

HYPRA-SOFT-BASIC V1.0

Dieses Programm stellt Ihnen kurz die neuen Befehle des Hypra-Soft-Basics vor. Dieses neue Basic ist ein modifiziertes Atari-Basic. Das heißt für den Benutzer, daß die Geschwindigkeit original die des Atari-Basic ist und das alle alten bekannten Befehle auch hier noch gültig sind. Hypra-Soft-Basic hat rund 60 neue Befehle/Funktionen, die einen schnelleren Programmfluß und ein Speicherplatz optimiertes Programmieren ermöglichen.

HYPRA-SOFT-BASIC V1.0

Es wurde insbesondere darauf geachtet, daß Befehle die im ständigen Umgang mit dem Basic notwendig sind, endlich eingeführt wurden. Dies sind z.B. die Disk-Befehle, die Player/Missile-Befehle und die Speicherbefehle zum Verschieben und Füllen von Speicherbereichen. Dazu kommen dann noch zahlreiche Luxusbefehle wie der Multicolor-modus in GR.0

HYPRA-SOFT-BASIC V1.0

Da Sie nun mit einem Befehl ausführen können, was früher nur durch aufwendige Routinen möglich war, können viele Programme schneller ablaufen und belegen weniger Speicherplatz. Die Benutzung von HS-BASIC kostet Sie zwar ganze 8 Kilobytes, die das Hypra Soft-Basic mehr belegt als das Atari-Basic, doch ist das HS-Basic dafür in einigen Bereichen leistungsstärker als selbst das Turbo-Basic-XL oder das BASIC-XL.

HYPRA-SOFT-BASIC V1.0

Es folgt nun eine kurze Beschreibung der neuen Befehle.

Damit die Anwendung klarer wird, sehen Sie sich bitte auch die DEMO-Programme an.

*

INFO

Der Befehl INFO zeigt an welche Speicherbereiche vom BASIC wie belegt sind. Der User muß diese Bereiche meiden, wenn er eigene Datenmengen im Speicher ablegen will.

Syntax:

INFO

VAR

VAR listet die aktuelle Variablen-tabelle aus. Es werden bei Zahlenvariablen der Inhalt, bei Strings die tatsächliche und die maximale Länge und bei Arrays die Dimensionierung mit angegeben.

Syntax:

VAR

Diskettenbefehle

DIR

LOCK

UNLOCK

ERASE

RENAME

Mit diesen Befehlen können die gängigsten DOS-Befehle nun direkt angesprochen werden.

Syntax:

DIR "D:FILENAME.EXT"

DIR FILE\$

HELP

Der Befehl HELP listet alle neuen Befehle und Funktionen des Hypra-Soft-Basics auf, die das alte Atari-Basic nicht kannte.

Syntax:
HELP

BOOT

Der Befehl BOOT löst einen Kaltstart aus, nachdem die Rückfrage positiv beantwortet wurde. Bei der Cartridge oder ROM-Version des Hypra-Soft-Basic bleibt diese im Speicher erhalten.

Syntax:
BOOT

BREAK

Mit diesem Befehl läßt sich die BREAK-Taste an und ausschalten. Hinter dem Befehl wird als Parameter ein boolescher Ausdruck gefordert. Ist dieser wahr=1, so ist die Taste an; ist er falsch=0, so ist die Taste aus.

Syntax:
BREAK 1 - BREAK 12=12 = an
BREAK 0 - BREAK 2=3 = aus

HANDLER

Der Befehl Handler listet alle aktiven Handler wie z.B. Editor, Keyboard, Screen oder DOS auf. Zum jeweiligen Identifikationszeichen (D: E: S: K:) wird noch der Anfang der zugehörigen Treiber tabelle angegeben.

Syntax:
HANDLER

IOCB

Der Befehl IOCB zeigt an welcher IO-Kanal von welchem Gerät belegt ist, und für welche Datenrichtung er geöffnet wurde.

Syntax:
IOCB

UHR

Der Befehl UHR bewirkt, daß bei Rechnern, die mit einem 16K-Bibomon ausgerüstet sind, oberhalb des Bildschirms die Uhr zeile ein- und ausgeblendet werden kann.

Syntax:
UHR 1 =an
UHR 0 =aus

SYSTEM

Der Befehl SYSTEM gibt wichtige Adress-Vektoren des Betriebssystems an. Der User kann hier sehen, ob und wie Interrupts eingesetzt werden, wo die Display-List liegt und einiges mehr.

Syntax:
SYSTEM

BGET / BPUT

Mit diesen Befehlen kann ein Datenblock geladen oder gespeichert werden. Dazu muß einer Variablen die Anfangsadresse der Daten und einer anderen die Länge des Blocks zugewiesen werden. Achtung es können nur Variablen, nicht aber Zahlen eingesetzt werden.

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

MOVE+ / MOVE-

Mit diesen Befehlen kann ich einen Datenblock im Speicher verschieben. Bei MOVE+ wird mit dem ersten Byte begonnen und mit dem letzten Byte aufgehört. Bei MOVE- ist dies genau andersherum. Der erste Parameter gibt die Quell Adresse, der zweite die Ziel-Adresse und der dritte die Länge des Blocks an.

Syntax:
MOVE+ Quell,Ziel,Len
MOVE- Quell,Ziel,Len

STORE

Mit dem Befehl STORE kann ein bestimmter Speicherbereich mit einem Wert gefüllt werden. Der erste Parameter ist hier die Startadresse, der zweite die Länge und der dritte das Füll-Byte.

Syntax:
STORE Start,Len,Byte

DPOKE

Viele Vektoren enthalten 16-Bit Adressen. Um nun einen solchen Vektor zu ändern mußten bislang Low und Highbyte der neuen Adresse ermittelt werden und getrennt in die jeweiligen Adressen geschrieben werden. Der DPOKE Befehl erledigt dies alles selbst.

Syntax:
DPOKE AD,BYTE 0<= BYTE <= 65535

SCREEN

Mit dem Befehl SCREEN kann der Bildschirm aus- und wieder eingeschaltet werden. Ist der Bildschirm aus, so ist der Rechner bis zu 30% schneller.

Syntax:
SCREEN 0 =aus
SCREEN 1 =an

BIB

Besitzer eines 16K-Bibomon können mithilfe dieses Befehls in den Monitor springen. Wenn Sie den Monitor dann mit dem Befehl Q verlassen, können Sie sogar ein unterbrochenes Basic-Programm fortsetzen.

Syntax:
BIB

SHELTER

Der Befehl SHELTER speichert ein BASIC-Programm in der Art und Weise ab, daß es nur noch mit RUN"D:*.*)" geladen und gestartet werden kann. Solche Programme können dann nicht mehr gelistet oder verändert werden.

Syntax:
SHELTER"D:FILENAME.BAS"

CLS

CLS steht für Clear Screen. Dieser Befehl löscht in Gr.0 den Bildschirm und in den anderen Grafikstufen das Textfenster.

Syntax:
CLS

DO / OD

Mit DO und OD können Sie eine neue Schleife einrichten. Stößt das Basic auf den Befehl OD, so beginnt das Programm wieder bei dem letzten DO. OD ohne vorheriges DO löst eine Fehlermeldung aus. Diese Schleife kann nur mit EXIT verlassen werden.

Syntax:
DO:... .. :OD

EXIT

Mit EXIT kann eine DO-OD Schleife verlassen werden. Nach EXIT muß die Zeilennummer stehen, an der das Programm fortgeführt werden soll.

Syntax:
EXIT Zeile

REPEAT - UNTIL

Die Befehle oder Zeilen, die zwischen REPEAT und UNTIL stehen, werden solange wiederholt, bis der boolesche Ausdruck hinter UNTIL wahr also 1 ist.

Bsp.:
REPEAT:?A:A=A+1:UNTIL A=11

Syntax:
REPEAT:.... :UNTIL bool

REN

REN steht für Renumber. Mit diesem Befehl können die Zeilennummern neu durchnummeriert werden. Der erste Parameter gibt die erste Zeile des "neuen" Programms an. Der zweite Parameter ist der Abstand zwischen den Zeilen.

Syntax:
REN Anfang,Abstand

BASIC 0

Hypra-Soft-Basic und ATARI-Basic sind bis auf einen Unterschied kompatibel. Damit ein Hypra-Soft Basic Programm, in dem keiner der neuen Befehle benutzt wurde auch in ATARI-BASIC läuft, muß vor dem Abspeichern BASIC 0 eingegeben werden. Aus den Zuordnungen wie A=10 wird dann Let A=10. Wird ein solches Programm dann vom ATARI Basic geladen sieht alles wieder normal aus und funktioniert richtig.

BASIC 1

Hat man ein normales Atari-Basic Programm geladen, so läuft dies natürlich einwandfrei auch in HS-Basic. Listet man aber ein solches Programm, so sieht man bei Wertzuordnungen anstelle von A=10 Let A=10. Der Befehl BASIC 1 macht daraus dann die normale Form A=10.

Player-Missile Grafik

Es folgen nun einige Befehle, die Ihnen das Arbeiten mit PM-Grafik erheblich vereinfachen. Die meisten Befehle funktionieren aber nur dann, wenn ganz zu Beginn der Befehl PMGRAPHICS 1 oder 2 ausgeführt wurde, da dieser Befehl neben der Initialisierung der PM-Grafik auch den Speicherbereich der PM-Grafik festlegt. Die Player haben die Nummern 0 bis 3; Missiles die von 4 bis 7.

PMGRAPHICS

Mit PMGRAPHICS wird die PM-Grafik ein und ausgeschaltet. PMGR.0 schaltet die PM Grafik aus. PMGR.1 schaltet die zweizeilige, Auflösung ein. PMGR.2 aktiviert die einzeilige, feinere Auflösung.

Syntax:
PMGRAPHICS modus

PMCLR

Der verlangte Parameter ist die Playernummer. Ist er zwischen 0 und 3 so wird der entsprechende Player gelöscht. Ist er größer als 3 so wird der gesamte Missile-Speicher gelöscht.

Syntax:
PMCLR nr

PMCOLOR

Der Befehl PMCOLOR legt die Farbe von Player und Missiles fest. Ein Paar wie Player 0 und Missile 0 etc. hat dabei immer die gleiche Farbe. Der erste Parameter ist die Player/Missile Nummer, dann folgt die Farbe 0-15 und die Helligkeit ebenfalls im Bereich von 0-15.

Syntax:
PMCOLOR nr,col,lum

PMWIDTH

Mit PMWIDTH wird die Breite eines Players/Missile festgelegt. 0 ist einfach, 1 ist doppelt und 2 ist vierfach.

Syntax:

PMWIDTH nr,breite

PMADR

Wurde mit PMGRAPHICS die PM-Grafik aktiviert, so läßt sich mit PMADR(nr) die Startadresse eines des Players/Missiles mit der Nummer nr ermitteln. Player haben die Nummern 0-3; Missiles die von 4 bis 7.

Syntax:

AD=PMADR(0):AD1=PMADR(1):....

PM

Mit PM(0) läßt sich ermitteln, ob und welche PM-Auflösung aktiv ist.

0 PM=aus

1 PM grob (zweizeilig)

2 PM fein (einzeilig)

Syntax:

A=PM(0)

PMPOS

Mit PMPOS kann die horizontale Position der Objekte festgelegt werden.

Syntax:

PMPOS nr,x

PMDOWN / PMUP

Mit diesen Befehlen kann ein Player/Missile um eine bestimmte „Strecke“ nach oben oder unten bewegt werden. Der erste Parameter ist wie immer die Player Nummer, der zweite gibt an um wieviele Bytes das Objekt nach oben verschoben werden soll.

Syntax:

PMUP nr,anz nach oben

PMDOWN nr,anz nach unten

MULTICOL

Dieser Befehl funktioniert nur in Gr.0. Er aktiviert eine ganze Reihe von Interrupts (24 DLIs und 1 VBI). Es kann nun für jede einzelne Gr.0 Zeile eine eigene Hintergrundfarbe definiert werden. Dies geschieht über den Befehl LINECOL.

Syntax:

MULTICOL 0 aus

MULTICOL 1 an

LINECOL

Ist MULTICOL aktiv, so kann nun für jede Zeile eine eigene Hintergrundfarbe gewählt werden. Der erste Parameter gibt dabei die Zeile (0 bis 23), der zweite die Farbe (0 bis 15) und der dritte die Helligkeit (0 bis 15) an.

Syntax:

LINECOL zeile,col,lum

KLICK

Mit diesem Befehl läßt sich bei den XL/XE-Geräten der Tastaturklick an- und abschalten.

Syntax:

KLICK 0 aus

KLICK 1 an

Funktionen

Es folgen nun die zusätzlichen Funktionen.

DPEEK

Dies ist das Gegenstück zu dem Befehl DPOKE. Mit dieser Funktion kann ein 16 bit Adress-Vektor direkt ausgelesen werden, ohne erst High und Low-Byte getrennt zu ermitteln und umzurechnen.

Syntax:

A=DPEEK(adr)

ERR / ERL

Mit diesen Funktionen läßt sich die Fehlernummer und die Zeilennummer nach einem Error ermitteln.

ERR = Fehler - ERL = Zeile

Syntax:

A=ERR(0)

A=ERL(0)

HSTICK

Mit HSTICK läßt sich der horizontale Bewegungsanteil des Joysticks ermitteln. links, links-oben, links-unten = -1 keine horizontale Bewegung = 0 rechts, rechts-oben, r.-unten = 1

Syntax:

A=HSTICK(0):B=HSTICK(1)

VSTICK

Mit VSTICK läßt sich der vertikale Bewegungsanteil des Joysticks ermitteln.

hoch, l.-hoch, r-hoch = -1

keine vertikale Bewegung = 0

runter, l-runter, r-runter = 1

Syntax:

A=VSTICK(0):B=VSTICK(1)

RAND

RAND(zahl) erzeugt eine Zufallszahl zwischen 0 und zahl.

Syntax:

A=RAND(16)

WSL

Die Funktion WSL kann nur 0 und 1 werden. Der Parameter von WSL muß zwischen 0 und 1 liegen. Dieser Parameter gibt die Wahrscheinlichkeit an mit der WSL 1 wird. Bei WSL(0.25) liegt diese bei 0.25, oder besser gesagt bei 1 zu 3. Die Funktion kann nun 0 oder 1 werden, aber es ist hier 3 mal so wahrscheinlich, daß sie 0 wird. WSL(0) ist immer 0 und WSL(1) ist immer 1.

Syntax:

A=WSL(zahl) 0<=zahl<=1

*

HYPRA-SOFT-BASIC V1.0

Dies ist erst einmal alles.

Mehr Informationen entnehmen Sie bitte dem zugehörigen Artikel im Compy-Shop-Magazin August 1989.