

```

*****
*                                     *
*   E x t e n d e d   B A S I C   c o m p i l e r   *
*                                     *
*****

```

Vážený uživateli ,

dostává se Ti do rukou dlouho očekávaný kompilátor grafického jazyka BASIC. Jedná se o BASIC kvalitativně vyšší úrovně, než byly prozatím všechny verze tohoto jazyka na počítačích Sharp. Pokud ho chceš efektivně používat, budeš si muset zvyknout na trochu jiný přístup k psaní programů, protože Extended BASIC obsahuje mnoho z filosofie PASCALU, od zápisu programu bez čísel řádků, přes proměnné různých typů až po strukturované příkazy. Než se pustíš do práce, prohlédni si dva krátké demo-programy, ukážou Ti, jak na to. První předvádí možnosti znakové 'grafiky' a použití zvukového generátoru ve formě hry. Druhý program využívá spojení jemné grafiky a strojového kódu k napodobení 'art studia' a je určen především pro uživatele ovládající alespoň základy strojového kódu. To samozřejmě neznamená, že by ho ostatní nemohli pochopit. Znovu upozorňuji, že oba programy mají posloužit k prvnímu seznámení s kompilátorem a nemají tedy žádný praktický význam.

Stručný popis všech příkazů napsal podle mých strohých poznámek Josef Havlíček. Při zjišťování syntaxe a činnosti povelů mi pomáhal Jiří Kepka, na vychytávání much z kazetové verze se vydatně podílel Jaroslav Randa.

Již předem se omlouvám za možnost výskytu nejrůznějších chybiček. Nad tímto výtvořem jsem strávil nejednu dlouhou noc a při existenci téměř stovky verzí, které na mé disketě za těch 14 dní vznikly, nemůžu vyloučit to, že jsem něco přehodil, vynechal nebo jinak zmotal. Zároveň bych rád odradil všechny, kteří by měli chuť program různě 'vylepšovat'. I mně trvalo za pomoci floppy disků a tiskárny několik desítek hodin, než jsem se v něm orientoval. Pokud by se přece jen našel nějaký zoufalec, který na to má čas, ať laskavě zanechá v programu mimo změn i své jméno.

Nakonec bych rád upozornil na to, že nebudu písemně ani telefonicky podávat jakékoliv další informace o tomto programu. Vše je popsáno dále a pokud bude něco méně jasné, tak využij svého intelektu a zkus na to přijít sám. Já jsem také žádný manuál nedostal.

Michal Kreidl

P.S.: Touto cestou se také omlouvám všem, kteří mi psali a marně čekají na odpověď. Kdybych měl odepisovat na všechny dopisy, co dostávám, tak už bych nestihl nic jiného a to by asi byla škoda. Uvědomte si, že jsem jenom amatér, který musí také studovat a zároveň si vydělávat peníze na tuto neuvěřitelně drahou zábavu.

P o p i s j a z y k a (verze 25.11.1988)

Proměnné typů :	integer, 16 bitů, bez znaménka	(bez označení)
	integer, 16 bitů, se znaménkem	%
	integer, 8 bitů, bez znaménka	#
	real, 32 bitů	!
	string	\$

Větvení programu :

	IF f THEN - :ELSE -		Větvení v této formě nelze vnořovat !

Cykly :	FOR TO (STEP)	FOR DOWNT0 (STEP)	Řídící proměnná může být
	.	.	jen typu integer bez zn.,

	.	.		zrovna tak i výraz za STEP.
	.	.		Při snižování hodnoty řídící
	NEXT	NEXT		proměnné je nutné použít
				cyklus s DOWNT0.
	REPEAT	WHILE f	LOOP	Cyklus REPEAT-UNTIL se provádí
	.	.	.	až do splnění podmínky f.
	.	.	.	WHILE-WEND se provádí, dokud je
	.	.	.	podmínka f splněna.
	UNTIL f	WEND	END LOOP	DO n LOOP-END LOOP je
				nekonečná smyčka.
				END DO DO-END DO se provede
				n-krát.
Rozšířené				
IF :	IF f	IF f		
	.	.		
	.	.		
	END IF	ELSE		
		.		
		.		
		END IF		

Skoky :

Návěští se definuje jako xxx::
Při volání se používá GOTO xxx , GOSUB xxx
Ekvivalentem GOSUB xxx je uvedení návěští jako povelu.

Přepínač :

ON n GTO seznam náv.
ON n GSUB seznam náv.

Povely :

```

GTO      - ON n GTO; lze použít i GOTO, ale není to hezké
GSUB     - ON n GSUB; lze použít i GOSUB, ale není to hezké
END
RUN
BOOT     - návrat do monitoru
WAIT n   - čeká n milisekund
GOTO
GOSUB
RETURN
RETURN symbol - návrat lze nasměrovat jinam než za volání podprogramu !
FOR
NEXT
REPEAT   - |
UNTIL    - | podmínka na konci |
WHILE    - |                   | cykly
WEND     - | podmínka na začátku |
LOOP     - nekonečná smyčka LOOP - END LOOP
IF
ELSE
ON       - přepínač nebo ON ERROR GOTO
CLS      - [ CLS FRAME n v módu 800 2,4 maže obrazovku 0,1]
COLOR n1,n2 - v módu 700 barva tisku n1, pozadí n2; v módu
              800 tisk n1 (n2 nepřípustné)
PRINT
PRINT[n] ... - n: bit0=1: obrazovka
               bit1,bit2=1: tiskárny
INPUT     - zná i formu INPUT "text"; ...
INPUT "text"(n); ... omezí délku vst.řádku,má význam hlavně
                   při čtení znakových proměnných.
CURSOR n1,n2 - pozice cursoru nastavená na n1,n2
GET x#     - test klávesnice, vrací ASCII klávesy nebo 0
GETC x#    - test klávesnice, čeká na stisk klávesy, zahrnuje
               autorepeat
LET
DIM       - deklarace polí, používají se hranaté závorky,indexování od 0

```

DIM seznam x\$[...],n,seznam y\$[...],m ... - při deklaraci řetězců lze omezit max.délku

READ
DATA
RESTORE - je vhodné používat i před čtením první položky, po překladu na kazetu to bez něj nefunguje !

CLR - nuluje proměnné
LOAD name\$ - nahraje soubor name\$
LOAD name\$,adr - nahraje soubor n\$ od zadané adresy
SAVE name\$,typ#,from,size,exec,from1 - zapíše soubor name\$, typu typ#, od adresy from, délky size, start. adresa exec, data k uložení se vybírají od adresy from1.

DIR
FORMAT f\$ - "4 znakový řetězec" určuje, jak se budou tisknout čísla při povelu PRINT
1. znak řetězce - udává typ čísla (viz začátek náv.)
2. znak řetězce - SP (mezera) .. tiskne se před každé číslo mezera
- - (mínus) .. netiskne nic před č.
3. znak řetězce - SP .. tiskne jen záporná znaménka a místo kladného znaménka je SP
- + .. tiskne vždy obě znaménka
- - .. tiskne jen znaménka u záporných čísel
4. znak řetězce - SP .. jako 2. znak, ale za číslem
- - .. jako 2. znak, ale za číslem

DELETE name\$ - smaže soubor name\$
RENAME name1\$ TO name2\$
DEFAULT device\$ - device\$:
v diskové verzi zařízení : R, F1, F2, F3, F4
v připravované kazetové verzi : R - RAM DISK
C1 CMT 1/1
C2 CMT 2/1
C3 CMT 3/1
C4 CMT 3/2
pro CMT fungují jen příkazy SAVE, LOAD, LOADH, LOADD, EXEC, VERIFY, DIR.

POKE# - |
POKE - |
POKE% - | ukládá čísla jako 1, 2 nebo 4 bytová
POKE! - |
POKE\$ - uloží řetězec, 1. byte = délka
INP n,x# - přečte x z portu n |
OUTP n,x# - zapíše x# na port n | pro 8255 a 8253 funguje jen v módu 800, n=<0,255>

CALL n - volání strojáku na adrese n
RESUME NEXT - návrat po ON ERROR GOTO za příkaz, který vyvolal chybu
RESUME label - návrat po ON ERROR GOTO
DO - DO n - END DO cyklus, daný počtem opakování
BREAK - ON BREAK ERROR | ON BREAK BREAK |
přepínají se dva režimy
1) při stisknutí BREAKu se program zastaví je-li použito ON BREAK BREAK
2) při stisknutí BREAKu se hlásí BREAK ERROR, ale pokud je ošetřeno pomocí ON ERROR GOTO tak se nezastaví. Pozor !!! V tomto případě musí být chybový podprogram překládán s OPTION "B-", jinak dojde k zacyklení.

OPTION o\$ - o\$: "A+", "A-", "B+", "B-", "O+", "O-" - ovlivnění překladače
B...break, 0...přetečení, A...indexování pole
"+" povoluje, "-" zakazuje
Pozor !!! Při kontrole Break se zároveň testuje zaplnění strojového zásobníku. Přetečení zásobníku při vypnuté kontrole vede k okamžitému zřícení systému. Na druhé straně kontrolování čehokoliv brzdí běh programu, a to velice znatelně. Proto je vhodné používat kontroly při odladování a pak je vypnout.

RESET - RND! | SP | FORMAT
RESET SP vyčistí zásobník
RESET RND! nastaví pokaždé stejnou posloupnost náhodného čísla
RESET FORMAT nastaví standartní formát pro tisk: "- -"

HOME - kurzor na pozici 0,0
BELL - zvukový signál
ML n1,n2 ... - zahrne kódy do překladu (inline stroják). Na místě n1,n2 ... mohou ležet konstanty <0,255>, jména proměnných (ne polí) nebo návěští. Jména proměnných kompilátor nahradí 2-bytovou adresou jejich uložení, návěští přímo jeho 2-bytovou adresou.

INPH - |

OUTPH - | jako INP, OUTP, ale adresa je 16-ti bitová
 TAB - tabelátor, ale jen uvnitř PRINT
 TEST - TEST BREAK provede se jednorázové otestování BREAKu
 (má význam je-li OPTION "B-").
 TEST SP testuje jestli není plný zásobník (asi
 500 položek)
 LENGTH n, seznam x\$ - definuje max.počet znaků řetězců (šetří paměť).
 RANDOM - nastaví náhodnou posloupnost náhodného čísla
 INIT - načte soubor FORMAT a spustí ho (v RAM DISKU asi
 smaže vše a připraví adresář)
 EDIT x\$ - editace řetězce x\$
 EDIT (n);x\$ - editace řetězce s omezením délky vstupního řádku.
 CRT n - nastavuje zobrazovací režim (0 - 700, 1 až 4 - 800)
 PLOT n1,n2 - bod (PLOT FRAME n určí,do které obrazovky bude tisknout PLOT
 a LINE - graf.módy 2,4).
 LINE n1,n2 TO n3,n4 - čára z n1,n2 (nebo graf. ukazatel,je-li LINE TO n3,n4)
 na n3,n4
 GMOVE n1,n2 - grafický ukazatel na souřadnice n1, n2
 SETORG n1,n2 - střed souřadnic do n1, n2 (vzhledem k dolnímu
 levému rohu)
 PAL n1,n2 - palety,v 16-ti barevném módu se nastavuje blok palet takto:
 PAL 4*číslo bloku+paleta,barva
 GMODE n - grafický mód (0 - replace, 1 - reset, 2 - or, 3 - xor)
 SYNC n - n=0 synchronizace vypnutá, n=1 synchronizace VBLK
 funguje pro přepínání borderu, změny obrazovky a pro
 výměnu palet.
 PCG n,n1#..n8# - modifikuje znak n generátoru,obrazy řádků znaku n1...n8
 n=<0,511>
 VPOKE n1,n2,n3,n4,n5 - v módy 700 uložení videokódu n3 na
 na souřadnice n1, n2, barva n4,
 pozadí n5, n=<0,511>
 SOUND n1,n2,n3 - n1=<0,1,2> - zvukový kanál, n2=<0,1023> -
 frekvence, n3=<0,15> - hlasitost, platí
 pro PSG
 SSOUND n1,n2,n3 - n1=<0,3> - frekvence, n2=<0,1> - typ šumu,
 n3=<0,15> - hlasitost šumu, platí pro PSG
 BORDER n - barva borderu
 SHOW FRAME n - n=0,1 nastavení zobrazované obrazovky
 LOADH name\$ - přečte hlavičku do paměti, nastaví funkce
 FADR, FSIZE, FNAME\$.atd.
 LOADD adr - přečte tělo souboru,jehož hlavička byla přečtena pomocí LOADH
 od adr, která může být dána i implicitně z hlavičky.
 FKEY n,string\$ - definice klíčů F1 až F10
 FREQ n - zapne tón 1.1 MHz/n, n=0 vypíná, jen v módu 700
 BVPOKE n1,n2,n3,n4,n5,n6,n7 - přenos ze znakového generátoru
 od kódu n5 do čtverce n1xn2 znaků
 na souřadnice n3,n4, barvy n6,n7
 SHARP n - 0 = CENTRONICS, 1 = MZ Printer
 LOCK name\$ - blokování proti zápisu
 UNLOCK name\$ - odblokování proti zápisu
 EXEC name\$ - nahraje soubor n\$ a spustí ho (stroják, ne BASIC)

Funkce:

CURX# - |
 CURY# - | vrací pozici kursoru
 ERRN# - číslo chyby při běhu programu
 ERRADR - adresa chyby při běhu programu
 MAXADR - adresa posledního bytu programu +1
 SIZE - velikost volné paměti
 RND! - náhodná čísla <0,1>
 TIME - čas v sekundách
 TI\$ - čas ve tvaru "hhmmss"
 PI! - číslo pí
 TRUE# - | 255
 FALSE# - | konstanty 0
 ERRDEV# - číslo zařízení, které způsobilo chybu (5=F1/C1,6=F2/C2 ...,15=R)
 MZ800# - test, zda se jedná o počítač MZ 800
 KEYP# - test stisku klávesy (0 - není, 255 - je)
 FADR - |
 FSIZE - |
 FEXADR - | údaje o souboru
 FTYPE# - |
 FNAME\$ - |

VR32# - test, je-li připojená VRAM 32K
 LEFT\$ (string\$,n)
 RIGHT\$ (string\$,n)
 MID\$ (string\$,n1,n2) n2 znaků počínaje n1
 POS\$ (string1\$,string2\$,n) - hledá string1 ve stringu2 od znaku n, n je volitelné
 POINT# (n1,n2) - čtení VRAM, funguje v obou módech
 STRING\$ (n,string\$) - n-násobné opakování řetězce
 HEX\$ - |
 BIN\$ - | převod čísel na řetězce
 DEC\$ - |
 MOD% - |
 MOD! - | MOD (a,b) počítá a mod b
 MOD - |
 ODD# - test na lichost parametru
 STR\$
 SPC\$ - mezery, počet daný parametrem
 VAL#
 VAL%
 VAL!
 VAL
 PEEK#
 PEEK%
 PEEK!
 PEEK\$ - viz POKE\$
 PEEK
 CONV# - |
 CONV% - |
 CONV! - | konverze
 CONV\$ - |
 CONV - |
 CHR\$
 ADRS (prom) - vrací adresu proměnné
 HB# - vyšší byte 16-ti bitového čísla bez znaménka
 LB# - nižší byte 16-ti bitového čísla bez znaménka
 LEN#
 ASC#
 ADRSL - vrací adresu návěští
 VPEEK n1,n2 - udává videokód na n1, n2 (jen v módu 700)
 SGN#
 MULT2# - n*2
 DIV2# - n/2
 JOY# - čtení směru joysticku 0 (klávesnice), 1, 2, směr vpravo = 1 ,
 dále proti směru hodin
 FIRE# - čtení firu joysticku
 SGN
 SQR
 MULT2
 DIV2
 SGN%
 ABS%
 SQR%
 MULT2%
 DIV2%
 SGN!
 ABS!
 INT! - nejmenší celé číslo
 CUT! - odřízne desetinnou část
 SQR! - 2. mocnina
 SQRT! - 2. odmocnina
 LN!
 LOG!
 EXP!
 SIN!
 DSIN! - počítá ve stupních
 COS!
 DCOS! - počítá ve stupních
 TAN!
 DTAN! - počítá ve stupních
 ATN!
 DATN! - počítá ve stupních
 ROUND! - zaokrouhlí na celé číslo
 POW10! - mocnina 10
 RAD! - převod stupňů na rad
 DEG! - převod rad na stupně
 MULT2!

DIV2!

UP\$ - převede text na velká písmena

LOW\$ - převede text na malá písmena

Povely monitoru

C[n] - change line (implicitně 1)
Zde i jinde lze používat místo čísla řádku znak "a"=řádek
za řádkem naposledy editovaným,"l"=poslední řádek.
Po chybě se za znak "a" dosadí číslo chybového řádku !!!

D[n1[,n2]] - delete line (implicitně vše)

F[n1[,n2]]"text" - find text

I[n] - insert line (vkládá před n, implicitně za
poslední řádek)

L[n1[,n2]] - list (implicitně vše)

N n - page len, n>10 (pro tiskárnu), n... dekadicky

P+S n1,n2...n5 - nastavení kódů pro odstránkování, počet 0 až 5
v hexa

P+B n1,n2...nj - vysílá kódy přímo na tiskárnu

P+X - typ tiskárny CENTRONIX

P+P - typ tiskárny MZ

P:povel - výstup povelu na tiskárnu

S - status list - vypisuje - volnou paměť pro BASIC
program (zdroják)

- page len (viz povel N)
- skip kódy (odstránkování
viz P+S)
- počet bytů vyhrazených
pro tabulku symbolů

ST n - nastaví počet bytů pro tabulku symbolů(n - hexa)

T n1,n2,n3 - přesune n1-n2 před n3

V4 - 40 znaků na řádek

V8 - 80 znaků na řádek

Z - smazání obrazovky

Q - opuštění BASICu, restart z monitoru G1200

J - přeloží a spustí přeložený program

JT - přeloží a nespustí přeložený program

JSR jm.zař."name" nahraje knihovnu (v paměti přitom nesmí být program)

JS - nahraje vlastní odladěný strojový program za knihovnu
!!! po tomto povelu je poničen vlastní editor)

Xod - o operace d zařízení
disková verze : d=R, F1, F2, F3, F4
operace pro R : S, D, K, R, X, L, F
pro F : S, B, D, K, R, P, U, X, L, F
kazetová verze : d=R, C1, C2, C3, C4
pro R : stejně jako u diskové verze
pro C : S, V, L, X, D

S....save [n1[,n2]]"name"

B....save [n1[,n2]]"name", pokud existuje, přepíše
(S hlásí chybu)

D....directory

K....kill "name" smazání souboru

R....rename "staré jméno"="nové jméno"

P....protect "name" blokování proti zápisu

U....unprotect "name" odblokování proti zápisu

X...."name" spustí program OBJ

L...."name" LOAD

F....format, viz povel INIT v BASICu

V....verify (jen pro kazetovou verzi)

Ovládací klávesy (jen pro kazetovou verzi)

šipka vpravo, vlevo - posun kursoru

SHIFT šipka vpravo, vlevo - posun kursoru na konec, začátek
řádky

šipka nahoru - o 4 znaky vpřed

šipka dolů - o 4 znaky vzad

SHIFT šipka nahoru - výmaz do konce řádky

SHIFT šipka dolů - výmaz celé řádky

INST - mód insert/overwrite

BLANK (nad CR) - výmaz znaku před kursorem, posun
kursoru vlevo

DEL - výmaz znaku pod kurzorem
F1 až F5, SHIFT F1 až F5 - funkční klíče 1 až 10
ESC - ESCAPE

Řídící kódy (pro print, nebo i z klávesnice)

ESC R0....US ASCII
ESC R1....US ASCII
ESC R2....Deutsch.
ESC R3....UK ASCII
ESC lomená závorka....Normal
ESC vlnovka....Inverse
ESC Ezvýrazněné písmo (jen v módu 800)
ESC Fnormální písmo (jen v módu 800)
ESC -0nepodtrhává (jen v módu 800)
ESC -1podtrhává (jen v módu 800)

03....CR, LF
04....kurzor vpravo
05....kurzor nahoru
07....BELL
09....TAB
0A....kurzor dolů
0C....CLS
0D....CR, LF
13....kurzor vlevo
18....kurzor dolů
1A....CLS
1B....ESC
1D....HOME
1E....není-li kurzor na 1. pozici řádku - CR, LF

Pro kazetovou verzi platí u jmen souboru (monitor i BASIC):

"?name" prohlíží kazetu, vypíše jména nalezených prog.
na obrazovku a pokud souhlasí jméno programu,
tak jej nahraje
"*name" to samé, ale nevypisuje nic na obrazovku
"? " nebo "*" nahraje první co najde na kazetě
"name" nahraje program uvedeného jména, ale hlásí
chybu, že soubor nebyl nalezen, pokud nesouhlasí
název u 1. nalezeného programu

Kódy čtené z klávesnice

(GET v BASICu)

TAB....09
NIC....11
CR0D
ESC....1B
INS....16
DEL....07
CLR....0C
BRK....03
HOME...1D
šipka vlevo....13
šipka vpravo...04
šipka nahoru...06
šipka dolů.....01
SHIFT šipka vlevo....05
SHIFT šipka vpravo...18
SHIFT šipka nahoru...17
SHIFT šipka dolů.....1A

Chybová hlášení

proměnná ERRDEV#:

1 = Editor
2 = Compiler
3 = Tiskárna
6 až 9 = Disky nebo Kazety (čtyři)
15 = RAM DISK
18 = ??? (vadná grafická operace)
19 = Chyba ve strojáku
20 = ???

proměnná ERRN#:

0 BRK break
1 S syntax error
2 V value error (chyba hodnoty)
3 SYMNE symbol not existing error
4 LNE line not existing error
5 MCAP memory capacity error
6 MPRO memory protection error
7 STRL string length error
8 SYMSTCAP symbol stack capacity error
9 LD loading error
10 SV saving error
11 VER verifying error
12 STYP file type error
13 SNE file not found error
14 SAE file already existing error
15 NSP no file space error
16 NE not existing error (u Ram Disku)
17 NFRM unformat error
18 WPRO write protection error
19 TMF too many files error
20 DVNE device name error
21 CHK check error (možná u Ram Disku)
22 HRD hardware error
23 NRED not ready error
24 MOD mode error (např. graf. operace v 700 režimu)
25 STO stack overflow error
26 CMSQ command sequence error (např. se přehází cykly)
27 STYP ???
28 O overflow error
29 SYMDD double defined symbol error
30 NES structure error (jako chyba sekvence)
31 RD read error (u READ,DATA)
32 TYPM type mismatch error
33 A array error (přetečení indexu pole)
34 LCK lock error (vadný adresář u disků)
35 DSKTYP disk type error
36 NPRTN unprintable error (číslo chyby mimo výše uvedenou oblast 1 až 35)

----- konec -----