 1,300,180TO300,190:DRAW1,50,190TO  
300,190:REM CILOVA CARA  
44 CHAR 1,18,23,"C I L":REM PISE CIL
```

Nechejte napsaný program jednou proběhnout. V pravém horním rohu uvidíte bílé závodní auto v pravouhlém rovnoběžníku.

nakreslena

Mimo to je výrazná závodní dráha v dolním rohu i s cílovou čarou. Auto zatím není Spritem, protože mu chybí schopnost pohybovat se. K tomu je potřeba uložit do paměti právě nakreslený obraz do řetězcové proměnné.

Vložte proto do svého programu následující řádek:

45 SSHAPE A\$,11,10,34,31:REM ULOZENI OBRAZU DO RETEZCE

Čísla 11,10,34,31 jsou souřadnice levého horního a pravého dolního rohu řbrazu. Pazici souřadnic můsíte určit přesně, jinak hrozí nebezpečí, že Sprite nebude uložen správně anebo vůbec.

Proměnná A\$ ukládá binární znakový řetězec, který se skládá z nul a jedniček a který bude odložen do paměti počítače. Každému bodu na obrazovce je přiřazen jeden Bit v grafické paměti počítače. Je-li Bit zap. /1/, pak je bod ve zvolené barvě popředí viditelný. Je-li Bit vyp. /0/, má bod zvolenou barvu pozadí. Obraz je nyní uložen. V dalším kroku se obraz uložený do A\$ přenese do Sprite.

Lze to realisovat následujícími instrukcemi:

SPRSAV n,v\$

SPRSAV v\$,n

Tento instrukci můžete vzor Bitu uložený do řetězcové proměnné použít jako Sprite - data ev. vzor Bitu jednoho Sprite předat do znakové řetězcové proměnné.

n. Celocíselná hodnota mezi 1 a 8 pro číslo Sprite
v\$ Libovolná znaková řetězcová proměnná, která přejímá obsahuj binární tvar .

Následující instrukce SPRSAV odkládají obraz uložený v A\$ do dvou Sprites.

Proveďte následující doplnění k dosavadnímu programu:

50 SPRSAV A\$,1:REM UKLADA OBSAH A\$ DO SPRITE 1

55 SPRSAV A\$,2:REM UKLADA OBSAH A\$ DO SPRITE 2

Předtím, než můžete Sprity uvidět, musíte je nejprve zaktivovat.

K tomu slouží instrukce SPRITE, která může také určit velikost objektu.

Mámo to se určí instrukcí SPRITE grafický modus buďto jako vysokorozlišitelný nebo jako vícebarevný, stejně jako se definiuje barva popředí i barva pozadí.

Následující programové řádky zaktivují Sprity:

```
60 SPRITE 1,1,7,0,0,0,0:REM AKTIVUJE SPRITE 1  
65 SPRITE 2,1,3,0,0,0,0:REM AKTIVUJE SPRITE 2
```

V následujícím Vám objasníme jednotlivé parametry instrukce SPRITE.

SPRITE n,[akt][,barva][,Pri][,xroztažení]
 [,yroztažení][,[modus]]]]]

n Celočíselná hodnota mezi 1 a 8 pro číslo Sprite

akt Celočíselná hodnota 0 nebo 1, která udává, je-li Sprite aktivován /1/ nebo dezaktivován /0/.

Barva Celočíselná hodnota mezi 1 a 16 pro barvu Sprite. Přiřazení kódů k barvám je popsáno v kap. 4.6.-instrukce COLOR.

Pri Celočíselná hodnota 0 nebo 1. Při 0 postupuje Sprite během pohybu okolo objektu, který je na obrazovce # dispozici, při 1, za objektem.

xroztažení Celočíselná hodnota 0 nebo 1, která udává, je-li Sprite v x-ovém směru roztažen /1/ nebo ne/0/.

yroztažení Celočíselná hodnota 0 nebo 1, která udává, je-li Sprite v y-ovém směru roztažen/1/ nebo ne /0/.

Modus Celočíselná hodnota 0 nebo 1, která udává, má-li být Sprite indikován ve vysokorozlišitelném /0/ nebo ve vícebarevném /1/ módu.

Instrukce MOVSPR uvede nyní Sprite do pohybu. Umíšťuje nebo pohybuje Sprite na určité místo obrazovky. Jeho všeobecná forma je:

```
MOVSPR n,x,y  
MOVSPR n,+/-x1,+/-y1  
MOVSPR n,úhel # rychlosť
```

n Celočíselná hodnota mezi 1 a 8, která udává číslo Sprite.

x,y Absolutní souřadnice, na kterých má být Sprite umístěn.

+/-x1,+/-y1 Relativní souřadnice k poslední pozici Sprite, na kterých má být Sprite umístěn.

úhel Celočíselná hodnota mezi 0 a 360, která udává, pod jakým úhlem se má Sprite pohybovat.

Vložte, prosím, do programu následující řádky:

70 MOVSPR 1,240,0:REM POZICE SPRITE 1
80 MOVSPR 2,120,0:REM POZICE SPRITE 2

Řádek 70 způsobí, že se Sprite 1 bude pohybovat k pozici 240/0 systému bodových souřadnic.

Řádek 80 způsobí analogicky, že se Sprite 2 bude pohybovat k pozici 120/0 systému bodových souřadnic.

Následující programové řádky pohybují Sprites podél definované čáry:

85 MOVSPR 1,180#6:REM pohybuje Sprite 1 shora dolů
87 MOVSPR 2,180#7:REM pohybuje Sprite 2 shora dolů

První hodnota této instrukce je číslo Sprite. Druhá hodnota /180/ je úhel, pod kterým se bude Sprite ze svého výchozího bodu vzdalovat. Poslední číslo udává rychlosť, jakou se bude Sprite pohybovat.

Nechejte svůj program proběhnout a uvidíte, že se Vám podařilo naprogramovat závodní dráhu se dvěma závodními vozy.

Nechejte volný průběh své fantazii a vyzkoušejte vlastní objekty.

Následuje ještě jednou kompletní program:

```
5 COLOR 0,1
10 GRAPHIC 1,1
15 BOX 1,2,2,45,45
20 DRAW 1,17,10TO28,10TO26,30TO19,30TO17,10
22 DRAW 1,11,10TO15,10TO15,18TO11,18TO11,10
24 DRAW 1,30,10TO34,10TO34,18TO30,18TO30,10
26 DRAW 1,11,20TO15,28TO15,28TO11,28TO11,20
28 DRAW 1,30,20TO34,20TO34,28TO30,28TO30,20
30 DRAW 1,26,28TO19,28
32 BOX 1,20,14,26,18,90,1
35 BOX 1,150,35,195,40,90,1
37 BOX 1,150,135,195,140,90,1
40 BOX 1,150,215,195,220,90,1
42 DRAW 1,50,180TO300,180:DRAW 1,50,180TO50,190
43 DRAW 1,300,180TO300,190:DRAW 1,50,190TO300,190
44 CHAR 1,18,23,"CIL"
45 SSHAPE A$,10,10,33,31
50 SPRSAV A$,1
55 SPRSAV A$,2
60 SPRITE 1,1,7,0,0,0,0
65 SPRITE 2,1,3,0,0,0,0
70 MOVSPR 1,240,0
80 MOVSPR 2,120,0
85 MOVSPR 1,180#6
87 MOVSPR 2,180#7
```

- Řádek 5 Pozadí obrazovky se začerní
Řádek 10 Zapíná vysokorozlišitelnou grafiku a maže obrazovku
Řádek 15 Kreslí pravoúhlý rovnoběžník vlevo nahore.
Řádky 20-32 Kreslí závodní auto
Řádky 35-44 Kreslí závodní dráhu a cílovou čáru.
Řádky 50-55 Převádí část znakových řetězcových proměnných do Sprite 1 a 2.
Řádky 70-80 Pohybují Sprite k hornímu okraji obrazovky
Řádky 85-97 Pohybují Sprite shora dolů.

PRIPOJENÍ SPRITU K VĚTŠÍMU OBJEKTU

Chcete-li vytvořit na obrazovce větší obrazce, musíte připojit jednotlivé Sprite k želkovému zvolenému obrazci. Následující návod Vám ukáže, jak lze sloučit dva Sprite:

- Namalujte znova libovolný obraz za pomoci instrukcí DRAW, BOX a PAINT. Zdvojte však tentokrát velikost plochy na 48x21 bodů.
- Použijte 2 instrukce SSHAPE k jednotlivému uložení dvou Sprites. Nyní rozdělte plochu obrazu 48x21 bodů do dvou stejně velikých ploch /24x21/ a uložete Sprites odpovídajícími souřadnicemi. Použijte dvě různé znakové řetězcové proměnné /AS/.
- Přeneste data obrazu do samostatných Sprites pomocí příkazu SPRSAV.
- Zaktivujte Sprity pomocí instrukce SPRITE.
- Umístěte druhý Sprite tak, aby se kryl první bod druhého Sprite s posledním bodem prvního. Nyní jste vzájemně spojili Sprites.

Vyzkoušejte ještě následující programové řádky:

100 MOVSPR 1,10,10

První hodnota je číslo Sprite. Tato instrukce umísťuje první Sprite na pozici x=10/y=10. Nyní umístěte druhý Sprite 24 bodů napravo od prvního:

200 MOVSPR 2,34,10

Nyní je Vám umožneno pohybovat Spritem po obrazovce k jedné ze zvolených pozic.

Závěrem ještě příklad spojených Sprites:

```
10 COLOR 0,1:REM POZADI CERNE
20 GRAPHIC 1,1:REM MAZE OBRAZOVKU
30 BOX 1,1,1,51,25:REM PRAVOUHLY ROVNOREZNIK OKOLO SPRITE
40 DRAW 1,3,3TO40,3TO47,12TO0,19TO3,3
45 REM KRESLI RAKETU
50 CHAR 1,1,1,"RAKT":REM PISE NA RAKETE NAPIS RAKT
60 SSHAPE CZ,2,2,25,22:REM JISTI CAST 1 DO CZ
70 SSHAPE DZ,26,2,48,22:REM JISTI CAST 2 DO DZ
80 SPRSAV CZ,1:REM UKLADA CZ DO SPRITE 1
90 SPRSAV DZ,2:REM UKLADA DZ DO SPRITE 2
100 SPRITE 1,1,2,0,0,0,0:REM AKTIVUJE SPRITE 1
110 SPRITE 2,1,2,0,0,0,0:REM AKTIVUJE SPRITE 2
120 MOVSPR 1,80,100:REM PRESOUVA SPRITE 1
130 MOVSPR 2,104,100:REM PRESOUVA SPRITE 2
```

4.8. ZVUKY A HUDBA S C 128

C 128 má ve srovnání s jinými mikropočítači nejvýkonnější syntetizátor hudby a hluku. Základní prvek zvuku /SID-Chip/ uzpůsobuje C 128 k vytváření 3 různých tónů současně. Každý tón může být mimo to hrán ve čtyřech různých zvukových zábranách. Syntetizátor umožnuje programovatelnou dobu úhozu, dobu slábnutí, dobu držení a dobu dozvívání /Parametry ADSR/ a má schopnost určité tony zdůraznit nebo utlumit. Mimo to obsahuje SID programovatelný filtr, kterým je možno zvuky a hluk změnit.

K plnému využití možností základního prvku SID nabízí Basic 7.0 u C 128 celou řadu nových instrukcí. Tyto instrukce programování hudby zrychlí a zjednoduší.

K sestavení Vašich vlastních "Videoclips" můžete kombinovat grafické programy s vlastní hudbou. K tomu jsou zde tyto nové instrukce pro hudbu:

SOUND	ENVELOPE
VOL	TEMPO
PLAY	FILTER

Tak jako v předcházejících částech, objasníme Vám krátce každý jednotlivý příkaz. V průběhu tohoto popisu vznikne opět malý program.

INSTRUKCE SOUND

Vám umožní vytvořit rychle a jednoduše tónové nebo zvukové efekty volitelné frekvence a trvání a vydat je přes monitor nebo televizor:

SOUND hlas,frekvence,doba trvání[,,[směr][,[maxfr][,[stupen][,[vlna]],impr]]]]]

Jednotlivé parametry znamenají:

Instrukce SOUND - pokračování

Hlas Můžete volit 1 až 3

- 1 Hlas 1 /tón/
- 2 Hlas 2 /tón/
- 3 Hlas 3 /tón/

Frekvence Celočíselná hodnota mezi 0 a 65535 volí frekvenci.

Doba trvání říká, jak dlouho má být tón podržen / v krocích po 60tině sekundy/. Celočíselná hodnota mezi 0 a 32767.

Směr Celočíselná hodnota mezi 0 a 2.

Maxfr Celočíselná hodnota mezi 0 a 65535

Stupeň Celočíselná hodnota mezi 0 a 32767.

Vlná Celočíselná hodnota mezi 0 a 3 s následujícím významem:

- 0 trojúhelník
- 1 pilovitý tvar
- 2 pravouhlý rovnoběžník
- 3 šum

Přednastavena je hodnota 2.

Impbr Celočíselná hodnota mezi 0 a 4095. Určuje tlačítkový poměr /poměr kláves/.

Bližší údaje o instrukci SOUND najdete v kap. 4.6.

Chcete-li např. vytvořit tón o 1 sekundě, musí být hodnota doby trvání jedné sekundy 60, 10 sekund 600 atd.

Nyní, prosím, vyzkoušejte následující krátký program:

```
10 VOL 5  
20 SOUND 1,512,60
```

Nechejte program proběhnout. C 128 zahraje krátký tón.

Řádek 10 určuje hlasitost. Číslo 512 v řádku 20 určuje frekvenci.

Nyní změňte výšku tónu následujícím řádkem:

```
30 SOUND 1,1000,60
```

Slyšíte, že tón je podstatně vyšší než předešlý. Ovšem oba tóny budou hrány pouze jednu sekundu.

Vyzkoušejte následující:

```
40 SOUND 1,0,60
```

Uslyšíte nejhļubší tón, jaký je možný.

Nyní proběhněte všechny tónové výšky od 0 do 65535 v jednom tahu / FOR...NEXT/

```
50 FOR I=0 TO 65535  
60 SOUND 1,I,1  
70 NEXT
```

celý

Tento díl programu proběhne ~~konstantní~~ definovatelný frekvenční rozsah zespadu nahoru.

Nahraďte nyní řádek 60 řádkem následujícím:

```
60 SOUND 3,I,1,,,3
```

Nyní proběhne program celý rozsah generátoru šumu. Nízké frekvence vyznějí jako rachocení, vysoké frekvence jako startující raketa.

Vyzkoušejte následující variantu. Jedná se tu o program, který simuluje různými šumy leteckou bitvu.

```
10REM LETECKA BITVA  
20 VOL 5  
30 FOR I=850 TO 1015  
40 SOUND 3,I,1  
50 SOUND 1,1015-I,1  
60 NEXT  
70 SOUND 3,965,60,,,3  
80 FOR I=1023TO850 STEP-1  
90 SOUND 3,I,1  
100 SOUND 1,I-150,1  
110 NEXT  
120 FOR J=1000TO0 STEP-7  
130 SOUND 3,J,1  
140 NEXT  
150 SOUND 3,965,60,,,3  
160 GOTO 30
```

Stisknutím klávesy STOP můžete program ukončit.

Nyní objasníme jednotlivé příkazové řádky:

Řádek 20 nastavuje hlasitost na 5.

Řádky 30 až 60 způsobí vlastní hluk letadla. Jak vidíte a slyšíte, na řádku 30 výška tónu ubývá, u řádku 50 přibývá.

Řádek 70 vytvoří hluk kulometu.

Řádek 80-110 produkuje hluk sestřeleného letadla. Aby pád letadla realističtěji působil, nasadí se rozdílné tónové výšky /viz řádek 90 a 100/

Řádky 120-140 vytvoří hluk výbuchu letadla při nárazu.

Řádek 150 odpovídá řádku 70.

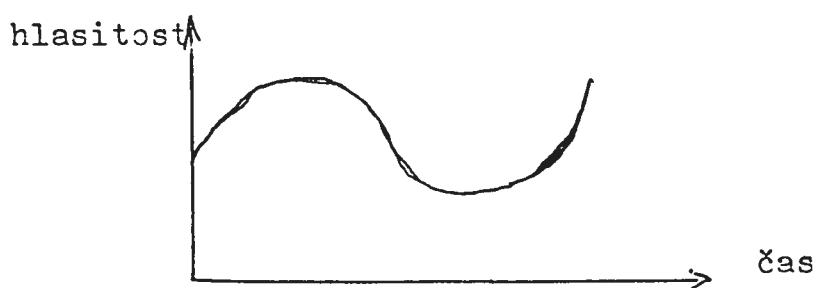
Řádek 160 umožní opakování programu od řádku 30.

Instrukce SOUND má význam hlavně při vytváření vedlejších efektů. Chcete-li hrát na C 128 písně, můžete použít jiné příkazy.

Hluk a hudba v módu C 128

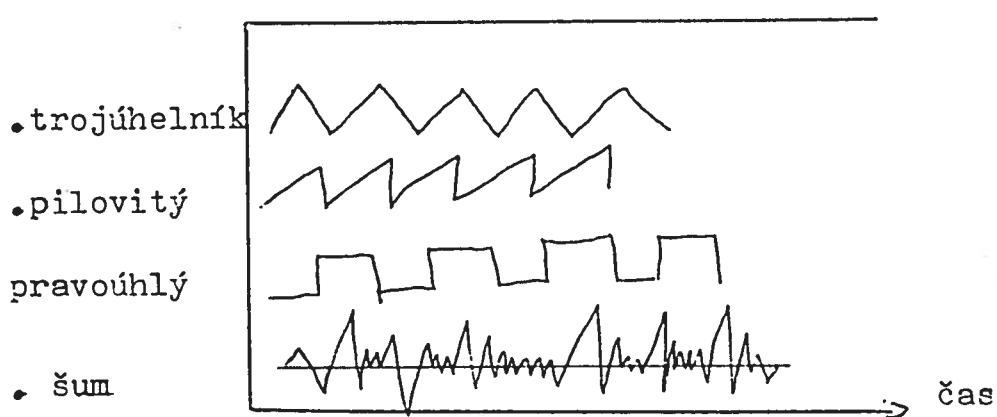
Úvodem nejnudnější základy:

Každý tón, který slyšíte je vlna, přenášená vzduchem. Tato tónová vlna se pohybuje určitou rychlostí /frekvence/.



Barva zvuku tónu je určena tvarem vlny.

C 128 může vytvářet 4 různé tvary vln:



VYTVAŘENÍ HUDBY S C 128

Ke změně barvy zvuku jednoho tónu můžete použít instrukci ENVELOPE. Tato instrukce kontroluje parametr ADSR a tvar vlny.

Všeobecný formát instrukce ENVELOPE je následovný:

ENVELOPE n,[An][, Ab][, Ha][, Au][, Wf]
[,[Ib]]])]

Jednotlivé parametry znamenají:

n číslo obalové křivky /0 až 9/

An doba úhozu /0 až 15/

Ab doba slábnutí /0 až 15/

Ha doba trvání /0 až 15/

Au doba doznívání /0 až 15/

Wf tvar vlny: 0=trojúhelník, 1=pilovitý 2=pravoúhlý; rovnoběž.
3=šum, 4=kruhová modulace

Ib šířka impulsu /0 až 4095/.

Nechejte nás, vysvětlit Vám jednotlivé pojmy jako je obalová křivka, tvar vlny a šířka impulsu.

Obalová křivka je souhra jednotlivých faktorů /parametr ADSR/ která vytváří průběh hlasitosti jednoho tónu / doba výhozu, doba slábnutí, doba držení a doba doznívání/. Kytara má např. jiné parametry ADSR jako flétna.

Pod pojmem tvar vlny rozumíme druh vln, které např. vytvářejí hudební nástroj. Skládají se z kombinací hudebních harmonií jednoho tónu. Harmonie je mnohonásobek základního tónu.

Šířky impulsu jsou změny pravoúhlého tvaru, přičemž se také mění poměr harmonií.

Vidíte tedy, že instrukce ENVELOPE umožňuje programování základních hudebních složek.

C 128 nabízí barvy zvuku 10 hudebních instrumentů, jejichž parametr ADSR je již přednastaven a který lze změnit instrukcí ENVELOPE. Stačí pouze zádat odpovídající číslo obalové křivky a C 128 poskytne přednaprogramované parametry sám.

Chcete-li vytvořit vlastní obalové křivky s novým tvarem zvuku, musíte změnit jednotlivé parametry instrukce ENVELOPE:

Další podrobnosti k přednaprogramovaným obalovým křivkám zjistíte z následující tabulky:

0	0	9	0	0	2	1536	klavír
1	12	0	12	0	1		harmonika
2	0	0	15	0	0		cirkusové varhany
3	0	5	5	0	3		buben
4	9	4	4	0	0		flétna
5	0	9	2	1	1		kytara
6	0	9	0	0	2	512	cembalo
7	0	9	9	0	2	2048	varhany
8	8	9	4	1	2	512	trumpeta
9	0	9	0	0	0		xylofon

Zadejte nyní následující programový řádek:

10 ENVELOPE 9,10,5,10,5,2,4000:PLAY" T9"

Další krok určí hlasitost Vašeho hudebního programu:

20 VOL 8

Instrukce VOL udává hlasitost mezi 0 a 15, přičemž 0 je minimum = vypnuto, a 15 je maximální hlasitost.

S instrukcí PLAY můžete hrát noty:

PLAY"znakový řetězec"

Znakový řetězec je znakový řetězec ze speciálních písmenových

příkazů, které určují vytváření tónů a sled zahrání. Význam jednotlivých parametrů je uveden v kap. 4,6.

Další krok kontroluje hrací tempo zvolených not. Pomůže Vám při tom instrukce TEMPO

Všeobecná forma instrukce:

TEMPO,,

n musí být celočíselná hodnota mezi 0 a 255. Chybí-li v programu tento příkaz, nastaví počítač pevnou délku 8. Skutečná doba trvání se vypočítá z:

doba trvání = $19.22/n$ v sekundách.

Nastavíte-li tempo /rychlosť, dobu/ = 0, nota se podrží.

Napište nyní následující programový řádek:

30 TEMPO 100

Nyní se naučíme, jak si můžete zahrát písničku. Děje se tak analogicky k příkazu PRINT, ve kterém zadáte v uvozovkách zvolené noty v instrukci PLAY.

Následnou instrukcí PLAY zahráje počítač stupnicí:

40 PLAY"C D E F G A B" Notě B odpovídá v češtině nota H.

Ted můžete před jednotlivé noty umístit následující znaky:

- nota bude zahrána o půltón výš

% - nota bude zahrána o půltón níže

. - prodlouží notu o polovinu její hodnoty

Mimo to můžete měnit dobu trvání tónů, přičemž umístíte před jednotlivé noty následující písmena:

W - Následující nota bude zahrána jako celá nota

H - Následující nota bude zahrána jako půlnota

Q - Následující nota bude zahrána jako čtvrtkōmá nota

I - Následující nota bude zahrána jako osminová nota

S - Následující nota bude zahrána jako šestnáctinová nota

Písmenem "R" umístěným v uvozovkách dosáhnete toho, že se tón podrží.

Dále je možné programovat syntetizátor C 12S doplňkovými řídicími znaky v instrukci PLAY pro různé efekty při hraní hudby.

Tyto znaky následují v tabulce:

Kontrolní znak	popis	rozsah	přednastavení
Un	hlasitost	0 - 9	9
Tn	obalová křivka	0 - 9	0
Vn	hlas	1 - 3	0
Xn	filtr	0=zap, 1=vyp.	0
On	oktáva	0 - 6	4

Normálně nepotřebujete řídící znaky vůbec používat. Můžete tím však značně vyčerpat možnosti syntetizátoru. Definujete-li v instrukci PLAY obalovou křivku s Tn, dbejte na to, že zvolená hodnota se týká obalové křivky, pro kterou jste užali ^{specifické} parametry instrukcí ENVELOPE.

Vzhledem k tomu, že má SID-Chip jen jeden filtr, používá se u všech tří hlasů.

Vložte následující programové řádku do svého programu a dávejte pozor na rozdíl s řádkem 40.

3. PLAY"U5 V1 05 C D E F G A B"

Tento řádek také zahráje stupnici, hlasitost je však snížena na 5, hlas 1 byl zvolen a noty budou zahrány o oktávě /5/ výše. Nyní hraje Váš program dva paralelní hlyasy.

3. hlas při zapnutém filtru dosáhnete následujícím řádkem:

60 PLAY"U7 V3 06 X1 C D E F G A B"

Vysvětlete si krátce jednotlivé znaky řádku 60.

U7 - nastaví hlasitost na 7

V3 - volí třetí hlas

06 - umístituje 3. hlas do 6.oktávy

X1 - ^{zapíná} ~~nastavuje~~ filtr.

Bližší výdaje najdete v kap. 4.6. instrukce PLAY.

Program nyní zahráje všechny tři hlyasy. Každý o oktávu výše než předešlý.

Až do tohoto momentu jsme hráli celé noty. V dalším příkladě programu by měly být hrány i noty půlkové, osminkové atd.

Změňte noty, přičemž vložíte výše jmenovaná písmena:

70 PLAY"U7 V2 06 X1 H CD E EF I GA S B"

Řádek 70 hraje v 6.oktávě, hlasem č.2 hlasitosti 7.Filtr zapnut.

Zde dlaší objasnění případných řídících znaků v instrukci PLAY.

C a D budou hrány jako poláviční noty

E a F budou hrány jako čtvrtkové noty

G a A budou hrány jako osminkové noty

B bude zahrána jako šestnáctinová nota.

Doplňte si teď noty, přičemž tóny zvýšíte o půl oktávy, snížíte o půl oktávy nebo tóny progloužíte.

80 PLAY"U8 V0 O4 X0 .H CD Q# EF IS GA .S#B"

C a D budou půlkové noty prodloužené o polovinu své hodnoty

E a F budou zahrány jako čtvrtkové noty o půl oktávy výše

G a A budou zahrány jako osminkové noty o půl oktávy níže

B se prodlouží o polovinu hodnoty a bude zahrána o půltón výš.

Jestliže jste již zvolili tvar vlny, ADSR-parametr, klasitost a tempa, měli byste zvolit filtr. Instrukce FILTER změní barvu a výšku zvuku.

Filtr zabudovaný v SID Vám dává možnost zdůraznit nebo eliminovat tvar vlny.

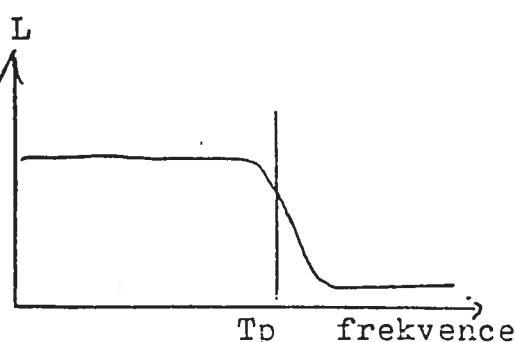
Tím mění filtr barvu zvuku tónu.

SID-Chip může simulovat 3 různé typy filtru:

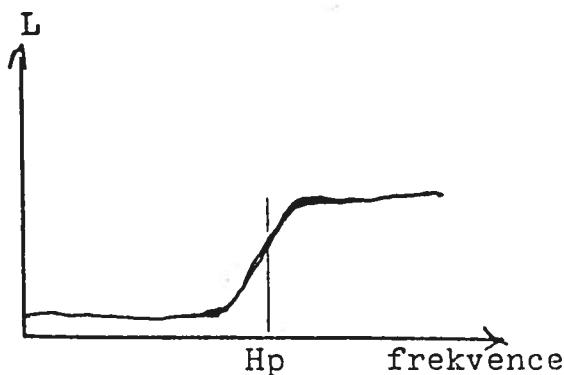
1. Filter s nízkokmitočtovou propustí

2. Filtr s vysokokmitočtovou propustí

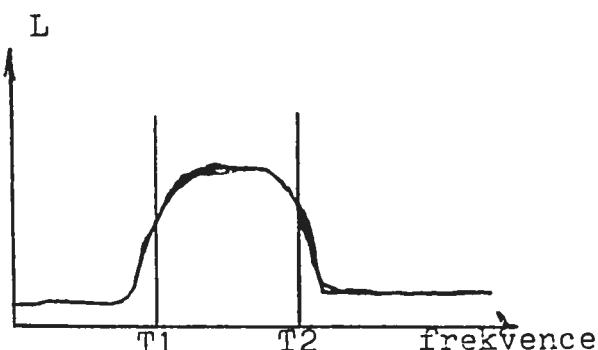
3. Pásmová propust



Nízkokmitočtová propust odřízne všechny frekvence, které leží ~~výše~~ nad mezním kmitočtem a propustí všechny tóny ležící pod. Tón zní až temně.



Analogicky k tomu propustí vysokokmitočtová propusť jen frekvence, které leží nad mezním kmitočtem. Všechny frekvence ležící pod se odříznou. Tón zní tence až ostře.



Pásmová propusť propustí jen ty frekvence, které leží nad dolním a pod horním mezním kmitočtem. Tón zní dutě nebo jako tlumený hlas.

Instrukce FILTER určuje mezní kmitočet, typ filtru stejně jako rezonanci.

Všeobecný tvar zní:

FILTER[frekv][,[Tp][,[Bp][,[Hp][,[Res]]]]]

Vysvětlení jednotlivých parametrů:

frekv Celocíselná hodnota mezi 0 a 2047, která určuje kmitočet filtru.

Tp přepíná nízkokmitočtovou propusť /0=vyp., 1=zap./

Bp přepíná pásmovou propusť /0=vyp., 1=zap./

Hp přepíná vysokokmitočtovou propusť /0=vyp., 1=zap./

Res Celocíselná hodnota mezi 0 a 15, která určuje rezonanci.

Doplňte si program o další řádek:

35 FILTER 1023,1,0,0,10

Řádek 35 nastavuje mezní frekvenci na 1023 a zapíná nízkokmitočtovou propusť. Vysokokmitočtová propusť a pásmová propusť se vypnou. Môžete to se nastaví rezonance na 10. Pošlechněte si, jak tyto tóny zní.

Nyní změňte následujícím způsobem řádek 35

35 FILTER 1023,0,1,0,10

Toto způsobí, že se současně vypne nízkokmitočtová propusť a zapne se pásmová propusť. Poslechněte si ještě jednou tyto tóny.

Obměňte ještě jednou řádek 35:

35 FILTER 1023,0,0,1,10

Dávejte pozor na rozdíl, kdy bude vypínán jeden filtr po druhém. Experimentujme ještě trochu s jednotlivými filtry. Zde jsou uvedeny některé příklady příkazů Filter s jinými mezními kmitočty.

35 FILTER 2000,0,0,1,15

35 FILTER 1500,0,1,0,5

35 FILTER 1000,1,0,0,8

První příklad nastaví mezní frekvenci na 2000 a volí rezonanci 15. Vysokokmitočtová /horní/ propust filru je zapnuta.

Druhý příklad nastavuje mezní kmitočet na 1500, stanová rezonanci 5 a zapíná pásmovou propust.

Ve třetím příkladě je stanovena mezní frekvence na 1000 a rezonance na 8. Dolní propust je zapnuta.

PSANÍ HUDEBNÍHO PROGRAMU

Sestavme jednotlivé součásti hudebního programu:

Uvádíme jeden příkladný program:

```
10 ENVELOPE 9,10,5,10,5,2,4000:PLAY"T9"  
20 VOL 8  
30 TEMPO 30  
35 FILTER 1024,0,1,3,10  
40 PLAY"C D E F G A E"  
50 PLAY"U5 V1 05 C D E F G A B"  
60 PLAY"U7 V2 06 X1 C D E F G A B"  
70 PLAY"U7 V2 06 X1 H CD Q#EF IS GA.S#B"  
80 PLAY"US V1 04 X0.H CD Q#EF IS GA.S#B"
```

Následuje systematické vysvětlení jednotlivých řádků programu.

Řádek 10 volí obalovou křivku 9. Instrukce PLAY přejímá nový parametr ADSR.

Řádek 20 nastavuje hlasitost na stupeň 8.

Řádek 30 určuje tempo.

Řádek 35 filtruje noty řádků 60-70

Mezní kmitočet je stanoven při 1024. Horní propust filru je zapnuta a rezonanční frekvence stanovena u 10.

Řádek 40 hraje noty CDEFGAB /tónová stupnice/

Řádek 50 hraje stejné noty jako řádek 40. Určuje oktávu a hlasitost na 5.

Řádek 60 hraje stejné noty jako řádek 40 a 50, jen se změněnou hlasitostí, jiným hlasem a v jiné oktévě. Filtr je zapnut.

Řádek 70 se liší od předchozích řádků, přičemž jeho tóny byly změněny na poloviční /půldobé/, čtvrtiční, osminové a šestnáctinové.

Řádek 80 hraje stejnou melodii jako řádek 70, filtr je včak vypnut.

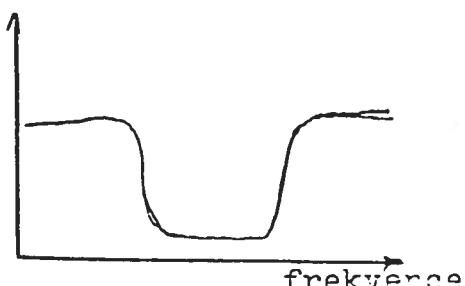
Nyní máte již všechny potřebné informacek které potřebujete k napsání vlastního hudebního programu.

Vyzkoušejte zadat do počítače noty známé písni a propojte grafický program se svým hudebním programem.

DALŠÍ MOŽNOSTI FILTRU

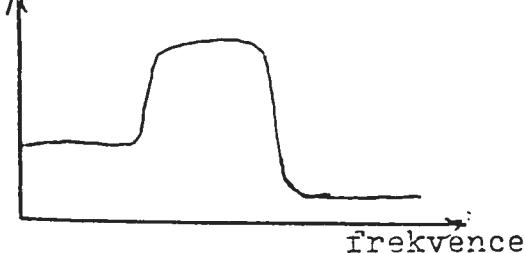
Doposud jste používali jen kombinaci filtru. Můžete však zkombinovat všechny tři filtrové možnosti.

L



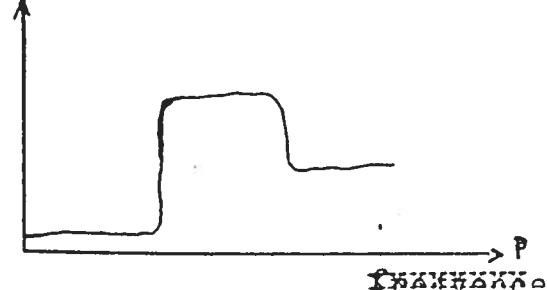
Vrubový filtr /Notch reject Filter/ propustí všechny kmitočty nad a pod mezní frekvencí mimo kmitočtu, které leží v rozsahu pásmové propusti.

L



Můžete spojit také horní, dolní a pásmovou propust. Spojte např. pásmovou a horní propust filtru, pozdvihnu se tak všechny frekvence, které leží v tomto rozsahu.

L



Kombinací pásmové a horní propusti filtru se zdůrazní všechny frekvence nad mezním kmitočtem.

Vyzkoušejte experimentovat s filtry, abyste dokonale poznali různé možnosti utváření hudby.