



## ZX81 \*\*Basic Version 1.0 Handbuch

\*\*Basic ist eine Erweiterung für das ZX81-Basic, die ohne Hardwareänderungen auf jedem ZX81 lauffähig ist.

Mit \*\*Basic ist es möglich, Befehle des ZX81-Basics zu ändern oder mit neuen Befehlen zu ergänzen.

Neuen Befehle werden mit dem Mathematik-Symbol des ZX81-Basic "\*\*\*" für Power bzw. Potenz eingegeben. Im Listing wird das "\*\*\*" unterdrückt, so dass der neue Befehl linksbündig erscheint.

Beispiel:

```
10 REM NEUE SYNTAX
20 STATUS
30 STOP
```

\*\*Basic erweitert dazu die Syntax um das neue Keyword "\*\*\*". Zur Eingabe eines neuen Befehls muss deshalb zuerst SHIFT-H gedrückt werden. Dies ergibt das Zeichen "\*\*\*" in der Editor-Zeile. Danach kann der neue Befehl in einzelnen Buchstaben eingegeben werden. Durch diese Syntax ist ein Programm, das unter \*\*Basic geschrieben wurde, auch mit dem normalen ZX81 Basic oder anderen PC-Programmen (ZXtool, ZX81List, etc.) als Listing darstellbar. Dort wird allerdings das "\*\*\*" nicht unterdrückt und erscheint im Listing.

Die neuen Befehle können im Direktmodus, im Zeileneingabemodus und natürlich zur Programmlaufzeit benutzt werden. Ein neuer Befehl wird allerdings bei der Eingabe noch nicht auf Syntaxfehler untersucht. Fehler werden erst im Programmlauf oder bei der Abarbeitung im Direktmodus festgestellt.

Zusätzlich zu den neuen Befehlen gibt es in \*\*Basic Hotkeys, mit denen das Verhalten von \*\*Basic während des Programmlaufs gesteuert werden kann.

Eine weitere Änderung ist, dass ein Programm-Abbruch bzw. BREAK nur noch mit SPACE und gleichzeitig gedrücktem NEWLINE erfolgt. Mit dieser Änderung ist SPACE oder £ über INKEY\$ in Programmen nutzbar.

Im Gegensatz zum original Sinclair-Basic hat \*\*Basic nun auch einen blinkenden K-, L-, F- und G-Cursor.

Zusätzlich dazu sind die Fehlermeldungen erweitert. \*\*Basic meldet sich beispielsweise nach einem fehlerfreien Programmlauf statt mit „0/10“ nun mit „READY“. Im Fehlerfall meldet es statt „2/10“ nun „ERROR 2 IN LINE 10“.

Damit kann man sofort sehen, dass \*\*Basic aktiv ist.

## ***Systemvoraussetzungen***

Für \*\*Basic wird ein ZX81 mit mindestens 16k Speicher und originale Sinclair 8k-Basic-Rom benötigt.

Zusätzlicher Speicher im Bereich 8k-16k oder über 32k (mit M1NOT-Schaltung) kann für \*\*Basic genutzt werden. Beim Einsatz von zusätzlichen Speicher bleibt der volle 16k Speicher für Basic frei.

Eine Grafikerweiterung des Speichers ist für die Funktion von \*\*Basic nicht notwendig. Selbstverständlich kann \*\*Basic mit HRG-Programme zusammen genutzt werden, solange es keine Adresskonflikte mit HRG-Daten oder HRG-Programmteilen gibt.

\*\*Basic läuft kann auch auf dem Emulator „EightyOne“ (Version 1.0 10/03/08) bei entsprechend eingestellten Hardware-Optionen für ROM und RAM genutzt werden.

## Speicherbelegung

Es gibt drei Versionen von \*\*Basic, die für unterschiedliche Speicherbereiche generiert sind.

### PowerBasic-8k.p

Diese Version lädt sich in den Speicherbereich zwischen 8k und 16k. Damit ist der komplette 16k Speicher für Basic nutzbar.

### PowerBasic-16k.p

Diese Version ist auf jedem ZX81 lauffähig. Sie läuft komplett im 16k Speicher. Dazu erniedrigt sie automatisch RAMTOP und kopiert sich darüber. RAMTOP muss deshalb vorher auf den Originalwert für 16k Speicher stehen. Andernfalls bricht die Installation mit einem Fehler ab.

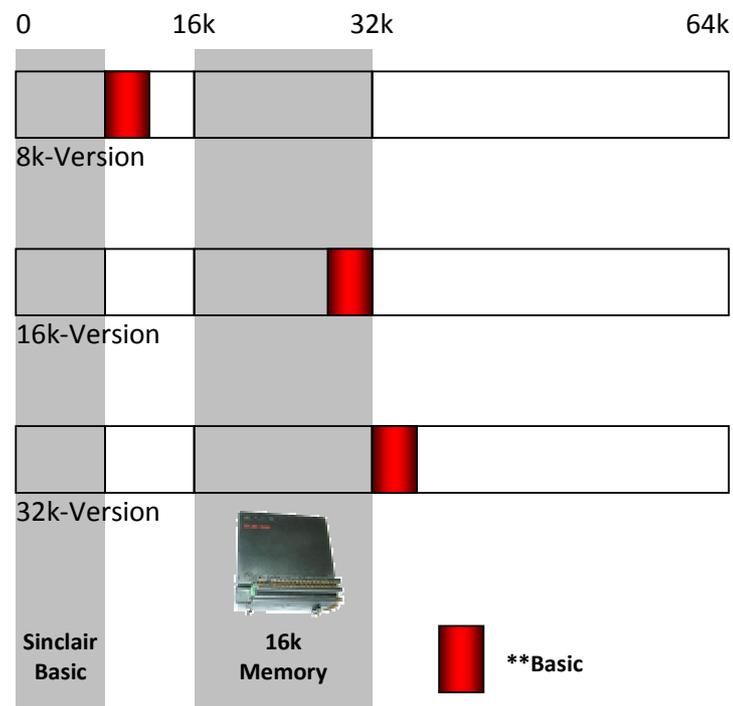
Für das Basic stehen nach der Installation ca. 12kByte zur Verfügung.

### PowerBasic-32k.p

Diese Version lädt sich in den Speicherbereich über 32k. Um dort ein Maschinenprogramm laufen zu lassen, muss eine M1NOT-Schaltung im ZX81 integriert sein! Für Basic steht dann der komplette 16k Speicher zur Verfügung.

## Adressbelegung

Das folgende Diagramm zeigt die Lage der verschiedenen Versionen im Adressraum.



## **Start von \*\*Basic**

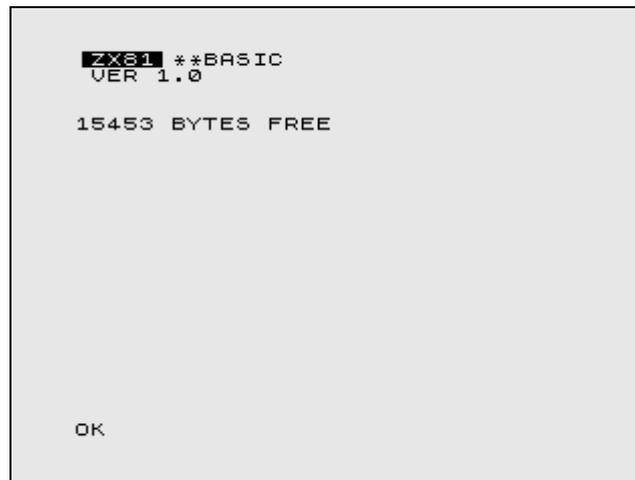
### **Installation**

Nach dem Laden der entsprechenden Programmversion muss diese mit RUN gestartet werden.

\*\*Basic installiert sich damit automatisch in den vorgesehenen Speicherbereich und ist sofort aktiviert.

Danach kann das Basic-Ladeprogramm mit NEW gelöscht werden.

Nach jedem NEW meldet sich \*\*Basic mit seiner Statusmeldung, die die Version von \*\*Basic und den für Basic verfügbaren Speicher anzeigt.

A screenshot of a ZX81 terminal window showing the status message for \*\*Basic. The text is displayed in a monospaced font on a light gray background. The message reads: 'ZX81 \*\*BASIC' on the first line, 'VER 1.0' on the second line, '15453 BYTES FREE' on the third line, and 'OK' on the fourth line.

```
ZX81 **BASIC
VER 1.0
15453 BYTES FREE
OK
```

Diese Statusmeldung kann auch mit dem Befehl \*\*STATUS angezeigt werden.

## **HotKeys**

Für die Debug-Funktionen: BREAK, Pause, Trace und Singlestep gibt es die folgenden Hotkeys:

NEWLINE + SPACE	= BREAK
SHIFT + T	= Trace an/aus
SHIFT + S	= Singlestep
SHIFT + SPACE	= Pause

Solange SHIFT + SPACE gedrückt wird, hält die Abarbeitung von Programmzeilen an. Beim Loslassen der Tasten läuft das Programm normal weiter.

**neue Befehle****\*\*STATUS**

Gibt die Version von **\*\*Basic** und den momentan freien Speicherbereich aus. Die angezeigte Speichermenge ist der verbleibende Adressraum zwischen den Variablen und dem Prozessorstapel.

**\*\*TRACEON****\*\*TRACEOFF**

Der Befehl **\*\*TRACEON** startet den Trace-Modus. Mit **\*\*TRACEOFF** wird der Trace-Modus wieder abgeschaltet. Im Trace-Modus wird die aktuell abgearbeitete Programmzeilen-Nummer am rechten Bildschirmrand angezeigt. Die Zeilennummern rollen dabei automatisch nach oben, so dass immer die letzten 20 Zeilennummern im Bildschirm ablesbar sind.

**\*\*STEPON****\*\*STEPOFF**

Der Befehl **\*\*STEPON** startet den Singlestep-Modus. Mit **\*\*STEPOFF** wird der Singlestep-Modus wieder abgeschaltet. Es wird jeweils nur eine Programmzeile ausgeführt und dann automatisch eine Pause gemacht. Nach Drücken von S wird jeweils eine weitere Zeile abgearbeitet. Der Singlestep-Modus kann auch mit SPACE frühzeitig beendet werden.

**\*\*DATA D1[, D2, D3, .. Dn]****\*\*RESTORE [Zeilennummer]****\*\*READ Variable**

Diese Befehle dienen der Datenspeicherung. In der **\*\*DATA**-Zeile können ein oder mehrere String- oder numerische Ausdrücke ggf. mit Komma getrennt stehen. In einem Programm können mehrere **\*\*DATA**-Zeilen stehen. Die einzelnen **\*\*DATA**-Zeilen müssen dabei nicht direkt aufeinanderfolgen. Es können beliebige andere Basic-Zeilen zwischen den **\*\*DATA**-Zeilen stehen.

**\*\*RESTORE** stellt den Lesezeiger auf eine **\*\*DATA**-Zeile. Ist keine Zeilennummer angegeben so wird die erste **\*\*DATA**-Zeile im Programm selektiert. Durch Angabe einer Zeilennummer kann der darauffolgende **\*\*READ** auf unterschiedliche Daten zugreifen. Vor der Benutzung von **\*\*READ** muss ein **\*\*RESTORE** ausgeführt sein.

**\*\*READ** liest den Wert eines Ausdrucks aus der **\*\*DATA**-Zeile in die angegebene Variable ein. Eine Stringvariable kann nur mit einem String-Ausdruck eingelesen werden. Eine numerische Variable kann nur mit einem numerischen Ausdruck eingelesen werden.

Ist der letzte Ausdruck einer **\*\*DATA** Zeile eingelesen so wird der folgende **\*\*READ** den ersten Ausdruck aus der darauffolgenden **\*\*DATA**-Zeile einlesen. Wenn keine weiteren Daten folgen, oder ein falscher Datentyp benutzt wird, bricht **\*\*READ** mit einer Fehlermeldung ab.

## **Kompatibilität**

### **Sinclair Basic**

Alle Basic-Befehle des Sinclair 8k-Rom sind weiter nutzbar. Große Teile des Roms wie z.B. der Zeileneditor, die Arithmetik, die Videoausgaberoutine und auch der Zeichensatz werden von \*\*Basic mitbenutzt. Deshalb gleicht \*\*Basic vom grundsätzlichen Verhalten dem Sinclair-Basic.

Die Funktionen und die Lage von Systemvariablen, der Programmzeilen, des Bildspeichers und der Basic-Variablen ist identisch mit dem Sinclair-Basic.

Ein mit \*\*BASIC geschriebenes Programm ist mit Sinclair-Basic lauffähig, solange keine \*\*-Befehle benutzt werden. Ein \*\*-Befehl führt unter Sinclair-Basic zu einer Fehlermeldung.

### **Kompatibilität mit HRG-ms Version 2.7**

\*\*Basic-16k kann nicht mit HRG-ms benutzt werden. Beide Programme liegen im selben Adressraum über RAMTOP.

\*\*Basic-32k kann mit Einschränkungen HRG-ms benutzen. Die Grafik-Bank 5 darf nicht benutzt werden, weil in diesem Speicherbereich \*\*Basic liegt.

\*\*Basic-8k kann ohne Einschränkungen HRG-ms nutzen. Es könne alle Grafik-Bänke ohne Gefahr benutzt werden.

### **Kompatibilität mit anderen Programmen**

Reine Basic-Programme sind uneingeschränkt unter \*\*Basic lauffähig.

Maschinensprache-Programme können ebenfalls ausgeführt werden. Allerdings können, wie im originalen Basic auch, Maschinenspracheprogramme möglicherweise die Funktionalität von \*\*Basic behindern, stören oder \*\*Basic sogar zum Absturz bringen.

## Beispielprogramme

### Trace

Der Lauf des folgenden Programms wird mit \*\*TRACEON sichtbar gemacht.

```

10  REM TESTPROGRAMM
20  TRACEON
30  GOTO 100+500*RND
100 GOTO 30
200 GOTO 30
300 GOTO 30
400 GOTO 30
500 GOTO 30
1000 TRACEOFF
1010 PRINT "FERTIG"

```

Ohne Trace kann der Programmlauf in diesem Beispielprogramm nicht eindeutig vorhergesagt werden. Mit TRACE wird er sofort sichtbar.

### SingleStep

In einigen Fällen ist es hilfreich, ein Programm zusätzlich zum TRACE Zeile für Zeile im Einzelschritt ausführen zu lassen und es dazu gezielt anzuhalten. Das ist mit \*\*STEPON möglich.

```

10  REM TESTPROGRAMM
20  TRACEON
100 FOR X=0 TO 100
150 IF X=50 THEN STEPON
200 PRINT ".";
300 NEXT X
320 STEPOFF
340 TRACEOFF
1010 PRINT "FERTIG"

```

Wenn x den Wert 500 erreicht, wird ab Zeile 150 jede weitere Zeile nur nach dem Druck der Taste S ausgeführt.

Mit einem Druck auf die Taste SPACE kann zu jeder Zeit der Einzelschritt frühzeitig beendet werden. Das Programm läuft ab dann wieder durchgängig weiter.

### Data, Restore, Read

Das folgende Programm zeigt, wie \*\*READ benutzt wird.

```

10  REM TESTPROGRAMM
20  DATA 3
30  DATA "PAUL",27
40  DATA "ESTHER",41
50  DATA "PETER",35
1000 REM
1010 RESTORE 20
1020 READ END
1030 FOR M=1 TO END
1040 READ N$
1050 READ AGE
1060 PRINT N$;" IST ";AGE
1070 NEXT M

```

Hier wird zuerst der numerische Wert für die Anzahl der Datensätze eingelesen. Danach werden in einer Schleife die Datensätze bestehend aus dem String für den Namen und dem numerischen Wert für das Alter eingelesen.