

## Hypra-Soft-Basic V1.0

Here is a new language which provides a superset of BASIC commands. In addition to those in Atari BASIC, there are approximately 60 new commands / functions available. Hypra-Soft BASIC enables faster program execution and optimises the space used for programs.

It was decided that several commands missing from Atari BASIC needed to be included. For example, there are commands for Player / Missile graphics and for memory manipulation.

In addition to this, there are numerous luxurious commands such as for a multi-colour mode in Graphics mode 0.

As you can now use pre-built routines, you will find that your code is faster and uses less memory.

The use of HS-BASIC will cost an additional 8KB over the amount of space used by Atari BASIC, however HS-BASIC is more powerful than Turbo-Basic-XL and BASIC-XL in some areas.

In order for everything to be a bit clearer, you may wish to consult the demo programs.

Here is a brief description of the new commands.

### Commands

#### INFO

The INFO command displays which areas of memory are occupied by the BASIC language. The user should avoid using these areas for storing their own data.

Syntax:        INFO

#### VAR

VAR lists the current variable table. Here you will see the variable name, content, actual length, maximum length and array dimensions.

Syntax:        VAR

#### Floppy Commands

DIR, LOCK, UNLOCK, ERASE and RENAME

These commands are similar to common DOS commands.

Syntax:

```
DIR "D:filename.ext"  
DIR FILE$
```

#### HELP

The HELP command lists all new commands and functions of Hypra-Basic which were not in Atari BASIC.

Syntax:        HELP

#### BOOT

This command completes a coldstart boot after the confirmation message is confirmed. If the cartridge or ROM version of Hypra-Soft-Basic is used, this is retained in memory.

Syntax:        BOOT

#### BREAK

This command allows the BREAK button to be switched on or off. A Boolean expression can be used as a parameter. If true, the button is on. If false, the button is switched off.

Syntax:        BREAK 1   ==> On  
              BREAK 0       ==> Off  
              BREAK 12=12 ==> On  
              BREAK 2=3   ==> Off

## HANDLER

The handler command lists all active handlers such as the editor, keyboard, screen or DOS. For each, it's identification letter is displayed (i.e. D:, E:, S:, K:).

Syntax:         HANDLER

## IOCB

The IOCB command displays which IO channel and data direction and open device is using.

Syntax:         IOCB

## UHR (German for 'CLOCK')

Computers which are equipped with the 16K Bibomon from CompyShop can display the time with this command.

Syntax:         UHR 1 ==> On  
                  UHR 0 ==> Off

## SYSTEM

The system command will dump information about the operating system vectors to the screen. You will find information such as interrupts and where the display list is located.

Syntax:         SYSTEM

## BGET / BPUT

These commands can load or save a block of data. This requires a handle for the channel to use, a variable for the start address of the block and another for the length of the data block. As a warning, only variables may be used, not actual values.

Syntax:         BGET #1,AD,LN  
                  BPUT #1,AD,LN

## MOVE+ , MOVE-

These commands will move a block of memory. With 'MOVE+', the first byte is copied first and the last byte last. With 'MOVE-', the opposite is true. The first parameter is the source address, the 2nd is the destination address and the 3rd is the length of the block of data to be copied.

Syntax:         MOVE+ source,destination,len  
                  MOVE- source,destination,len

## STORE

The STORE command will fill an area of memory with a value. The first parameter is the start address, the 2nd is the length of bytes to fill and third is the value to be used for filling.

Syntax:         STORE start,len,byte

## DPOKE

Many vectors use 16-bit addressing. This command will poke the high and low bytes of a vector.

Syntax:         DPOKE AD,BYTE (0 <= byte <= 65535)

## SCREEN

In order to speed up the Atari by 30%, you can switch the display on or off.

Syntax:         SCREEN 0     ==> Off  
                  SCREEN 1     ==> On

## BIB

Owners of the 16K-Bibomon can use this command to jump into the monitor. Using the Q key, a BASIC program can continue its execution.

Syntax:         BIB

## SHELTER

The SHELTER command writes a BASIC file out so that only RUN "D:\*.\*)" will be able to load and start it. Such programs can no longer be listed or changed in any way.

Syntax:         SHELTER "D:FILENAME.BAS"

## CLS

CLS stands for 'CLear Screen'. This command clears a text screen or a graphics mode's text window.

Syntax: CLS

## DO / OD

With DO and OD you can create a new loop. To get out of the loop, the EXIT command needs to be used. If 'OD' is used without a prior 'DO', an error is thrown.

Syntax: DO: ..... :OD

## EXIT

EXIT can be used to exit from DO / OD loop. After the EXIT command, a line number is required to show where the program should continue from.

Syntax: EXIT line

## REPEAT - UNTIL

The commands between the 'REPEAT' and 'UNTIL' will run until the Boolean expression is true.

Example: REPEAT: A=A+1: UNTIL A=11

Syntax: REPEAT: ... .. :UNTIL bool

## REN

REN stands for 'RENUMBER'. With this command, the line numbers will be renumbered. The first parameter is the first line of the new program. The 2nd parameter is the spacing between the lines.

Syntax: REN beginning,distance

## BASIC 0

HS-BASIC and Atari BASIC are up to a point compatible. If 'BASIC 0' is ran before executing a BASIC program with none of the new commands in, the program will be ran as if it is in Atari BASIC.

## BASIC 1

If you have a normal Atari BASIC program loaded, this will be ran flawlessly in HS-BASIC.

## Player-Missile-Graphics

Here are some commands to simplify working with player missile graphics (PMGs). The commands will work if PMGGRAPHICS 1 or 2 was already executed in order to switch on the PMGs. In addition, the storage area for the PMGs needs to be initialised. The players use the numbers 0 to 3 and the missiles use 4 to 7.

## PMGRAPHICS

With PMGRAPHICS, the PMGs can be switched on or off.

PMGRAPHICS 0 - switches them off

PMGRAPHICS 1 - switches them on in double line resolution

PMGRAPHICS 2 - switches them on in single line finer resolution

## PMCLR

The mandatory parameter for this is the player number. If the parameter is 0 to 3, the respective player is cleared. If a number above 3 is supplied, the whole of missile memory is cleared.

Syntax: PMCLR nr

## PMCOLOR

The PMCOLOR command sets the colour and players and missiles. The first parameter is the player / missile number, then the 2nd is the colour (0-15) and the third is the brightness/luminance (0-15).

Syntax: PMCOLOR No,col,lum

## PMWIDTH

PMWIDTH sets the width of a player / missile set. 0 is normal, 1 is double and 2 is quadruple width.

Syntax: PMWIDTH nr,width

## PMPOS

This is the horizontal position of the object.

Syntax: PMPOS nr,x

## PMDOWN/ PMUP

These commands control where the the player/missile is vertically. The first parameter is the player / missile to work on and the 2nd is for how many scanlines that they should move.

Syntax: PMUP nr,value ==> value = number of scanlines to move up  
PMDOWN nr,value ==> value = number of scanlines to move down

## PMADR

When the PMGs are activated, this will return the address of the graphics for each player or missile.

Players (0-3)

Missiles (4-7)

Syntax: AD=PMADR(0):AD=PMADR(1)

## PM

A PMG resolution can be decided with this command.

0 ==> PM off

1 ==> PM coarse (double sized)

2 ==> PM fine (single sized)

Syntax: PM(nr)

## Features

Here are additional useful commands.

## DPEEK

This is the opposite of DPOKE. This is given a 16 bit address and a byte value is returned.

Syntax: A = DPEEK(adr)

## ERR / ERL

These command give you the error number and error line.

Syntax: A=ERR(0)

A=ERR(1)

## HSTICK

With HSTICK one can determine the direction of a joystick. The joystick number needs to be specified as a parameter.

Left / Left-Up / Left-Down = -1

No horizontal movement = 0

Right / Right-Up / Right-Down = 1

Syntax: A=HSTICK(0)

B=HSTICK(1)

## VSTICK

The same as HSTICK except for vertical movement of the joystick.

## RAND

RAND(number) generates a random number between 0 and the specified number.

Syntax: A=RAND(16)

## MULTICOL

This command only works in graphics mode 0. It activates DLIs and the VBI in order to give each of the 24 rows a different background colour. The colour is defined with the command LINECOL.

Syntax: MULTICOL 0 ==> Off

MULTICOL 1 ==> On

## LINECOL

In conjunction with MULTICOL, the background colours in graphics mode 0 can be set. The first parameter is the line number (0-23), the 2nd is the colour (0-15) and the 3rd is the brightness/luminance (0-15).

## CLICK

On XL/XE machines, this will switch the keyboard click on or off.

Syntax:           CLICK 0                   ==> Off  
                  CLICK 1                   ==> On

## WSL

The WSL function can only produce a value of 0 or 1. The parameter that you pass indicates the chance of a 0 or a 1 being produced. For example, WSL(0.25) will in a 3:1 ratio produce zeroes and ones. WSL(0) will always produce a zero and WSL(1) will always produce a 1.

Syntax:           A=WSL(number)

## **HYPRA-SOFT-BASIC V1.0**

This is the first publication about HYPRA-SOFT-BASIC. To find more information, you can take a look at the articles in Compy-Shop-Magazin in the August 1989 issue.

**Uwe Roeder** (Original version in German)

**Steve Nicklin** (English translation and rejig 12/06/2015 v1.0.0)