

## **Beschreibung des BASICODE-Übersetzers für BIC A5105**

Im RBASIC des BIC kann der Bascoder von Diskette (BLOAD "BACD3BIC.RMC",r) geladen werden. Nach dem Laden erscheint das Startbild mit einem Hinweis auf die erforderliche Versionsnummer des RBASIC (2.0). Nach ca. 12 Sekunden bzw. nach dem Druck einer beliebigen Taste erfolgt der Übergang ins Hauptmenü.

Um eine Kopie des Bascoders anzufertigen, muß dieser ohne den Zusatz ',r' geladen werden. Danach kann mit BSAVE "\_[CAS:]name",&h8800,&hC0EF,&h8800 eine Kopie auf Diskette bzw. Kassette angelegt werden. Für 'name' gelten dabei die Festlegungen des verwendeten Externspeichers.

Beim erstmaligen Start des Bascoders wird der RBASIC-Programmspeicher gelöscht. Beim Restart des Bascoders (mit PF3 bzw. CMD u. ENTER) bleibt dagegen ein evtl. vorhandenes RBASIC-Programm sowie der Inhalt des BASICODE-Textspeichers erhalten.

Im oberen Teil des Hauptmenüs wird der Ablauf des ausgewählten Modus schematisch dargestellt. In der unteren Hälfte erfolgt jeweils die Bedienerführung. 17 Modi sind mittels der Tasten A..Q wählbar. Mit ENTER kann ein gewählter Modus gestartet werden. Mit ESC kann in das Extra-Menü verzweigt werden. Die Möglichkeiten dieses Menüs werden ebenfalls in der unteren Bildhälfte angeboten.

Der Bascoder verwaltet für BASICODE-Programme bzw. -Dateien einen eigenen Textspeicher. Die Größe des noch verfügbaren und bereits belegten Speicherbereiches wird im Editor des Bascoders ständig in der obersten Zeile angezeigt. Vom Hauptmenü aus gelangt man mit K in diesen Editor.

Folgende Betriebsarten des Bascoders sind wählbar:

Modus ! Bedeutung

- 
- |   |   |
|---|---|
| A | ! Einlesen eines BASICODE-Programmes in den Textspeicher. Bei fehler- |
|   | ! freiem Lesevorgang erfolgt anschließend die Übertragung in den      |
|   | ! RBASIC-Speicher. Die benötigten Subroutinen werden automatisch      |
|   | ! eingefügt.  |
|   | ! Im Fehlerfall (Pruefsummenfehler !) erfolgt die Ausschrift          |
|   | ! 'Lesefehler'.   |
|   | !   |
| B | ! Übersetzen eines RBASIC-Programmes in die von BASICODE-3 geforderte |
|   | ! Form (ab Zeile 1000) in den Textspeicher. Bei erkannten Fehlern     |
|   | ! werden diese markiert und danach erfolgt der automatische Übergang  |
|   | ! in den Editor. Beginnend bei der ersten fehlerhaften Zeile wird der |
|   | ! Inhalt des Textspeichers angezeigt. Die fehlerhaften Stellen werden |
|   | ! invers dargestellt. Wird der Cursor auf eine solche Position be-    |
|   | ! wegt, so erscheint in der untersten Zeile eine entsprechende        |
|   | ! Fehlermeldung.  |
|   | ! War das übersetzte Programm fehlerfrei, so wird der Inhalt des      |
|   | ! Textspeichers im BASICODE-Format auf Kassette geschrieben. Im       |
|   | ! unteren Teile des Bildschirms erfolgt dabei die Bedienerführung.    |
|   | ! Der Nutzer erhält während des Schreibens Informationen über die     |
|   | ! Gesamtlänge des Programmes sowie über die Anzahl der noch zu        |
|   | ! schreibenden Bytes.   |
|   | !   |
| C | ! Einlesen eines BASICODE-Programmes in den Textspeicher.             |
|   | !   |
| D | ! Inhalt des Textspeichers (ASCII) im BASICODE-Format auf Kassette    |
|   | ! schreiben. Sind im Text noch Fehler (durch eine vorherige           |
|   | ! Übersetzung markiert !) vorhanden, so erfolgt keine Kassetten-      |
|   | ! ausgabe !   |
|   | !   |

```

E ! Einlesen einer BASICODE-Datei (Blöcke aus je 1024 Bytes) in den
! Textspeicher.
!
F ! Inhalt des Textspeichers (ASCII) als BASICODE-Datei im geforderten
! Format auf Kasette schreiben
!
G ! Inhalt des Textspeichers (ab Zeilennummer >= 1000) in die interne
! RBASIC-Form übersetzen und in den RBASIC-Programmspeicher laden
! (dieser wird zuvor gelöscht). Die benötigten Subroutinen werden
! eingefügt.
!
H ! Übersetzen eines RBASIC-Programmes in den Textspeicher (ASCII) mit
! gleichzeitiger Überprüfung der von BASICODE-3 geforderten Syntax.
!
I ! Inhalt des Textspeichers wird in den RBASIC-Programmspeicher
! eingefügt. Es werden keine Subroutinen erzeugt. Dabei wird auf
! vorhandene Zeilennummern und zulässige Zeilenlänge (<256 Zeichen)
! getestet.
!
J ! Inhalt des RBASIC-Programmspeichers in den BASICODE-Textspeicher
! (ASCII) übersetzen. Dabei werden alle Zeilen übernommen. Das
! RBASIC-Programm wird an ein ggf. vorhandene BASICODE-Programm
! angefügt.
!
K ! Es wird der BASICODE-Editor aufgerufen (verlassen mit ESC).
!
L ! Inhalt des Textspeicherinhaltes an einen Drucker ausgeben.
!
M ! Ausgabe des Textspeicherinhaltes im internen BIC-Format auf
! Kasette. Diese Datei kann im RBASIC mit LOAD "CAS:name" (wobei
! 'name' max. 6 Zeichen lang sein darf).
!
N ! Einlesen eines RBASIC-Programmes über den internen Kassettentreiber
! in den BASICODE-Textspeicher. Das Programm muß im RBASIC mit SAVE
! "CAS:name",a (d.h. im ASCII-Format) auf die Kasette geschrieben
! worden sein.
!
O ! Einlesen einer Diskettendatei (ASCII-Datei !) in den BASICODE-
! Textspeicher. Im unteren Bildteil wird zuvor das Inhaltsverzeichnis
! der Diskette (Laufwerk A) angezeigt.
!
P ! Inhalt des Textspeichers auf Diskette schreiben. Es wird ebenfalls
! zuvor das Inhaltsverzeichnis angezeigt.
!
Q ! Verlassen des Bascoders und Übergang ins RBASIC.

```

---

Bei der Arbeit mit Kassettendateien im BASICODE-Format werden keine Dateinamen verwendet. Deshalb sollte man für jede Datei den Stand des Bandzählwerkes notieren (falls vorhanden). Die von Kasette eingelesenen Daten werden während des Einlesens fortlaufend in der unteren Bildhälfte angezeigt. Fehlerhafte Zeichen werden dabei durch ein inverses Fragezeichen ausgeben. Im Fehlerfall erfolgt nach beendetem Einlesen die Ausschrift 'Lesefehler', ansonsten erfolgt die Ausschrift 'Keinen Fehler entdeckt'.

Der erkannte Anfang einer Datei wird durch die inverse Ausschrift 'Vorton' angezeigt. Wird das Einlesen an einer falschen Bandposition begonnen, so erfolgt dennoch die fortlaufende Ausgabe der gelesenen Daten auf dem Bildschirm. Jedoch kommt am Ende zwangsläufig die entsprechende Fehlermeldung. Wird beim Einlesen nicht der geforderte Typ (BASICODE-Programm an Stelle einer geforderten BASICODE-Datei bzw. umgekehrt) erkannt, so erfolgt ebenfalls eine dementsprechende Fehlermeldung.

Beim Einlesen von im RBASIC erzeugten Kassettendateien (Modus M) wird mit 'Welchen Namen ?' auf die Eingabe eines Dateinamens gewartet. Es werden Groß- und Kleinbuchstaben unterschieden ! Wird kein Name eingegeben, d.h. nur ENTER eingegeben, so wird die nächste (auf Kassette vorhandene Datei) eingelesen. Falls ein Dateinamen eingegeben wurde, so wird auf die Datei mit dem angegebenen Namen gewartet. In jedem Fall erfolgt die Ausschrift 'Ich suche..' . Wird eine Datei mit einem anderen Namen gefunden, so erfolgt keine Ausschrift in der sonst im RBASIC üblichen Form ('SKIP name'). Mit CTRL+STOP können die Datenübertragungen mit dem Kassettenrecorder jederzeit unterbrochen werden !

Sollen ins RBASIC übersetzte BASICODE-Programme gestartet werden, so ist im Laufwerk A die Zeichengeneratordatei 'ZS.ZG' erforderlich. In den beim Übersetzen automatisch eingefügten Subroutinen steht in Zeile 17 das Kommando zum Laden dieser Datei. (Dadurch ergeben sich im Grafikschild des BASICODE-3-Programmes besser lesbare Zeichen.) Falls der Bascoder ohne angeschlossene Disketteneinheit verwendet werden sollte, so muß in diesem Fall in der Zeile 17 das Kommando zum Laden der Datei (BLOAD "ZS.ZG",z) vorm Starten des Programmes gelöscht werden.

Im Editor des Bascoders kann der Cursor mit den Kursortasten gesteuert werden. Weiterhin kann mit den Funktionstasten PF1..PF4 im Text geblättert werden. Mit PF5 kann jeweils eine Zeile gelöscht werden. Mit DEL wird das Zeichen, auf welchen sich der Cursor befindet, gelöscht. Mit der ESC-Taste kann der Editor verlassen und wieder ins Hauptmenue gegangen werden.

Innerhalb des Bascoders werden nach verschiedenen Operationen im unteren Bildteil Meldungen ausgegeben. Diese werden nach einer angemessenen Pause gelöscht, allerdings kann durch Drücken einer beliebigen Taste das vorzeitige Löschen dieser Meldungen und damit eine schnellere Weiterarbeit erreicht werden. Notwendige Eingabe werden durch das Blinken des Cursors angefordert !