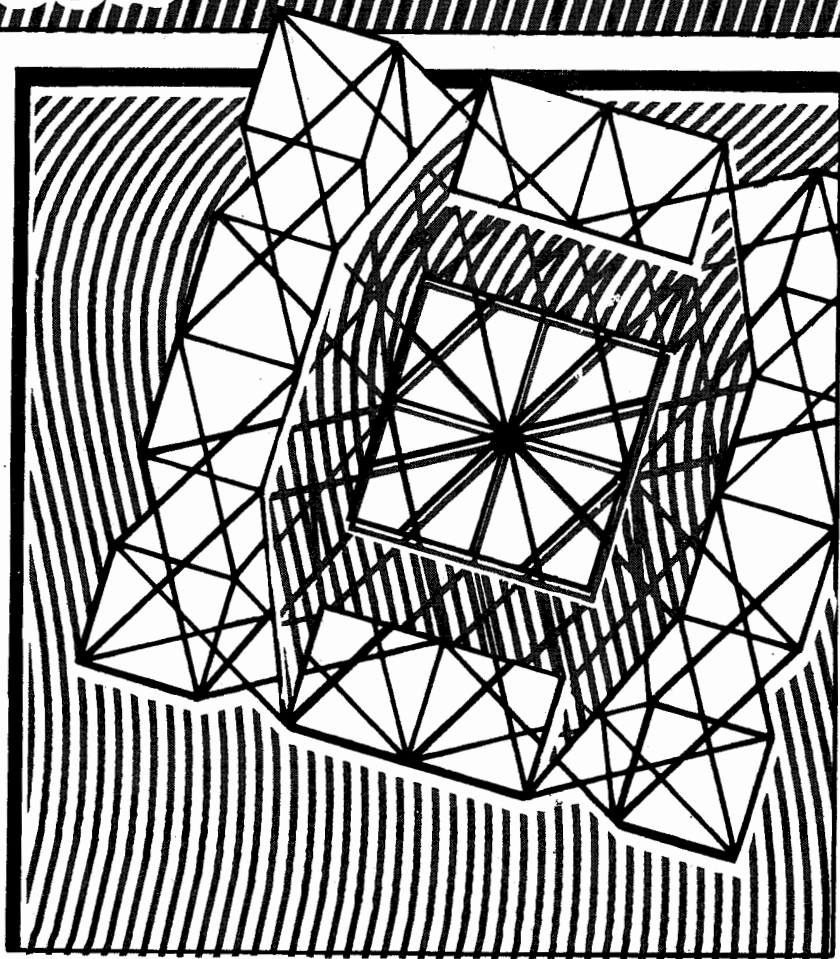




602

COMMODORE

2  
88



## BASIC a ty ostatní.

### 3. část

1. a 2. část našeho miniseriálu jsme věnovali programům EXBASIC a METABASIC, které slouží především pro větší komfort při tvorbě a ladění programů. Do této skupiny by bylo možno zařadit i snad nejrozšířenější program, tj. Simon's Basic, který obsahuje nejméně 40 příkazů typu AUTO, KEY, DUMP, RESUME, DISPLAY apod., avšak Simon's Basic jsme zařadili mezi skupinu programů pro práci s grafikou. Commodore 64 umožňuje v režimu s vysokou rozlišitelností (tzv. HIRES-módu) zobrazení 320 x 200 bodů a v režimu MULTICOLOR počet poloviční 160 x 200 bodů. To je oproti 40 x 25 znaků ve znakové ROM podstatný rozdíl. Přejedem mezi oběma typy je 4x jemnější pseudografika, kterou jsme popsali u EXBASICU. V říjnovém čísle časopisu 64'er byl popsán program TEGRA-Grafik s rozlišitelností 160 x 100 bodů.

Toto číslo zpravodaje je určeno popisu ULTRA-BASICu a především jeho porovnání s některými dalšími programy pro grafiku. Tabulka zachycuje pouze rámcově příkazy programů Ultrabasic, Simon's Basic a Hires-Master a vyplývají z ní základní rozdíly mezi programy.

program:	Ultrabasic	Simon's Basic	Hires-Master
obrazce	7x Dot,Draw Box,Block,Fill Circle,Char,...	10x Plot,Line Rec,Angl,Paint Circle,Text,...	11x Plot,Line Box,Block,Fill Circle,Text,...
Sprite	7x Sprite,Bit Place,Rotate	10x Mmob,Mem Design,Check,...	nemá
"želva"	11x Turtle,Tup	pouze Draw\$	pouze Figure\$
obraz. efekty	nemá	12x Fill,Left Flash,Up,...	15x Page,Option Revers,Effect,...
uložení HIRES	Dump,Gread	nemá	Gsave,Gload

Při programování grafiky je třeba pečlivě uvážit požadavky na grafický program. Pro práci se SPRITE je nejlépe vybaven Simon's Basic, který také jako jediný obsahuje řadu efektů pro obrazovku jako blikání, střídání barev (FLASH, BFLASH) rolování do stran, svisle apod. Hires-Master má zvláštnost v tom, že umožňuje definovat několik grafických obrazovek v adresovém prostoru hex. 2000, 4000, 6000, 8000, E000, jejich výměnu, překrývání, kombinace apod. Ultrabasic jako jediný

obsahuje 11 příkazů pro grafiku typu "želva" - Turtle-grafik (TURTLE, TUP, TDOWN, MOVE, TURN...) a v kombinaci s funkcí JOY dovolí sestavit poměrně jednoduchý program, který umožňuje snadné kreslení obrázku na obrazovku pomocí joysticku. Ultrabasic má příkazy DUMP a GREAD pro uložení a načtení HIRES dat, zatímco např. Simon's Basic umí uložit pouze LOWRES data. Další výhodou Ultrabasicu je příkaz TIC, který pro rychlou orientaci na obrazovce zobrazí jako rastr popis osy X a Y. Velkou výhodou Ultrabasicu je možnost kdykoli přepínat grafickou a normální obrazovku funkcemi tlačítka F5 a F7. Na rozdíl od Simon's Basicu zůstává grafika zapnuta po provedení každého grafického příkazu, zatímco Simon's Basic se vždy vrací do normálního rozlišení, což je pro kreslení obrázků mnohem náročnější.

Ultrabasic kromě grafických příkazů obsahuje i 5 příkazů pro tvoření hudby (obdobně jako Simon's Basic), má funkce JOY, PADDLE, PEN a navíc oproti Simon's Basicu nastavení timeru SCTR a čtení - funkci CTR. Grafická obrazovka Ultrabasicu je definována v prostoru hex. A000. Určitou nevýhodou Ultrabasicu ve srovnání se Simon's Basicem je čtení polohy joysticku.

Třetí část našeho miniseriálu končí jako obvykle tabulkou se stručným popisem všech příkazů Ultrabasicu a navíc krátkým programem, který umožňuje kreslit na obrazovku joystickem.

```

10 REM - KRESLENI POMOCI JOYSTICKU
20 HIRES 3,5: TURTLE 4
30 FLG=0: TDOWN
40 IF JOY(1)=0 THEN GOTO 40
50 Q=JOY(1)
60 IF Q=16 THEN GOSUB 180
70 IF Q=1 THEN W=0
80 IF Q=9 THEN W=45
90 IF Q=8 THEN W=90
100 IF Q=10 THEN W=135
110 IF Q=2 THEN W=180
120 IF Q=6 THEN W=225
130 IF Q=4 THEN W=270
140 IF Q=5 THEN W=315
150 TURN TO W
160 IF TPOS(X)>320 OR TPOS(Y)>280 THEN STOP
170 GOTO 40
180 IF FLG=0 THEN GOTO 200
190 TDOWN: FLG=0: RETURN
200 TUP: FLG=1: RETURN

```

(P.Ko.)

## ULTRABASIC - příkazy

Základem je grafická obrazovka 319 x 199 bodů, B udává barvu.

**HIRES Bo,Bp** - inicializace grafiky, barva obrazu a pozadí

**MULTI Bo,Bp** - varianta MULTICOLOR

2. barva +100

3. barva +200

**CONT** - pokračování

**TIC x,y,B** - Popis osy x a y

**DOT x,y,B** - Bod

**DRAW x1,y1,x2,y2,B** - linka

**BOX x1,y1,x2,y2,B** - obdélník

**BLOCK x1,y1,x2,y2,B** - plocha

**CIRCLE x,y,r,B** - kružnice o poloměru r

**CHAR t,x,y,B,var** - popis textem, var je string nebo číslo, t je typ písma (1-velká, 2-inverzní, 3-malá, 4-inverzní)

**FILL x,y,B** - Vyplnění prostoru s bodem x,y

**MODE m** - charakter (1-normální, 2-mazací, 3-inverzní)

**COPY N,L** - přiřadí blok čísla N od řádky L pro sprite

**BIT** - řádka obsahuje v uvozovkách bod po bodu sprite

**HEX**

**SDATA**

**SPRITE Sc,N,m,P,x,y,B1,B2,B3** - sprite, Sc=číslo, N=blok, m=@ jednobarevný, m=1 vícebarevný, P=@ priorita před pozadím, P=1 priorita za pozadím, x,y=velikost (1=dvojitá)

**PLACE Sc,x,y** - změna polohy sprite

**OFF Sc** - zřízení spritu

**ROTATE D,t,Sc** - rotace spritu typu t ve směru D

**DUMP název, pp** - uložení grafické obrazovky, pp=1(8)

**GREAD název, pp** - načtení z periferie

DUMP je F2

GREAD je F4

**PIXEL (x,y)** - funkce, hodnota je 0-3; udává typ zápisu bodu

**TURTLE Bt,x,y;** - želva; není-li pozice určena, je ve středu

**MOVE D** - počet bodů pohybu

**TURN n** - otočení o n stupňů

**TURN TO n** - nastavení směru na určitý úhel

**TUP** - zvednutí pera

**TDOWN** - spuštění pera

**TCOLOR B** - barva zápisu

**BYE** - želva zmizí, zápis je rychlejší

**TPOS(x)** - funkce udává polohu v x

**TPOS(y)** - funkce udává polohu v y

**TPOS(A)** - funkce udává směr natočení

Pro možnost opakování určitého příkazu slouží varianta v hranatých závorkách s dvojtečkou, před níž je číslo kolikrát se má příkaz opakovat. [p:.....]

**HARD pp** - hardcopy, pp=4 není-li jinak

**JOY(p)** - funkce čte JOY portu p

**PADDLE (p,PD)** - funkce čte paddle v portu p, hodnotu x,y

**PEN(d)** - funkce čte světlé pero

**SCOLL (S1,S2[,S3...])**

**BCOLL (S1[,S2...])** - jsou dvě funkce pro hodnocení kolize spritu s pozadím a mezi sebou, hodnota=@, při kolizi=1

**SOUND G,P,D** - Nastavení generátoru číslo G, trvání tónu P (1/60sec. pro P=1), D je výška (1-255)

**GEN A,B,C,D,F,G,H,I** - charakter tónu

A 1-3 číslo generátoru

B 1,2,4,8 tvar tónu

C 0-15 rychlost náběhu

D 0-15 rychlost zeslabení

E 0-15 trvání úrovně

F 0-15 doba doznění

G 0-15 trvání zapnuto

H synchronizace (0=vypnuta)

I modulace (0=vypnuta)

**VOL A,B,C,D,E** - hlasitost tónu

A 0-15 hlasitost

B 1,2,4,8 modus filtru

C 1,2,4,8 řízení filtru

D 0-15 frekvence filtru

E 0-15 rezonance filtru

**SET**

**TDATA**

**TUNE**

**SCTR T,v** - nastavení timeru T na čas v (1-65000)

**CTR(t)** - funkce čte hodnotu timeru

## ABRACALC

Program ABRACALC je programovatelná výpočtová tabulka o velikosti 40 řádků a 26 sloupců (1040 boxů)  
Dotaz COMMAND očekává vstup příkazu pro box označený kurzorem.

### Designové příkazy:

F1 . . . . . změna barvy rámu obrazovky  
F3 . . . . . změna barvy pozadí tabulky

### Pohyb kurzoru:

CRSR . . . . . normální funkce  
HOME . . . . . nastavení boxu **a1**  
> . . . . . skok do zvoleného boxu  
← . . . . . kurzor do sloupce **a**  
↑ . . . . . kurzor do řádku **1**

### Vstup obsahu boxu

**znakový (LABEL)** . . . Za příkaz COMMAND v editačním řádku se запиše text (max. 9 znaků). Zapisuje-li se více znaků přejde kurzor automaticky na do následujícího boxu v řádku.

**číselný (VALUE)** . . . Za COMMAND se запиší číslice. Je nutno dodržet 9 míst (včetně znaménka a desetinné tečky). Znaménko plus je nepřipustné.

### Vstup pozadí boxu

**aritmetický výraz (ARITHMETIC)** . . . Za COMMAND se запиše znaménko plus potom následuje krátká aritmetická operace začínající označením požadovaného boxu. Může obsahovat souřadnice boxů a aritmetické operátory +, -, \*, / (ale max. 2).

Př: kurzor je na pozici **c3**

zapišeme: **+a1+a2** (RETURN)

box **c3** bude obsahovat součet obsahů **a1** a **a2**

**výpočtová formule (FORMULA)** . . . Za COMMAND se запиše zavináč (Ⓢ) a dále jeden z příkazů:

**a** . . . . (ave) aritmetický průměr

**s** . . . . (sum) součet

**c** . . . . (con) počet numerických boxů (value)

**mi** . . (min) minimum

**ma** . . (max) maximum

Zápis souřadnic boxů vypadá takto '**xx..xx**' (RETURN) kde **xx** je souřadnice boxu (písmeno a pak cifra).

Př. Box jehož pozadí obsahuje **@ave(b3..b15)** udává aritmetický průměr čísel ve sloupci b3 až b15.

Př. **@con(h5..x5)** udává počet boxů v řadě h5 až x5, které mají numerickou náplň (value).

### Vstup příkazů

ZA příkaz COMMAND se zapiše znak /, čímž je vyvolán seznam příkazů v editačním řádku. Stisknutím příslušného symbolu se okamžitě daný příkaz vyvolá. Příkazové menu se ruší stisknutím znaku /.

**A...AUTOCALC** - Nastaví (ON) nebo (OFF) průběžné provádění početních operací v tabulce. Nastavení OFF tabulku zmrazí a výpočty se provedou až příkazem RECALCULATION.

**B...BLANK** - Vymaže obsah boxu označeného kurzorem.

**C...CLEAR** - Vymaže obsah boxů podle doplnění příkazu:

t...textový obsah (label)

v...numerický (value)

f...boxy s formulemi a aritmetickým pozadím (formula)

a...všechny boxy (all)

DEL, RUN/STOP...ruší příkaz

**D...DELETE** - Vymaže řádek nebo sloupec s kurzorem.

l ... řádek (line)

c ... sloupec (column)

RETURN...ruší příkaz

**F...FREE MEMORY** - Zobrazí velikost volné paměti.

**G...GRAPH** - Graficky znázorňuje hodnoty boxů (xx..xx) (Syntaxe jako FORMULA).

**I...INSERT** - Vloží sloupec nebo řádek do pozice kurzoru. Vzniklý prostor neobsahuje žádné hodnoty.

l ... řádek (line)

c ... sloupec (column)

**J...JUSTAGE** - Modifikuje číselné zobrazení:

i ... zobrazení čísel ve tvaru INTEGER.

r ... zobrazení čísel se 2 desetinnými místy (je-li des. část nenulová).

libra ... nastavuje původní zobrazení.

**L...LINE** - Kurzorový box proškrtne čarou.

**P...PRINT** - Výstup na tiskárnu. (Ize zrušit DEL).

Heading: y...definice a tisk záhlaví

n...tisk bez záhlaví

Záhlaví může obsahovat 10 řádků (max. 30 znaků).

Grid ref. - tisk masky (y/n)

- Q...QUIT** - Po následné odpovědi **y** se program ukončí.
- R...REPLICATE** - Obsah boxu označeného kurzorem se přepíše do boxů v řadě (sloupci) až k boxu jehož souřadnice je zadána za R. Má-li box pozadí, pak se přenáší formule:  
 Relative: y... Písmena a čísla označující box se ve formuli modifikují v závislosti na souřadnicích boxu do něhož je přepis uskutečňován. Je-li pozadí typu arithmetic je aktivován pouze absolutní přepis (tj. odpověď n).  
 n... Překopíruje se skutečná formule z pozadí.
- Př. Relativní přenos. V boxu a1 je @min(a1..a10) a kurzor je v pozici a1. Při zadání / R y d1 (číslo 1 se dosadí implicitně) bude:  
 box b1 obsahovat @min(b1..b10)  
 box c2 obsahovat @min(c1..c10)  
 box d2 obsahovat @min(d1..d10)  
 atd.
- Př. Relativní přenos. V boxu a1 je @sum(a1..d1) a kurzor je na pozici a1. Po zadání / R y a5 RETURN bude:  
 box a2 . . . @sum(a2..d2)  
 box a3 . . . @sum(a3..d3)  
 atd.
- S...STORAGE** - Práce s periferií (disk nebo kazeta).  
 load...nahrát data  
 save...uložit data
- V...VERSION** - Verze programu ABRACALC.
- W...WINDOW** - Nastavení počtu řádků, které se zobrazují. Pro window větší než 10 se HELP zobrazí pomocí Y.
- Z...WINDOW FORM** - Místo HELP se zobrazí zvolená část tabulky určená souřadnicí boxu v levém horním rohu této části.

### Ostatní příkazy

- FB** - Nastavuje (ON) nebo (OFF) aktivace módu, ve kterém je blokován zápis do boxů s vyplněným pozadím (formule a aritmetické výrazy).
- !** (RECALCULATION) - Je-li blokován režim AUTOCALC (tj. A OFF) nejsou výpočty prováděny průběžně. Recalculation provede jednorázové přepočítání celé tabulky.

\*\*\* IP \*\*\*

## Počítač ! A co s ním ?

(2. díl)

V prvním dílu našeho seriálu jsme si objasnili rozdíly mezi pamětí ROM a RAM a řekli jsme si, proč pro označování adres používáme hexadecimální soustavu. Dnes se podíváme, jaké programové vybavení je ukryto v naší paměti ROM. Ovšem dříve si musíme vysvětlit, co je to program a programovací jazyk a co je třeba k tomu, aby mu náš mikroprocesor porozuměl.

Program je vlastně algoritmus řešení úlohy zapsaný ve formě zpracovatelné počítačem. Minule jsme si řekli, že jediné čemu náš mikroprocesor rozumí je instrukce strojového kódu. Taková instrukce je v našem případě, stejně jako u všech ostatních osmibitových počítačů, osmibitová informace, kterou lze zapsat ve formě dvoumístného hexadecimálního čísla.

Uznáte ale, že programovat ve strojovém kódu, to jest zadávat ve správné posloupnosti hexadecimální čísla je pracné a nepraktické. Proto byl vyvinut jazyk symbolických adres, který dovoluje nahradit těžko zapamatovatelná hexadecimální čísla mnemotechnickými značkami (jejich význam a použití si vysvětlíme v některém dalším pokračování). Zápis programu v jazyku symbolických adres je nutno před spuštěním přeložit do strojového kódu. Tento překlad, který se provádí pomocí programu zvaného assembler, je poměrně jednoduchý, neboť každé mnemotechnické značce odpovídá jedna instrukce strojového kódu.

Daleko náročnější práci má překladač v případě, že používáme k programování vyššího jazyka. V tomto případě se nahrazuje jeden příkaz vyššího jazyka celým souborem instrukcí strojového kódu (podprogramem, subroutine). Podle způsobu, jakým se zdrojový program překládá do strojového kódu rozdělujeme překladače na interpretery a kompilátory.

Kompilátor nejprve námi zadaný zdrojový program přeloží do strojového kódu, tedy vytvoří tzv. cílový program. Tento cílový program je teprve srozumitelný počítači a lze jej spustit. Nevýhodou tohoto typu překladače je především ta skutečnost, že jakákoli chyba lze opravit pouze ve zdrojovém programu a pak se musí znovu provést překlad. Výhodou je, že program přeložený do strojového kódu pracuje pak velmi rychle.



Interpreter analyzuje každou řádku, kterou mu zadáváme z klávesnice. Případné chyby v zápisu řádky interpreter ihned ohlásí, proto se tento způsob překladač nazývá též interaktivní. Zdrojový program zapsaný do paměti překládá interpreter až po spuštění programu a to postupně. Nejprve načte příslušný příkaz vyššího jazyka, přeloží jej do strojového kódu, vykoná a postupuje k dalšímu příkazu. Z toho také vyplývá, že rychlost provádění programu je v tomto případě podstatně nižší než při použití kompilátoru. Výhodou je však možnost rychlé změny resp. opravy programu. V podstatě jediným používaným interpreterem je interpreter jazyka Basic.

A nyní se vraťme zpět k našemu počítači a k jeho paměti ROM. Naše paměť ROM zabírá adresový prostor od adresy 8000H do adresy FFFFH. Její organizace je následující:

8000H	-	CFFFH	Basic interpreter
D000H	-	D3FFH	Generátor znaků - mód velkých písmen
D400H	-	D7FFH	Generátor znaků - mód malých písmen
D800H	-	FBFFH	Provozní systém
FC00H	-	FCFFH	Banka ROM programů
FF00H	-	FF3FH	TED CHIP register
FF40H	-	FFFFH	Tabulka skoků

A nyní k jednotlivým částem podrobněji:

#### a) **Basic interpreter**

Co je jeho úkolem jsme si již vysvětlili v úvodní části. Připomeňme si, že v našem počítači je vestavěn překladač jazyka Basic V 3.5, což je varianta, která dokáže rozpoznat a do strojového kódu přeložit 120 různých povelů a příkazů.

#### b) **Generátor znaků**

Toto je část paměti, kde jsou uloženy tvary všech znaků, které umí náš počítač zobrazit. V adresovém prostoru od D000H do D3FFH jsou uloženy znaky, které lze vyvolat pracujeme-li v módu velkých písmen a od adresy D400H jsou uloženy tvary znaků použitelných v módu malých písmen.

Podívejme se jakým způsobem jsou zde znaky uloženy. Víme, že rozlišovací schopnost našeho počítače je 320 x 200 bodů. V textovém módu lze využít 25 řádek a 40 sloupců. Znamená to, že jeden znak zabírá plochu 8x8 bodů tj. jeden pixel. Každý z těchto bodů může mít buď barvu pozadí nebo barvu písma tzn. může mít hodnotu 0 nebo 1. K vyjádření jednoho pixelu tedy potřebujeme osm osmibitových informací, tedy 8 Byte. V každém módu máme k dispozici 128 různých znaků, jejichž tvary lze uchovat v 128x8=1024Byte, tedy právě 400H.

Příklad uložení znaku "A":

D008H	00011000	18H
D009H	00111100	3CH
D00AH	01100110	66H
D00BH	01111110	7EH
D00CH	01100110	66H
D00DH	01100110	66H
D00EH	01100110	66H
D00FH	00000000	00H

### c) Provozní systém

V této části paměti ROM jsou uloženy programy, které jsou vyvolávány jednak vlastním systémem (např. program zjišťující rozsah volné paměti RAM po zapnutí počítače, programy zajišťující funkce vstupu a výstupu), jednak použitím některých speciálních funkcí (např. ESC funkce) a jednak použitím vestavěného monitoru.

### d) Tabulka skoků

Jak již název napovídá jsou v tabulce skoků uloženy startovací adresy nejdůležitějších systémových programů.

K podrobnému popisu jednotlivých částí provozního systému i tabulky skoků se v našem časopisu ještě vrátíme. V další části seriálu se budeme zabývat pamětí RAM a TED CHIP registerem.

(3. díl)

Ve druhém dílu jsme se seznámili s naší pamětí ROM. Dnes se podíváme jak je organizována naše paměť RAM.

Hned v prvním díle našeho seriálu jsme si řekli, že náš počítač má kapacitu paměti RAM 16kb (resp. 64kb). Zapneme-li náš počítač ohlásí se hlavičkou, ve které inzeruje pouze 12271 Byte (resp. 60671 Byte) volných. Jak je to možné?

Odpověď je velmi jednoduchá. pro potřeby uživatele (tedy pro zápis programů) je vyhrazena pouze část paměti RAM (i když podstatná). Zbytek využívá sám systém pro svou práci.

V případě, že používáme textový mód, je organizace paměti RAM následující.

0000H - 07FFH	systémové proměnné
0800H - 0BFFFH	barvonosná RAM pro obrazovku
0C00H - 0FFFFH	obrazová RAM
1000H - 3FFFFH	(resp. FD00H) Basic RAM
FD00H - FF3FH	adresy vstupu/výstupu TED CHIP register

V případě, že používáme grafický mód, který umožňuje ovládat každý z 64000 bodů obrazovky separátně, je volná Basic RAM organizována následovně:

1000H -	17FFH	Basic RAM (pozn. 1)
1800H -	1BFFH	Stupeň jasu jednotlivých pixelů obrazovky
1C00H -	1FFFH	kód barvy jednotlivých pixelů obrazovky
2000H -	3FFFH	aktivace jednotlivých bodů obrazovky (1 bod= 1 bit)
4000H -	FD00H	Basic RAM (pozn. 2)

Pozn. 1: U verze 16kb je tedy volná kapacita paměti při použití grafického módu 2kb.

Pozn. 2: U verze 64kb se při použití grafického módu posouvá počátek Basic RAM na adresu 4000H a adresový prostor od 1000H do 17FFH zůstává nevyužitý.

Nyní se podívejme na jednotlivé části RAM podrobněji:

#### a) systémové proměnné

Jak již napovídá sám název, ukládá si počítač do této části paměti údaje o stavu systému. Ukládají-li se zde adresy, které jak víme jsou šestnáctibitové, ukládají se systémem LO-HI. To znamená, že na adresu nižší se ukládá dolních osm bitů a na adresu vyšší se ukládá horních (mocnějších) osm bitů.

Jako příklad uvedu uložení některých zajímavých proměnných:

002BH -	002CH	ukazatel počátku Basic RAM
002DH -	002EH	ukazatel počátku paměti pro proměnné
0037H -	0038H	ukazatel konce Basic RAM
00c8H		kód právě stisknuté klávesy
0534H		návěští SHIFT: 1 - klávesa SHIFT 2 - klávesa C 4 - klávesa CTRL

07F8H                      přepínání zobrazované paměti pro  
TEDMON: ROM (00) a RAM (80).

Uveřejnit na stránkách našeho časopisu celý seznam systémových proměnných z kapacitních důvodů nelze, ale je možno nechat jej rozmnožit a poskytnout případným zájemcům.

#### b) Basic RAM

Do této části paměti se ukládá vlastní program a všechny typy proměnných, které využívá. Jakým způsobem se zde tyto informace uchovávají probereme ve zvláštní kapitole.

#### c) TED CHIP register

Proměnné, jejichž hodnoty jsou uchovány v této části paměti RAM lze rovněž zařadit mezi systémové proměnné. Jejich přehled přineseme v příloze příštího zpravodaje.

## GHOSTBUSTERS

Nejprve se na obrazovce objeví nápis GHOSTBUSTERS. Stiskněte RETURN a máte možnost vybrat si některá zařízení, která se Vám mohou hodit na Vaši cestě. Na první výzvu počítače napište své jméno a příjmení a stiskněte RETURN. Druhou otázkou se Vás počítač táže zdali máte založeno konto v bance. Hrajete-li poprvé nebo novou hru od začátku odpovězte stisknutím klávesy N (no) a RETURN. Počítač Vám do začátku daruje 10 000 \$. Máte-li konto založeno stiskněte Y (yes) a RETURN. Poté udejte počítači číslo Vašeho konta.

Nyní si vyberete vozidlo, se kterým budete jezdit. Vozidla si můžete prohlédnout opakovaným stisknutím klávesy SPACE.

### Nabídka vozidel

1. Malé auto. Nejvyšší rychlost 75 mil/hod. Uveze 5 pastí na duchy. Kupní cena 2 000 \$.
2. Pohřební vůz z roku 1963. Max. rychlost 90 mil/hod. Uveze 9 pastí. Cena 4 800 \$.
3. Dodávka. Max. rychlost 110 mil/hod. Uveze 11 pastí na duchy. Cena 6 000 \$.
4. Výkonný vůz. Max. rychlost 160 mil/hod. Uveze 7 pastí. Cena 15 000 \$.

Auto koupíte tak, že stisknete číslo příslušného auta a RETURN.

### Další zařízení

1. Hlídací zařízení: PK ENERGY DETECTOR - varuje před přicházejícím duchem (nazývaným SLIMER) tím, že budova zrůžoví, když ji míváte. **IMAGE INTENSIFIER** (zesilovač obrazu) - dělá duchy lépe viditelné, když se je snažíte chytit. **MARSHMALLOW SENZOR** - varuje před příchodem ibiškového muže (MARSHMALLOW MAN). Budova u které se nacházíte se stane bílou.

U každého přístroje je vpravo napsaná cena. Joystickem lze ovládat nakládací vozík a koupená zařízení ukládat do svého auta. Chcete-li přejít do další části nákupu stiskněte 2.

2. Chytací zařízení: GHOST BAIT - vnaidlo na duchy (zvané ROMERS), kteří se shlukují, aby vytvořili MARSHMALLOW MANA. Bez vnaidla ho nemůžete zastavit. **GHOST TRAPS** - používají se na chytání a ukládání duchů (SLIMER). Do pastí se vejde jen jeden SLIMER. Bez pastí nemůžete získat peníze a proto je nezbytné koupit alespoň jednu. **GHOST VACUUM** - nasává potulné duchy (ROMERS) při Vaší cestě ulicemi. Způsob nákupu je stejný jako u hlídacích zařízení. Poté stiskněte 3 a RETURN.

**3. Ukládací zařízení: PORTABLE LASER** - uvězní 10 duchů ve vozidle, aniž byste se museli vracet do GHO.

Jestliže jste koupili vše potřebné stiskněte RETURN a objeví se Vám úsek mapy města.

### Mapa

Na obrazovce se objeví mapa města se strašidelným zámekem ZUULA uprostřed a Vaše stanoviště GHO je dole. Rudě blikající budovy ukazují přítomnost ducha (SLIMER) v této budově. Své vozidlo řiďte co nejkratší cestou k jakékoli blikající budově. Cestou se pokuste zmrázit co nejvíce ROMERS, kteří se pohybují k ZUULU tak, že se jich dotknete za jízdy autem.

Chcete-li se dostat do budovy, pohněte joystickem do daného směru a stiskněte spoušť.

### Jízda ulicemi

Navádějte vozidlo na ROMERS, které jste při cestě mapou zmrázili. Stisknutím spouště je můžete vsát.

### Chytání duchů

Po příjezdu na místo uděláte toto:

- a) Nasměřujte svého prvního muže směrem ke středu budovy a stisknutím spouště položte past na zem. Pak muže odveďte vlevo a znovu stiskněte spoušť.
- b) Objeví se druhý muž. Zaveďte ho doprava a otočte čelem k pastí. Stiskněte spoušť a oba chytači zapnou své ionizátory.
- c) Přibližujte své muže k sobě, aby se Vám podařilo dostat ducha mezi proudy paprsků. POZOR, paprsky se nesmí zkřížit!!
- d) Když je duch nad pastí, opět stiskněte spoušť. Jestliže past stáhne ducha je vše v pořádku, jestliže ho však nezasáhne, duch zabije jednoho muže.

Každý chycený SLIMER zvyšuje částku na Vašem účtu. Množství vydělaných peněz závisí na rychlosti Vaší reakce.

### Důležité bezpečnostní rady

Stisknutím klávesy. SPACE dostanete zprávu o Vašem stavu.

Když začne v dolní části obrazovky blikat nápis MARSHALLOW ALERT, ROMERS se začnou rychle sbíhat, aby vytvořili MARSHALLOW MANA. Stiskněte klávesu B (odhodíte kus vnaidla) dříve, než MAN zničí kteroukoli budovu.

### Konec hry ! Svatyně Zuulu

Hra končí jedním ze tří způsobů:

- 1) Dveřník a klíčník se spojí u svatyně Zuulu. Nevydělali jste více peněz než jste měli na začátku.
- 2) Dveřník a klíčník se spojí u Zuulu. Máte dostatek peněz, ale nedostanete dva ze svých mužů do vchodu Zuulu.
- 3) Provedl jste dva svoje muže vchodem a dosáhl vrcholu svatyně Zuulu.

## KUDI 64

**BORDER a,b** (dva parametry): Těmito parametry nastavíte hranice na ose x. Hranice na ose y se změní sami tak, aby vznikla čtvercová síť. (Jednotka na ose y = jednotce na ose x.) Osa x se posune doprostřed obrazovky. Můžete tím vytvořit na obrazovce kruh.

**BORDER a,b,c** (tři parametry): Podobně jako u případu se dvěma parametry. X-ové souřadnice nastavíte nyní pevně a Y-ové souřadnice se posunou tak, aby byl zachován rozsah původní obrazovky.

**BORDER a,b,c,d** (čtyři parametry): Nastavíte všechny čtyři hranice.

Při psaní parametrů menších než jedna dejte pozor na správné psaní teček u číslic.

**DISCUSS**: Toto je nejučinější příkaz, který nám KUDI 64 poskytuje. Jako u příkazu DRAW se nám vykreslí zadaná funkce i s první a druhou derivací, přičemž se automaticky vymaže předešlá grafická obrazovka. Současně se však ještě vypočítají nulové body, extrémy, inflexní body a definiční obor na BORDERem zadaném intervalu.

Po provedení příkazu DISCUSS zůstane na CRT zobrazena grafická obrazovka. Stisknete-li klávesu <RUN/STOP> CRT se vymaže a vymažou se i všechny zadané funkce. Stisknete-li <RETURN> dostanete se také do textové obrazovky, ale nevymažou se zadané funkce. Stisknete-li jakékoli jiné tlačítko, můžete si libovolně přepínat grafický a výpočtový mód.

Zadáte-li příkaz DISCUSS v programu, počítač reaguje trochu odlišně. Nejdříve se také zobrazí grafika jako v předešlém případě. Potom se derivace (jako u příkazu DERIVE) nadefinují jako BASIC-funkce, vypočítané nulové body, extrémy, inflexní body a body kde se mění konkávnita funkce jsou nadefinovány jako indexované proměnné. Viz. tabulka:

NS(a)	(a)-tý spočítaný nulový bod
NS	Počet spočítaných nulových bodů
MI(a)	(a)-té spočítané minimum
MI	Počet spočítaných minim
MA(a)	(a)-té spočítané maximum
MA	Počet spočítaných maxim
TP(a)	(a)-tý spočítaný inflexní bod (Terassenpunkt)
TP	Počet spočítaných inflexních bodů

WP(a)	(a)-tý spočítaný bod, kde se mění konkávitá nebo konvexita funkce (wendepunkt)
WP	Počet spočítaných wendepunktů (viz.↑)

Ukázka BASIC programu s příkazem DISCUSS:

```

10 DISCUSS X↑2-X-2+1/X
15 PRINT "F (1)="; FNF0(1)
20 PRINT "F' (1)="; FNF1(1)
25 PRINT "F'' (1)="; FNF2(1)
30 PRINT
35 PRINT "Nulové body :"

```

Také u příkazu DISKUSS můžete do hranatých závorek zadat stupně libovolných derivací, které chcete spočítat, ale pak nebudou spočítány extrémy, a inflexní body, které pro svůj výpočet potřebují první nebo druhé derivace.

**ADD:** Tento příkaz se velice podobá příkazu DISKUSS. Použijeme ho tehdy, chceme-li současně zpracovávat dvě funkce. ADD tedy nesmaže obrazovku, a nekreslí derivace současně s funkcí, ale s časovým odstupem.

Používáte-li příkaz ADD v programu zpracovává danou funkci skoro stejně jako DISKUSS, jen s tím rozdílem, že funkce zpracovávaná pod ADD se nazývá  $g(x)$  místo  $f(x)$ .

Např.: »PRINT " G(1)="; FNG0(1)« vám napíše hodnotu funkce  $g(x)$  v bodě 1.

Funkci zadanou jako první můžete stále používat nadefinovanou pod funkcí  $f(x)$ , ale její extrémy, nulové a inflexní body budou vymazány. (Můžete si je však převést do jiných proměnných.)

*Petr Bacek*



## V příštím čísle:

Programování ve strojovém kódu, 5. část  
Basic a ty ostatní, 4.část  
Basic v2.0 kompilátory



Zavádíme **novou službu pro členy**. Na každé pondělní schůzce je možno vypůjčit si na týden diskety s programovým vybavením C64 (Print Fox, Vizawrite, Ace, F-15, Hi Eddie+, Hypra Platos). Mimopražským členům budou diskety na požádání zasilány poštou.

**Hledáme** členy klubu ochotné pravidelně nebo jednorázově přispívat do zpravodaje. Příspěvky je možno předat na kterékoli schůzce nebo zaslat písemně na adresu klubu. Přednost dostávají články na kazetě či disketě ve formě souboru textového editoru Printfox nebo Vizawrite. Pokud uvedené programy nevlastníte, můžete je získat na každé burze klubu nebo v nově zavedené výpůjční službě.

**Formátovací řádek** textového editoru Print Fox pro příspěvky do zpravodaje: z=10 x=80 l=510 y=40 h=2 v=2 s=1 g=0 t=25  
Maximální délka stránky po přeložení do grafiky je 720 bodů.

K dispozici je český překlad manuálu k programu **DATAMAT**. Současně s návodem je možno získat i nejnovější verzi tohoto kvalitního programu na zpracování dat. Bližší informace na kterékoli schůzce klubu.

Toto číslo Zpravodaje bylo napsáno a editováno textovým procesorem PRINT FOX a přispěli do něj:

MUDr. Petr Kocna (P. Ko.)  
Ivan Paulíček (\*\*\* IP \*\*\*)  
Jiří Strnad (J.S.)  
Uítězslav Uěříš (U. U.)  
Petr Uacek

---

Commodore 602, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Uydáová 602. ZD Svazarmu pro potřeby vlastního aktivu, zodpovědný redaktor Ing. Pavel Bukovský, sestavení čísla Ivan Paulíček. Adresa redakce: 602. ZD Svazarmu, Wintrova 8, Praha 6, 160 41. Telefon: 32 85 63. Povoleno UUTEI pod evidenčním číslem 87 006. Cena 12,- Kčs dle ČCÚ č. 1030/202/86. Náklad 500 výtisků. Praha, duben 1988