



Allen-Bradley

***BASIC-
Entwicklungs-
software***

(Bestellnummer 1747-PBASE)

Programmier- handbuch

AB Spares

Wichtige Anwendungshinweise

Elektronische Geräte unterscheiden sich in ihrer Arbeitsweise von elektromechanischen Geräten. Die Publikation "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Controls" (Publication SGI-1.1) beschreibt einige wichtige Unterschiede zwischen elektronischen und festverdrahteten, elektromechanischen Geräten. Aufgrund dieser Unterschiede und der vielfältigen Einsatzmöglichkeiten elektronischer Geräte müssen Sie als Verantwortlicher für die Anwendung dieses Gerätes sicherstellen, daß jede Anwendung dieses Gerätes die jeweiligen Anforderungen erfüllt.

Für Schäden, die direkt oder indirekt durch die Verwendung oder den Einatz dieser Geräte entstehen, kann Allen-Bradley Company keinesfalls verantwortlich oder haftbar gemacht werden.

Die Beispiele und Abbildungen in diesem Handbuch sind ausschließlich zur besseren Texterläuterung aufgeführt. Aufgrund der vielfachen Möglichkeiten und Anforderungen jedes einzelnen Verwendungszwecks kann die Allen-Bradley Company keine Verantwortung oder Haftung für tatsächliche Einsätze, die auf diesen Beispielen und Abbildungen beruhen, übernehmen.

Allen-Bradley Company übernimmt keine Patenhaftpflicht für die Anwendung von Informationen, Schaltungen, Geräten bzw. Softwareprogrammen, die in diesem Handbuch beschrieben werden.

Eine Wiedergabe des Inhalts dieses Handbuchs, ganz oder auszugsweise, ist ohne die schriftliche Genehmigung von Allen-Bradley nicht gestattet.

Besondere Hinweise in diesem Handbuch sollen den Anwender auf Sicherheitsmaßnahmen aufmerksam machen.



ACHTUNG: Weist auf Informationen und Verfahrensweisen oder Umstände hin, die zu Körperverletzungen oder sogar Lebensgefahr, zu Sachschaden oder wirtschaftlichem Verlust führen können.

Achtungshinweise helfen Ihnen:

- eine Gefahr festzustellen
- die Gefahr zu vermeiden
- die Konsequenzen zu erkennen

Wichtig: Weist auf Informationen hin, die äußerst wichtig für die erfolgreiche Anwendung und für das gründliche Verstehen des Produktes sind.

PLC ist ein eingetragenes Warenzeichen der Allen-Bradley Company, Inc.
SLC, SLC 500, SLC 5/01 und SLC 5/02 sind Warenzeichen der Allen-Bradley Company, Inc.
IBM ist ein eingetragenes Warenzeichen der International Business Machines, Incorporated.
LOGITECH ist ein Warenzeichen der LOGITECH, Inc.
Toshiba ist ein Warenzeichen der Toshiba America, Inc.
Microsoft ist ein eingetragenes Warenzeichen von Microsoft.
MOUSE SYSTEMS ist ein eingetragenes Warenzeichen der MOUSE SYSTEMS Corporation.
Epson ist ein eingetragenes Warenzeichen der Epson America, Inc.
HP ist ein eingetragenes Warenzeichen der Hewlett-Packard Company.

Zusammenfassung der Änderungen

Die nachstehenden Informationen fassen die Änderungen zusammen, die seit der letzten Drucklegung vorgenommen wurden.

Neue Informationen

Die folgende Tabelle führt Abschnitte auf, die neue Merkmale und zusätzliche Informationen bestehende Funktionen dokumentieren, und zeigt, an welcher Stelle im Handbuch diese neuen Hinweise zu finden sind.

Neue Informationen	Kapitel/Anhang
Dokumentationssatz für das BASIC-Modul	Vorwort
Hinweise zum Benutzerhandbuch des BASIC-Moduls	im gesamten Handbuch
Hinweise zum Design- und Integrationshandbuch des BASIC-Moduls SLC 500	im gesamten Handbuch
Sichern des Modulabbildes	9
Abrufen des Modulabbildes	9
Backplane-Makroaufrufe	Anhang A
Funktionen der 1771-DB/B-Backplane (Aufrufe zur Zahlenumwandlung und Backplane-Aufrufe)	Anhang A
Zeichenkettenroutinen	Anhang A

Inhaltsverzeichnis

Wichtige Anwendungshinweise	<u>I</u>
Zusammenfassung der Änderungen	<u>P-1</u>
Neue Informationen	<u>P-1</u>
Vorwort	<u>V-1</u>
Leser dieses Handbuchs	<u>V-1</u>
Zweck dieses Handbuchs	<u>V-2</u>
Dokumentationssatz des BASIC-Moduls	<u>V-3</u>
Literaturhinweis	<u>V-4</u>
Begriffe und Abkürzungen	<u>V-5</u>
Konventionen in diesem Handbuch	<u>V-6</u>
Technische Unterstützung durch Allen-Bradley	<u>V-7</u>
Produktspezifische Unterstützung	<u>V-7</u>
Technische Unterstützung	<u>V-7</u>
Fragen und Kommentare zu diesem Handbuch	<u>V-7</u>
Installation der Software	<u>1-1</u>
Funktionsmerkmale der Software	<u>1-1</u>
Anforderungen an die Hardware	<u>1-3</u>
Hardware für den Anschluß des Personalcomputers an das BASIC-Modul	<u>1-3</u>
Inhalt der Disketten	<u>1-4</u>
Installationsverfahren	<u>1-4</u>
Installation von 5.25-Zoll-Disketten	<u>1-5</u>
Installation von 3.5-Zoll-Disketten	<u>1-6</u>
Verzeichnisstruktur	<u>1-7</u>
Sichern der Software	<u>1-7</u>
Einführung in die Entwicklungssoftware	<u>2-1</u>
Starten der Software	<u>2-1</u>
Toplevel-Menü	<u>2-2</u>
Hauptmenü	<u>2-4</u>
Statuszeile	<u>2-4</u>
Meldungszeile	<u>2-5</u>
Funktionstasten	<u>2-5</u>
Aufruf des Hilfesystems außerhalb eines Programms	<u>2-6</u>
Aufruf des Hilfesystems von einem Programm aus	<u>2-7</u>
Ein typisches Pulldown-Menü	<u>2-7</u>
Wechseln zwischen Fenstern	<u>2-8</u>

Editieren eines BASIC-Programms	2-11
Generelles Editieren	2-11
Sichern eines Programms	2-12
Übersetzen eines Programms	2-13
Anschluß an das BASIC-Modul	2-14
Wechseln in den Terminal-Modus	2-14
Herunterladen eines Programms	2-15
Verlassen eines BASIC-Programms	2-16
Konfiguration der Software	3-1
Wahl des Menüs "Setup and Configuration"	3-2
Zusätze der Befehlszeile	3-2
Laden von Files von der Befehlszeile aus	3-4
Konfigurieren der Editieroptionen	3-4
Konfiguration der Anzeigeoptionen	3-7
Konfiguration der Maus- und Tastaturoptionen	3-7
Konfiguration der Filenamenerweiterung	3-7
Konfiguration der Druckeroptionen	3-11
Konfiguration der Speicheroptionen	3-11
Sicherungsfiles, Zwischenfiles und automatische Sicherung (Autosave)	3-13
Sicherungsfiles	3-13
Zwischenfiles	3-13
Automatische Sicherung (Autosave)	3-13
Standardeinstellung der Such- und Ersetzfunktion	3-14
Wahl des Terminalemulationsmodus	3-14
Sicherung der Konfiguration und Beenden	3-15
Editieren eines Programms	4-1
Anwendung einfacher Editiermethoden	4-1
Suchfunktionen	4-2
Anwendung von Blockfunktionen	4-4
Textformatierung, Rückgängigmachen und Aufheben	4-6
Erweiterte Editierfunktionen	4-8
Linienzeichnen	4-9
Verwendung der Taschenrechnerfunktion	4-10
ASCII-Tabelle	4-10
Verwendung von Fenstern	4-10
Tastenmakros	4-12
Ändern von aufgezeichneten Tastenfolgen	4-14
Editieren von Tastenmakros	4-15
Verwendung von Cursormarkierungen	4-17

Bearbeiten von Files	5-1
Das File-Menü	5-1
Wahl von Projektfiles	5-3
Anwendermenü	5-3
Anwenderdefinierte Programme	5-4
Anwendermenü	5-5
Benennung eines Programms	5-5
Wahl eines Filenamens	5-6
Wahl der Speicherauslagerung	5-6
Speicheranforderungen	5-8
Die Option "Use COMMAND.COM"	5-8
Anfangsverzeichnis (Starting Directory)	5-8
Programmieren mit der BASIC-Entwicklungssprache	6-1
Überblick über die Sprache	6-1
Übersetzungsanweisungen	6-2
Kommentar (Anmerkung)	6-2
Include	6-3
Label	6-4
Text	6-5
Zeilennummer	6-5
Inkrement-Anweisung	6-6
Fixpunkt (Breakpoint)	6-6
Debugging aktiviert Debugging deaktiviert	6-7
Makros	6-8
Erstellung eines Programmiermakros	6-8
Einlesen von Programmen aus separaten Files	6-9
Unterprogramm CALL 70, CALL 71	6-10
Programmiermakros	6-10
Verknüpfen eines Programmfiles	6-11
Erstellen eines zweiten Programms	6-11
Eingabe des Programms	6-11
Übersetzen	6-14
Zweites Programm nach der Übersetzung (SECOND.BAS)	6-15
Herunterladen	6-16
Ausführung	6-16
Drucken eines Programms	7-1
Einstellung des Druckers	7-1
Wahl des Druckers	7-2
Druckereinstellung	7-3
Druckergerät/-file	7-3
Serielle Drucker	7-4

Verwendung der Druckersteuerungen	7-5
Anzahl der Ausdrücke	7-5
Numerierung der Zeilen	7-5
Drucken eines Randes	7-5
Seitenvorschub	7-6
Drucken eines vollständigen Dokumentes bzw. eines Auszugs	7-6
Drucken im Hintergrund	7-6
Übersetzen eines Programms	8-1
Übersetzen eines Programms	8-1
Zweites Programm vor der Übersetzung (SECOND.BDL)	8-3
Zweites Programm nach der Übersetzung (SECOND.BAS)	8-4
Beseitigen von Übersetzungsfehlern	8-5
Suche nach dem nächsten Kompilierfehler	8-5
Konfiguration des Übersetzers	8-5
Kommunikation mit dem BASIC-Modul	9-1
Wechseln in den Terminal-Modus	9-1
Herunterladen von Files	9-3
Hochladen von Files	9-5
Übertragungen von Hexadezimalfiles	9-6
Hochladen von hexadezimalen Files	9-7
Herunterladen von hexadezimalen Files	9-8
Sichern des Modulabbildes	9-9
Wiederherstellen des Modulabbildes	9-10
Programm-Debugging	9-11
Debugging-Beispiel	9-12
Programmkorrekturen	9-15
Einrichtung der RS-232-Kommunikation	9-16
Autobaud	9-17
Allgemeine Einrichtungsparameter	9-17
Terminal	9-18
Einstellung des COM-Ports	9-18
Farben	9-19
Sichern der Einstellung	9-19
Einrichtung der DH-485-Kommunikation	9-19
Anschließen	9-20
Allgemeine Einrichtungsparameter	9-20
Terminal	9-21
Einstellung des COM-Ports	9-21
WHO ACTIVE (Aktive Netzknotten mit Terminal)	9-21
WHO LISTEN (Aktive Netzknotten ohne Terminal)	9-22
Farben	9-23
Sichern der Einstellung	9-23

Bibliothek der BASIC-Makros	A-1
Die Bibliothek im Überblick	A-1
Bildschirmfunktionen	A-2
Tastenfunktionen	A-3
SLC-Backplane-Funktionen	A-4
Uhrfunktionen	A-5
Batteriepufferung	A-6
Programmsteuerung	A-7
Zeichenkettenroutinen	A-7
[Ctrl C]-Funktionen	A-8
Portsteuerungsfunktionen	A-8
DH-485-Funktionen	A-9
1771-DB/B-Backplane-Funktionen	A-9
Die DOS-Verzeichnis-Shell	B-1
Verwendung der DOS-Verzeichnis-Shell	B-1
Funktionen der DOS-Verzeichnis-Shell	B-2
File-Funktionen der DOS-Verzeichnis-Shell	B-4

Vorwort

Lesen Sie dieses Vorwort, um sich mit dem restlichen Handbuch vertraut zu machen. Es werden die folgenden Themen behandelt:

- Leser dieses Handbuchs
- Zweck dieses Handbuchs
- Begriffe und Abkürzungen
- Konventionen in diesem Handbuch
- Technische Unterstützung durch Allen-Bradley

Leser dieses Handbuchs

Dieses Handbuch soll dem Bediener und Programmierer bei der Auslegung, Installation, Programmierung und Fehlersuche im Zusammenhang mit Steuersystemen, die speicherprogrammierbare Steuerungen von Allen-Bradley enthalten, helfen.

Sie sollten grundlegende Kenntnisse über SLC 500-Geräte, PLC- und 1771-E/A-Geräte von Allen-Bradley besitzen und mit programmierbaren Steuerungen vertraut sein. Ferner sollten Sie in der Lage sein, die zur Steuerung einer Anwendung erforderlichen Kontaktplanbefehle zu interpretieren. Verfügen Sie nicht über die erforderlichen Kenntnisse, setzen Sie sich bitte vor der Verwendung dieses Gerätes mit Ihrer Allen-Bradley-Geschäftsstelle in Verbindung, die Sie gerne über Ausbildungsseminare informiert.

Zweck dieses Handbuchs

Dieses Handbuch dient als Programmierhilfe bei der Anwendung der BASIC-Entwicklungssoftware (Bestellnummer 1747-PBASE).

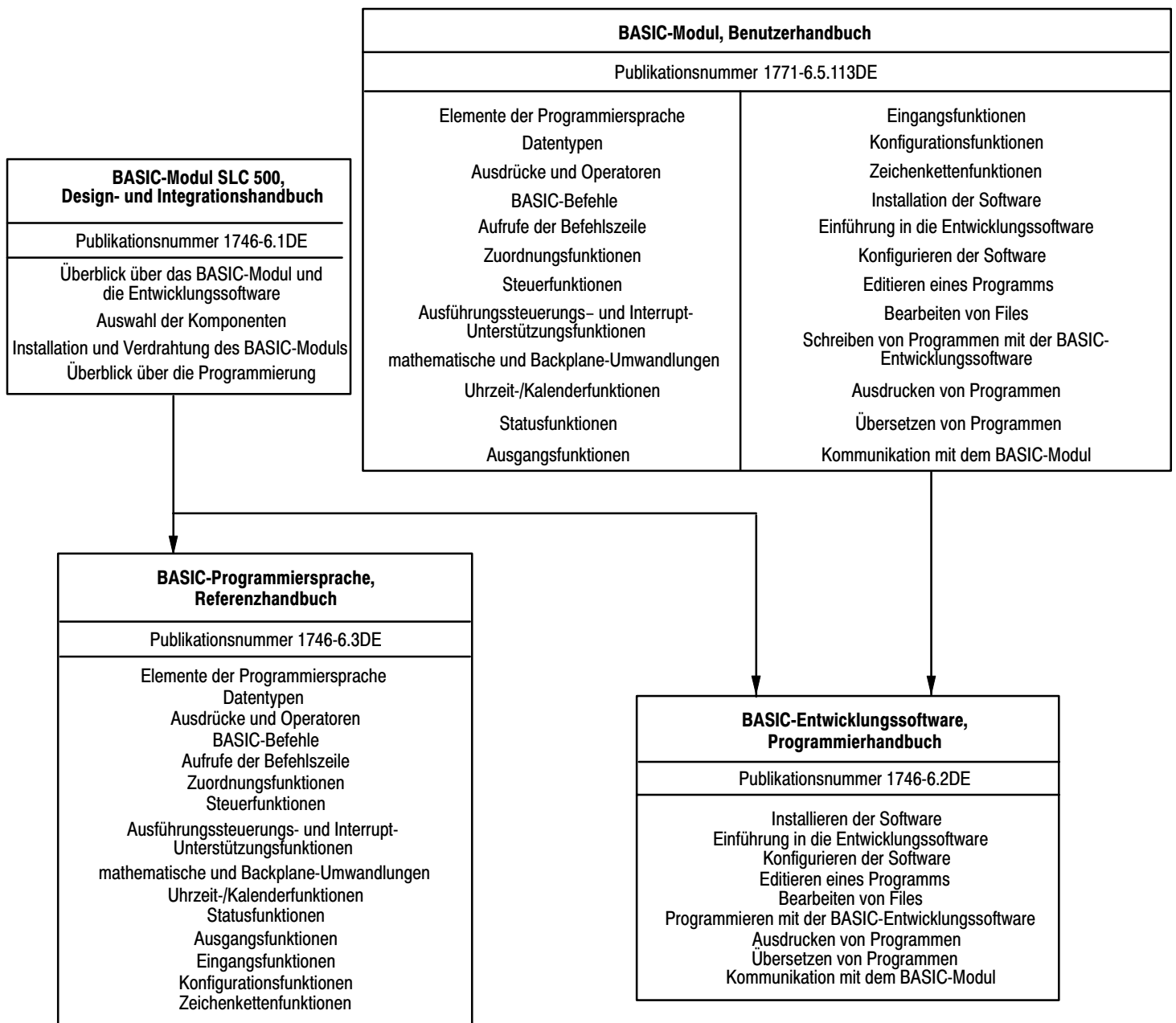
Tabelle V.1
Inhalt dieses Handbuchs

Kapitel	Titel	Zusammenfassung
	Vorwort	Hinweise zum Zweck, Hintergrund, Umfang und beabsichtigte Leserschaft dieses Handbuchs.
1	Installieren der Software	Installation der BASIC-Entwicklungssoftware von einer 3.5- bzw. 5.25-Zoll-Diskette auf den Personalcomputer.
2	Einführung in die Entwicklungssoftware	Beschreibung der Grundkonzeption der BASIC-Entwicklungssoftware, einschließlich der Funktionsweise des Hilfesystems, der wesentlichen Aspekte der Pulldown-Fenster und der Entwicklung eines ersten einfachen Programmes.
3	Konfigurieren der Software	Informationen zu einer Reihe von Konfigurationsoptionen, u.a. Maus, Anzeige, Drucker, Speicher, Funktionstasten und Terminal-Modus.
4	Editieren eines Programms	Erklärung aller Editierfunktionen, einschließlich Suchfunktion, Verschiebung von Textblöcken, Textformatierung sowie einer Reihe erweiterter Editierfunktionen.
5	Bearbeiten von Files	Informationen zur Fileverwaltung, vom einfachen Laden, Sichern und Verknüpfen bis zur Entwicklung von eigenen Anwendermenüs, die häufig verwendete Files oder Programme enthalten.
6	Programmieren mit der BASIC-Entwicklungssoftware	Überblick über die BASIC-Programmiersprache, -Makros und den Programmaufbau. Schreiben, Übersetzen und Herunterladen eines zweiten Programms.
7	Drucken eines Programms	Alle Aspekte der Druckerkonfiguration und des Druckvorgangs.
8	Übersetzen eines Programms	Vorbereitung des BASIC-Entwicklungsprogramms zum Herunterladen in das BASIC-Modul.
9	Kommunikation mit dem BASIC-Modul	Informationen zur Aktivierung des Terminal-Modus, zum Hoch- und Herunterladen vom/an das BASIC-Modul und zur Fehlersuche im BASIC-Programm.
Anhang A	Bibliothek der BASIC-Makros	Auflistung der in der BASIC-Entwicklungssoftware zur Programmoptimierung verfügbaren Makros.
Anhang B	DOS-Verzeichnis-Shell	Erläuterung der Anzeige eines DOS-Verzeichnisbaums und eines Verzeichnisses zur Ausführung einer Vielzahl von DOS-Befehlen.

Dokumentationssatz des BASIC-Moduls

Der Dokumentationssatz des BASIC-Moduls ist entsprechend den Aufgabenbereichen in verschiedene Handbücher untergliedert. Zu diesen Aufgabenbereichen gehören die Auslegung, Integrierung und Programmierung des BASIC-Moduls. In Abbildung V.1 sind die einzelnen BASIC-Dokumentationen und die in den einzelnen Handbüchern enthaltenen Informationen aufgeführt.

**Abbildung V.1
Dokumentationssatz des BASIC-Moduls**



Literaturhinweis

Die folgenden Dokumentationen enthalten zusätzliche Informationen über die SLC- und PLC-Geräte von Allen-Bradley und sind von Ihrer Allen-Bradley-Geschäftsstelle bzw. von Ihrem Allen-Bradley-Händler erhältlich.

**Tabelle V.2
Themenverwandte Dokumentationen**

Thema	Dokumentation	Dokumentationsnummer
Überblick über die Geräte der Reihe SLC 500	SLC 500, Systemübersicht	1747-2.30DE
Beschreibung der Installation und Anwendung der modularen speicherprogrammierbaren Steuerung SLC 500	Installations- und Benutzerhandbuch für die modulare Hardware-Konfiguration speicherprogrammierbarer Steuerungen	1747-6.2DE
Beschreibung der Installation und Anwendung der fest konfigurierten speicherprogrammierbaren Steuerungen SLC 500	Installation & Operation Manual for Fixed Hardware Style Programmable Controllers	1747-NI001
Ein verfahrensbezogenes Handbuch für technisches Personal zur Entwicklung von Steueranwendungen mit der APS-Software	Benutzerhandbuch für die erweiterte Programmiersoftware (APS) von Allen-Bradley	1747-6.4DE
Referenzhandbuch mit Statusfiledaten, Befehlssatz- und Fehlersuchinformationen zur APS-Software	Bedienerhandbuch für die erweiterte Programmiersoftware (APS) von Allen-Bradley	1747-6.11DE
Detaillierte Informationen über die Erdung und Verdrahtung von speicherprogrammierbaren Steuerungen von Allen-Bradley	Richtlinien zur Erdung und Verdrahtung von speicherprogrammierbaren Steuerungen von Allen-Bradley	1770-4.1DE
Beschreibung der wichtigsten Unterschiede zwischen elektronischen speicherprogrammierbaren Steuerungen und festverdrahteten elektromechanischen Geräten	Application Considerations for Solid-State Controls	SGL-1.1
Artikel über Leitergrößen und -arten zur Erdung von elektrischen Geräten	National Electrical Code	Veröffentlichung durch National Fire Protection Association of Boston, MA.
Eine komplette Auflistung aktueller Dokumentationen von Allen-Bradley, einschließlich Bestellinformationen. Gibt ferner an, ob die Dokumentationen auf CD-ROM bzw. in anderen Sprachen erhältlich sind.	Allen-Bradley Publication Index	SD499
Glossar mit Begriffen und Abkürzungen im Bereich der industriellen Automatisierungstechnik	Industrielle Automatisierung Glossar	AG-7.1DE
Beschreibung der Installation und Anwendung eines Moduls, das als Übergangseinheit zwischen DH-485-Netzwerken und Geräten, die das DF1-Protokoll einsetzen, fungiert.	DH-485/RS-232C Interface Module User's Manual	1747-6.12

Begriffe und Abkürzungen

Die folgenden Begriffe und Abkürzungen beziehen sich speziell auf dieses Gerät. Eine vollständige Auflistung der englischen Allen-Bradley-Fachterminologie befindet sich in Publikation AG-7.1DE, Industrielle Automatisierung Glossar.

- **A-STACK error** — tritt ein, wenn zu viele Ausdrücke mit einer PUSH-Anweisung in den Argumentstapel eingespeichert werden oder wenn versucht wird, dem Stapelspeicher nicht vorhandene Daten mit einer POP-Anweisung zu entnehmen
- **BASIC-Entwicklungssoftware** — BASIC-Entwicklungssoftware (Bestellnummer 1747-PBASE)
- **BASIC-Entwicklungssprache** — die erweiterte BASIC-Sprache, die im Lieferumfang der BASIC-Entwicklungssoftware (Bestellnummer 1747-PBASE) enthalten ist und zur Programmierung des BASIC-Moduls verwendet wird
- **BASIC-Modul** — SLC 500 BASIC-Modul (Bestellnummer 1746-BAS) und das 1771-DB/B
- **Copy file or files** — der hervorgehobene File wird kopiert
- **Delete file or files** — der hervorgehobene File wird gelöscht
- **DH-485** — Netzwerk-Kommunikationsprotokoll
- **Dialogfeld** — ein Feld auf der Bildschirmanzeige der BASIC-Entwicklungssoftware, das die verfügbaren Menüoptionen des gewählten Menüelementes enthält
- **EMS** — bezieht sich auf den erweiterten Speicherbereich, der im Personalcomputer installiert werden kann. Die BASIC-Entwicklungssoftware belegt einen *erweiterten Speicher* und nicht einen *Zusatzspeicher*.
- **EPROM** — löschbarer, programmierbarer Nur-Lese-Speicher
- **File** — ein BASIC-Programmfile
- **Filenamenerweiterung** — die Zeichen nach dem Filenamen und Punkt
- **Hypertext-Hilfe** — ein kontextuelles Hilfesystem, das den Zugriff auf das Hilfesystem von praktisch jeder Stelle der BASIC-Entwicklungssoftware aus ermöglicht
- **Load file or files** — der hervorgehobene File wird in das aktuelle Fenster geladen, wobei zuvor im Fenster vorhandene Files überschrieben werden.
- **MTOP** — Systemsteuerwert, der die zuletzt gültige Speicheradresse enthält

- **Print file or files** — der bzw. die hervorgehobenen Files werden gedruckt. Anschließend kann die DOS-Verzeichnis-Shell verlassen werden, ohne daß der Druckvorgang unterbrochen wird
- **Programmierport** — Port zur Programmierung des BASIC-Moduls. Entweder der PRT1- oder der DH485-Port kann als Programmierport konfiguriert werden.
- **Programmiermakros** — Programmierbefehle, die von der BASIC-Entwicklungssoftware zur Optimierung des Programms zur Verfügung gestellt werden
- **QuickStroke** — bezieht sich auf eine Taste bzw. auf ein Tastenpaar, mit dem eine Funktion schnell ausgeführt bzw. ein Menü schnell aufgerufen werden kann. Diese Tasten sind in eckigen Klammern [] angezeigt (Beispiele: [Alt-F10] und [Ctrl-F1])
- **Rename file** — der hervorgehobene File wird umbenannt
- **RS-232/423** — serielle Kommunikationsschnittstelle
- **RS-422** — Differentialkommunikationsschnittstelle
- **RS-485** — Netzwerk-Kommunikationsschnittstelle
- **SLC 500** — SLC 500-Steuerungen mit fester und modularer Hardware-Konfiguration
- **Sort Files** — die Files innerhalb eines Verzeichnisses werden neu angeordnet
- **Tastenmakros** — reduziert die Anzahl der zur Ausführung einer Funktion erforderlichen Tastenanschläge
- **View file** — ermöglicht das schnelle Betrachten des aktuellen Files in einem Nur-Lese-Popup-Fenster

Konventionen in diesem Handbuch

In diesem Handbuch werden die folgenden Konventionen verwendet:

- Mit Aufzählungspunkten wie hier gekennzeichnete Auflistungen enthalten Informationen und nicht Verfahrensweisen.
- Numerierte Auflistungen enthalten sequentielle Schritte bzw. hierarchisch angeordnete Informationen.
- *Kursivschrift* wird zur Hervorhebung von Informationen verwendet.
- Text in **diesem Font** kennzeichnet Worte bzw. Sätze, die Sie eingeben müssen.

- Tastenbezeichnungen entsprechen den dargestellten Bezeichnungen und erscheinen als fettgedruckte, von eckigen Klammern umgebene Großbuchstaben (Beispiel: [**ENTER**]).
- Der Aufruf einer Bildschirmanzeige über den Menüpfad wird wie folgt dargestellt: **Path: Toplevel Menu → Edit mode → File → load file in cur. win.** Drücken Sie **Cancel <ESC>**, um zur Toplevel-Menüebene zurückzukehren.
- Der Aufruf einer Bildschirmanzeige durch einen Schnellaufruf wird wie folgt dargestellt: **QuickStroke: <ShftF5>**.

Technische Unterstützung durch Allen-Bradley

Mit 75 Geschäftsstellen in den Bereichen Verkauf/Technische Unterstützung, 512 autorisierten Distributoren und 260 autorisierten Systemintegratoren allein in den USA bietet Allen-Bradley weltweite technische Unterstützung. Außerdem ist Allen-Bradley in jedem größeren Land der Welt vertreten.

Produktspezifische Unterstützung

Ihre Allen-Bradley-Vertretung berät Sie gerne über:

- Verkauf und Beratung
- technische produktspezifische Ausbildung
- Garantieansprüche
- Unterstützungs-/Serviceverträge

Technische Unterstützung

Bevor Sie technische Unterstützung von Allen-Bradley anfordern, lesen Sie bitte zuerst die Informationen im entsprechenden Kapitel durch. Setzen Sie sich anschließend mit Ihrer Allen-Bradley-Vertretung in Verbindung.

Fragen und Kommentare zu diesem Handbuch

Wenn Sie Verbesserungsvorschläge zum Aufbau dieses Handbuchs haben, können Sie uns unter der folgenden Adresse erreichen:

Allen-Bradley Company, Inc.
Automation Group
Technical Communication, Dept. J602V, T121
P.O. Box 2086
Milwaukee, WI 53201-2086, USA

AB Spares

Installation der Software

In diesem Kapitel wird zunächst eine generelle Beschreibung der BASIC-Entwicklungssoftware und anschließend eine schrittweise Anleitung zur Installation der Software auf einem Personalcomputer gegeben. Die Hauptthemen umfassen:

- eine generelle Beschreibung der BASIC-Entwicklungssoftware
- die für die Installation erforderliche Hardware
- die Diskettensätze der BASIC-Entwicklungssoftware
- die Installation der Software
- das Sichern der Diskettendaten

Funktionsmerkmale der Software

Dieser Abschnitt dient als genereller Leitfaden zu den Funktionsmerkmalen der BASIC-Entwicklungssoftware. Zu den standardmäßigen Merkmalen gehören:

- Funktionen des Kommunikationsterminals
- Debugging-Funktion zur Fehlersuche in der Software, die im BASIC-Modul ausgeführt wird
- Übersetzer-Dienstprogramm zur Umwandlung von Programmen in BASIC-Entwicklungssprache, um das Herunterladen an das BASIC-Modul zu ermöglichen
- Fenster, die das gleichzeitige Editieren von bis zu 100 Files (oder verschiedenen Teilen eines Files) ermöglichen
- Nachbarfenster-Funktion zum einfachen Wechseln zwischen geteilten Fenstern
- Pulldown-Menüsystem und Schnellaufruf-Tasten (QuickStroke)
- dynamische Anzeige der Funktionstastenbezeichnungen, die durch Drücken der Tasten [shift] [ctrl] und [alt] eingeblendet werden
- anwenderbelegbare Tastenmakros, durch die häufig verwendete Tastensequenzen mit nur einer Taste bzw. Tastenkombination ersetzt werden
- Rückgängigmachen von bis zu 65535 Operationen und Aufhebung der letzten Rückgängigmachung

- Zeilen-, Zeilensequenzen- und Spaltenblockoperationen, einschließlich Kopieren, Verschieben, Ausschneiden und Einfügen zwischen Fenstern
- Suchen und Ersetzen in mehreren Files in verschiedenen Verzeichnissen bzw. Files
- DOS-Shell und DOS-Verzeichnis-Shell mit Mehrfachverzeichnisanzeige
- Hypertext-Hilfe bietet ein kontextuelles Online-Hilfesystem, das jederzeit aufgerufen werden kann
- automatische Erweiterung, die bestimmte Standardeinstellungen zum Editieren, Übersetzen und Kompilieren ermöglicht
- Programmiersprachenunterstützung (automatischer Einzug, Übereinstimmung des Befehlsaufbaus) für BASIC
- Linienfunktion ermöglicht die Programmierung von Grafikbildschirmen, die in BASIC übersetzt und vom BASIC-Modul ausgeführt werden können
- einfache Textformatierung, u.a. intelligenter Wortumbruch, Neuformatierung sowie positiver und negativer Einzug
- Mausunterstützung zur Beschleunigung der Menüauswahl
- Popup-Taschenrechner mit Hexadezimal-, Dezimal-, Oktal- und Binär-Rechenfunktionen, einem simulierten Kontrollstreifen–ausdruck und der Fähigkeit, das Ergebnis in ein Programm einzufügen
- Popup-ASCII-Tabelle mit hexadezimalen und dezimalen Äquivalenten
- vollständige EMS-Unterstützung, d.h. große Files können vollständig im EMS-Speicher editiert werden.
- Auslagerung des Speichers in den EMS-Speicher oder auf die Festplatte, während andere Programme oder umfangreiche Kompilierungen ablaufen
- Anwendermenüs zur schnellen Wahl der am häufigsten editierten Files bzw. ausgeführten Programme, die auf dem Personalcomputer installiert sind
- anwenderkonfigurierbare Änderung von Vorgabeeinstellungen der Such- und Ersetzfunktion sowie der Suchfunktion für mehrere Files

- mehrfache Standardverzeichnisse für Files mit einer bestimmten Erweiterung (.bas-Files in einem und .bdl-Files in einem anderen Verzeichnis)

Anforderungen an die Hardware

Für die BASIC-Entwicklungssoftware ist ein IBM-PC-kompatibler Computer mit einer RAM-Speichergröße von mindestens 640 KByte und einer Festplattenspeicherkapazität von mindestens 2 MByte erforderlich. Der Personalcomputer muß unter DOS-Version 3.1 oder höher laufen.

Die BASIC-Entwicklungssoftware unterstützt fast alle Videokarten und -Betriebsarten und ist für die EGA-Betriebsart mit 43 Zeilen und VGA-Betriebsart mit 50 Zeilen konfiguriert. Andere Video-Betriebsarten können von der Befehlszeile aus konfiguriert werden. Weitere Hinweise zur Konfiguration der Video-Betriebsart sind in Kapitel 3 enthalten.

Die Software unterstützt Microsoft, MOUSE SYSTEMS, LOGITECH und jede andere kompatible Maus sowie die erweiterte Speicherfunktion Expanded Memory Specification 4.0 und höher.

Hardware für den Anschluß des Personalcomputers an das BASIC-Modul

Am PC muß mindestens ein Kommunikationsport zur Verfügung stehen. Für den Anschluß des Personalcomputers an das BASIC-Modul ist eine der folgenden Zusatzgerätekombinationen erforderlich.

Zur Herstellung der Verbindung des Kommunikationsports (COM1 oder COM2) des PCs mit dem DH485-Port des BASIC-Moduls empfiehlt sich ein Schnittstellenwandler 1747-PIC von Allen-Bradley:

- für den direkten Anschluß an den DH485-Port des BASIC-Moduls
- in Verbindung mit einem isolierten Verbindungskoppler 1747-AIC von Allen-Bradley, wenn der Personalcomputer in ein DH-485-Netzwerk eingebunden werden soll, an das das BASIC-Modul angeschlossen ist

Für die Verbindung zwischen dem Kommunikationsport (COM1 bis COM8) des Personalcomputers und dem PRT1-Port des BASIC-Moduls sollte für das Nullmodemkabel 1771 DB/B ein 25-poliger Buchsenstecker/25-poliger Stiftstecker bzw. für das Nullmodemkabel 1746-BAS ein 9-poliger Buchsenstecker/25-poliger D-Stecker verwendet werden.

Für die Kommunikation mit dem BASIC-Modul ohne die Verwendung eines Kommunikationsports des Personalcomputers kann mit einer DH-485-Schnittstellenkarte 1784-KR und einem Verbindungskoppler 1747-AIC von Allen-Bradley eine direkte Verbindung mit dem DH485-Port des BASIC-Moduls hergestellt werden.

Nähere Hinweise zur Verkabelung und Brückeneinstellung für diese verschiedenen Konfigurationen sind im des BASIC-Modul SLC 500, Design- und Integrationshandbuch (Publikationsnummer 1746-6.1DE) enthalten.

Inhalt der Disketten

Der Diskettensatz ist im doppelseitigen 5.25-Zoll-Format doppelter Dichte (360 KByte) und im 3.5-Zoll-Format doppelter Dichte (720 KByte) lieferbar. Der Diskettensatz im 5.25-Zoll-Format enthält drei Disketten, und der Satz im 3.5-Zoll-Format enthält zwei Disketten.

Tabelle 1.A
Diskettenformate

Disketten- nummer	Inhalt der 5.25-Zoll-Disketten	Inhalt der 3.5-Zoll-Disketten
1	INSTALL.EXE IDENTIFY.TXT HELP.EXE	INSTALL.EXE IDENTIFY.TXT MACROS.EXE HELP.EXE
2	IDENTIFY.TXT MACROS.EXE	IDENTIFY.TXT BASICEX.EXE TRANS.EXE MISC.EXE
3	IDENTIFY.TXT BASICEX.EXE TRANS.EXE MISC.EXE	-

Installationsverfahren

Vor der Installation der BASIC-Entwicklungssoftware ist die Datei autoexec.bat im Personalcomputer so zu modifizieren, daß sie das Verzeichnis \ABBASIC\BDS in ihrem Pfadbefehl enthält. Auf diese Weise wird die Initialisierung der BASIC-Software von einem beliebigen Verzeichnis der Festplatte aus ermöglicht.

Diese Anweisung wird mit einem beliebigen Datei-Editor (die meisten DOS-Systeme enthalten den Editor EDLIN) in die Pfadzeile der Datei autoexec.bat eingefügt. Detaillierte Informationen über die Strukturierung und Bearbeitung dieser Datei können im DOS-Handbuch des Personalcomputers nachgeschlagen werden. Das folgende Beispiel stellt eine typische Pfadanweisung dar, in der dieses Verzeichnis enthalten ist:

```
PATH=C: \ ; C: \DOS ; C: \MOUSE ; C: \ABBASIC\BDS ;
```

Wichtig: Bei der Modifizierung des Pfadbefehls in der Datei autoexec.bat dürfen aus dem bestehenden Pfad *keine* anderen Verzeichnisse gelöscht werden. Fügen Sie `C:\ABBASIC\BDS;` am Ende der bestehenden Pfadzeile ein. Hierbei wird vorausgesetzt, daß die BASIC-Entwicklungssoftware auf Festplatte C: installiert wird. Bei der Installation auf einer anderen Festplatte ist C: durch den entsprechenden Buchstaben zu ersetzen.

Überprüfen Sie die Datei config.sys, um sicherzustellen, daß sie die Angabe von mindestens 20 Dateien und 20 Zwischenspeichern enthält. Bei den meisten DOS-Systemen kann diese Datei gelesen werden, indem `TYPE CONFIG.SYS` eingegeben wird. Fügen Sie, falls erforderlich, die folgenden Zeilen in diese Datei ein:

```
FILES=20  
BUFFERS=20
```

Wichtig: Die *Mindestanzahl* der erforderlichen Dateien und Zwischenspeicher ist 20. Ist die angezeigte Zahl größer als 20, sollte sie nicht auf 20 verringert werden. Dies könnte sonst dazu führen, daß andere auf dem Personalcomputer installierte Software nicht mehr betriebsfähig ist.

Installation von 5.25-Zoll-Disketten

Ein automatisches Installationsprogramm vereinfacht die Installation. Es erstellt automatisch Unterverzeichnisse und kopiert Dateien von den 5.25-Zoll-Disketten auf die Festplatte.

Wichtig: Das Installationsverfahren kann jederzeit durch gleichzeitiges Drücken der Tasten `[ctr1]` und `[c]` abgebrochen werden. *In diesem Fall ist die Installation unvollständig, und die Software ist nicht betriebsfähig.*

Zum Ausführen des Installationsprogramms:

1. legen Sie Diskette 1 in das Laufwerk ein
2. geben Sie `A:INSTALL` ein, und drücken Sie die `[ENTER]`-Taste.

Wichtig: Wenn das 5.25-Zoll-Laufwerk auf Ihrem PC nicht Laufwerk A: ist, geben Sie anstelle von A: den entsprechenden Buchstaben ein.

Befolgen Sie die angezeigten Anleitungen.

Für die Installation dieser Software sind 2 MByte Speicher erforderlich. Wenn auf der Festplatte weniger als 2 MByte Speicherplatz vorhanden ist, erscheint die folgende Meldung:

```
There is not enough room on your disk drive C.  
The minimum number of free bytes must be: 2000000.
```

```
WARNING: Software was not properly installed.  
C:\>
```

Hiermit wird angezeigt, daß das Installationsverfahren abgebrochen wurde. Soll die Software dennoch auf dieser Festplatte installiert werden, muß durch Löschen überflüssiger Dateien freier Speicherplatz geschaffen werden und anschließend die Installation erneut durchgeführt werden.

Sie können auch eine andere Festplatte wählen. Verfügt die gewählte Festplatte über ausreichenden Speicherplatz, wird das Installationsverfahren fortgesetzt. Sie werden aufgefordert, das Verzeichnis, in dem die Software installiert werden soll, zu bestätigen oder zu ändern.

```
Specify main directory in which to install files.  
C:\ABBASIC
```

Drücken Sie die [ENTER]-Taste, um den Verzeichnisnamen zu bestätigen, oder geben Sie einen neuen Verzeichnisnamen ein, und drücken Sie anschließend die [ENTER]-Taste.

Wichtig: Wenn nicht das Standardverzeichnis ABBASIC, sondern ein anderes Verzeichnis gewählt wird, müssen Sie die Datei autoexec.bat, wie am Beginn des Abschnittes "Installationsverfahren" beschrieben, entsprechend ändern.

Nach dem Laden der Disketten werden Sie daran erinnert, die Dateien config.sys und autoexec.bat, wie im vorhergehenden Abschnitt beschrieben, zu modifizieren.

Installation von 3.5-Zoll-Disketten

Das Installationsverfahren für 3.5-Zoll-Disketten ist identisch mit dem Verfahren für 5.25-Zoll-Disketten, wobei die folgenden Ausnahmen gelten:

- Bei vielen Personalcomputern, die sowohl über ein 5.25-Zoll- als auch ein 3.5-Zoll-Laufwerk verfügen, ist das 3.5-Zoll-Laufwerk das Laufwerk B:. Ist dies auch bei Ihrem PC der Fall, sollten Sie nicht **A:INSTALL**, sondern **B:INSTALL** eingeben.
- Da der Diskettensatz im 3.5-Zoll-Format nur zwei Disketten enthält, werden Sie nicht, wie beim Diskettensatz im 5.25-Zoll-Format, aufgefordert, eine dritte Diskette einzulegen.

AB Spares

Verzeichnisstruktur

Die Standardverzeichnisstruktur lautet nach der Installation wie folgt:

- \ABBASIC - Hauptverzeichnis
- \ABBASIC\BAS - Unterverzeichnis zum Speichern der Anwender-BASIC-Quellprogramme. Dies ist das "Arbeitsverzeichnis".
- \ABBASIC\BDS - Unterverzeichnis zum Speichern der BASIC-Entwicklungssoftware-Dateien
- \ABBASIC\BDS\HELP - Unterverzeichnis zum Speichern der Hilfe-Dateien

Wechseln Sie vor Aufruf von PBASE in das Arbeitsverzeichnis. Zusätzliche Arbeitsverzeichnisse können an beliebigen Stellen auf der Festplatte erstellt werden.

Sollten die Standardbezeichnungen der Unterverzeichnisse nicht akzeptabel sein, können sie nach der Beendigung des Installationsverfahrens geändert werden. Das HELP-Unterverzeichnis muß direkt unter dem BDS-Unterverzeichnis bzw. einem entsprechenden Verzeichnis angeordnet sein. Das HELP-Unterverzeichnis darf nicht umbenannt werden.

Wichtig: Bei Änderung eines Verzeichnis- oder Unterverzeichnisnamens ist es erforderlich, die Datei autoexec.bat, wie am Beginn des Abschnittes "Installationsverfahren" beschrieben, entsprechend zu ändern.

Sichern der Software

Die Firma Allen-Bradley gewährt dem Käufer eine Lizenz, diese Software auf einem Personalcomputer zu installieren und anzuwenden und Sicherungskopien anzufertigen. Es wird keine Lizenz für die Anwendung der Software auf mehr als einem Personalcomputer oder für den Vertrieb erteilt. Hinweise zur Lizenzerteilung sind auf der Verpackung der Software enthalten.



ACHTUNG: Diese Software ist nach dem Urheberrecht der Vereinigten Staaten geschützt. Eine nicht autorisierte Reproduktion von urheberrechtlich geschützter Software verstößt gegen dieses Urheberrecht. Ein Verstoß kann zu Geld- oder Gefängnisstrafen führen.

- Für die Sicherungskopie der Original-BASIC-Entwicklungssoftware im 5.25-Zoll-Format werden drei leere, doppelseitige Disketten doppelter Dichte (ca. 360 KByte je Diskette) benötigt.
- Für die Sicherungskopie der Original-BASIC-Entwicklungssoftware im 3.5-Zoll-Format werden zwei leere, doppelseitige Disketten doppelter Dichte (ca. 720 KByte je Diskette) benötigt.

Eine Formatierung der leeren Disketten ist nicht erforderlich. Weitere Hinweise sind im DOS-Handbuch enthalten.

AB Spares

Einführung in die Entwicklungssoftware

Dieses Kapitel soll Ihnen einen Überblick über die BASIC-Entwicklungssoftware zum besseren Verständnis der allgemeinen Struktur geben. Die folgenden Themen werden erörtert:

- Starten der Software
- Toplevel-Menü
- Hauptmenü
- Editieren und Beenden eines Programms

Starten der Software

Nach der Installation wird die Software wie folgt gestartet:

1. Booten Sie den Personalcomputer erneut. Dies ist zum Aktivieren der in den Dateien autoexec.bat und config.sys vorgenommenen Änderungen erforderlich.
2. Wurde der Maustreiber nicht automatisch durch die Datei autoexec.bat installiert, führen Sie den erforderlichen Befehl aus, um die Maus-Software vor dem Starten der BASIC-Entwicklungssoftware zu installieren.
3. Wechseln Sie in das Arbeitsverzeichnis. Wurden die standardmäßigen Verzeichniszuweisungen während der Softwareinstallation angenommen (siehe Kapitel 1), so steht nun das Unterverzeichnis BAS als Arbeitsverzeichnis zur Verfügung. Dieses Verzeichnis wird durch die folgende Eingabe am DOS-Prompt aufgerufen: > `CD\ABBASIC\BAS`. Drücken Sie anschließend die [ENTER]- bzw. [Return]-Taste (diese wird in diesem Handbuch als [ENTER]-Taste bezeichnet). Das Zeichen > befindet sich nun am Ende der Zeile: `CD\ABBASIC\BAS`.
4. Geben Sie am DOS-Prompt > `PBASE` ein, und drücken Sie die [ENTER]-Taste. Die Software ermittelt den Monitortyp und zeigt das Toplevel-Menü an.

Wichtig: Wenn der PBASE-Befehl vom Personalcomputer nicht erkannt wird, wurde der Pfadbefehl in der Datei autoexec.bat möglicherweise nicht korrekt modifiziert.

Ist eine Maus installiert, die jedoch nicht funktioniert, kann zu diesem Zeitpunkt mit den Pfeiltasten fortgefahren werden. Sie können auch überprüfen, ob der Maustreiber über die Datei `autoexec.bat` oder durch eingegebene Befehle installiert wurde.

Wichtig: Im Menü “Setup and Configuration” sind weitere Mauseinrichtungsoptionen aufgeführt, u.a. eine Maus-EIN/AUS-Option (ON/OFF). Weitere Informationen zu diesen Menüs sind in Kapitel 3 enthalten.

Toplevel-Menü

Im Toplevel-Menü werden BASIC-Programme entwickelt und editiert.

Abbildung 2.1
Toplevel-Menü

```
Toplevel Menu
Select a project file
Edit mode
Translate/compile
teRminal [RS-232]
User defined program
Configuration menu
shell to Dos
Quit
```

- **select a project file** — Wahl eines Projektfiles über ein anwenderdefiniertes Menü. Diese Option ist auch im Other-Menü verfügbar. Eine nähere Beschreibung ist in Kapitel 5 enthalten.
- **edit mode** — Aufruf des Edit-Modus und Öffnen eines neuen Filefensters oder Zugriff auf das derzeit aktivierte Fenster. In diesem Modus sind alle Funktionen des Hauptmenüs verfügbar.
- **Translate/compile** — sofortiges Übersetzen/Kompilieren eines gewählten Files, vorausgesetzt, daß die Erweiterung des Filenamens und das Übersetzerprogramm korrekt konfiguriert wurden. Diese Option ist auch im Other-Menü verfügbar. Eine nähere Beschreibung ist in Kapitel 8 enthalten.
- **teRminal [232]** — Umschalten des Personalcomputers in den Terminal-Modus, um eine Kommunikationsverbindung mit dem BASIC-Modul oder einem anderen konfigurierten Gerät herzustellen. Der Terminal-Modus kann auch vom Other-Menü aus aktiviert werden und ist in Kapitel 9 näher beschrieben.

AB Spares

- **User defined program** — Vorübergehendes Verlassen der BASIC-Entwicklungssoftware und Ausführung eines Programms auf der DOS-Ebene des Personalcomputers. Das Programm wird von einem anwenderdefinierten Menü gewählt. Diese Option ist auch im Other-Menü verfügbar und ist in Kapitel 5 näher beschrieben.
- **Configuration menu** — Anzeige eines Menüs mit Einrichtungsoptionen für Maus, Anzeige, Drucker, Editierfunktion usw. Eine nähere Beschreibung dieses Menüs ist in Kapitel 3 enthalten.
- **shell to dos** — Vorübergehendes Verlassen der BASIC-Entwicklungssoftware und Aufruf von DOS zur Abarbeitung von Programmen bzw. Ausführung von DOS-Befehlen. Eine Rückkehr zur Entwicklungssoftware wird durch Eingabe von **exit** am DOS-Prompt veranlaßt. Diese Option ist auch im Other-Menü verfügbar.
- **Quit** — Verlassen der PBASE-Software und Rückkehr zu DOS.

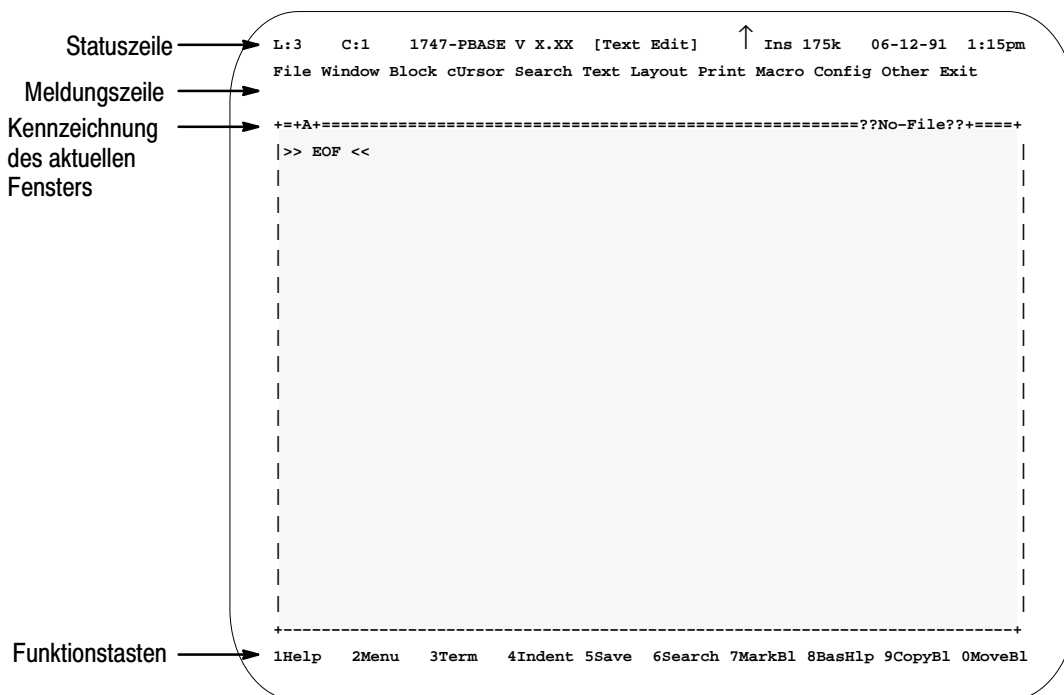
Hauptmenü

Das Hauptmenü wird angezeigt, sobald Sie vom Toplevel-Menü aus den Edit-Modus aufrufen.

1. Wählen Sie im Toplevel-Menü mit dem Cursor oder der Maus die Option **edit mode**. Alternativ hierzu können Sie auch [E] eingeben, da dieser der hervorgehobene Buchstabe dieses Menüelementes ist.
2. Drücken Sie bei Verwendung des Cursors die [ENTER]-Taste, nachdem Sie "Edit mode" hervorgehoben haben. Klicken Sie bei Verwendung einer Maus die linke Maustaste, wenn sich der Mauscursor auf dem Element **edit mode** befindet.

Das Hauptmenü wird wie unten abgebildet angezeigt. Der Edit-Modus ist nun aktiviert, und Sie können ein Programm schreiben oder editieren.

Oben links vom Hauptmenü wird der Buchstabe **A**, und oben rechts **?No-File?** angezeigt. Hierbei handelt es sich um Kennzeichnungen des aktuellen Fensters, das beim Aufruf des Edit-Modus geöffnet wurde, d.h. die Anzeige enthält Fenster A, welches derzeit keinen File enthält.



Statuszeile

Die Statuszeile ist die erste Zeile des Hauptmenüs. Im folgenden wird die Statuszeile der obenstehenden Abbildung erläutert:

AB Spares

- **L:3** — zeigt an, daß sich der Cursor in der dritten Zeile des aktivierten Fensters befindet
- **C:1** — zeigt an, daß sich der Cursor in der ersten Spalte des aktivierten Fensters befindet
- **1747-PBASE** — zeigt an, daß Sie mit der BASIC-Entwicklungssoftware arbeiten
- **[TEXT EDIT]** — zeigt an, daß der Texteditier-Modus aktiviert ist
- **R** — kommt in diesem Beispiel nicht vor; erscheint jedoch bei der Erstellung eines Tastenmakros (Beschreibung in Kapitel 4) zur Anzeige der Aufzeichnung
- **↑** — zeigt an, daß Großbuchstaben eingegeben werden. Die **[cap lock]**-Taste der Tastatur wurde betätigt
- **Ins** — schaltet zwischen dieser Anzeige und **ovr** hin und her, um anzuzeigen, ob der Einfügemodus (Insert) oder Überschreibmodus (Overstrike) aktiviert ist
- **175K** — zeigt den derzeit zur Verfügung stehenden RAM-Speicherplatz des Personalcomputers an
- **06-12-91** — das aktuelle Datum entsprechend der PC-Uhr
- **1:15pm** — die aktuelle Uhrzeit entsprechend der PC-Uhr

Meldungszeile

Die Meldungszeile ist die dritte Zeile des Hauptmenüs, direkt unter den Pull-down-Menüs. Die Software zeigt in dieser Zeile entsprechend Ihrer Auswahl und Eingabe Normalbetrieb-Meldungen, Anweisungen oder Fehlermeldungen an.

Funktionstasten

Die Funktionstasten (1 bis 10) sind am unteren Bildschirmrand aufgeführt.

Durch Drücken von **[shift]**, **[Alt]** oder **[Ctrl]** vor und während der Betätigung der Funktionstasten stehen zusätzliche Funktionen zur Verfügung. Beispiel: Mit **[F2]** wird der Cursor nach oben auf das Hauptmenü bewegt, während durch Drücken von **[shift-F2]** im aktivierten Fenster das Datum und die Uhrzeit angezeigt werden. Mit **[Alt-F2]** wird die Taschenrechner-Funktion aktiviert, und mit **[Ctrl-F2]** wird bei der Untersuchung einer geladenen Fehlerdatei der nächste Fehler gesucht.

Beachten Sie bitte, daß sich die Belegung der Funktionstasten mit der Wahl verschiedener Menüs und auch nach dem Drücken einer bestimmten Funktionstaste ändert. Auf diese Weise können die am häufigsten verwendeten Funktionen über Funktionstasten aufgerufen werden. Beispiel: Wenn mit [F8] das Hilfesystem des BASIC-Moduls aufgerufen wird, ändert sich auch die Belegung der Funktionstasten.

Aufruf des Hilfesystems außerhalb eines Programms

1. Drücken Sie [F1] oder klicken Sie mit der linken Maustaste, wenn sich der Cursor auf `1 help` in der unteren linken Bildschirmcke befindet. Das Hilfesystem ist kontextuell, d.h. der angezeigte Hilfebildschirm befaßt sich mit dem aktuellen Thema. Wird derzeit kein bestimmtes Thema bearbeitet, erscheint die folgende Anzeige: `1747-PBASE REFERENCE GUIDE - TABLE OF CONTENTS`.
2. Bewegen Sie den Cursor bzw. die Pfeiltaste auf `B. editor Quick Reference`, und drücken Sie die [ENTER]- oder die linke Maustaste. Nun sollte auf dem Bildschirm die Anzeige `1747-PBASE QUICK REFERENCE` erscheinen. Hilfebildschirme können einige Sätze oder mehrere Paragraphen lang sein.
3. Verwenden Sie zum Einsehen dieses Hilfebildschirms die Auf/Ab-, Rechts-/Links-Cursortasten (Pfeiltasten) oder die Bild-Auf-/Bild-Ab-Tasten. Äußerst rechts auf dem Hilfebildschirm befindet sich eine Spalte mit jeweils einem Pfeil am oberen und unteren Rand. Setzen Sie den Mauscursor auf den oberen bzw. unteren Pfeil, und klicken Sie die linke Maustaste, um den Hilfebildschirm zu durchlaufen. Beachten sie bitte, daß in vielen Hilfebildschirmen Wörter bzw. Elemente hervorgehoben (bzw. auf einem Farbmonitor in einer anderen Farbe dargestellt) sind. Diese Elemente beziehen sich auf ähnliche Themen, für welche zusätzliche Hilfeinformationen vorhanden sind.
4. Setzen Sie den Cursor auf `goto Line number`, und drücken Sie die [ENTER]-Taste oder die linke Maustaste. Der Hilfebildschirm für `goto Line number` wird angezeigt. Er enthält weitere verwandte Themen, die nun auf dieser Anzeige gewählt werden können.
5. Drücken Sie die [Backspace]-Taste oder setzen Sie den Mauscursor auf `Back-track<BackSpace>`, und klicken Sie die linke Maustaste. Nun werden die zuvor aufgerufenen Hilfebildschirme und schließlich `1747-PBASE REFERENCE GUIDE - TABLE OF CONTENTS` wieder angezeigt. Von dieser Anzeige aus können andere Themen gewählt werden. Außerdem steht ein umfassender Hilfeindex zur Verfügung, der ebenfalls gewählt werden kann.
6. Drücken Sie die [Esc]-Taste oder setzen Sie den Mauscursor auf `Done<ESC>`, und klicken Sie die linke Maustaste, um das Hilfesystem zu verlassen. Nun wird das Hauptmenü wieder angezeigt.

Aufruf des Hilfesystems von einem Programm aus

1. Geben Sie vom Hauptmenü aus einen Befehl wie beispielsweise `GOTO` ein. (Für `GOTO` steht ein vollständiger Abschnitt der Hypertext-Hilfe zur Verfügung.)
2. Drücken Sie `[F8]` oder setzen Sie den Mauscursor auf `8 BasHlp` am unteren Bildschirmrand, um den Hilfetext über das BASIC-Modul und die BASIC-Programmiersprache aufzurufen.
3. Drücken Sie die `[ESC]`-Taste oder bewegen Sie den Mauscursor auf `Done<ESC>`, und klicken Sie die linke Maustaste, um den Hilfebildschirm für `GOTO` zu verlassen.
4. Bewegen Sie den Cursor nach oben bzw. unten, bis er sich nicht mehr in derselben Zeile wie die `GOTO`-Anweisung befindet.
5. Drücken Sie `[F8]` oder bewegen Sie den Mauscursor auf `8 BasHlp`, und klicken Sie die linke Maustaste. Sofern sich der Cursor nicht auf einem Wort befindet, wird die Meldung `BASIC Language Reference Manual - TABLE OF CONTENTS` angezeigt. Von diesem Inhaltsverzeichnis aus kann eine große Vielfalt an Hilfebildschirmen über das BASIC-Modul, einschließlich der Hardwarekonfiguration und Programmierung, aufgerufen werden.
6. Drücken Sie die `[ESC]`-Taste oder bewegen Sie den Mauscursor auf `Done<ESC>`, und klicken Sie die linke Maustaste, um das Hilfesystem zu verlassen. Nun wird wieder das Hauptmenü angezeigt.

Ein typisches Pulldown-Menü

Mit Hilfe eines Pulldown-Menüs ermöglicht PBASE den Aufruf der erforderlichen Optionen.

Beispiel: Das File-Menü ist verfügbar, nachdem der Edit-Modus vom Toplevel-Menü aus aktiviert wurde.

Path: `Toplevel Menu` → `Edit mode` → `File`

QuickStroke: `available for individual items within the File menu`

Bei Verwendung einer Maus bewegen Sie den Cursor auf das File-Menü oben links und klicken die linke Maustaste, um das File-Menü herunterzuziehen. Ist keine Maus vorhanden, drücken Sie die Funktionstaste `[F2]`, um den Cursor auf die Pulldown-Menüs zu bewegen. Bewegen Sie anschließend den Cursor nach rechts bzw. links, bis `File` hervorgehoben ist, und drücken Sie die `[ENTER]`-Taste. Das `FILE`-Dialogfeld wird nun angezeigt:

```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91   1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----FILE-----+
|Edit new file... |=====No-File?+=+
|Load file into current window... <ShftF5>|
|Save file in current window |
|save file As... <F5> |
|Information about current file... |
|-----|
|Merge file from disk... |
|(save Block to disk... ) |
|-----|
|DOS directory shell <CtrlF4>|
|Quit <AltX> |
+-----Cancel<ESC>-----+
|
|
|
|
|
+-----+
1 Help 2 3 4 5 6 7 8 9 0

```

Das **FILE**-Dialogfeld enthält eine Auflistung aller verfügbaren Optionen. Bewegen Sie den Cursor mit den Auf/Ab-Pfeiltasten auf die gewünschte Option. Drücken Sie die [**ENTER**]-Taste, um die Option zu wählen. Wenn das Dialogfeld verlassen werden soll, ohne daß eine Option gewählt wird, drücken Sie **cancel <ESC>**.

Bei einigen Optionen sind rechts QuickStroke-Zuordnungen aufgeführt. Beispiel: Anstatt das File-Menü herunterzuziehen, drücken Sie [**shift F5**], um einen File in das aktivierte Fenster zu laden.

Wechseln zwischen Fenstern

Die Fenster sind in der oberen linken Ecke alphabetisch gekennzeichnet. Die BASIC-Entwicklungssoftware kann bis zu 100 Fenster gleichzeitig öffnen. Jedes davon kann einen eigenen File enthalten oder mit anderen Fenstern verknüpft sein und verschiedene Teile eines großen Files enthalten. Diese Fenster können unabhängig voneinander geöffnet und geschlossen werden.

In der folgenden Abbildung sind drei gleichzeitig geöffnete Fenster dargestellt. Fenster A enthält einen File mit dem Namen BASE#1, und Fenster B und C enthalten einen File mit dem Namen BASE#2. In diesem Fall sind B und C miteinander verknüpft, d.h. sie enthalten jeweils verschieden Teile desselben Files.

```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91 1:15pm
File Window Block Cursor Search Text Layout Print Macro Config Other Exit

++A+=====D:\ABBASIC\BAS\BASE#1=+
|
|100 (this is BASE #1)           | ++B+↑ =====D:\ABBASIC\BAS\BASE#2===+
|110 for x = 1 to 100           | |100 (this is BASE #2)           |
|120 a = x * 2                  | |110 for x = 1 to 50             |
|130 if a = 60 then 180         | |120 a = x * 2 + 5               |
|140 next x                      | |130 if a = 70 then 180          |
|150 goto 200                    | |140 next x                       |
|180 print "a = 60"              | |150 goto 190                     |
|200 end                          | |180 print "a = 70"               |
|                                | |                                  |
|>> EOF <<                       | ++C+↑ =====D:\ABBASIC\BAS\BASE#2===+
|                                | |550 for x = 1 to 100             |
|                                | |560 a = x * 2                     |
|                                | |570 if a = 60 then 600            |
|                                | |580 next x                         |
|                                | |590 goto 700                       |
|                                | |600 print "a = 60"                 |
|                                | |700 end (end of BASE #2)           |
|                                | |>> EOF <<                           |
|                                | |                                  |
+-----+
1 Help 2Menu 3Term 4Indent 5Save 6Search 7MarkB1 8BasHlp 9CopyB1 0MoveB1

```

Eine ausführlichere Erläuterung der im Windows-Menü zur Verfügung stehenden Optionen ist in Kapitel 4 enthalten. An dieser Stelle soll zunächst nur das Konzept, das hinter der Fensterfunktion steht, erläutert werden. Modifizieren Sie hierzu dieses Fenster:

1. Verschieben Sie den Mauscursor auf das Window-Menü (oben links), und klicken Sie die linke Maustaste. Wenn keine Maus verwendet wird, drücken Sie [F2] auf der Tastatur, um auf die Pulldown-Menüs zuzugreifen. Bewegen Sie den Cursor nach rechts oder links, um das Window-Menü hervorzuheben, und drücken Sie die [ENTER]-Taste.


```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91   1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----WINDOW-----+
+==A+|Open          |=====C:\ABBASIC\BAS\FIRST.BDL==+
|   |Close         |
|   |Split         |
|   |List...      <AltF1> |
|   |-----|
|   |Next         <ShftF1> |
|   |Previous    <CtrlF1> |
|   |(Adjacent  <AltW>) |
|   |-----|
|   |Hide         |
|   |Modify size <ScrollLockOn>|
|   |Zoom        <CtrlF3> |
|   |mInimize    |
|   |-----|
|   |linK...     |
|   |(Unlink)    |
|   +-----Cancel<ESC>-----+
|
|
+-----+
1 Help 2 3 4 5 6 7 8 9 0

```

2. Bewegen Sie den Cursor nach unten bzw. setzen Sie den Mauscursor auf **modify size** im Window-Menü. Wenn dieses Element hervorgehoben ist, drücken Sie die **[ENTER]**-Taste oder klicken Sie die linke Maustaste.

Der Rand des Fensters A ist nun hervorgehoben (oder nimmt eine andere Farbe an), womit angezeigt wird, daß die Größe bzw. Position verändert werden kann. Beachten Sie bitte, daß die Meldungzeile direkt über Fenster A nun Anweisungen enthält.

3. Drücken Sie **[shift]** und **[pfeil links]**. Fenster A wird nach links verkleinert. Wenn **[pfeil links]** losgelassen und **[pfeil auf]** gedrückt wird, wird der Bildschirm nach oben verkleinert.
4. Lassen Sie **[shift]** los, und drücken Sie **[pfeil rechts]**, um Fenster A zu positionieren.
5. Drücken Sie **[esc]**, nachdem Sie Größe und Form des Fensters A eingestellt haben.
6. Ziehen Sie das Window-Menü erneut herunter. Wählen Sie mit dem Cursor das Element **zoom** im Window-Menü, um Fenster A wieder in voller Größe anzuzeigen.

Editieren eines BASIC-Programms

Die BASIC-Entwicklungssprache (BASIC Development Language – BDL) optimiert die BASIC-Programmierung durch eine Reihe von Bildschirm-Editierfunktionen, unformatierten Zeileneingaben und Makros. Makros sind Befehle, die eine komplexere Folge von BASIC-Befehlen umfassen.

Nachdem Sie das Programm in der BASIC-Entwicklungssprache geschrieben haben, muß es in die BASIC-Programmiersprache übersetzt/kompiliert werden, um vom BASIC-Modul verstanden zu werden. Auf diesem Wege wird ein BAS-File erstellt, der anschließend in das Modul heruntergeladen werden kann.

Das Übersetzen und das Herunterladen sind im folgenden Abschnitt und in den Kapiteln 8 und 9 beschrieben.

Generelles Editieren

Im letzten Abschnitt wurde die Positionierung und Größeneinstellung von Fenster A beschrieben. Hier wird nun erläutert, wie Sie ein BASIC-Programm editieren.

1. Bewegen Sie den Mauscursor auf das File-Menü (oben links), und klicken Sie die linke Maustaste, um das File-Menü herunterzuziehen. Wenn keine Maus verwendet wird, drücken Sie [F2] auf der Tastatur, um auf die Pulldown-Menüs zuzugreifen. Bewegen Sie den Cursor nach rechts oder links, um **File** hervorzuheben, und drücken Sie die [ENTER]-Taste.
2. Wählen Sie mit dem Cursor **save file as** im File-Menü, und drücken Sie anschließend die [ENTER]-Taste, oder klicken Sie die linke Maustaste. Alternativ hierzu hätten Sie auch **a** drücken können, da dies der hervorgehobene Buchstabe ist, der diesem Menüelement entspricht.

Auf dem Bildschirm wird nun das Dialogfeld **SAVE FILE AS** angezeigt. In der ersten Zeile wird der Name des Files im aktivierten Fenster **?No-File?** angezeigt.

3. Geben Sie **FIRST.BDL** ein.
4. Drücken Sie die [ENTER]-Taste oder bewegen Sie den Mauscursor auf **OK<ENTER>**, und klicken Sie die linke Maustaste.

Das Dialogfeld **SAVE FILE AS** wird nun ausgeblendet und in der Meldungszeile (dritte Zeile von oben) wird nun **"C:\ABBASIC\BAS\FIRST.BDL saved"** angezeigt. Wenn die Software nicht auf Festplatte C: installiert ist oder wenn andere Verzeichnisnamen verwendet werden, lautet die Meldung entsprechend anders.

Sie haben nun einen File gespeichert, der die `GOTO`-Anweisung unter dem Namen `FIRST` mit der Erweiterung `BDL` enthält. `BDL` ist die Abkürzung für BASIC-Entwicklungssprache, und die Eingabe der Fileerweiterung `.BDL` ermöglicht es, daß die Software den korrekten Kompilierer für die Übersetzung des Programms in die BASIC-Sprache erkennen kann. Filenamenerweiterungen sind in Kapitel 3 näher beschrieben.

5. Verwenden Sie das bereits gespeicherte Programm:

```
FIRST.BDL
```

6. Bewegen Sie den Cursor in die obere linke Ecke des Fensters.
7. Geben Sie das folgende Programm ein:

```
{THIS IS A FIRST PROGRAM}  
REM FIRST PROGRAM  
FOR X=1 TO 100  
  A=X*2  
  PRINT A  
  NEXT X  
END
```

Beachten Sie bitte, daß hier keine Zeilennummern eingegeben werden, und daß die Zeilen innerhalb der For/Next-Schleife eingezogen sind. Der Grund dafür ist, daß die BASIC-Entwicklungssoftware die Eingabe eines Programms in der BASIC-Entwicklungssprache erkennt (da die Filenamenerweiterung `BDL` verwendet wurde), und die Standardparameter auf einen programmierbaren Einzug eingestellt sind. In Kapitel 3 wird eine Reihe anderer verwandter Parameter für `BDL`-Files beschrieben.

Sichern eines Programms

Path: Edit mode → File → save file as

Path: Edit mode → File → save file as → save file in current window

QuickStroke: [F5]

Es empfiehlt sich, größere Programme während der Bearbeitung in regelmäßigen Abständen zu sichern, um im Falle einer Systemstörung oder eines Stromausfalls den Verlust größerer, bereits eingegebener Programmteile zu vermeiden.

Die BASIC-Entwicklungssoftware verfügt über eine automatische Sicherungsfunktion, mit der Files während der Bearbeitung auf zweierlei Arten automatisch gesichert werden können. Sie verfügt auch über eine Backup-Funktion, die mehrere Möglichkeiten zur Anfertigung von Sicherungskopien bietet. Diese Funktionen sind in Kapitel 3 näher beschrieben.

1. Bewegen Sie den Mauscursor auf das File-Menü (oben links), und klicken Sie die linke Maustaste. Wenn keine Maus verwendet wird, drücken Sie [F2], um auf die Pulldown-Menüs zuzugreifen. Bewegen Sie den Cursor nach rechts oder links, um das File-Menü hervorzuheben, und drücken Sie die [ENTER]-Taste.
2. Verschieben Sie den Cursor auf `save file in current window`, und drücken Sie die [RETURN]-Taste, oder klicken Sie die linke Maustaste.

Da Sie den File bereits zuvor im aktivierten Fenster benannt und gesichert haben und `save file in current window` gewählt haben, wird davon ausgegangen, daß Sie den File unter dem bereits vorhandenen Namen sichern. Der Text in der Meldungszeile signalisiert, daß der Sicherungsvorgang abgeschlossen ist.

Übersetzen eines Programms

Path: Toplevel Menu → Translate/compile

Path: Toplevel Menu → Edit mode → Other →

Translate/compile

QuickStroke: [CTRL F8]

1. Bewegen Sie den Mauscursor auf das Other-Menü, und klicken Sie die linke Maustaste. Wenn keine Maus verwendet wird, drücken Sie [F2], um auf die Pulldown-Menüs zuzugreifen. Bewegen Sie den Cursor nach rechts oder links, um das Other-Menü hervorzuheben, und drücken Sie die [ENTER]-Taste.
2. Positionieren Sie den Cursor auf Translate/compile, und drücken Sie die [ENTER]-Taste, oder klicken Sie die linke Maustaste. Es erscheint ein Dialogfeld mit der Aufforderung, einen Übersetzer/Kompilierer zu wählen. Zu diesem Zeitpunkt steht nur eine Option zur Verfügung, welche die .BDL-Erweiterung, unter welcher der File gesichert wurde, erkennt.
3. Verschieben Sie den Cursor auf `1747-PBASE Translator`, und drücken Sie die [ENTER]-Taste bzw. die linke Maustaste.

Sofern bei der Eingabe dieses Programms keine Fehler unterlaufen sind, wird die Übersetzung erfolgreich durchgeführt. Waren Fehler enthalten, erscheint in der Meldungszeile eine Fehlermeldung, und der Cursor wird zur Fehlerstelle im Programm verschoben.

Nach dem erfolgreichen Abschluß der Übersetzung eines .BDL-Files wird ein neuer File mit demselben Namen erstellt, dessen Fileerweiterung allerdings .BAS lautet. Dieser File enthält das Programm in der BASIC-Programmiersprache, einschließlich Zeilennummern. Im vorliegenden Beispiel ist nun ein File mit dem Namen FIRST.BDL und ein File mit dem Namen FIRST.BAS vorhanden.

Wichtig: Eine Übersetzung ist nur erforderlich, wenn das Programm in der BASIC-Entwicklungssprache (BDL) geschrieben wurde. Wurde der File allerdings in der BASIC-Programmiersprache erstellt (siehe Referenzhandbuch zur BASIC-Programmiersprache, Publikationsnummer 1746-6.3DE, oder Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771-6.5.113DE), ist eine Übersetzung nicht erforderlich.

Anschluß an das BASIC-Modul

Nun kann das Programm FIRST.BAS an das BASIC-Modul heruntergeladen und dort ausgeführt werden. Hierzu muß das Modul an einen seriellen Port (COM1 bis COM8) des Personalcomputers angeschlossen werden. Ist im PC eine 1784-KR-Karte von Allen-Bradley installiert, kann die Verbindung anstelle zu einem seriellen Port zu dieser Karte hergestellt werden.

Nähere Informationen zur Terminalauswahl sind in Kapitel 3 und Hinweise zur Einrichtung des Terminal-Modus sind in Kapitel 9 enthalten. Einzelheiten über Hardware-Anschlüsse können dem BASIC-Modul SLC 500, Design- und Integrationshandbuch (Publikationsnummer 1746-6.1DE) oder dem Benutzerhandbuch des BASIC-Moduls (Publikationsnummer 1771-6.5.113DE) entnommen werden.

Wechseln in den Terminal-Modus

Path: Toplevel Menu → teRminal [232]

Path: Toplevel Menu → Edit mode → Other → teRminal [232]

QuickStroke: [F3]

1. Bewegen Sie den Mauscursor auf das Other-Menü, und klicken Sie die linke Maustaste. Wenn keine Maus verwendet wird, drücken Sie [F2]. Bewegen Sie den Cursor nach rechts oder links, um das Other-Menü hervorzuheben, und drücken Sie die [ENTER]-Taste.
2. Setzen Sie den Cursor auf teRminal [RS-232], und drücken Sie die [RETURN]-Taste, oder klicken Sie die linke Maustaste. Das Hauptmenü wird ausgeblendet, und der folgende Bildschirm "Terminal Mode" wird angezeigt:

AB Spares

```
HELP<F1> |MENU<F2> |EXIT<F3> |COM1 1200 8 1 N |LOG CLOSED |ANSI |07-04-91 06:33pm
File Setup Display deBug Exit
Edit
PBASE RS-232 TERMINAL MODE Version x.xx
```

Die Angabe [RS-232] hinter der Terminalauswahl zeigt an, daß die BASIC-Entwicklungssoftware derzeit für die RS-232-Kommunikation konfiguriert ist. Je nach Systemkonfiguration erscheint an dieser Stelle [RS-232] oder [DH-485].

3. Sobald Sie die [ENTER]-Taste drücken, sollte das Aufforderungszeichen > erscheinen, um anzuzeigen, daß eine Kommunikationsverbindung mit dem BASIC-Modul hergestellt wurde. Wenn das Aufforderungszeichen nicht erscheint, bedeutet dies, daß keine Kommunikationsverbindung mit dem BASIC-Modul hergestellt wurde. Überprüfen Sie die Terminalwahl (Kapitel 3), die Kommunikationskonfiguration (Kapitel 9) und die Brückeneinstellungen am Modul, sowie die Verdrahtung (siehe BASIC-Modul SLC 500, Design- und Integrationshandbuch, Publikationsnummer 1746-6.1DE, oder Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771-6.5.113DE).

Herunterladen eines Programms

Path: Toplevel Menu → teRminal [232] → File → Download from host to module

Path: Toplevel Menu → Edit mode → Other → teRminal [232] → File → Download from host to module

QuickStroke: [PgDn]

1. Bewegen Sie den Mauscursor auf dem Terminal-Modusbildschirm auf **File**, und klicken Sie die linke Maustaste. Wenn keine Maus verwendet wird, drücken Sie [F2]. Bewegen Sie den Cursor nach rechts oder links, um das File-Menü hervorzuheben, und drücken Sie die [ENTER]-Taste.
2. Positionieren Sie den Cursor auf **Download from host to module**, und drücken Sie die [ENTER]-Taste, oder klicken Sie die linke Maustaste. Es erscheint ein Dialogfeld, in dem Sie aufgefordert werden, einen Filenamen zum Herunterladen einzugeben.
3. Durch Drücken der [ENTER]-Taste wird die übersetzte Version (FIRST.BAS) des Files, den Sie aktiv bearbeiten (FIRST.BDL), an das BASIC-Modul heruntergeladen. *Nur Files mit der Erweiterung .BAS können an das Modul heruntergeladen werden.*

Verlassen eines BASIC-Programms

Verlassen Sie ein Programm wie folgt:

1. Drücken Sie [F3], um den Terminal-Modus zu verlassen. Nun wird wieder das Hauptmenü bei aktiviertem Edit-Modus angezeigt.
2. Bewegen Sie den Mauscursor in der zweiten Bildschirmzeile auf **Exit**, und klicken Sie die linke Maustaste. Wenn keine Maus verwendet wird, drücken Sie [F2] auf der Tastatur, um die Pulldown-Menüs aufzurufen. Bewegen Sie den Cursor nach rechts oder links, um **Exit** hervorzuheben, und drücken Sie die [ENTER]-Taste. Somit kehren Sie zum Toplevel-Menü zurück.
3. Verschieben Sie den Mauscursor im Toplevel-Menü auf **quit**, und klicken Sie die linke Maustaste. Sie können den Cursor auch nach oben oder unten bewegen, um **quit** hervorzuheben und anschließend die [ENTER]-Taste drücken. Somit wird die BASIC-Entwicklungssoftware verlassen, und Sie kehren zur DOS-Ebene zurück.

Konfiguration der Software

In diesem Kapitel wird die Konfiguration der Software beschrieben. Die wichtigsten Themen umfassen:

- Editieroptionen
- Anzeigeeoptionen
- Maus- und Tastaturoptionen
- Filenamenerweiterungen
- Druckeroptionen
- Speicheroptionen
- File-Backup, Zwischenfiles und automatisches Sichern
- Standardeinstellung der Such- und Ersetzfunktion
- Wahl des Terminalemulationsmodus
- Sicherung der Konfiguration und Beenden

Wahl des Menüs "Setup and Configuration"

Das Menü "Setup and Configuration" kann vom Toplevel-Menü aus oder vom Hauptmenü aus gewählt werden.

Path: Toplevel Menu → Configuration menu

Path: Main Menu → Config

Zum Aufruf des Menüs "Setup and Configuration" beispielsweise vom Toplevel-Menü aus bewegen Sie den Mauscursor auf die Option **Configuration menu** und klicken die linke Maustaste. Wenn keine Maus verwendet wird, bewegen Sie den Cursor nach oben oder unten, um die Option **Configuration menu** hervorzuheben und drücken anschließend die [ENTER]-Taste.

```
L:3 C:1 1747-PBASE V x.xx [Text Edit] Ins 175k 06-12-91 1:15pm
File Window Block cursor Search Text Layout Print Macro Config Other Exit
+-----SETUP AND CONFIGURATION-----+
+|DOS=5.0 CPU=80386 |=====No-File?==+
||Video Card = VGA Color |
||PBASE Path = C:\ABBASIC\BDS\ |
||Expanded Memory N/A |
||Serial # PBASE1 |
||-----|
||Edit settings... |
||Display setup... |
||Mouse / Key repeat setup... |
||Filename extensions... |
||Printer... |
||sWapping / Expanded memory... |
||Backups / Temp files / Autosave... |
||Search and Replace defaults |
||Terminal Selection Menu... |
||saVe configuration and exit |
|| |
|+=====Exit Setup<ESC>=====+
|
+-----+
1 Help 2 3 4 5 6 7 8 9 0
```

Zusätze der Befehlszeile

Mit der BASIC-Entwicklungssoftware können fast alle Optionen über das Konfigurationsmenü eingestellt werden. Ist jedoch die zwangsweise Einstellung einer Konfiguration oder die vorübergehende Änderung einer Einstellung erforderlich, kann dies mit einem Zusatz der Befehlszeile erreicht werden.

Die folgenden Zusätze der Befehlszeile sind nach dem Starten der BASIC-Entwicklungssoftware verfügbar. Soll ein Zusatz verwendet werden, muß zuerst **PBASE** und anschließend der gewünschte Zusatz bzw. die gewünschten Zusätze eingegeben werden. (Wahlweise kann zwischen **PBASE** und dem Schrägstrich / eine Leerstelle eingegeben werden. Bei der Eingabe von mehreren Zusätzen müssen diese durch eine Leerstelle getrennt werden.)

Tabelle 3.A
Zusätze der Befehlszeile

Zusatz	Bedeutung	Zusammenfassung
/MN	Deaktivierung der Maus	Die BASIC-Entwicklungssoftware stellt automatisch fest, ob ein Maustreiber anwesend ist. Sie erfordert DOS-Version 3.1 oder höher.
/MY	Aktivierung der Maus	
/KN	Tastengeschwindigkeit bleibt unverändert	Bei einigen Personalcomputern mit besonderen ANSI.SYS-Ersatzdateien treten Störungen ein, wenn versucht wird, die Tasten- bzw. Tastaturwiederholungsgeschwindigkeit zu ändern. Treten Tastaturstörungen auf, empfiehlt sich der Zusatz /KN (/KY ist die Standardvorgabe). Wird das Problem dadurch behoben, muß diese Konfiguration vom Konfigurationsmenü aus gespeichert und das Menü verlassen werden. Somit wird /KN als Standardvorgabe gespeichert, d.h., diese Zusätze der Befehlszeile müssen nicht erneut verwendet werden. Um die vorherige Konfiguration wieder aufzurufen, muß die Software mit dem Zusatz /KY gestartet und diese Konfiguration erneut gesichert werden.
/KY	Tastengeschwindigkeit wird verändert	
/KR	1784-KR-Karte ist installiert	Wenn diese Karte verwendet werden soll, muß zuerst sichergestellt werden, daß ihr Treiber im Haupt-Unterverzeichnis der BASIC-Entwicklungssoftware (ABBASIC\BDS, sofern er während der Installation nicht umbenannt wurde) enthalten ist. Anschließend muß dieser Zusatz der Befehlszeile jedesmal, wenn die BASIC-Entwicklungssoftware läuft, verwendet werden.
/EN	erweiterte Tastatur darf nicht verwendet werden	Einige Personalcomputer und einige besondere speicherresidente Programme funktionieren nicht, wenn versucht wird, über die BASIC-Entwicklungssoftware erweiterte Tastaturfunktionen aufzurufen. Treten tastaturbezogene Störungen ein, sollte der Zusatz /EN programmiert werden.
/NR	Wiederherstellen deaktiviert	Deaktiviert die Wiederherstellungsfunktion der BASIC-Entwicklungssoftware. Dieser Zusatz sollte dann verwendet werden, wenn die Wiederherstellungsfunktion (im Menü "Setup and Configuration") aktiviert ist und wenn die BASIC-Entwicklungssoftware abgearbeitet werden soll, ohne daß der vorherige Status des Editors wiederhergestellt wird.
/B	Schwarzweiß	Veranlaßt, daß die Bildschirmanzeige der Software zwangsweise in schwarzweiß und nicht in Farbe dargestellt wird.
/Xn	Einstellung der Bildschirmanzeigengröße auf n	Mit den Zusätzen /X und /Y wird die automatische Erkennung der Bildschirmanzeigengröße übersteuert und die Größe neu definiert. Dies ist besonders beim Einsatz von speziellen Video-Betriebsarten einiger SUPER- und ENHANCED EGA/VGA- (und anderen) Karten von Vorteil. Die für n eingegebenen Werte müssen für die jeweilige spezielle Video-Betriebsart zulässig sein. Bei Verwendung eines dieser Zusätze werden bei Wahl der Anzeigeeinrichtung aus dem Konfigurationsmenü die Optionen der Video-Betriebsart deaktiviert.
/Yn	Einstellung der Bildschirmanzeigenbreite auf n	
/N	Keine Schneeunterdrückung bei einem CGA-Bildschirm	Die Nachvollziehung der CGA-Video-E/A darf nicht abgewartet werden. Hierdurch wird die Schneeunterdrückung deaktiviert.

Zusatz	Bedeutung	Zusammenfassung
/EMSOFF	Der erweiterte Speicher darf nicht belegt werden	Der erweiterte Speicher darf nicht belegt werden. Anderenfalls werden alle erweiterten Speicheroptionen, die über das Konfigurationsmenü verfügbar sind, deaktiviert.

Laden von Files von der Befehlszeile aus

Von der Befehlszeile aus können Files, die geladen werden sollen, mit dem **PBASE**-Befehl spezifiziert werden. Bei der Spezifizierung von mehreren Files (jeweils durch eine Leerstelle getrennt) werden für jeden weiteren File Fenster erstellt. DOS-Platzhalterzeichen (* und ?) sind zulässig.

Beispiele:

Mit **PBASE *.BAS** werden Fenster geöffnet, die zum Laden aller .BAS-Files in das aktuelle Verzeichnis erforderlich sind.

Mit **PBASE TEST.BDL TEST.BAS** wird ein Fenster für TEST.BDL und anschließend ein Fenster für TEST.BAS geöffnet.

Konfigurieren der Editieroptionen

Path: Toplevel Menu → Configuration menu → Setup and Configuration menu → Edit settings

Path: Main Menu → Config → Setup and Configuration menu → Edit settings

Durch Anwählen der Option **Edit settings** im Menü "Setup and Configuration" erscheint die folgende Anzeige:

```

L:3 C:1 1747-PBASE V x.xx [Text Edit] Ins 175k 06-12-91 1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----SETUP AND CONFIGURATION-----+
+|DOS=5.0 CPU=80386 |=====No-File?|=+
||Video+-----EDIT SETTINGS-----+| | |
||PBASE|Page break string..... 0|
||Expan|Word delimits..... ( ) ' # $ @ ! % * { } ? / | 9|
||Seria|Max undo count..... 50|
||-----|
||Edit |
||Displ|Cursor:
||Mouse| Insert Overwrite Options:
||Filen| (.) - ( ) Underline [ ] Truncate spaces
||Print| ( ) - (.) 1/2 block [ ] Control-Z at EOF
||sWapp| ( ) - ( ) 1/3 block [ ] CR/LF at EOF
||Backu| ( ) - ( ) Full block [ ] Restore previous status
||Searc|
||Termi|Tab expand: Column block move style: Default mode:
||SaVe | (.) Tabs (.) Delete space (.) Insert
|| | ( ) Spaces ( ) Leave space ( ) Overwrite
|+=====|
| | OK<ENTER> Cancel<ESC> Help<F1>
| +-----+
+-----+
STRING INPUT: Next field<TAB>, Previous<ShftTAB>, Select<Spacebar>

```

AB Spares

Im Dialogfeld **EDIT SETTINGS** können verschiedene Editor-Parameter konfiguriert werden, die sich auf den Betrieb und die Anwendung der BASIC-Entwicklungssoftware auswirken. Drücken Sie **[Tab]**, um die nächste Option zu wählen.

- **Page break string** — ermöglicht die Konfiguration einer anwenderdefinierten Seitenumbruch-Zeichenkette mit maximal zehn Zeichen. Diese Zeichenkette wird von der BASIC-Entwicklungssoftware als Seitenumbruch behandelt. Die standardmäßige Seitenumbruch-Zeichenkette ist ein Formularvorschubzeichen (ASCII 12), das generell mit der Tastenkombination **[Ctrl-L]** eingegeben wird. Nähere Einzelheiten sind in den Hilfe-Bildschirmtexten enthalten.
- **Word delimits** — ermöglicht die Eingabe einer Zeichenkette zur Definition der Wortbegrenzung. Die standardmäßige Wortbegrenzungszeichenkette lautet `()"'',#$$!%*{}?/|9`, wobei das **[Leerzeichen]** mit der Leertaste und **|9** mit der **[Tab]**-Taste eingegeben wird.

Die als Wortbegrenzer gewählten Zeichen stehen in der Regel nach einem Wortende. Beim Verschieben des Cursors um ein Wort nach links **[Ctrl - Pfeil links]** oder um ein Wort nach rechts **[Ctrl - Pfeil rechts]** wird er auf das erste Zeichen rechts vom Wortbegrenzer gesetzt. Stehen zwei oder mehrere Wortbegrenzer nebeneinander, wird der Cursor auf das erste Zeichen rechts vom äußerst rechten Wortbegrenzer gesetzt.

Im folgenden Beispiel lautet die Wortbegrenzerzeichenkette: **[space] ? : .** Der Cursor stoppt an jedem **w**, da ein Wortbegrenzer vorhanden ist:

inter winter?winter:winter

- **Max undo count** — ermöglicht die Konfiguration der Rückgängigmachungsschritte für jedes Fenster. Die Höchstanzahl beträgt 65535. Bei Eingabe einer 0 wird diese Funktion deaktiviert. Die Standardeinstellung lautet 50. Je größer diese Anzahl ist, desto kleiner ist der zum Editieren verfügbare Speicher.
- **Insert/Overwrite** — ermöglicht die Wahl zwischen vier verschiedenen Cursorgrößen. Für den Einfüge- und Überschreibungscursor kann jeweils eine verschiedene Größe gewählt werden, um leichter erkennen zu können, welcher Modus aktiviert ist. Wählen Sie die gewünschte Cursorgröße mit dem Mauscursor aus, und klicken Sie die linke Maustaste. Wenn keine Maus vorhanden ist, drücken Sie die **[Tab]**-Taste, um dieses Feld aufzurufen, positionieren Sie den Cursor mit den Pfeil-Auf- und Pfeil-Ab-Tasten auf das gewünschte Feld, und drücken Sie die **[Tab]**-Taste erneut, um ein Element zu wählen und dann zum Optionsfeld zu wechseln.

- **options** — Zur Auswahl dieser verschiedenen Editieroptionen bewegen Sie den Mauscursor auf das gewünschte Feld und klicken die linke Maustaste. Wenn keine Maus vorhanden ist, drücken Sie die [Tab]-Taste, um auf dieses Feld zuzugreifen, positionieren den Cursor mit den Pfeil-Auf- und Pfeil-Ab-Tasten und drücken anschließend die [Leertaste] zur Auswahl der gewünschten Elemente.
 - **Truncate spaces** — Diese Option legt für die 1747-PBASE fest, ob Leerstellen im Anschluß an Textzeilen abgeschnitten werden sollen oder nicht. Dieser Vorgang erfolgt nur beim Laden eines Files.
 - **ctrl-z at EOF** — Einige Programme erfordern am Ende eines Textfiles die Tastenkombination [ctrl-z] (ASCII 26). Diese Funktion kann mit dieser Option ein- und ausgeschaltet werden.
 - **CR/LF at EOF** — Mit dieser Option wird festgelegt, ob am Fileende ein Wagenrücklauf/Zeilenvorschub gesetzt werden soll. Dieser Abschluß wird in der Regel für DOS-Dateien verwendet.
 - **Restore previous status** — ermöglicht das Ein- und Ausschalten der Wiederherstellungsfunktion. Wenn diese Funktion aktiviert ist, wird der gesamte STATUS des Editors beim Verlassen der BASIC-Entwicklungssoftware gesichert. Hierzu gehören geladene Files, Fenster, Cursorpositionen, markierte Blöcke und Aufzeichnungen fast sämtlicher Meldungen.

Wichtig: Wenn ein zu ladender File auf der Befehlszeile spezifiziert wird, wird die Wiederherstellungsfunktion nicht aktiviert. Diese Funktion kann auch bei der Initialisierung der BASIC-Entwicklungssoftware von der DOS-Befehlszeile aus mit dem /NR-Zusatz unterdrückt werden.

- **Tab expand** — Mit dieser Option kann konfiguriert werden, wie die 1747-PBASE Tabulatorzeichen und die [Tab]-Taste abarbeitet. Wenn **tabs** gewählt ist, wird mit der [Tab]-Taste veranlaßt, daß ein Tabulatorzeichen eingefügt und der Cursor auf die nächste Tabulatorposition verschoben wird. Wenn **spaces** gewählt ist, wird mit der [Tab]-Taste veranlaßt, daß bis zum nächsten Tabulator Leerstellen (ASCII 32) eingefügt werden.

Wichtig: Wenn **tabs** gewählt ist, kann die Tabulatoreinstellung jederzeit geändert werden. Wenn **spaces** gewählt ist, kann die Tabulatoreinstellung für den File nicht vom Layout-Menü aus durch Eingabe einer neuen Tabulatoreinstellung geändert werden.

- **Column block move style** — Mit dieser Option wird gewählt, ob die Leerstelle der ursprünglichen Position des Spaltenblocks erhalten oder gelöscht werden soll. Der Hilfetext enthält ein Beispiel.
- **Default mode** — Mit dieser Option kann konfiguriert werden, ob die 1747-PBASE ursprünglich im Insert- oder Overwrite-Modus aufgerufen wird. Diese Konfiguration kann im Edit-Modus durch Drücken der [Insert]-Taste umgeschaltet werden.

AB Spaces

Wichtig: Die Wortumbruchfunktion sollte beim Editieren von .BAS- und .BDL-Files ausgeschaltet sein. Sie kann jedoch bei allgemeinen Textverarbeitungsanwendungen verwendet werden.

Konfiguration der Anzeigeoptionen

Path: Toplevel Menu → Configuration menu → Setup and Configuration menu → Display setup
Path: Main Menu → Config → Setup and Configuration menu → Display setup

Durch Anwahl der Option `Display setup` im Menü “Setup and Configuration” kann der Videomodus auf 25 oder 43/50 Zeilen eingestellt werden. Bei der Einstellung auf 43/50 Zeilen können Sie einen größeren Teil des Programms auf dem Bildschirm einsehen, wobei jedoch die Schrift erheblich kleiner ist. Nähere Einzelheiten sind auf den Hilfebildschirmen enthalten.

Konfiguration der Maus- und Tastaturoptionen

Path: Toplevel Menu → Configuration menu → Setup and Configuration menu → Mouse / Key repeat setup
Path: Main Menu → Config → Setup and Configuration menu → Mouse / Key repeat setup

Durch Anwahl der Option `Mouse / Key repeat setup` im Menü “Setup and Configuration” können Sie die Maus aktivieren bzw. deaktivieren, die Empfindlichkeit der Maus einstellen und den Mauscursor bei Verwendung der Tastatur ausschalten. Ferner kann die Wiederholrate der Tastatur (Verzögerung zwischen den Tastenanschlägen bei gedrückter Taste) und die Wiederholungsverzögerung (Verzögerung, bevor bei gedrückter Taste mit der Wiederholung begonnen wird) eingestellt werden. Einzelheiten hierzu sind auf den Hilfebildschirmen enthalten.

Konfiguration der Filenamenerweiterung

Path: Toplevel Menu → Configuration menu → Setup and Configuration menu → Filename extensions
Path: Main Menu → Config → Setup and Configuration menu → Filename extensions

Die Erweiterungen der Filenamen werden wie folgt konfiguriert:

1. Wählen Sie im Menü “Setup and Configuration” die Option `Filename extensions`. Es wird eine Auflistung von Filenamenerweiterungen angezeigt, die derzeit von der BASIC-Entwicklungssoftware erkannt werden. Dieser Liste können Erweiterungen hinzugefügt (Create) werden; ähnlich können Erweiterungen aus der Liste gestrichen (Delete) werden.

Auswirkung auf das Programmieren: Nachdem ein File geladen wurde, sucht die Software diese Liste nach der Erweiterung des geladenen Files ab.

```

L:3  C:1  1747-PBASE V x.xx  [Text Edit]  ↑  Ins 175k  06-12-91  1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----SETUP AND CONF+-----EXTENSION SETUP-----+
+|DOS=3.30 CPU=80386 | Create<Ins> Delete<Del> |=====No-File?==+
||Video Card = VGA Colo| Copy<F4> Edit<F3> | |
||PBASE Path = C:\ABBAS| | |
||Expanded Memory N/A | DEFAULT | |
||Serial # 1234567 | BDL | ↑ |
||-----| BAS | |
||Edit settings... | DOC | ↓ |
||Display setup... | | |
||Mouse / Key repeat se| Edit<ENTER> Done<ESC> Help<F1> |
||Filename extensions..+=====+
||Printer... | |
||sWapping / Expanded memory... | |
||Backups / Temp files / Autosave...| |
||Search and Replace defaults | |
||Terminal Selection Menu... | |
||saVe configuration and exit | |
|| | |
|+=====Exit Setup<ESC>=====+
|
+-----+
LIST BOX: Next field<TAB>, Previous<ShftTAB>, ↑↓<PgUp><PgDn><Home><End>

```

2. Drücken Sie die [TAB]-Taste, um die nächste Option hervorzuheben. Drücken Sie die [ENTER]-Taste, um eine Option zu wählen.
3. Heben Sie in der Liste der Filamenerweiterungen die zu bearbeitende Erweiterung hervor (in diesem Beispiel BDL), und wählen Sie edit<F3>. Das Dialogfeld EXTENSION SETUP für die Erweiterung BDL wird angezeigt. Aus diesem Feld sind für BDL-Files die folgenden Einstellungen zu ersehen: der Edit-Modus lautet Text, der Einzug (Indent Style) ist programmierbar (smart), der Tabulatorabstand (Tab spacing) ist 2, der rechte Rand (Right margin) ist 78 und der Wortumbruch (Word wrap) ist ausgeschaltet.

Auswirkung auf das Programmieren: Wählen Sie beispielsweise ein Programm mit dem Namen FIRST.BDL, das in ein Editierfenster geladen werden soll, sucht die Software die Auflistung der Filamenerweiterungen ab und findet BDL. Sie bezieht sich beim Laden des Files auf die Parameter, die für BDL in das Dialogfeld EXTENSION SETUP eingegeben wurden. Beim Editieren dieses Files werden die Einzüge, Tabulator-, Rand- und Wortumbrucheinstellungen entsprechend diesen Einstellungen ausgeführt.

```
+-----EXTENSION SETUP-----+
| Extension:  BDL                   |
| Edit mode:          Indent style:  Tab spacing:  2  |
| (.) Text           ( ) Off         Right margin: 78 |
| ( ) Document       ( ) Auto        |
|                   (.) Smart        [ ] Word wrap   |
|
| Language type:    [1747_PBASE    ] ↓             |
| Default directory: .....          |
| Compiler-Program setup            |
|
|                               OK<ENTER> Cancel<ESC> Help<F1> |
+-----+
```

4. Heben Sie mit dem Cursor die Option "Language type" der .BDL-Erweiterung hervor. (Verwenden Sie hierzu die Maus, oder drücken Sie die [Tab]-Taste). Drücken Sie [Pfeil ab], um das Auflistungsfeld der Programmiersprachen (Language) aufzurufen. Beachten Sie hier, daß für die Erweiterung .BDL die Sprache **1747_PBASE** aus dieser Liste gewählt wurde.

Auswirkung auf das Programmieren: Beim Editieren von Programmen erkennt die Software die Programmiersprache **Language type** aus der Konfigurationstabelle der Fileerweiterung und löst entsprechende Reaktionen aus. Einzelheiten können dem Hilfebildschirm "Language Specific Support" (Sprachenspezifische Hilfe) entnommen werden.

```
+-----LANGUAGE-----+
| Create<Ins> Delete<Del>         |
|
| 1747-PBASE                      |
| NONE                            |
| BASIC                            |
|                                  |
|                               ↑ ↓ |
|                               |   |
|                               |   |
|                               |   |
| OK<ENTER> Cancel<ESC> Help<F1> |
+-----+
```

5. Drücken Sie die [ESC]-Taste, um das Auflistungsfeld "Language" zu verlassen und zum Dialogfeld **EXTENSION SETUP** zurückzukehren. Mit dem nächsten Parameter, **Default directory**, werden die DOS-Verzeichnispfade zum Laden und Sichern von Files definiert. An dieser Stelle kann ein Laufwerk und/oder Verzeichnis- und Unterverzeichnispfad eingegeben werden. Einzelheiten sind auf dem Hilfebildschirm "Default directory" aufgeführt.

Auswirkung auf die Programmierung: Wenn der Name eines Files eingegeben wird, der in ein Fenster geladen werden soll, kann zusammen mit dem Filenamen auch ein DOS-Pfad eingegeben werden. Anderenfalls wird das derzeit aktive Verzeichnis nach dem jeweiligen Filenamen abgesucht. Wird der Name nicht gefunden, wird das spezifizierte Vorgabeverzeichnis abgesucht.

Beim Sichern eines Files von einem geöffneten Fenster aus kann außer dem Filenamen auch noch ein DOS-Pfad eingegeben werden. Anderenfalls wird der File in dem hier eingegebenen Standardverzeichnis gesichert.

6. Drücken Sie die [TAB]-Taste, um mit dem Cursor die Option **Compiler-Program setup** für die Erweiterung .BDL hervorzuheben. Drücken sie anschließend die [ENTER]-Taste. Das Auflistungsfeld "Compiler/Program setup" wird angezeigt, aus dem der gewünschte Kompilierer ausgewählt werden kann. In diesem Beispiel steht für die .BDL-Erweiterung nur ein Kompilierer zur Verfügung: 1747-PBASE Translator.

Auswirkung auf das Programmieren: Beim Versuch, einen File zu übersetzen/kompilieren, bezieht sich die Software nochmals auf die Erweiterungseinstellungstabelle. Dort liegt die Information vor, daß Files mit der Erweiterung .BDL das Kompilierprogramm "1747-PBASE Translator" verwenden. Dieser Kompilierer wird zum Übersetzen/Kompilieren verwendet.

```
+-----COMPILER/PROGRAM SETUP-----+
|      Create<Ins> Delete<Del>      |
|      Copy<F4> Edit<F3>            |
|                                     |
|      1747-PBASE Translator        |
|                                     |
|                                     |
|      Edit<ENTER> Done<ESC> Help<F1> |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

7. Drücken Sie die [ENTER]-Taste. Das Dialogfeld **COMPILER/PROGRAM SETUP** wird angezeigt. Hier können mehrere Parameter, die sich auf den Kompilierer beziehen, definiert werden. Details zu den Parametern in diesem Dialogfeld sind in Kapitel 8 enthalten.

Auswirkung auf das Programmieren: Das Kompilieren wird durch die Einstellungen des Dialogfeldes **COMPILER/PROGRAM SETUP** gesteuert. Diese Parameter bestimmen unter anderem, ob der File zuerst gesichert, die Bildschirmanzeige ausgeblendet und Kompiliererfehler bearbeitet werden und auf welche Weise der Speicher während des Kompilierens zugeordnet wird.

```

+-----COMPILER/PROGRAM SETUP-----+
| Description: 1747-PBASE Translator      |
| Command line: BTRAN <FILE> /I\ABBASIC\BAS..... |
| Program type: [1747-PBASE              ] |
| Memory requited in k bytes(0=all): 0    |
|                                         |
| Swap method:                            Options: |
| (.) Use default                          [ ] Clear screen |
| ( ) No swap                              [X] Save all files |
| ( ) To disk only                         [ ] Reload file   |
| ( ) To EMS only                          [ ] Do NOT process compiler errors |
| ( ) To EMS then to disk                  |
|                                         |
|                                         OK<ENTER> Cancel<ESC> Help<F1>
+-----+

```

Konfiguration der Druckeroptionen

Path: Toplevel Menu → Configuration menu → Printer
 Path: Main Menu → Config → Printer

Bei der Wahl der Option **Printer** im Menü “Setup and Configuration” wird eine Auflistung der üblichen Drucker angezeigt. Sie können einen Drucker aus dieser Auflistung wählen oder mit der Option **Create<Ins>** der Liste einen nicht aufgeführten Drucker hinzufügen. Mit der Option **Edit<F3>** können die Parameter eines aufgeführten bzw. hinzugefügten Druckers geändert werden.

Weitere Hinweise zum Drucken und zur Druckerkonfiguration sind in Kapitel 7 enthalten.

Konfiguration der Speicheroptionen

Path: Toplevel Menu → Configuration menu →
 sWapping/Expanded memory
 Path: Main Menu → Config → sWapping/Expanded memory

Wenn Sie im Menü “Setup and Configuration” die Option **sWapping/Expanded memory** wählen, werden auf dem Bildschirm mehrere speicherbezogene Parameter angezeigt, mit denen die Belegung eines konventionellen Speichers (die ersten 640 KByte des RAM-Speichers), des erweiterten Speichers und des PC-Festplattenspeichers optimiert werden kann.

Wichtig: Die BASIC-Entwicklungssoftware verwendet einen *Erweiterungs-* und keinen *Zusatzspeicher*. Die meisten Personalcomputer auf 80386-Basis werden mit einer Software geliefert (in der Regel mit der Bezeichnung EMM – Expanded Memory Manager), die es ermöglicht, einen Teil des Speichers als Erweiterungsspeicher zu verwenden. Bei den meisten Computern auf 80386- und 80286-, 8088-, 8086-Basis ist für einen Erweiterungsspeicher eine besondere Platine erforderlich. Näheres ist in der Hardware- und DOS-Dokumentation Ihres Computers aufgeführt.

Beim Editieren bzw. Bearbeiten eines Files wird der konventionelle Speicher des PCs gewöhnlich vollständig belegt. Wenn dies geschieht, versucht die BASIC-Entwicklungssoftware, Teile eines Files in den Erweiterungsspeicher oder auf eine Festplatte abzuspeichern. Dieser Auslagerungsvorgang wird transparent ausgeführt.

Bevor Sie einen dieser Parameter definieren, sollten Sie feststellen, wie groß der Erweiterungsspeicher (EMS) Ihres Personalcomputers ist. Wenn Sie mit der Anwendung dieser Software vertrauter werden, und beginnen, Shell-DOS- oder anwenderdefinierte Programmfunktionen zu verwenden, sollten Sie die voraussichtliche maximale Programmgröße festlegen und einen entsprechenden Speicherbereich zuweisen.

Wichtig: Prüfen Sie vor der Wahl einer Speicheroption, wie groß der Erweiterungsspeicher Ihres Computers ist. Wenn nicht genügend Speicherplatz zur Verfügung steht, besteht die Möglichkeit, daß ein Systemfehler eintritt.

Die hier angezeigten Parameter der Speicherauslagerung ähneln denen des Kompilierer-Einrichtungsmenüs, das beim Einrichten eines Kompilierers für eine bestimmte Filenamenerweiterung aufgerufen wird (siehe Kapitel 8). Beachten Sie bitte, daß die für einen Kompilierer gewählte Speicherauslagerungsmethode vor der hier gewählten Methode Vorrang hat, wenn dieser Kompilierer verwendet wird. Details sind auf den Hilfebildschirmen enthalten.

AB Spares

Sicherungsfiles, Zwischenfiles und automatische Sicherung (Autosave)

Path: Toplevel Menu → Configuration menu → Backups/Temp files/Autosave

Path: Main Menu → Config → Backups/Temp files/Autosave

Wenn Sie im Menü "Setup and Configuration" die Option **Backups/Temp files/Autosave** wählen, erscheint ein Dialogfeld mit mehreren Parametern zur Anfertigung von Sicherungsfiles, Zwischenfiles und zum Sichern von offenen Files.

Sicherungsfiles

Sicherungsfiles sind zuvor gesicherte Versionen von Files, die editiert wurden und im Falle von Störungen oder nicht gewünschten Änderungen als ursprünglicher File wiederhergestellt werden können. Für die Art und Häufigkeit der Sicherungsaufzeichnung stehen mehrere Optionen zur Verfügung.

Zwischenfiles

Zwischenfiles werden während der Speicherauslagerung oder beim Aufruf der DOS-Shell erstellt. Obwohl der Zugriff auf diese Files beschränkt ist, und sie beim Verlassen der BASIC-Entwicklungssoftware gewöhnlich gelöscht werden, können sie in einem bestimmten Verzeichnis gespeichert werden. Über dieses Verzeichnis können auch Files an eine RAM-Diskette übertragen werden, sofern dieses Dienstprogramm auf dem Computer installiert ist.

Wichtig: Wenn für die Zwischenfiles kein Verzeichnispfad eingegeben wird, werden sie auf dem derzeit aktiven Laufwerk und im derzeit aktiven Verzeichnis gesichert. Hierdurch können jedoch Störungen verursacht werden, wenn Sie von der BASIC-Entwicklungssoftware aus beispielsweise die DOS-Shell aufrufen, in ein anderes Diskettenlaufwerk überwechseln und anschließend wieder die Software aufrufen. Die Zwischenfiles werden dann auf diese Diskette geschrieben, und es tritt eine Störung ein, wenn die Diskette voll ist. Um solch eine Störung zu vermeiden, empfiehlt es sich, den Namen des Verzeichnisses einzugeben, in dem die Software installiert wurde, z.B. C:\ABBASIC\BDS.

Automatische Sicherung (Autosave)

Mit der automatischen Sicherungsfunktion Autosave werden Files nach einem bestimmten Zeitraum, in dem keine Eingaben über die Tastatur erfolgen, automatisch gesichert. Hierdurch wird verhindert, daß im Falle eines Stromausfalls oder Systemfehlers ein großer Teil des soeben geschriebenen Programms verlorengeht.

Einzelheiten und eine nähere Beschreibung dieser Parameter sind auf den Hilfebildschirmen enthalten.

Standardeinstellung der Such- und Ersetzfunktion

Path: Toplevel Menu → Configuration menu → Search and Replace defaults

Path: Main Menu → Config → Search and Replace defaults

Wenn im Menü “Setup and Configuration” die Option **search and replace defaults** gewählt wird, erscheint ein Auflistungsfeld mit drei Suchfunktionen. Wenn eine dieser drei Funktionen hervorgehoben und gewählt wird, erscheint ein Dialogfeld, das die Parameter der jeweiligen Suchfunktion enthält.

Diese Parameter sind auf Standardwerte eingestellt, durch die festgelegt wird, wie die Suche durchgeführt wird. Wird die Suche im Edit-Modus durchgeführt, wird ein nahezu identischer Bildschirm mit den von Ihnen eingegebenen Parametern angezeigt. Die Parameter können während der Suche geändert werden, wobei die Änderung solange in Kraft bleibt, bis die BASIC-Entwicklungssoftware verlassen wird. Wenn die BASIC-Entwicklungssoftware anschließend erneut aufgerufen wird, gelten wieder die hier vorgegebenen Standardwerte.

Mit Hilfe von anwenderdefinierten Tastenmakros können gewünschte Einstellungen der Such- und Ersetzfunktion gewählt werden. Tastenmakros sind in Kapitel 4 beschrieben.

Die Hilfebildschirme enthalten nähere Einzelheiten und eine Beschreibung dieser Parameter.

Wahl des Terminalemulationsmodus

Path: Toplevel Menu → Configuration menu → Terminal Selection Menu

Path: Main Menu → Config → Terminal Selection Menu

Mit der Option **Terminal Selection Menu** im Menü “Setup and Configuration” kann der für die Kommunikation mit dem BASIC-Modul gewünschte Terminalemulationsmodus gewählt werden. Hierzu stehen die drei folgenden Terminal-Modi zur Verfügung:

- **RS-232 Terminal Mode** — ermöglicht die Kommunikation mit dem RS-232-Programmierport (PRT1) des BASIC-Moduls
- **DH-485 Terminal Mode** — ermöglicht die Kommunikation mit dem DH485-Port des BASIC-Moduls
- **General Purpose Communications Terminal Mode** - ermöglicht den Einsatz des Computers als Allzweck-Kommunikationsterminal. In diesem Modus kann der Personalcomputer als ASCII-Terminal verwendet werden, das den Telefon-/Modemwählbetrieb unterstützt. Dieser Modus sollte *nicht* für die Kommunikation mit dem BASIC-Modul verwendet werden.

Informationen über Verbindung und Kommunikation mit dem BASIC-Modul und anderen an einen Port angeschlossenen Geräten im Terminal-Modus sind im BASIC-Modul SLC 500, Design- und Integrationshandbuch (Publikationsnummer 1746-6.1DE) und im Benutzerhandbuch des BASIC-Moduls (Publikationsnummer 1771-6.5.113DE) enthalten.

Sicherung der Konfiguration und Beenden

Path: Toplevel Menu → Configuration menu → `save configuration and exit`

Path: Main Menu → Config → `save configuration and exit`

Einige Konfigurationsänderung werden sofort bei Zuweisung gesichert, andere hingegen erst, wenn Sie im Menü "Setup and Configuration" die Option `save configuration and exit` wählen. Sie werden dann aufgefordert, die Konfiguration zu sichern, wenn Änderungen vorgenommen, jedoch noch nicht gesichert wurden.

Wenn Sie das Menü verlassen, ohne die Konfigurationsänderungen zu sichern, bleiben sofort durchgeführte Änderungen unverändert. Nur die Änderungen, die vor dem Beenden des Programms nicht gesichert werden, bleiben unberücksichtigt.

Wichtig: Die BASIC-Entwicklungssoftware enthält den File `DEFAULTS.BAT` zur Wiederherstellung der Vorgabewerte. Mit diesem File werden *alle* Werte so wiederhergestellt, als ob die Software soeben auf dem Computer installiert worden wäre. Dieser File wird durch Eingabe von `DEFAULTS` (von dem Verzeichnis aus, in dem sich `DEFAULTS.BAT` befindet) ausgeführt, bevor die BASIC-Entwicklungssoftware aufgerufen wird.



ACHTUNG: Durch das Abarbeiten des Files `DEFAULTS.BAT` werden alle Änderungen gelöscht, die an den Parametern der BASIC-Entwicklungssoftware vorgenommen wurden, und durch die ursprünglichen Vorgabewerte ersetzt. `DEFAULTS.BAT` darf nicht von der DOS-Shell aus aufgerufen werden.

Editieren eines Programms

Dieses Kapitel enthält eine Einführung in die Funktionen der BASIC-Entwicklungssoftware zum Editieren der Files. Es enthält eine allgemeine Beschreibung der Funktionsmerkmale, die Sie beim Schreiben eines Programms verwenden können. Zu den wichtigsten Themen gehören:

- einfache Editiermethoden
- Suchfunktionen
- Blockfunktionen
- Textformatierung
- erweiterte Editierfunktionen

Anwendung einfacher Editiermethoden

Wenn im Toplevel-Menü die Option `edit mode` gewählt wird, erscheint das Hauptmenü, und ein Fenster wird geöffnet. Dieses Fenster (Fenster A) steht sofort zum Editieren eines Files zur Verfügung. Auch kann ein vorhandener File geöffnet und in dieses Fenster geladen werden. Über die Tastatur stehen Ihnen einige einfache Editiermethoden zur Verfügung.

Wichtig: Bei der Verwendung des numerischen Tastenfeldes des Personalcomputers muß `[NumLock]` AUSGESCHALTET sein.

Die folgenden Tasten werden zum Editieren von Texten verwendet:

Tabelle 4.A
Tastenfunktionen

Taste	Funktion
[Pfeil rechts]	Der Cursor wird nach rechts bewegt.
[Pfeil links]	Der Cursor wird nach links bewegt.
[Pfeil auf]	Der Cursor wird nach oben bewegt.
[Pfeil ab]	Der Cursor wird nach unten bewegt.
[Ctrl - Pfeil rechts]	Der Cursor wird um ein Wort nach rechts bewegt.
[Ctrl - Pfeil links]	Der Cursor wird um ein Wort nach links bewegt.
[Home]	Der Cursor wird an den Beginn einer Zeile bewegt.
[End]	Der Cursor wird an das Ende einer Zeile bewegt.
[Bild auf]	Die Bildschirmanzeige wird um eine vollständige Bildschirmseite nach oben verschoben.

Taste	Funktion
[Bild ab]	Die Bildschirmanzeige wird um eine vollständige Bildschirmseite nach unten verschoben.
[Ctrl - Bild auf]	Die Bildschirmanzeige wird an den Seitenbeginn verschoben.
[Ctrl - Bild ab]	Die Bildschirmanzeige wird an den Beginn der nächsten Seite verschoben.
[Ctrl - Home]	Die Bildschirmanzeige wird an den Beginn des Files verschoben.
[Ctrl - End]	Die Bildschirmanzeige wird an das Ende des Files verschoben.
[Löschen]	Löscht das Zeichen, auf dem sich der Cursor befindet.
[Backspace]	Löscht das Zeichen links vom Cursor.
[Einfügen]	Umschaltung zwischen dem Einfügemodus (der neue Text wird links vom Cursor eingefügt) und dem Überschreibmodus (der neue Text überschreibt das Zeichen, auf dem sich der Cursor befindet).

Der Cursor kann auch durch die Eingabe einer Zeilennummer innerhalb des Programms bewegt werden. Wählen Sie hierzu `goto Line number`, wenn Sie das Cursor-Menü herunterziehen.

Path: Toplevel Menu → Edit mode → cUrsor → goto Line number

QuickStroke: [Alt F8]

Es stehen eine Reihe von anderen QuickStroke-Funktionen und einfachen Editiermethoden zur Verfügung. Diese sind in der Hypertext-Hilfe enthalten, die vom Hilfeindex bzw. vom Inhaltsverzeichnis aus durch Anwahl von Quick Reference (Schnellinformationen) aufgerufen werden können.

Suchfunktionen

Im Search-Menü stehen mehrere Methoden zur Suche von verschiedenen Ausdrücken zur Verfügung. Die hier vorliegende Beschreibung beschränkt sich auf eine allgemeine Erläuterung der Optionen dieses Menüs. Nähere Einzelheiten zur Implementierung der Menüoptionen sind in den Hilfebildschirmen aufgeführt. Hinweise zur Verwendung der Hilfebildschirme sind in Kapitel 2 enthalten.

Path: Toplevel Menu → Edit mode → Search

QuickStroke: available for individual items within the Search menu


```

L:3 C:1 1747-PBASE V x.xx [Text Edit] Ins 175k 06-12-91 1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----SEARCH-----+
+=+A+=====|Search for text... <F6> |==?No-File?==+
|>> EOF << |search and Replace... <ShftF6>| | | | | | | | |
| |repeat Previous search <CtrlF6>| |
| |-----| |
| |multiple File search... <AltF> | |
| |List files from last search...<AltG> | |
| |-----Cancel<ESC>-----| |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
+-----+
1 Help 2 3 4 5 6 7 8 9 0

```

Das Search-Menü ist in zwei Abschnitte unterteilt. Im ersten Abschnitt sind Suchoptionen enthalten, die sich auf den aktuellen, im Editierfenster geladenen File beziehen. Der zweite Abschnitt enthält Optionen zur Suche in mehreren Files des spezifizierten Pfades, unabhängig davon, ob der File in ein Editierfenster geladen wurde oder nicht.

Es folgt eine Auflistung der Optionen des Search-Menüs sowie eine kurze Erläuterung der jeweiligen Funktion. Weitere Einzelheiten sind in der Hypertext-Hilfe enthalten.

- **search for text** — ermöglicht die Suche innerhalb eines Fensters in Vorwärts- und Rückwärtsrichtung. Für die Suche nach verschiedenen Ausdrücken, Buchstaben oder Wortfolgen stehen eine Reihe verschiedener Suchmodi zur Verfügung.
- **search and Replace** — ermöglicht das Ersetzen von gesuchten Elementen mit einem gewählten Element. Diese Funktion entspricht der Funktion **search for text**, wobei jedoch Zeichen während der Suche ersetzt werden können.
- **repeat Previous search** — wiederholt die letzte Funktion **search for text** bzw. **search and Replace**

- **multiple File search** — durchsucht eine oder alle Files in einem Laufwerk bzw. auf einer Festplatte nach einer spezifizierten Zeichenkette, wobei eine Reihe unterschiedlicher Suchoptionen angewandt werden. Files, welche die gesuchte Zeichenkette enthalten, werden in einem Auflistungsfeld angezeigt. Von diesem Feld aus können Files gewählt und in ein Fenster geladen werden.
- **List files from last search** — listet die Files auf, die während der letzten Suche in mehreren Files gefunden wurden.

Anwendung von Blockfunktionen

Das Block-Menü stellt ein nützliches Hilfsmittel beim Schreiben von Programmen dar. Es enthält zahlreiche Funktionen, die beim Editieren von BASIC-Programmen mit einer Textverarbeitungssoftware zur Erhöhung der Geschwindigkeit und Genauigkeit eingesetzt werden können.

Path: Toplevel Menu → Edit mode → Block

QuickStroke: available for individual items within the Block menu

```
L:3 C:1 1747-PBASE V x.xx [Text Edit] Ins 175k 06-12-91 1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----BLOCK-----+
++A+====|(Copy block <F9>)|=====No-File?++
|>> EOF << |(Move block <F10>)|
| |(Delete block <CtrlF10>)|
| |-----|
| |cut and Paste|
| |(save Block to disk...<CtrlF5>)|
| |(Indent line block <AltI>)|
| |(Undent line block <AltU>)|
| |-----|
| |Window copy... <ShftF9>|
| |window moVe... <ShftF10>|
| |-----|
| |mark Lines of text <F7>|
| |mark cOlumns of text <ShftF7>|
| |mark Stream of text <AltF7>|
| |-----Cancel<ESC>-----|
+-----+
1 Help 2 3 4 5 6 7 8 9 0
```

Bevor ein Block bearbeitet (kopiert, verschoben, gelöscht usw.) werden kann, muß er zunächst definiert werden. Deshalb müssen Sie zu Beginn eine der zuletzt aufgelisteten drei Markierungsoptionen wählen.

Wichtig: Wenn es nicht möglich ist, Zeichen in einem File zu ändern, wurde dieser File wahrscheinlich gesperrt. Files können über das File-Menü gesperrt (d.h. als Nur-Lese-Files konfiguriert) oder zugänglich gemacht werden. Verschieben Sie den Cursor auf die Option **Information About Current File**, und drücken Sie die **[ENTER]**-Taste bzw. die linke Maustaste. Unter der Beschreibung des aktuellen Files sind die Sperr- und Entsperr-Optionen aufgeführt. Drücken Sie die **[Tab]**-Taste und/oder die Pfeiltasten, um eine dieser Optionen zu wählen, und drücken Sie anschließend die **[ENTER]**-Taste bzw. die linke Maustaste.

Es folgt eine Auflistung der Optionen, die im Block-Menü verfügbar sind, sowie eine Erläuterung ihrer jeweiligen Funktion. Weitere Hinweise sind in der Hypertext-Hilfe enthalten.

- **mark lines of text [F7]** — ermöglicht die Markierung bestimmter Zeilen eines Files ab (oder bis zu) der Zeile, auf der sich der Cursor vor der Wahl dieses Menüelementes befindet. Klicken Sie die linke Maustaste, um diese Cursorposition im File zu definieren.
- **mark columns of text [shft-F7]** — ermöglicht die Markierung bestimmter Spalten eines Files ab (oder bis zu) der Spalte, auf der sich der Cursor vor der Wahl dieses Menüelementes befindet. Klicken Sie die linke Maustaste, um diese Cursorposition im File zu definieren. Mit dieser Option können Sie den Cursor in die gewünschte Richtung bewegen, um die gewählte Spalten- und Reihbreite zu bestimmen.
- **mark stream of text [Alt-F7]** — ermöglicht die Markierung eines bestimmten Textabschnitts eines Files ab (oder bis zu) dem Zeichen, auf dem sich der Cursor vor der Wahl dieses Menüelementes befindet. Klicken Sie die linke Maustaste, um diese Cursorposition im File zu definieren. Diese Option ist der Zeilenmarkierung ähnlich, wobei hier jedoch Teile von Zeilen markiert werden können.
- **End block [F7]** — Diese Option wird nur während der Textmarkierung angezeigt und sollte gewählt werden, nachdem der Block wie gewünscht markiert wurde.
- **turn marking off [Ctrl-F9]** — Diese Option wird nur nach Beendigung der Textmarkierung angezeigt. Mit dieser Funktion können alle Blockmarkierungen im File des aktiven Fensters entfernt werden.

**MARK
LINES
OF TEXT
<F7>**

**MARK
COLUMNS
OF TEXT
<ShftF7>**

**MARK
STREAM
OF TEXT
<AltF7>**

Nachdem Sie einen Block definiert haben, stehen eine Reihe von verschiedenen Funktionen zur Verfügung, die am Block ausgeführt werden können. Diese Methoden der Blockmanipulation sind im folgenden Abschnitt erläutert:

- **Copy block [F9]** — Kopieren (Duplizieren) eines Blocks an die gegenwärtige Cursorposition innerhalb eines Fensters
- **Move block [F10]** — Verschieben eines Blocks an die gegenwärtige Cursorposition innerhalb eines Fensters
- **Delete block [Ctrl-F10]** — Löschen des Blocks innerhalb des derzeit aktivierten Fensters
- **cut and Paste** — Verschieben, Kopieren oder Anhängen eines Blocks in einen versteckten Pufferspeicher zum Einfügen an einer anderen Stelle in einem aktivierten Fenster. Es kann jeweils nur ein Block im Pufferspeicher abgelegt werden.
- **save Block to disk [Ctrl-F5]** — Sichern eines Blocks auf ein Laufwerk als separater File
- **Indent line block [Alt-I]** — Verschieben eines Blocks um eine Tabulatorposition nach rechts. Diese Funktion kann nur bei Blöcken verwendet werden, die durch die Markierung von Textzeilen erstellt wurden.
- **Undent line block [Alt-U]** — Verschieben eines Blocks um eine Tabulatorposition nach links. Diese Funktion kann nur bei Blöcken verwendet werden, die durch die Markierung von Textzeilen erstellt wurden.
- **window copy [shft-F9]** — Kopieren eines Blocks in das derzeit aktivierte Fenster
- **window move [shft-F10]** — Verschieben eines Blocks in das derzeit aktivierte Fenster

Textformatierung, Rückgängigmachen und Aufheben

Im Text-Menü ist die Eingabe und Formatierung von Text unter Verwendung verschiedener zeitsparender Funktionen möglich. Hierzu gehören auch die Funktionen Rückgängigmachen (Undo) und Wiederherstellen (Redo), die es Ihnen ermöglichen, unerwünschte Eingaben schrittweise rückgängig zu machen.

Path: Toplevel Menu → Edit mode → Text

QuickStroke: available for individual items within the Text menu

```

L:3  C:1  1747-PBASE V x.xx  [Text Edit]  Ins 175k  06-12-91  1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----TEXT-----+
+==A+=====|Undo          <CtrlF7> |===?No-File?==+
|>> EOF <<  |Redo          <CtrlDEL>| |
|            |-----| |
|            |Sort text...  | |
|            |upper Case  | |
|            |Lower case  | |
|            |cenTer line| |
|            |time/Date stamp <ShftF2> | |
|            |-----| |
|            |re-Format paragraph <AltR> | |
|            |-----| |
|            |rePeat command n times<CtrlR> | |
|            |-----| |
|            |-----Cancel<ESC>-----| |
+-----+
1 Help 2 3 4 5 6 7 8 9 0
    
```

Wichtig: Wenn es nicht möglich ist, Zeichen in einem File zu ändern, wurde dieser File wahrscheinlich gesperrt. Files können über das File-Menü gesperrt (d.h. als Nur-Lese-Files konfiguriert) oder zugänglich gemacht werden. Verschieben Sie den Cursor auf die Option **Information About Current File**, und drücken Sie die [ENTER]-Taste bzw. die linke Maustaste. Unter der Beschreibung des aktuellen Files sind die Sperr- und Entsperr-Optionen aufgeführt. Drücken Sie die [tab]-Taste und/oder Pfeiltasten, um eine dieser Optionen zu wählen, und drücken Sie anschließend die [ENTER]-Taste bzw. die linke Maustaste.

Im folgenden werde die Optionen des Text-Menüs aufgeführt und ihre jeweiligen Funktion kurz erläutert. Nähere Hinweise sind in der Hypertext-Hilfe zu finden.

- **Undo [Ctrl-F7]** — Rückgängigmachung der letzten Editiereingaben, ausgenommen Sicherung und Laden von Files. Die Anzahl der Rückgängigmachungsschritte kann im Menü “Setup and Configuration” auf einen Wert zwischen 0 und 65535 eingestellt werden.
- **Redo [Ctrl-De1]** — Wiederherstellung der rückgängig gemachten Editiereingaben
- **sort text** — Sortieren der Zeilen eines Files in steigender oder fallender, alphabetischer oder numerischer Reihenfolge. Sie können bestimmen, welche Zeichenspalte sortiert werden soll. Außerdem kann die Sortierfunktion auf einen bestimmten, als Block definierten Bereich beschränkt werden.

- **upper case** — Umwandlung eines Wortes, einer Zeile oder eines Blocks in Großbuchstaben
- **lower case** — Umwandlung eines Wortes, einer Zeile oder eines Blocks in Kleinbuchstaben
- **center line** — Zentrieren der derzeit gewählten Zeile zwischen Spalte 1 und dem rechten Rand
- **time/date stamp [shft-F2]** — Einfügen des aktuellen Datums und der Zeit an der Cursorposition
- **re-format paragraph [Alt-R]** — Einfügen des Textes in einen Absatz, der sich vom linken bis zum rechten Rand erstreckt und in dem die Zeilen nach Bedarf umgebrochen werden. Diese Funktion wird generell für Textabsätze verwendet.
- **repeat command n times [Ctrl-R]** — Eingeben einer Wiederholungsrate, gefolgt von dem zu wiederholenden Zeichen. Wenn Sie beispielsweise nach Anwahl der Option **repeat command n times [Ctrl-R]** den Wert **74x** eingeben, wird der Buchstabe **x** an der aktuellen Cursorposition 74 mal eingefügt. Wenn Sie den Wert **20** eingeben und anschließend die Löschtaste drücken, werden die nächsten zwanzig Zeichen gelöscht.

Erweiterte Editierfunktionen

In diesem Abschnitt werden die erweiterten Editierfunktionen beschrieben, die für die BASIC-Entwicklungssoftware zur Verfügung stehen. Die behandelten Themen umfassen:

- Linien zeichnen
- Verwendung der Taschenrechnerfunktion
- ASCII-Tabelle
- Verwendung von Fenstern
- Tastenmakros
- Verwendung von Cursormarkierungen

Die ersten drei hier beschriebenen erweiterten Editierfunktionen können über das Other-Menü aufgerufen werden.

Path: Toplevel Menu → Edit mode → Other
QuickStroke: available for individual items within the Other menu

```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91   1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----OTHER-----+
+=+A+=====|Select a project file... <AltENTER>|=+
|>> EOF << |User defined program... | |
| |-----| |
| |Translate/Compile <CtrlF8> | |
| |find neXt error <CtrlF2> | |
| |-----| |
| |Calculator <AltF2> | |
| |Ascii table <AltA> | |
| |shell to DOS <AltF5> | |
| |Linedraw <AltL> | |
| |-----| |
| |teRminal [RS-232] <F3> | |
| |-----Cancel<ESC>-----+
+-----+
1 Help 2 3 4 5 6 7 8 9 0
    
```

Linienzeichen

Path: Toplevel Menu → Edit mode → Other → Linedraw
 QuickStroke: Toplevel Menu → Edit mode → [Alt-L]

Die Funktion “Linien zeichnen” ist über das Other-Menü aufrufbar und dient zur Erstellung von Feldern, Balkendiagrammen und Tabellen. Diese Linien werden (in einer Druckanweisung) zur Verwendung in einem BASIC-Programm übersetzt, das vom BASIC-Modul ausgeführt wird.

Mit dieser Funktion können Sie in einem Fenster einfache und doppelte Linien sowie Felder zeichnen. Bei Verwendung des 1747-PBASE Translators werden die Linien mit dem File gesichert und in den BASIC-Code übersetzt. Beachten Sie bitte, daß für einige Drucker besondere Linienzeichen definiert werden müssen, um diese Linien drucken zu können.

Nähere Einzelheiten zu dieser Funktion sind in der Hypertext-Hilfe enthalten.

Ein Beispielpogramm einer Linienzeichnung, die mit Programmiermakros erstellt wurde, ist in der BASIC-Entwicklungssoftware enthalten. Dieses Programm heißt **LINEDRAW.BDL** und befindet sich im BAS-Unterverzeichnis. Es kann in ein Fenster geladen und als Beispiel nachvollzogen werden.

Verwendung der Taschenrechnerfunktion

Path: Toplevel Menu → Edit mode → Other → Calculator
QuickStroke: Toplevel Menu → Edit mode → [Alt-F2]

Die Taschenrechnerfunktion (Calculator) kann vom Other-Menü aus aktiviert werden und dient zur schnellen Durchführung mathematischer Operationen direkt auf dem Bildschirm, wobei das Ergebnis in den aktivierten File eingefügt werden kann.

Die Taschenrechnerfunktion unterstützt das Dezimal-, Hexadezimal-, Oktal- und Binärformat und kann Additionen, Subtraktionen, Multiplikationen und Divisionen durchführen. Ferner können logische AND-, OR- und exklusive OR-Operationen ausgeführt werden. Kontrollstreifen ermöglicht das Einsehen der Einträge, und eine Einfügefunktion ermöglicht das Einfügen des errechneten Ergebnisses an der letzten Cursorposition im aktivierten Fenster.

Weitere Hinweise zu dieser Funktion sind in der Hypertext-Hilfe enthalten.

ASCII-Tabelle

Path: Toplevel Menu → Edit mode → Other → ASCII table
QuickStroke: Toplevel Menu → Edit mode → [Alt-A]

Die ASCII-Tabelle kann nur eingesehen werden und umfaßt drei Seiten mit Zeichen-, Dezimal- und Hexadezimalspalten.

Verwendung von Fenstern

Die Fenster sind in ihrer linken oberen Ecke alphabetisch gekennzeichnet. Die BASIC-Entwicklungssoftware kann bis zu 100 Fenster gleichzeitig öffnen. Dabei kann jedes Fenster einen anderen File enthalten oder die Fenster können miteinander verknüpft sein und jeweils einen Teil desselben Files enthalten. Die Fenster können unabhängig voneinander geöffnet und geschlossen werden.

Eine ausführlichere Erläuterung der Fensterfunktionen wäre hier nicht angebracht. Zum besseren Verständnis ihrer Anwendung ist es am sinnvollsten, sie beim Editieren von Files zu verwenden. Nützliche Hinweise sind in der Hypertext-Hilfe enthalten.

Path: Toplevel Menu → Edit mode → Window
QuickStroke: available for individual items within the Window menu


```

L:3      C:1      1747-PBASE V x.xx      [Text Edit]  Ins 175k  06-12-91  1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----WINDOW-----+
+==C+|Open          |=====C:\ABBASIC\BAS\FIRST.BDL==+
| |Close          |
| |Split          |
| |List...    <AltF1> |
| |-----|
| |Next      <ShftF1> |
| |Previous  <CtrlF1> |
| |Adjacent  <AltW>   |
| |-----|
| |Hide          |
| |Modify size <ScrollLockOn>|
| |Zoom        <CtrlF3> |
| |mInimize    |
| |-----|
| |linK...     |
| |(Unlink)    |
| |-----Cancel<ESC>-----+
|
|
|
+-----+
1 Help  2      3      4      5      6      7      8      9      0
    
```

Nachfolgend werden die Optionen des Window-Menüs aufgeführt und ihre Funktionen beschrieben:

- **open** — Ein neues Fenster wird geöffnet. Gleichzeitig kann ein neuer File geladen werden.
- **close** — Das derzeit aktivierte Fenster wird geschlossen. Wurde der File in diesem Fenster editiert, aber nicht gesichert, werden Sie nun aufgefordert, ihn zu sichern oder unverändert zu belassen.
- **split** — Das aktivierte Fenster wird horizontal oder vertikal geteilt. Hierbei kann ein neuer File geladen werden. Wird kein neuer File geladen, wird der bestehende File in das zweite Fenster kopiert und mit dem Original verknüpft.
- **List [Alt-F1]** — Auflistung der offenen Fenster in alphabetischer Reihenfolge unter Angabe des Filenamens, Status und Pfads. Fenster können gewählt, gelöscht, verdeckt, gesichert und im Vorschaumodus anhand dieser Liste eingesehen werden.
- **Next [Shft-F1]** — Das nächste Fenster in alphabetischer Reihenfolge wird gewählt (sofern es nicht verdeckt ist).
- **Previous [Ctrl-F1]** — Das vorherige Fenster in alphabetischer Reihenfolge wird gewählt (sofern es nicht verdeckt ist).
- **Adjacent [Alt-w]** — Mit dieser Option werden Sie aufgefordert, eine Richtung einzugeben, um ein Nachbarfenster zu aktivieren.

Tastenmakros sollen die Arbeit erleichtern und beschleunigen, indem Eingabewiederholungen reduziert werden. Beispiel: Bei der Erstellung eines Programmfiles werden dieselben Grundinformationen in den Kopf eingegeben. Eine zeitsparende Alternative wäre, einen separaten File mit diesen Grundinformationen zu erstellen und den File in jeden neuen Programmfile einzufügen. Hierzu würden Sie einen File mit den Kopfinformationen erstellen, auf der Festplatte unter dem Namen `header` speichern und dann aus dem File-Menü die Option `Merge file from disk option` wählen.

Die Arbeit wird hierdurch zwar beschleunigt, jedoch ist es immer noch erforderlich, den File aufzurufen und in das Programm einzufügen. Die Programmierung eines Tastenmakros hingegen ermöglicht es Ihnen, den gesamten Vorgang mit einem Tastenanschlag durchzuführen.

Wichtig: Verwenden Sie bei der Aufzeichnung von Tastenfolgen zur Erstellung eines Tastenmakros nicht die Maus, um Menüelemente zu wählen oder den Cursor zu positionieren. Verwenden Sie vielmehr die Tasten selbst (einschließlich der Funktionstasten und der für die Wahl von Menüelementen definierten Buchstabentasten).

Wichtig: Bei der Aufzeichnung von Tastenfolgen sollten die Cursortasten möglichst nicht verwendet werden, um Menüelemente oder Filenamen zu wählen. Beinhaltet die Aufzeichnung z.B., daß mit dem Cursor der sechste File einer Auflistung gewählt werden soll, so könnte ein Fehler eintreten, wenn diese Auflistung vergrößert oder verkleinert wird und dadurch der falsche File gewählt wird.

Im folgenden sind die einzelnen Schritte zur Erstellung eines Tastenmakros erläutert, der die Aufgaben im o.g. Beispiel zusammenfassen soll. Weitere Hinweise sind in der Hypertext-Hilfe enthalten.

1. Drücken Sie [`A1t-F10`], um mit der Tastenaufzeichnung zu beginnen.

Beachten Sie bitte das invertierte R (Recording), das in der Statuszeile am oberen Bildschirmrand blinkt. Es signalisiert, daß alle Tastenanschläge aufgezeichnet werden, bis [`A1t-F10`] erneut gedrückt wird.

2. Drücken Sie die Tasten, die zum Einfügen eines Files verwendet würden:
 - [`F2`] Aufruf des Hauptmenüs
 - [`F`] Wahl des File-Menüs
 - [`M`] Wahl der Option `Merge file from disk`
 - [`geben sie den Namen des zu ladenden Files ein`]
 - [`ENTER`]

3. Drücken Sie [**Alt-F10**] erneut, um die Aufzeichnung von Tastenfolgen zu beenden. Nach Beendigung der Aufzeichnung kann der Tastenmakro gesichert werden. Wählen Sie hierzu die Option **yes**. Das Dialogfeld **EDITING KEYSTROKE MACRO** wird angezeigt. Sie werden aufgefordert, eine Beschreibung und eine QuickStroke-Zuordnung einzugeben und einen Modus für den Tastenmakro zu spezifizieren.
4. Geben Sie für diesen Tastenmakro die Bezeichnung **Firmenkopf** ein. Drücken Sie noch *nicht* die [**ENTER**]-Taste, da der Tastenmakro sonst ohne QuickStroke-Zuordnung gesichert wird. Wurde die [**ENTER**]-Taste dennoch gedrückt, kann mit der Funktionstaste [**F3**] das Dialogfeld **EDITING KEYSTROKE MACRO** erneut aufgerufen werden.
5. Bewegen Sie den Mauscursor auf das QuickStroke-Zuordnungsfeld, und klicken Sie die linke Maustaste. Wenn keine Maus vorhanden ist, drücken Sie die [**Tab**]-Taste, um den Cursor auf dieses Feld zu setzen, und drücken Sie anschließend die [**Leertaste**]. Das Dialogfeld **EDIT QUICKSTROKE** wird angezeigt.
6. Drücken Sie [**Alt-B**], die QuickStroke-Tastenkombination, die diesem Tastenmakro zugewiesen werden soll. Gültige QuickStroke-Zuordnungen sind im nächsten Abschnitt "Editieren von Tastenmakros" beschrieben.
7. Drücken Sie die [**Tab**]-Taste, um den Cursor auf das Mode-Feld zu verschieben. Dieses Feld wird ebenfalls im Abschnitt "Editieren von Tastenmakros" beschrieben.
8. Drücken Sie die [**ENTER**]-Taste, um das Mode-Feld unverändert zu belassen und zum Dialogfeld **MACRO MANAGER** zurückzukehren. Die Auflistung der Tastenmakros enthält nun den Namen "Firmenkopf" und rechts davon [**Alt-B**], die QuickStroke-Zuordnung, die diesen Tastenmakro aktiviert.

Mit der Tastenkombination [**Alt-B**] wird nun ab sofort an der aktuellen Cursorposition der Firmenkopf eingefügt.

Ändern von aufgezeichneten Tastenfolgen

Auf Wunsch kann die aufgezeichnete Tastenfolge vor dem Verlassen des Dialogfeldes **MACRO MANAGER** geändert werden. Beispielsweise werden Sie feststellen, daß der Firmenkopf nach dem Einfügen in einen File mit der QuickStroke-Tastenkombination [**Alt-B**] hervorgehoben bleibt. Da diese Blockmarkierung vor dem Editieren eines weiteren Blocks deaktiviert werden muß, erkennen Sie nun, daß Sie sie bereits während der ursprünglichen Aufzeichnung des Tastenmakros deaktivieren hätten sollen.

1. Heben Sie im Dialogfeld **MACRO MANAGER** den Firmenkopf-Tastenmakro hervor, und drücken Sie [F3], um ihn zu editieren.
2. Wählen Sie im Dialogfeld **EDITING KEYSTROKE MACRO** mit der Maus die Option “Edit keystrokes”, und klicken Sie die linke Maustaste. Wenn keine Maus vorhanden ist, drücken Sie die [Tab]-Taste, um diese Option mit dem Cursor hervorzuheben. Anschließend drücken Sie die [Leertaste] oder [Pfeil ab].
3. Verschieben Sie den Cursor in der Auflistung der Tastenkombinationen auf die Zeile *direkt vor* der Stelle, an der die Einfügung vorgenommen werden soll.
4. Drücken Sie:
 - [INS-F2], um die Taste für den Aufruf der Pulldown-Menüs einzufügen,
 - [INS-B], um die Taste für den Aufruf des Blockmenüs einzufügen,
 - [INS-F], um die Taste zur Deaktivierung der Blockmarkierung einzufügen.
5. Drücken Sie die [ENTER]-Taste, um das Editieren der Tastenfolge zu beenden.
6. Drücken Sie die [ENTER]-Taste, um das Editieren dieses Tastenmakros zu beenden.
7. Drücken Sie die [ENTER]-Taste, um das Dialogfeld **MACRO MANAGER** zu verlassen.

Der gesamte Vorgang des Einfügens von Kopfinformationen in einen neuen File ist nun auf eine Tastenkombination, [Alt-B], reduziert worden und ist auf dem aktivierten Laufwerk gespeichert.

Editieren von Tastenmakros

Im letzten Abschnitt wurde Schritt für Schritt erläutert, wie ein Tastenmakro aufgezeichnet, gesichert, umbenannt und editiert wird. In diesem Abschnitt soll ein Überblick über alle Optionen gegeben werden, die im Dialogfeld **MACRO MANAGER** zum Erstellen, Löschen, Kopieren, Editieren oder Wiedergeben eines Tastenmakros zur Verfügung stehen.

- **Create [INS]** — Erstellung eines neuen Tastenmakros. Mit dieser Option wird das Dialogfeld **EDITING KEYSTROKE MACRO** angezeigt. Dieses ist in vier Abschnitte unterteilt:
 - **Description** — Geben Sie mit maximal 25 Zeichen eine Beschreibung der Funktion dieses Tastenmakros ein.

- **QuickStroke assignment** — Drücken Sie die [Tab]-Taste, um dieses Feld zu aktivieren, und anschließend die [Leertaste], um es zu editieren. Geben Sie die Tastenkombination ein, durch die der Tastenmakro ausgelöst werden soll.

Sie können jede Tastenkombination verwenden, die mit der [Alt]-Taste beginnt, sofern die Kombination nicht bereits belegt ist. Wenn die gewählte Tastenkombination bereits belegt ist, werden Sie durch eine Meldung aufgefordert, eine andere Kombination einzugeben. Wenn in der Hypertext-Hilfe die Option "Quick Reference" gewählt wird, erscheint eine Auflistung der QuickStroke-Tastenzuordnungen.

Zusätzlich zu den [Alt]-Tastenkombinationen stehen, je nach der Tastatur des Personalcomputers, möglicherweise weitere Sonderfunktionstasten zur Verfügung. Beispielsweise können die Tasten [F11], [F12] und [Zentrieren] (Taste [5] des numerischen Tastenfeldes, wenn [NumLock] deaktiviert ist) bei AT- und PS/2-Tastaturen als QuickStroke-Tasten zugewiesen werden.

- **Mode** – Verschieben Sie den Cursor mit der [Tab]-Taste auf dieses Feld, und drücken Sie anschließend die [Leertaste] oder die [Pfeil-Ab]-Taste, um die verfügbaren Optionen anzuzeigen. Mit der Modus-Wahl wird festgelegt, ob das Tastenmakro vom Edit- oder vom Terminal-Modus aus aufgerufen werden kann.

In einigen Fällen ist es sinnvoll, Tastenmakros zur Verwendung im Terminal-Modus zuzuordnen. Sie könnten geschrieben werden, um Befehle wie RUN, CONT, NEW, LIST usw. an das BASIC-Modul zu senden.

- **Edit Keystroke** – Drücken Sie die [Tab]-Taste, um dieses Feld zu aktivieren, und anschließend die [Leertaste] oder [Pfeil-Ab]-Taste, um eine Auflistung der Tasten anzuzeigen, die für diesen Tastenmakro programmiert wurden. Der Cursor kann in dieser Auflistung nach oben und unten bewegt werden, und Tasten können nach Bedarf eingefügt [INS], gelöscht [DEL] und geändert [F3] werden.
- **Delete [DEL]** — Löschen eines Tastenmakros, der zuvor gesichert wurde
- **Copy [F4]** — Kopieren eines Tastenmakros. Sie werden aufgefordert, einen beschreibenden Namen für die Kopie einzugeben und die QuickStroke-Taste zu ersetzen oder neu zuzuweisen.
- **Edit [F3]** — Editieren eines bestehenden Tastenmakros. Mit dieser Option wird das Dialogfeld **EDITING KEYSTROKE MACRO**, das oben unter **Create [INS]** beschrieben wurde, aktiviert.
- **Playback [ENTER]** — Sofortige Ausführung des hervorgehobenen Tastenmakros und Rückkehr in den Edit-Modus. Mit dieser Option kann ein aufgelisteter Tastenmakro, auch wenn ihm keine QuickStroke-Taste zugeordnet wurde, ausgeführt werden.

- **Done [ESC]** — Verlassen des Dialogfeldes **MACRO MANAGER** und Rückkehr zum Edit-Modus

Verwendung von Cursormarkierungen

Im Cursor-Menü stehen während des Programmierens mehrere Optionen zum Markieren und Abruf der Cursorposition zur Verfügung. Mit diesen Funktionen kann die Geschwindigkeit und Genauigkeit der Programmierung erhöht werden.

Beispiel: Wenn beim Editieren eines sehr großen Files häufig auf verschiedene Abschnitte zugegriffen werden muß, kann eine Cursorpositionsmarkierung an einer geeigneten Stelle plaziert und der Cursor dann sofort mit nur einem Tastenanschlag jederzeit an dieser Stelle positioniert werden. Diese Markierungen können bis zu einer Stapeltiefe von 10 eingespeichert werden; somit können dann bis zu zehn Positionen in umgekehrter Reihenfolge abgerufen werden. In einem File können auch mehrere Zufallsmarkierungen gesetzt werden, um entsprechend der Markierungsnummer auf die Position zuzugreifen.

Path: Toplevel Menu → Edit mode → cUrsor

QuickStroke: available for individual items within the Cursor menu

```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91 1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit

+-----CURSOR POSITION-----+
++A+====|Push position onto marker stack<AltF3> |=====No-File?++
|>> EOF << |Get position from marker stack <ShftF3> |
|-----|
|Set random access mark |
|Retrieve random access mark |
|-----|
|goto Line number... <AltF8> |
+-----Cancel<ESC>-----+

1 Help 2 3 4 5 6 7 8 9 0
    
```

Es folgt eine Auflistung der Optionen des Cursor-Menüs, sowie eine Erläuterung ihrer Funktionen. Weitere Details sind in der Hypertext-Hilfe enthalten.

- **Push position onto marker stack [Alt-F3]** — Speichern von maximal zehn Cursorpositionen in einem Markierungsstapel. Sie werden in der Reihenfolge gespeichert, in der sie gesichert werden, wenn diese Option gewählt wird.
- **Get position from marker stack [shft-F3]** — Eine gespeicherte Cursorposition wird dem Markierungsstapel entnommen, und der Cursor wird auf diese Position verschoben. Die Positionen werden in *umgekehrter* Reihenfolge zur Eingabe abgerufen. Die erste gespeicherte Cursorposition wird somit als letzte Position abgerufen.
- **Set random access mark** — Speichern von bis zu zehn Cursorpositionen für den wahlfreien Abruf in jedem Fenster. Hierbei handelt es sich *nicht* um dieselben Cursorpositionen wie beim Markierungsstapel. Sie werden mit einer Markierungsnummer zwischen 0 und 9 gespeichert und bei Wahl dieser Menüoption abgerufen.
- **Retrieve random access mark** — Abruf der gewählten, gespeicherten Cursorposition und Platzieren des Cursors an dieser Position. Sie werden bei dieser Menüoption aufgefordert, die gewünschte Markierungsnummer (zwischen 0 und 9) einzugeben.
- **goto Line number [Alt-F8]** — Aufforderung zur Eingabe einer Zeilennummer und anschließende Positionierung des Cursors in dieser Zeile des Files. Die eingegebene Nummer ist *nicht* eine Zeilennummer des BASIC-Programms, sondern vielmehr die absolute Zeilennummer ab Filebeginn.

Bearbeiten von Files

Dieses Kapitel soll Sie mit den Funktionen der BASIC-Entwicklungssoftware vertraut machen, die zum Bearbeiten von Files verwendet werden. Zu den wichtigsten Themen gehören:

- das File-Menü
- Wahl von Projektfiles
- anwenderdefinierte Programme

Das File-Menü

Das erste Pulldown-Menü zur linken Seite des Hauptmenüs ist das File-Menü, über welches der größte Teil der Filebearbeitung erfolgt.

Path: Toplevel Menu → Edit mode → File

QuickStroke: available for individual items within the File menu

```

L:3   C:1   1747-PRASE V x.xx   [Text Edit]   Ins 175k   06-12-91  1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----FILE-----+
|Edit new file...          |=====No-File?+
|Load file into current window... <ShftF5>|
|Save file in current window          |
|save file As...          <F5>      |
|Information about current file...    |
|-----|
|Merge file from disk...          |
| (save Block to disk...          )  |
|-----|
|DOS directory shell          <CtrlF4>|
|Quit                          <AltX> |
+-----Cancel<ESC>-----+
|
|
|
|
|
|
|
|
|
|
|
+-----+
1 Help  2      3      4      5      6      7      8      9      0

```

Es folgt eine Auflistung der Optionen des File-Menüs und eine kurze Beschreibung ihrer Funktionen:

- **Edit new file** — Erstellen eines neuen Fensters beim Laden eines neuen bzw. bereits vorhandenen Files. Im angezeigten Dialogfeld wird der Filename eingegeben oder aus einer Verzeichnisliste ausgewählt. In der Verzeichnisliste können die in einem beliebigen Verzeichnis eines beliebigen Systemlaufwerks enthaltenen Files angezeigt werden. Es stehen auch Optionen zum Sortieren der Verzeichnisliste zur Verfügung, die es ermöglichen, die Files nach Erweiterung, Größe, Erstellungsdatum und -uhrzeit oder Namen zu sortieren.
- **Load file into current window [Shift-F5]** — Laden eines Files in das aktivierte Fenster, wobei ein bereits im Fenster vorhandener File ausgeblendet wird. Wurde der zuvor aktuelle File geändert, jedoch nicht gesichert, werden Sie gefragt, ob die Änderungen gesichert werden sollen. Anschließend wird ein Dialogfeld angezeigt (z.B. **Edit new file**), in dem der zu ladende neue File gewählt werden kann.
- **Save file in current window** — Sichern des im aktivierten Fenster enthaltenen Files. Der File wird unter seinem ursprünglichen Namen gesichert, wobei zuvor gesicherte Versionen überschrieben werden.
- **save file As [F5]** — Mit dieser Option wird der im aktivierten Fenster enthaltene File unter einem anderen Namen gesichert. Es wird ein Dialogfeld eingeblendet, in dem Sie einen neuen Namen mit Erweiterung eingeben oder einen bereits vorhandenen Namen aus der Verzeichnisliste wählen können. Wenn Sie den Namen eines bereits vorhandenen Files wählen, werden Sie vor Ausführung der Sicherungsaufzeichnung gefragt, ob Sie den vorhandenen File überschreiben wollen.
- **Information about current file** — Angabe des aktuellen Filenamens, des Pfades und der Größe des belegten Speicherplatzes. Es wird ein Dialogfeld eingeblendet, in dem Sie den Filenamens ändern und den File sperren können, so daß er nicht geändert werden kann.
- **Merge file from disk** — Einfügen eines anderen Files in den bearbeiteten File an der aktuellen Cursorposition. Es wird ein Dialogfeld angezeigt (z.B. **Edit new file**), aus dem Sie den einzufügenden File wählen können.
- **save Block to disk** — Mit dieser Option wird ein Block als separater File auf einem Laufwerk gesichert. Ein Block ist ein Teil eines Programms, der markiert wurde, um kopiert, verschoben, gelöscht usw. zu werden. Einzelheiten zur Markierung eines Blocks sowie eine Beschreibung des Block-Menüs sind in Kapitel 4 enthalten.

- **DOS directory shell [Ctrl-F4]** — Anzeige der DOS-Verzeichnisse und Verzeichnisbäume, aus denen Files kopiert, gelöscht, gedruckt usw. werden können. Weitere Hinweise zur DOS-Verzeichnis-Shell sind in Anhang B enthalten.
- **Quit [Alt-X]** — Mit dieser Option verlassen Sie nicht nur das Edit-Menü, sondern auch die BASIC-Entwicklungssoftware. Wurden geöffnete Files geändert, jedoch nicht gesichert, werden Sie gefragt, ob die Änderungen gesichert werden sollen.

Wahl von Projektfiles

Mit der Option **select a project file** können Sie ein eigenes Menü für häufig verwendete Files anlegen, um diese schnell aufzufinden und zu wählen. Zudem können Files übersichtlicher angeordnet und schneller geladen werden. Die Option **select a project file** kann entweder vom Toplevel-Menü oder vom Other-Menü (unter dem Hauptmenü) aus aufgerufen werden.

Anwendermenü

Path: Toplevel Menu → Select a project file
 Path: Toplevel Menu → Edit mode → Other → Select a project file
 QuickStroke: Toplevel Menu → Edit mode → [Alt-ENTER]

```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91  1:15pm
File Window Block Cursor Search Text Layout Print Macro Config Other Exit
+==A+=====No-File?= ?+
|>> EOF <<   +-----User Menu-----+
|              | Create<Ins> Delete<Del> Copy<F4> |
|              |           Edit<F3>           |
|              |                               |
|              |                               |
|              | Example Linedraw Program with macros |
|              |                               |
|              |                               |
|              | Go<ENTER> Cancel<ESC> Help<F1> |
|              +-----+
|
|
|
+-----+
LIST BOX:  Next field<TAB>, Previous<ShftTAB>, ↑↓<PgUp><PgDn><Home><End>
    
```

Hinter dem Anwendermenü (User Menu) befindet sich Fenster **A**, und oberhalb des Anwendermenüs befindet sich das Hauptmenü, in dem angezeigt wird, daß der Edit-Modus aktiviert ist.

Mit den Cursortasten Pfeil-Auf und Pfeil-Ab und durch Anklicken des rechten Rollbalkens können Sie verschiedene Files hervorheben. Wenn der zu ladende File hervorgehoben ist, drücken Sie die [**ENTER**]-Taste oder klicken die Option **Go<ENTER>** an, um den File zu laden. Der File wird in das nächste verfügbare Fenster geladen. Falls erforderlich, wird ein neues Fenster erstellt.

Beim Laden eines Files in das Anwendermenü müssen Sie einen Filenamen spezifizieren, unter dem er im Anwendermenü aufgeführt werden soll. Der tatsächliche Name dieses Files wird hierdurch nicht geändert.

Beispiel: Sie bearbeiten zwei BASIC-Files, HAER1.BDL und HAER2.BDL, zur Steuerung eines Härteofens. Werden diese Files dem Anwendermenü hinzugefügt, können sie mit der Option **select a project file** schnell geladen werden. Sie können diesen Files auch einen beschreibenden Namen geben, die für andere Anwender der Software aussagekräftiger sind. HAER1.BDL könnte beispielsweise als STEUERUNGSPROGRAMM HAERTEOFEN, HAUPTOFEN, SUEWAND und HAER2.BDL könnte als STEUERPROGRAMM HAERTEOFEN, ERSATZOFEN, WESTWAND aufgeführt werden.

Laden von Files in das Anwendermenü:

1. Wählen Sie im Anwendermenü die Option **Create<Ins>**.
2. Geben Sie rechts neben der Option **Name** einen beschreibenden Filenamen ein.
3. Geben Sie rechts neben der Option **select a filename** den tatsächlichen Filenamen ein.
4. Wählen Sie zum Beenden die Option **OK<ENTER>**.

Beachten Sie, daß der beschreibende Name nun im Anwendermenü enthalten ist.

In das Anwendermenü können nicht nur Files aufgenommen werden, sondern auch bereits vorhandene Files editiert, gelöscht und kopiert werden.

Anwenderdefinierte Programme

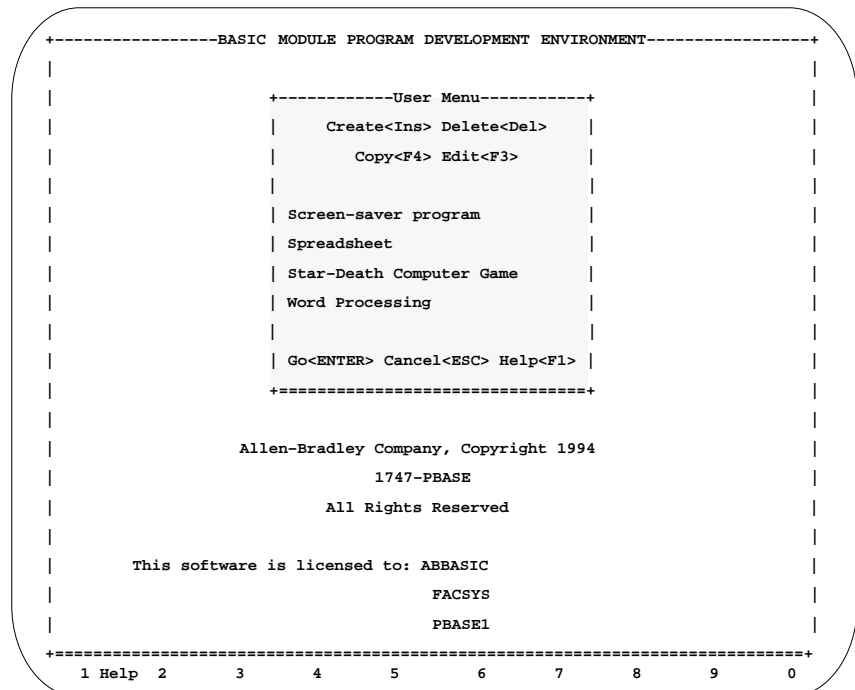
Mit Hilfe der Option **user defined program** im Toplevel-Menü können Sie die BASIC-Entwicklungssoftware vorübergehend verlassen, um andere Programme auf dem Personalcomputer auszuführen. Zu anwenderdefinierten Programmen zählen Textverarbeitungsprogramme, File- oder Laufwerkverwaltungsprogramme sowie Kalkulationstabellen.

Anwendermenü

Anwenderdefinierte Programme werden ebenso wie die zuvor beschriebenen Projektfiles im Anwendermenü (User Menu) angezeigt. Hierbei handelt es sich jedoch um ein völlig anderes Anwendermenü, und die darin enthaltenen Programme sind generell eigenständige Programme, die auf der DOS-Ebene des Personalcomputers ausgeführt werden können.

Path: Toplevel Menu → User defined program

Path: Main Menu → Edit mode → Other menu → User defined programs



Die Aufnahme von Programmen in das Anwendermenü erfolgt ähnlich wie im letzten Abschnitt beschrieben, mit der Ausnahme, daß einige speicher- und verzeichnisbezogene Eingaben gemacht werden müssen.

Benennung eines Programms

Bei Aufnahme eines Programms in das Anwendermenü müssen Sie einen beschreibenden Namen für das Programm eingeben, unter dem es dann im Menü aufgeführt ist. Der tatsächliche Filename bleibt dabei unverändert.

Eingabe eines beschreibenden Filenamens:

1. Wählen Sie im Anwendermenü die Option Create<INS>.



2. Geben Sie einen beschreibenden Namen für das Programm ein (bis zu 120 Zeichen), der im Anwendermenü angezeigt werden soll.

Wahl eines Filenamens

Damit eine anwenderdefinierte Funktion richtig funktioniert, müssen Sie den tatsächlichen Filenamens des Programms, so wie er im Verzeichnis des Personalcomputers angezeigt wird, eingeben. In der Regel ist es erforderlich, an dieser Stelle den ganzen DOS-Pfad einzugeben, sofern der File nicht im aktuellen Verzeichnis enthalten ist.

Wahl des aktuellen Filenamens:

1. Heben Sie die Option `select a filename` hervor, und drücken Sie die `[ENTER]`-Taste oder klicken Sie die linke Maustaste.

Es wird ein Dialogfeld angezeigt, in dem eine Auflistung der auf dem Computer vorhandenen Verzeichnisse eingesehen werden kann. Durchsuchen Sie die Verzeichnisse und Laufwerke dieser Auflistung nach dem eigentlichen Programmfile, der in das Anwendermenü aufgenommen werden soll.

2. Heben Sie das gewünschte Programm hervor, und drücken Sie die `[ENTER]`-Taste, oder wählen Sie das Programm mit der Maus an.

Der Filename wird für das gewählte Programm mit seinem vollständigen DOS-Pfad eingetragen.

Nachdem Sie den tatsächlichen Filenamens gewählt haben, bewegen Sie den Cursor nach unten auf die Option `swap method`.

Wahl der Speicherauslagerung

Die Option `swap method` dieses Dialogfeldes bezieht sich auf die Speicherzuordnung bei der Ausführung eines anwenderdefinierten Programms, wobei die aktiven Files und die aktuellen Bildschirmanzeigen der BASIC-Entwicklungssoftware beibehalten werden.

Es folgt eine Auflistung der Optionen, die unter der Option `swap method` verfügbar sind, sowie eine kurze Erläuterung ihrer jeweiligen Funktion:

- **No swap** — Bei der Ausführung eines anwenderdefinierten Programms wird durch das Auslagern der BASIC-Entwicklungssoftware auf eine Festplatte oder in einen EMS-Speicher kein zusätzlicher Speicherplatz frei. EMS bezieht sich auf den Erweiterungsspeicher des Computers, sofern vorhanden).

- **to disk only** — Bei der Ausführung eines anwenderdefinierten Programms wird durch das Auslagern der BASIC-Entwicklungssoftware auf eine Festplatte zusätzlicher Speicherplatz geschaffen.
- **to EMS only** — Bei der Ausführung eines anwenderdefinierten Programms wird durch das Auslagern der BASIC-Entwicklungssoftware in den Erweiterungsspeicher (EMS) zusätzlicher Speicherplatz geschaffen.
- **to EMS then to disk** — Bei der Ausführung eines anwenderdefinierten Programms wird durch das Auslagern der BASIC-Entwicklungssoftware in einen EMS und anschließend bei vollem EMS auf eine Festplatte zusätzlicher Speicherplatz geschaffen.
- **use default method** — Als Auslagerungsmethode wird die für **shelling to DOS** gewählte Methode verwendet.

Wichtig: Es ist möglich, daß ein anwenderdefiniertes Programm zuvor ohne Schwierigkeiten ausgeführt werden konnte und nun zu groß für den RAM-Speicher des Personalcomputers ist. Dies ist darauf zurückzuführen, daß das Programm den RAM-Speicher nun mit der BASIC-Entwicklungssoftware teilen muß.

Bei umfangreichen anwenderdefinierten Programmen, die einen Großteil des RAM-Speichers belegen, müssen in der Regel die Optionen **to EMS** oder **to disk only** gewählt werden, wenn dieser File dem Anwendermenü hinzugefügt wird.

Wahl der Auslagerungsmethode **swap method**:

1. Heben Sie die Option **swap method** hervor.
2. Wählen Sie die gewünschte Auslagerungsmethode **swap method**.

Positionieren Sie den Cursor nach der Wahl der Auslagerungsmethode unter **swap method** auf der Option **memory required**.

Speicheranforderungen

Die Option **Memory required** dieses Dialogfeldes bezieht sich auf den zum Abarbeiten des anwenderdefinierten Programms erforderlichen Speicher. Die BASIC-Entwicklungssoftware überprüft den zur Verfügung stehenden Speicherplatz, bevor das anwenderdefinierte Programm abgearbeitet wird. Ist der verfügbare Speicherplatz größer als die mit dieser Option spezifizierten Speicheranforderungen, wird die BASIC-Entwicklungssoftware nicht in den EMS bzw. auf eine Festplatte ausgelagert.

Die Option **Memory required** wird wie folgt gewählt:

1. Heben Sie die Option **Memory required** hervor.
2. Geben sie den für das anwenderdefinierte Programm erforderlichen Speicherplatz ein.

Nachdem die Speicheranforderungen gewählt wurden, verschieben Sie den Cursor nach unten auf **options**.

Die Option "Use COMMAND.COM"

Mit der Option **use COMMAND.COM** können interne DOS-Befehle, wie z.B. DIR, DEL, COPY usw., vom Menü "User Defined Program" aus verwendet werden.

Die Option **use COMMAND.COM** wird wie folgt gewählt:

1. Heben Sie die Option **use COMMAND.COM** hervor.
2. Wählen Sie mit der [Leertaste] zwischen [X] = Ja und [] = Nein.

Wählen Sie **options** und positionieren Sie den Cursor auf **starting directory**.

Anfangsverzeichnis (Starting Directory)

Wenn für den oben eingegebenen Filenamen kein vollständiger Verzeichnispfad eingegeben wurde, kann dies hier nachgeholt werden. Beispiel: Sie könnten **c:\word** eingeben, wenn der oben gewählte Name des Textverarbeitungsfiles tatsächlich im **word**-Verzeichnis des Laufwerks **c:** enthalten ist. Nähere Hinweise zur Definition von Verzeichnissen und Verzeichnispfaden sind im DOS-Handbuch enthalten.

Programmieren mit der BASIC-Entwicklungssprache

In diesem Kapitel wird das Schreiben von Programmen mit der BASIC-Entwicklungssoftware erläutert. Zunächst wird ein Überblick über die Sprache gegeben. Weitere Themen umfassen:

- Übersetzungsanweisungen
- Tasten- und Programmiermakros
- Programmintegration aus separaten Files
- Erstellen eines zweiten Programms

Überblick über die Sprache

Die Syntax, die den Kern der BASIC-Entwicklungssprache darstellt, wird im Referenzhandbuch der BASIC-Sprache (Publikationsnummer 1746-6.3DE) und dem Benutzerhandbuch für das BASIC-Modul (Publikationsnummer 1771-6.5.113DE) definiert. Dieser Kern wird mit Übersetzungsanweisungen und Programmiermakros ergänzt, die ein einfaches, offenes Format besitzen, das keine Zeilennummern erfordert. Anstelle von Zeilennummern werden beim Steuerungstransfer beschreibende Kennzeichnungsreferenzen als Ziel verwendet.

Als Beispiel ist nachstehend ein Programm in der BASIC-Entwicklungssprache zur Ausgabe des Datums dargestellt. Es enthält eine Übersetzungsanweisung und einen Programmiermakro und wird in ca. neun BASIC-Zeilen übersetzt. Dies bedeutet, anstelle der drei links abgebildeten Zeilen müßten *ohne* Verwendung der BASIC-Entwicklungssprache die rechts dargestellten elf Zeilen eingegeben werden.

```

REM EXAMPLE PROGRAM
{$I INCLUDE.BDL}
PRINT_DATE( )
0 REM ...
10 REM EXAMPLE PROGRAM
20 PUSH A
30 CALL 44
40 POP A
50 P. A, "/ ",
60 POP A
70 P. A, "/ ",
80 POP A
90 P. A, " ",
100 POP A

```

Die erste Anweisung `{ $\$$ I INCLUDE.BDL}` ist eine Übersetzungsanweisung. Im nächsten Abschnitt dieses Kapitels werden mehrere Übersetzungsanweisungen beschrieben. Diese Anweisung muß jedoch am Beginn der in der BASIC-Entwicklungssprache geschriebenen Programme stehen, wenn BASIC-Programmiersmakros verwendet werden sollen. Der 1747-PBASE-Übersetzer wird hierdurch während der Übersetzung auf den File INCLUDE.BDL verwiesen. Dieser File enthält die in Anhang A dieses Handbuchs beschriebenen Programmiersmakros.

Wichtig: Bei Verwendung des 1771-DB/B muß die Zeile `{ $\$$ I INCLUDEBB.BDL}` in das Programm aufgenommen werden.

Der Befehl `PRINT_DATE()` gehört zu den BASIC-Programmiersmakros des Files INCLUDE.BDL. Ein Programmiersmakro ist ein bereits geschriebenes BASIC-Unterprogramm. Wie aus Anhang A ersichtlich ist, stehen fast 100 Makros zur Verfügung, einschließlich solcher mit einzelnen und mehrfachen Variablen.

Zusammenfassend kann gesagt werden, daß die BASIC-Sprache der Kern der BASIC-Entwicklungssprache ist. Die BASIC-Entwicklungssprache erweitert die BASIC-Sprache mit Übersetzungsanweisungen und Programmiersmakros in einem offenen Format, das keine Zeilennumerierung erfordert.

Übersetzungsanweisungen

Ein in der BASIC-Entwicklungssprache geschriebenes Programm muß in die BASIC-Programmiersprache übersetzt werden, um an das BASIC-Modul heruntergeladen werden zu können. Die Übersetzung wird durch den 1747-PBASE-Übersetzer ausgeführt.

Diese Übersetzungssoftware wird von Übersetzungsanweisungen gesteuert, die in einem Programm enthalten sind. Mit diesen Übersetzungsanweisungen wird das vereinfachte, offene Format der BASIC-Entwicklungssprache in die Befehle und in das Format der BASIC-Programmiersprache übersetzt. Es folgt eine Auflistung der BASIC-Übersetzungsanweisungen:

Kommentar (Anmerkung)

Bei Text in geschweiften Klammern `{ }` handelt es sich um Programmkommentare, die *nicht* im übersetzten Programm enthalten sind. Anmerkungen (die keine Übersetzungsanweisungen sind und hier nur zum Vergleich erwähnt werden) hingegen *sind* im Übersetzungsprogramm enthalten. Beide können bei der Fehlersuche und Programmdokumentation als Hilfsmittel verwendet werden. Bei begrenztem Speicherplatz im BASIC-Modul empfiehlt es sich, mehr Kommentare und weniger Anmerkungen zu verwenden.

Syntax: { }

Syntax: REM

Beispiel:

Vor der Übersetzung:

Nach der Übersetzung:

```
{This is a comment}  
REM THIS IS A REMARK
```

```
0 REM ...  
10 REM THIS IS A REMARK
```

Wichtig: Wenn unmittelbar nach der linken geschweiften Klammer ein \$-Zeichen erscheint, wird der Inhalt der Klammer nicht als Kommentar interpretiert. Ferner ist es nicht zulässig, Kommentare in andere Kommentare zu verschachteln. Beispiele ungültiger Kommentare:

```
{$ comment}  
{{comment}}
```

Wichtig: Der 1747-PBASE-Übersetzer erstellt für jedes übersetzte .BDL-Programm eine Anmerkung (Zeile 0) (in diesem Handbuch durch 0 REM ... gekennzeichnet). Diese Anmerkung enthält den Filenamen des Übersetzers (BTRAN), die Revisionsnummer, den Pfad und den Namen des übersetzten Programms sowie das Datum und die Uhrzeit. Beispiel: Aus der Übersetzung des Programms FIRST (siehe Kapitel 2) resultiert:

```
0 REM BTRAN V X.XX C:\ABBASIC\BAS\FIRST.BDL, 06-12-91  
06:35pm
```

Include

Mit der Include-Anweisung lädt der Übersetzer einen weiteren Quellfile in das aktuelle Programm. Durch diese Art der Fileeinbeziehung können häufig verwendete Routinen in zahlreiche verschiedene Programme aufgenommen werden, ohne daß ein Code neu eingegeben werden muß oder Zeilennummern editiert werden müssen. Dieser eingefügte File kann beliebige Kombinationen von Programm- und Übersetzungsanweisungen enthalten.

Syntax: {\$I [filename.ext]}

Beispiel:

Vor der Übersetzung:

Nach der Übersetzung:

```
REM INCLUDE EXAMPLE  
{$I FIRST.BDL}
```

```
0 REM ...  
10 REM INCLUDE EXAMPLE  
20 FOR X=1 TO 100  
30 A=X*2  
40 P. A  
50 NEXT X  
60 END
```

In diesem Beispiel wurde der File FIRST.BDL einbezogen. Sein Name wurde in eine Include-Anweisung eingegeben, und nach der Übersetzung war das vollständige Programm enthalten.

Während der Übersetzung durchsucht der Übersetzer zunächst die Anweisung nach einem DOS-Pfad, der angibt, wo sich der File FIRST.BDL befindet. Beispiel:

`{ $I A:\FIRST.BDL }` kann spezifiziert werden, wenn sich der File FIRST.BDL auf Laufwerk A: befindet,

oder

`{ $I C:\ABBASIC\BDS\FIRST.BDL }` kann spezifiziert werden, wenn sich der File FIRST.BDL im BDS-Unterverzeichnis der BASIC-Entwicklungssoftware befindet.

Enthält die Include-Anweisung keinen DOS-Pfad, überprüft der Übersetzer seine Befehlszeile auf einen DOS-Pfad. Nähere Erläuterungen hierzu sind in Kapitel 8 enthalten.

Ist weder in der Include-Anweisung noch in der Befehlszeile des Übersetzers ein DOS-Pfad enthalten, sucht der Übersetzer den File FIRST.BDL im aktuellen Arbeitsverzeichnis.

Wird der File dennoch nicht aufgefunden, wird in der Meldungszeile während der Übersetzung der Fehler **Error 25: File not found** angezeigt.

Wichtig: Der File INCLUDE.BDL ist in der BASIC-Entwicklungssoftware enthalten. Er enthält BASIC-Programmiermakros und die zur Übersetzung dieser Makros erforderlichen Daten. Deshalb sollte die Anweisung `{ $I INCLUDE.BDL }` am Beginn jedes Programms stehen, in dem die Programmiermakros verwendet werden sollen.

Label

Anstelle von Zeilennummern werden als Bezugnahme auf eine Zielposition Labels im Programm verwendet. Eine symbolische Kennzeichnung wird dort verwendet, wo Sie in BASIC eine Programmzeilennummer zur Bezugnahme auf eine Verzweigung oder einen Sprung verwenden würden. Somit lassen sich Programme leichter lesen und verstehen. Auch ist es nach Durchführung von Änderungen nicht erforderlich, die Programmzeilen neu zu nummerieren.

Der Name des Labels besteht aus einer Zeichenkette mit maximal 30 ASCII-Zeichen. Der Zeichensatz ist als [0..9, A..Z, a..z, _] festgelegt. Der Name des Labels darf keine Tabulatorzeichen oder Leerzeichen enthalten und muß mit einem Buchstaben beginnen.

Syntax: `{ $L [labelname] }`

Beispiel:

Vor der Übersetzung:

```
REM LABEL EXAMPLE
IF A>C THEN GOTO GET_C_VALUE
A=55
{$L GET_C_VALUE}
C=107
```

Nach der Übersetzung:

```
0 REM ...
10 REM LABEL EXAMPLE
20 IF A>C THEN GOTO 40
30 A=55
40 C=107
```

Text

Mit der Text-Anweisung können Befehle wie z.B. RUN und LIST in das Programm eingefügt werden. Diesen Befehlen wird bei der Übersetzung *keine* Zeilennummer zugeordnet, sondern sie werden in den Ausgangsfile aufgenommen. Beim Herunterladen des Files an das BASIC-Modul werden diese Anweisungen unverzüglich nach ihrem Auffinden ausgeführt.

Syntax: {\$T [command for BASIC module]}

Beispiel:

Vor der Übersetzung:

```
REM TEXT COMMAND EXAMPLE
{$I INCLUDE.BDL}
BACKGROUND_BLACK()
BACKGROUND_RED()
CLS()
A=55
C=107
{$T RUN}
```

Nach der Übersetzung:

```
0 REM ...
10 REM TEXT COMMAND EXAMPLE
20 PRINT CHR(27), "[ 30m",
30 PRINT CHR(27), "[ 41m",
40 PRINT CHR(27), "[ 2J",
50 A=55
60 C=107
RUN
```

Im obenstehenden Beispiel wird sofort nach dem Herunterladen des Files der RUN-Befehl vorgefunden und ausgeführt.

Zeilennummer

Mit der Zeilennummeranweisung kann eine beginnende Zeilennummer spezifiziert werden, die der Übersetzer bei der Zuordnung von Zeilennummern verwendet. Die Zeilennummer kann eine beliebige Zahl zwischen 1 und 65535 sein.

Zeilennummeranweisungen können an einer beliebigen Stelle im Programm eingefügt werden. Die Zeile nach dieser Anweisung beginnt mit der neuen Nummer.

Syntax: {\$N [Zeilennummer]}

Beispiel:

Vor der Übersetzung:

```
REM LINE NUMBERING
{$N 500}
IF A>C THEN GOTO GET_C_VALUE
A=55
{$L GET_C_VALUE}
{$N 800}
C=107
```

Nach der Übersetzung:

```
0 REM ...
10 REM LINE NUMBERING
500 IF A>C THEN GOTO 800
510 A=55
800 C=107
```

Inkrement-Anweisung

Mit der Inkrement-Anweisung spezifizieren Sie einen numerischen Wert, um den die Zeilennummern erhöht werden sollen. Der Übersetzer verwendet diesen Wert bei der Zuordnung von Zeilennummern. Der Wert muß zwischen 1 und 100 liegen.

Diese Anweisung kann im Programm so oft wie erforderlich verwendet werden. Die erste Zeile direkt nach dieser Anweisung ist entsprechend dem vorherigen Erhöhungswert nummeriert. Alle folgenden Zeilen sind entsprechend dem neuen Inkrementierungswert nummeriert.

Syntax: {\$s [Zahl]}

Beispiel:

Vor der Übersetzung:

```
REM INCREMENT LINES BY 5'S
{$S 5}
5'S
IF A>C THEN GOTO GET_C_VALUE
A=55
REM INCREMENT LINES BY 10'S
{$S 10}
{$L GET_C_VALUE}
C=107
END
```

Nach der Übersetzung:

```
0 REM ...
10 REM INCREMENT LINES BY
5'S
20 IF A>C THEN GOTO 35
25 A=55
30 REM INCREMENT LINES By
10'S
35 C=107
45 END
```

Fixpunkt (Breakpoint)

Mit der Fixpunkt-Anweisung wird während der Übersetzung eine STOP-Anweisung in das Programm eingefügt. Die STOP-Anweisung wird nur dann eingefügt, wenn die Debugging-Anweisung aktiviert ist {\$D+}. Nach dem Herunterladen des Programms an das BASIC-Modul wird die Ausführung bei Auftreten einer STOP-Anweisung angehalten. Durch Eingabe des Befehls CONT (CONTinue – fortsetzen) wird die Ausführung an derselben Stelle fortgesetzt.

Fixpunkte können als Hilfsmittel bei der Fehlersuche oder zur Anzeige bzw. Modifizierung von Variablen verwendet werden.

Syntax: `{$B}`

Beispiel:
Siehe Debugging-Beispiel.

Debugging aktiviert **Debugging deaktiviert**

Bei aktivierter Debugging-Anweisung `{$D+}` werden während der Übersetzung an allen Fixpunkten `{$B}` STOP-Anweisungen eingefügt. Wenn an einer beliebigen Stelle im Programm `{$D+}` eingefügt wird, wird die Debugging-Funktion aktiviert, bis `{$D-}` vorgefunden wird. Standardmäßig ist die Debugging-Funktion deaktiviert `{$D-}`.

Wichtig: Zwischen der Debugging-Anweisung und der Fehlersuche des Terminal-Modus besteht kein Zusammenhang. Wenn die Fehlersuchfunktion des Terminal-Modus (siehe Kapitel 9) verwendet wird, sollte die Debugging- und die Fixpunkt-Anweisung *nicht* verwendet werden. Die Debugging- und Fixpunkt-Anweisungen sind zur Fehlersuche in Programmen des Terminal-Modus nützlich, ohne daß die Fehlersuchfunktion des Terminal-Modus verwendet wird.

Syntax: `{$D+}` to turn debug on
`{$D-}` to turn debug off

Beispiel:
Vor der Übersetzung:

```
REM DEBUGGING
{$B}
{$D+}
IF A>C THEN GOTO GET_C_VALUE
{$B}
A=55
{$L GET_C_VALUE}
{$B}
C=107
```

Nach der Übersetzung:

```
0 REM ...
10 REM DEBUGGING
20 IF A>C THEN GOTO 50
30 STOP
40 A=55
50 STOP
60 C=107
```

Bitte beachten Sie im vorausgehenden Beispiel, daß der erste Fixpunkt nicht in einen STOP-Befehl übersetzt wird. Der Grund liegt darin, daß die Debugging-Funktion an dieser Stelle noch nicht eingeschaltet war.

Makros

Die BASIC-Entwicklungssoftware stellt drei Arten von Makros zur Verfügung:

- Tastenmakros — vorübergehende oder permanente Aufzeichnung einer Tastenfolge, die durch eine Tastenkombination ersetzt wird. Eine nähere Beschreibung der Tastenmakros ist in Kapitel 4 enthalten.
- Programmiermakros — Optimierung der Programmierung und einfacheres Lesen und Verstehen der Programme. Die BASIC-Entwicklungssoftware enthält den File INCLUDE.BDL, der nahezu 100 Programmiermakros enthält. Eine Auflistung und detaillierte Beschreibung dieser Makros ist in Anhang A enthalten. Die Backplane-Aufrufe des 1771-DB/B sind im File INCLUDEBB.BDL enthalten. Eine Auflistung und detaillierte Beschreibung dieser Backplane-Aufrufe ist in Anhang A enthalten.

Wenn am Beginn eines Programms die Anweisung `{ $I INCLUDE.BDL }` steht, kann jeder beliebige Programmiermakro lediglich durch Eingabe seines Aufrufs, z.B. `foreground_green()` verwendet werden. Bei der Übersetzung des Programms greift der 1747-PBASE-Übersetzer auf den File INCLUDE.BDL zu, in dem sich das BASIC-Unterprogramm für den jeweiligen Makro befindet.

- anwendererstellte Programmiermakros — Unterstützung individueller Programmieranforderungen. Während die oben beschriebenen Programmiermakros bereits in der BASIC-Entwicklungssoftware enthalten sind, werden diese Makros vom Anwender erstellt. Hierzu wird die im nächsten Abschnitt beschriebene Übersetzungsanweisung `$MACRO` verwendet.

Erstellung eines Programmiermakros

Bei der Makroanweisung handelt es sich um eine Übersetzungsanweisung, mit der BASIC-Anweisungen als Makros definiert und im Programm eingesetzt werden können.

Ein Programmiermakro muß definiert werden, bevor er aufgerufen werden kann. Anderenfalls tritt ein Syntaxfehler ein. Die maximale Anzahl und Größe der in einem Quellfile spezifizierten Makros hängt vom verfügbaren Speicherplatz des Personalcomputers zum Zeitpunkt der Übersetzung ab. Programmiermakros können *nicht verschachtelt werden*.

In einem Makro können Labels definiert und angewandt werden, wobei jedoch nur innerhalb des Makros auf diese Labels Bezug genommen werden kann. Bei der Programmübersetzung weist der Übersetzer jedem Label im Makro eine separate Zeilennummer zu.

Der Makroaufruf besteht aus einer Zeichenkette, die bis zu 30 ASCII-Zeichen lang ist. Der Zeichensatz ist mit [0..9, A..Z, a..z, _] definiert. Der Makroaufruf darf keine Tabulatorzeichen und Leerstellen enthalten und muß mit einem Buchstaben beginnen.

Syntax: `{ $MACRO [macrocall(%1,%2...%n)] }`
`{ $ENDM }`

Die Variablen %1, %2... beziehen sich auf Parameterwerte, die an den Programmiermakro übertragen werden. Die maximale Anzahl der übertragenen Parameter, %n, ist auf 20 begrenzt. Die Anweisung `{ $ENDM }` kennzeichnet das Ende des Makros.

Beispiel:

In diesem Beispiel wird ein Programmiermakro zur Einstellung des Datums der Systemuhr geschrieben. Der Makro erhält die Bezeichnung "Setdate" und wird in den BASIC-Zeilen zwischen `{ $MACRO... }` und `{ $ENDM }` definiert. Nachdem der Programmiermakro definiert worden ist, kann er beliebig oft aufgerufen und an einer beliebigen Stelle des Programms eingefügt werden, indem lediglich der Name "Setdate" eingegeben wird.

Beispiel:

Vor der Übersetzung:

Nach der Übersetzung:

<code>REM CREATE PROGRAM MACRO</code>	<code>0 REM ...</code>
<code>{ \$MACRO Setdate(%1, %2, %3) }</code>	<code>10 REM CREATE PROGRAM MACRO</code>
<code>PUSH %1</code>	<code>20 D=16</code>
<code>PUSH %2</code>	<code>30 M=06</code>
<code>PUSH %3</code>	<code>40 Y=90</code>
<code>CALL 41</code>	<code>50 PUSH D</code>
<code>{ \$ENDM }</code>	<code>60 PUSH M</code>
<code>D=16</code>	<code>70 PUSH Y</code>
<code>M=06</code>	<code>80 CALL 41</code>
<code>Y=90</code>	
<code>Setdate(D,M,Y)</code>	

Ist eine große Anzahl anwenderdefinierter Programmiermakros bzw. eine bestimmte Anzahl häufig verwendeter Makros vorhanden, empfiehlt es sich, einen File zu erstellen, der ausschließlich diese Makros enthält. Dieser File kann dann in alle Programme eingefügt werden. Beispiel: Sie haben einen File mit der Bezeichnung `mine.bdl` erstellt, der Programmiermakros enthält. Alle Programme können dann mit dem Befehl `{ $I mine.bdl }` begonnen werden. Somit würde dieser File während der Übersetzung miteinbezogen werden, und die anwenderdefinierten Programmiermakros könnten an jeder beliebigen Stelle des Programms verwendet werden.

Einlesen von Programmen aus separaten Files

Je nach Anwendung ist es u.U. wünschenswert, von einem Programm aus andere Programme aufzurufen oder andere Programme direkt in das vorhandene Programm zu kopieren. Hierzu stehen mehrere Methoden zur Verfügung.

Berücksichtigen Sie bei der Entscheidung, ein Programm als Unterprogramm aufzurufen oder es in das Hauptprogramm einzufügen, folgendes:

- die Größe des aufzurufenden bzw. einzufügenden Programms
- die Häufigkeit, mit der es vom Hauptprogramm aus ausgeführt werden soll
- ob das Programm ein Unterprogramm anderer Programme ist bzw. sein kann
- ob das Einfügen des Programms dazu führt, daß das Hauptprogramm zu umfangreich und unübersichtlich wird

Wichtig: Wenn bestimmte Files nicht verknüpft bzw. editiert werden können, wurden diese wahrscheinlich gesperrt. Files können mit der Option `Information about current file` im File-Menü gesperrt (Nur-Lese-Files) bzw. geöffnet werden.

Unterprogramm CALL 70, CALL 71

In einem Hauptprogramm wird durch die BASIC-Befehle `CALL 70` und `CALL 71` veranlaßt, daß die Ausführung in ein Unterprogramm verzweigt wird. Das Unterprogramm muß zuvor im Speichermodul des BASIC-Moduls abgelegt worden sein. Mit `CALL 72` kann die Ausführung des Hauptprogramms wieder fortgesetzt werden. Weitere Informationen hierzu sind im Referenzhandbuch der BASIC-Programmiersprache (Publikationsnummer 1746–6.3DE) und im Benutzerhandbuch des BASIC-Moduls (Publikationsnummer 1771–6.5.113DE) enthalten.

Wichtig: Das BASIC-Modul unterscheidet nicht zwischen Unterprogramm- und Hauptprogrammvariablen. Wird eine im Hauptprogramm verwendete Variable in einem Unterprogramm verändert, wirkt sich dies auf das Hauptprogramm aus.

Programmiermakros

In Anhang A werden die in der BASIC-Entwicklungssoftware enthaltenen Programmiermakros beschrieben. Bei diesen Makros handelt es sich um Unterprogramme, die über die BASIC-Entwicklungssprache verwendet werden können. Mit der in diesem Kapitel beschriebenen `$Macro`-Übersetzungsanweisung können Sie eigene Programmiermakros erstellen.

Verknüpfen eines Programmfiles

Programmfiles können mit der Übersetzungsanweisung `$Include` oder der Option `Merge file from disk` im File-Menü verknüpft werden. `$Include` kann nur in Programmen, die in der BASIC-Entwicklungssprache geschrieben sind, verwendet werden. Diese Anweisung ist in diesem Kapitel im Abschnitt "Übersetzungsanweisungen" näher beschrieben.

`Merge file from disk` kann für jeden Programmfile verwendet werden. Sie werden aufgefordert, einen Filenamen einzugeben. Anschließend wird dieser File an der aktuellen Cursorposition in das Programm eingefügt.

Wichtig: Wenn zwei .BAS-Files mit gleichlautenden Zeilennummern verknüpft werden, bleiben die doppelten Zeilennummern erhalten, bis der File an das BASIC-Modul heruntergeladen wird. Ist zu diesem Zeitpunkt mehr als eine Zeile mit derselben Nummer vorhanden, ersetzt die zuletzt heruntergeladene Zeile alle vorhergehenden Zeilen mit derselben Nummer. Bei .BDL-Files braucht dieser Aspekt nicht berücksichtigt zu werden, da eine Zeilennumerierung nicht zulässig ist.

Erstellen eines zweiten Programms

In Kapitel 2 wurde die Erstellung eines einfachen Programms erläutert. Im folgenden ist ein etwas komplexeres zweites Programm enthalten, um die o.g. Programmierfunktionen zu veranschaulichen. Es werden Programmiermakros, Labels und die Linienzeichenfunktion verwendet.

Nach der Eingabe des Programms in der BASIC-Entwicklungssprache wird das Übersetzen, Herunterladen und Ausführung im BASIC-Modul kurz erläutert.

Wenn ein File mit einer .BDL-Erweiterung gesichert wird, bevor er bearbeitet wird, werden die Attribute dieser Erweiterung während des Editierens verwendet. Funktionen wie Einzug, Tabulatorabstand, rechter Rand, Wortumbruch und Sprachentyp werden für Sie erstellt.

Eingabe des Programms

1. Ziehen Sie das File-Menü herunter.
2. Wählen Sie die Option `save file as`. Das Dialogfeld `SAVE FILE AS` wird auf dem Bildschirm angezeigt.
3. Geben Sie `SECOND.BDL` ein.
4. Drücken Sie die `[ENTER]`-Taste oder positionieren Sie den Mauscursor auf `OK<ENTER>`, und klicken Sie die linke Maustaste. Das Dialogfeld `SAVE FILE AS` wird ausgeblendet, und in der Meldungszeile erscheint die Meldung
`C:\ABBASIC\BAS\SECOND.BDL saved.`

5. Geben Sie die ersten zwei Zeilen des zweiten Programms ein:

```
REM SECOND PROGRAM  
  {will print screen below then check operator number}
```

Bitte beachten Sie, daß es sich bei diesen Zeilen um Kommentare handelt, da sie in geschweiften Klammern { } stehen. Kommentare werden nicht übersetzt. Sollen diese Zeilen im übersetzten BASIC-File enthalten sein, müssen Sie **REM** anstatt der geschweiften Klammern { } verwenden. Damit die Kommentare in diesem Beispiel leichter erkennbar sind, wurden sie um fünf Leerstellen eingerückt.

6. Geben Sie `{ $I Include.bdl }` ein.

```
{ $I Include.bdl }
```

Beachten Sie bitte, daß `{ $I Include.bdl }` kein Kommentar ist, da unmittelbar auf die geschweifte Klammer { ein \$ folgt. `$I` ist die Include-Übersetzungsanweisung, mit der angefordert wird, daß während der Übersetzung ein separater File (in diesem Beispiel `Include.bdl`) miteinbezogen wird. `Include.bdl` ist der File der BASIC-Entwicklungssoftware, der die Programmiermakros enthält. Nachdem Sie mit der BASIC-Entwicklungssoftware vertraut geworden sind, können Sie einen eigenen File, z.B. `mine.bdl`, erstellen, der Ihre eigenen Programmiermakros enthält. Dieser File kann dann während der Übersetzung in das Programm aufgenommen werden, indem Sie am Beginn des Programms den Befehl `{ $I Mine.bdl }` eingeben.

7. Geben Sie den Kommentar "screen setup" und die darauffolgenden Bildschirmanweisungen ein. Bei diesen Befehlen handelt es sich um Programmiermakros (siehe Anhang A dieses Handbuchs).

```
  {screen setup}  
  background_Black()  
  High_Intensity()  
  foreground_Yellow()  
  cls()
```

8. Geben Sie den Kommentar "print statements" und den ersten Druckbefehl ein. Druckbefehle können mit **P.** abgekürzt werden, oder es kann der vollständige Befehl **PRINT** eingegeben werden. Ein Druckbefehl auf einer separaten Zeile führt dazu, daß eine Leerzeile gedruckt wird.

```
  {print statements}  
  print
```

Nun können die Druckbefehle eingegeben werden, welche bei Ausführung dieses Programms die Anzeige einer eingerahmten Meldung bewirken. Da in diesem Fall die Linienzeichenfunktion verwendet wird, empfiehlt es sich, zuerst den Text einzugeben und anschließend die Linien zu zeichnen. Das Linienzeichnen erfolgt nur im Überschreib-Modus. Deshalb dürfen in dem Bereich, in dem die Linien gezeichnet werden sollen, nur Leerstellen und kein Text vorhanden sein.

9. Geben Sie die Druckanweisungen (jedoch nicht die Linien) wie folgt ein. Fügen Sie in dem Bereich, in dem die Linien gezeichnet werden sollen, Leerstellen ein.

```
P.tab(10),"                               "  
P.tab(10),"                               WARNING      "  
P.tab(10),"   Use of this machine is restricted  "  
P.tab(10),"                               "
```

10. Ziehen Sie zum Zeichnen der Linien das Other-Menü herunter, und wählen Sie **Linedraw**.
11. Positionieren Sie den Cursor mit den Pfeiltasten an einer Ecke des Rahmens. Linien werden erst dann gezeichnet, wenn die [shift]-Taste gedrückt gehalten und die Pfeiltasten betätigt werden. Drücken Sie zum Zeichnen doppelter Linien die Funktionstaste [F2]. Drücken Sie diese Taste nochmals, um die Zeilen zu löschen. Drücken Sie diese Taste erneut, um wieder einfache Linien zu zeichnen.
12. Halten Sie die [shift]-Taste gedrückt, während Sie mit den Pfeiltasten ein Diagramm zeichnen. Beim Zeichnen einer Ecke bzw. beim Überkreuzen zweier Linien wird die gerade Linie durch die jeweils erforderlichen Zeichen ersetzt. Unterläuft Ihnen ein Fehler, drücken Sie die Funktionstaste [F2], um die unerwünschte Linie mit den Pfeiltasten zu löschen, während Sie die [shift]-Taste gedrückt halten.
13. Drücken Sie abschließend die [ESC]-Taste, um die Linienzeichenfunktion zu beenden.

14. Geben Sie nun den Rest des Programms ein:

```
hit_any_key()
    {pause and wait for keystroke}
cls()
{$L loop1}
Print "Enter Your Operator Number"
    {pause and wait for input}
get_key(op)
    {check for input of 49 or 50 which are ASCII decimal
      for 1 or 2}
if op=49 then goto op1setup
if op=50 then goto op2setup
    {print error and loop back for another entry}
cls()
print "Invalid Entry .. Operator number not recognized"
goto loop1
{$L op1setup}
get_time(d1,m1,y1)
    {setup for operator #1}
goto acknowledge
{$L op2setup}
get_time(d2,m2,y2)
    {setup for operator #2}
{$L acknowledge}
cls()
cursor_down(11)
print_time()
print "Operator #",op-48," acknowledged"
    {second program complete}
end
```

Übersetzen

Da dieses Programmbeispiel in der BASIC-Entwicklungssprache geschrieben ist, muß es in die BASIC-Sprache übersetzt werden, bevor es an das BASIC-Modul heruntergeladen wird. Hierzu wählen Sie im Other-Menü die Option "Translate/Compile" und anschließend die Option "1747-PBASE Translator". Das umstehende Programm ist die Übersetzung des oben dargestellten Programms.

Zweites Programm nach der Übersetzung (SECOND.BAS)

```

0 REM ...
10 REM SECOND PROGRAM
20 P. CHR(27),"[40m",
30 P. CHR(27),"[1m",
40 P. CHR(27),"[33m",
50 P. CHR(27),"[2J",
60 P.
70 P. TAB(10),"
80 P. TAB(10),"
90 P. TAB(10),"
100 P. TAB(10),"
110 P.
120 P. "Hit any key to continue....",
130 IF (EOF) THEN GOTO 130
140 PUSH A
150 A=GET
160 POP A
170 P.
180 P. CHR(27),"[2J",
190 P. "Enter Your Operator Number"
200 IF (EOF) THEN GOTO 200
210 OP=GET
220 IF OP=49 THEN GOTO 270
230 IF OP=50 THEN GOTO 320
240 P. CHR(27),"[2J",
250 P. "Invalid Entry .. Operator number not recognized"
260 GOTO 190
270 CALL 46
280 POP D1
290 POP M1
300 POP Y1
310 GOTO 360
320 CALL 46
330 POP D2
340 POP M2
350 POP Y2
360 P. CHR(27),"[2J",
370 P. CHR(27),"[11B",
380 PUSH A
390 CALL 46
400 POP A
410 P. A,":",
420 POP A
430 P. A,":",
440 POP A
450 P. A," ",
460 POP A
470 P. "Operator #",OP-48," acknowledged"
480 END

```

<p>WARNING Use of this machine is restricted</p>
--

Sofern keine Fehler auftreten, wird nach abgeschlossener Übersetzung in der Meldungszeile die Meldung "BASIC translator Version x.xx" angezeigt. Sollten Fehler aufgetreten sein, müssen Sie beseitigt werden, bevor Sie das Programm erneut übersetzen. Nähere Einzelheiten zum Übersetzen von Programmen und zur Eingrenzung von Übersetzungsfehlern sind in Kapitel 8 enthalten.

Herunterladen

Zum Herunterladen eines Files muß der Personalcomputer an das BASIC-Modul angeschlossen sein. Hierbei gilt die in diesem Handbuch, im BASIC-Modul SLC 500, Design- und Integrationshandbuch (Publikationsnummer 1746–6.1DE) und im Benutzerhandbuch des BASIC-Moduls (Publikationsnummer 1771–6.5.113DE) beschriebene Konfiguration. Die Brücken des Moduls sollten so eingestellt werden, daß sie diese Konfiguration unterstützen. Siehe BASIC-Modul SLC 500, Design- und Integrationshandbuch, Publikationsnummer 1746–6.1DE) oder Benutzerhandbuch des BASIC-Moduls (Publikationsnummer 1771–6.5.113DE).

Vor dem Herunterladen müssen Sie über das Konfigurationsmenü (siehe Kapitel 3) auch den korrekten Terminaltyp wählen und das Terminal über das Setup-Menü (aufrufbar vom Terminal-Modus aus) einstellen. Genaue Einzelheiten zur Einstellung des Terminals und zum Herunterladen sind in Kapitel 9 enthalten.

Laden Sie einen File nach der Durchführung dieser Schritte und ordnungsgemäßer Konfiguration folgendermaßen herunter:

1. Wählen Sie im Other- oder im Toplevel-Menü den **terminal**-Modus.
2. Wählen Sie vom Terminal-Modus aus das File-Menü.
3. Wählen Sie **Download from host to module**.
4. Wenn **second.bdl** der zuletzt übersetzte File ist, ist dessen übersetzte Version (**second.bas**) beim Herunterladen der Vorgabefile. Sie brauchen lediglich die [**ENTER**]-Taste zu drücken. Anderenfalls geben Sie den Namen des herunterzuladenden Files ein, in diesem Beispiel **second.bas**. Es darf nur ein übersetzter File heruntergeladen werden, d.h. ein File, dessen Erweiterung **.BAS** lautet.

Ausführung

Nachdem ein File erfolgreich in das BASIC-Modul heruntergeladen wurde, kann es ausgeführt werden, indem Sie rechts vom Aufforderungszeichen > den Befehl **run** eingeben. Das Programm kann mit der Funktionstaste [**F3**] verlassen werden. Die Ausführung kann mit der Tastenkombination [**ctr1 c**] unterbrochen werden, sofern diese Unterbrechungsfunktion nicht deaktiviert wurde. Die Fehlersuche in einem Programm des BASIC-Moduls ist in Kapitel 9 dieses Handbuchs beschrieben.

Drucken eines Programms

In diesem Kapitel wird die Einstellung des Druckers und das Drucken eines Programms beschrieben. Zu den wichtigsten Themen gehören:

- Wahl und Einstellung eines Druckers
- Verwendung der Druckersteuerungen
- Drucken eines vollständigen Dokumentes bzw. eines Auszugs

Einstellung des Druckers

Das Print-Menü wird vom Hauptmenü aus aufgerufen und dient zur Einstellung des Druckers sowie zur Steuerung des Ausdrucks eines Files. In diesem Kapitel werden hauptsächlich die Funktionen dieses Menüs behandelt.

Path: Toplevel Menu → Edit mode → Print
QuickStroke: none

```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91   1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----PRINT-----+
+==A+=====|print current File |==+
|>> EOF << | |(print marked Block) | | |
| | |-----| | |
| | |printer Setup... | |
| | |printer Type...IBM | |
| | |printer Device/file...LPT1 | |
| | |-----| | |
| | |Copies to print...1 | |
| | |Line numbering Off | |
| | |print Margin...1 | |
| | |eject Page | |
| | |-----Cancel<ESC>-----+
+-----+
1 Help 2 3 4 5 6 7 8 9 0

```

Wahl des Druckers

In der BASIC-Entwicklungssoftware sind bereits die Einstellungsparameter für eine Reihe von gängigen Druckern definiert. Außerdem ist die Software so flexibel konzipiert, daß fast jeder Drucker konfiguriert werden kann. Die Druckereinstellung beginnt mit der Wahl des Druckers.

Path: Toplevel Menu → Configuration menu → Printer

Path: Main Menu → Config → Printer

Path: Main Menu → Print → Printer type

```

L:3   C:1   1747-PBASE V x.xx   [Text Edit]   Ins 175k   06-12-91 1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----PRINT-----+
==+=====|print current File      |==+
|>> EOF <<| (print marked Block)          | | |
|          |-----| |
|          |printer Setup...      | |
|          |printer Type...IBM    | |
|          | +----SELECT A PRINTER TYPE----+|
|          | | Create<Ins> Delete<Del>    ||
|          | | Copy<F4> Edit<F3>        ||
|          | | +-----+|
|          | | IBM                      ||
|          | | OKIDATA                   ||
|          | | C.ITOH                    ||
|          | | EPSON                     ||
|          | | TOSHIBA                   ||
|          | | NEC                       ||
|          | | HP Laserjet II            ||
|          | | NONE                      ||
|          | |                           ||
|          | | OK<ENTER> Cancel<ESC> Help<F1> ||
|          | +-----+|
+-----+
LIST BOX: Next field<TAB>, Previous<ShftTAB>, ↑<PgUp><PhDn><Home><End>

```

Wählen Sie Ihren Drucker aus der angezeigten Auflistung. Sollte er hier nicht enthalten sein, kann mit der Option **Create<Ins>** der Bildschirm PRINTER DRIVER SETUP aufgerufen werden. Dort können dem Drucker ein Name, eine Initialisierungszeichenkette und Drucksteuerungscodes zugewiesen werden.

Ist der Drucker auf der angezeigten Liste aufgeführt, funktioniert er aber nicht ordnungsgemäß bzw. wunschgemäß (möglicherweise, weil es sich um ein älteres Modell mit geringfügig veränderten Steuerungscodes handelt), können die Steuerungscodes mit **<F3>** modifiziert werden.

Wichtig: Wird die **[ENTER]**-Taste gedrückt, nachdem die Bildschirmanzeige PRINTER DRIVER SETUP modifiziert wurde, werden die vorgenommenen Änderungen gesichert. Wurden Änderungen vorgenommen, die nicht gesichert werden sollen, oder wollen Sie die Bildschirmanzeige PRINTER DRIVER SETUP einfach verlassen, drücken Sie die **[ESC]**-Taste.

Der folgende Abschnitt enthält eine Beschreibung der Parameter, die Sie auf der Bildschirmanzeige PRINTER DRIVER SETUP erstellen und ändern können. Weitere Details sind im Hilfe-Bildschirmtext enthalten.

- **printer type** — wird bei der Erstellung einer neuen Druckertreibereinstellung eingegeben.
- **initialization string** — wird zur Initialisierung des Druckers eingegeben (nähere Einzelheiten sind in der Dokumentation des Druckers enthalten).

Für jeden Drucker können Sie bis zu 19 verschiedene Druckerodes spezifizieren. Jeder Druckercode enthält die folgenden definierbaren Datenfelder:

- **code name** — eine Zeichenkette, die den Druckercode kennzeichnet. Sie kann bis zu 17 Zeichen lang sein. Beispielsweise können Sie die Namen der Schriftarten eingeben, die den einzelnen Codes entsprechen. Die hier eingegebene Zeichenkette erscheint auf dem Druckereinstellungsbildschirm.
- **printer control code** — die eigentliche Zeichenkette, die an den Drucker übertragen werden soll. Hierbei handelt es sich in der Regel um eine Escape-Folge, die eine bestimmte Funktion des Druckers, unter anderem z.B. eine Änderung der Schriftart oder des Zeilenabstandes, initialisiert.

Druckereinstellung

Path: Main Menu → Print → **printer Setup**
QuickStroke: none

Bei der Wahl von **printer Setup** wird ein Menü der druckerspezifischen Steuerungs-codes angezeigt. Diese Codes stammen aus der o.g. Bildschirmanzeige PRINTER DRIVER SETUP.

In der Regel werden durch diese Codes Einstellungen wie z.B. Buchstabenqualität, Schriftart und Zeilenabstand festgelegt. Druckerodes sind je nach Drucker unterschiedlich. Verschieben Sie den Cursor auf den gewünschten Druckercode, und drücken Sie die [ENTER]-Taste, oder klicken Sie die linke Maustaste.

Druckergerät/-file

Path: Main Menu → Print → **printer Device/file**
QuickStroke: none

Bei Wahl der Option **printer Device/file** erscheint eine Bildschirmanzeige, aus der Sie ein Druckergerät oder einen Druckerfile wählen können, an das bzw. den die Druckausgabe übertragen werden soll. Sie können die vorhandenen Gerätenamen editieren und bei Bedarf neue Gerätenamen erstellen.

An dieser Stelle wählen Sie entweder einen Filenamen oder einen Druckerport. Bei der Wahl eines Filenamens, werden die zu druckenden Daten in diesen File übertragen. Bei der Wahl eines Ports werden die zu druckenden Daten direkt an den Drucker gesendet, der an diesen Port angeschlossen ist.

Die Kommunikation zwischen dem Personalcomputer und dem Drucker erfolgt in fast allen Fällen über den Parallelport (gewöhnlich LPT1).

Serielle Drucker

Einige Drucker kommunizieren über den seriellen Port und einige sind für die serielle und die parallele Kommunikation ausgerüstet. In diesem Abschnitt wird nur die serielle Druckerkonfiguration erläutert.

Mit der BASIC-Entwicklungssoftware kann sowohl die serielle als auch die parallele Kommunikation verwendet werden. Durch die Portkonfiguration des Personalcomputers wird festgelegt, über welchen Port des Computers die Druckausgabe erfolgt. Weitere Angaben zur Portkonfiguration sind in der Dokumentation des Personalcomputers und des Druckers enthalten.

Wichtig: In den folgenden Beispielen wird COM2 als der serielle Druckerport verwendet. Sie müssen sicherstellen, daß der spezifizierte serielle Druckerport nicht bereits als Schnittstelle für die Maus oder das BASIC-Modul verwendet wird.

Beim Drucken mit einem seriellen Drucker muß die Baudrate, Parität, Anzahl der Datenbits und Anzahl der Stoppbits des seriellen Ports mit den entsprechenden Parametern des Druckers übereinstimmen. Diese Anpassung kann u.a. mit dem DOS-**mode**-Befehl vorgenommen werden. Beispiel:

```
mode COM2:9600,n,8,1
```

Durch erneutes Programmieren des DOS-**mode**-Befehls kann der Druckerausgang vom LPT1-Port an einen seriellen Kommunikationsport umgeleitet werden. Beispiel:

```
mode LPT1:=COM2:
```

Soll der Druckerausgang nicht umgeleitet werden, kann dem Print-Menü ein serieller Drucker hinzugefügt werden.

```
Path: Toplevel Menu → Edit mode → Print  
QuickStroke: none
```

Ein neuer Drucker wird wie folgt spezifiziert:

1. Wählen Sie **printer device/file**, um das Dialogfeld **PRINTER DEVICE** aufzurufen.

2. Wählen Sie `Create<Ins>`, um einen neuen Drucker einzugeben.
3. Geben Sie den Namen des seriellen Kommunikationsports Ihres Personalcomputers ein, der als Druckerport verwendet werden soll, und drücken Sie die `[ENTER]`-Taste.

Beispielsweise würden Sie `COM2` eingeben.

Das angezeigte Dialogfeld `PRINTER DEVICE` enthält nun den eingegebenen Namen des Kommunikationsports am Beginn der Druckerauflistung.

4. Heben Sie den als seriellen Druckerport spezifizierten Anschluß hervor, und drücken Sie die `[ENTER]`-Taste.

Alle Druckausgaben werden nun an den spezifizierten seriellen Port gesendet.

Verwendung der Druckersteuerungen

Die Druckereinstellung wurde bereits erläutert. Vor dem eigentlichen Ausdrucken eines Files wird in diesem Abschnitt erörtert, welche Druckersteuerungen durch die BASIC-Entwicklungssoftware zur Verfügung stehen. Weitere Einzelheiten sind in der Hypertext-Hilfe enthalten.

Anzahl der Ausdrücke

`Path: Main Menu → Print → Copies to print`
`QuickStroke: none`

Mit dieser Druckersteuerung können Files oder Blöcke mehrmals ausgedruckt werden. Der Vorgabewert ist ein einmaliger Ausdruck.

Numerierung der Zeilen

`Path: Main Menu → Print → Line numbering`
`QuickStroke: none`

Mit dieser Druckersteuerung werden an der linken Seite des Ausdrucks Zeilennummern ausgedruckt. Hierbei handelt es sich *nicht* um BASIC-Zeilennummern, sondern lediglich um eine Numerierung der ausgedruckten Zeilen, beginnend mit Nummer 1.

Drucken eines Randes

`Path: Main Menu → Print → print Margin`
`QuickStroke: none`

Mit dieser Druckersteuerung wird ein linker Rand nur zum Ausdruck definiert (zusätzlich zu Rand- und Tabulatoreinstellungen, die im File vorgenommen wurden). Der Vorgabewert ist 1.

Seitenvorschub

Path: Main Menu → Print → eject Page
QuickStroke: none

Diese Druckersteuerung veranlaßt einen Seitenvorschub.

Drucken eines vollständigen Dokumentes bzw. eines Auszugs

Über das Print-Menü, das vom Hauptmenü aus aufgerufen wird, kann ein Block bzw. ein File direkt von einem Fenster aus ausgedruckt werden.

Path: Main Menu → Print → print current File
QuickStroke: none

Es folgt eine kurze Erläuterung der zwei Druckoptionen, die im Print-Menü verfügbar sind. Nähere Einzelheiten sind in der Hypertext-Hilfe enthalten.

- **print current file** — Drucken des gesamten Files des aktivierten Fensters. Der Druckvorgang kann durch Drücken der [Esc]-Taste abgebrochen werden.
- **print marked Block** — Drucken des derzeit markierten Blocks. Mit dieser Option kann anstelle des vollständigen Files ein Ausschnitt eines Files gedruckt werden. Weitere Informationen zu den drei Methoden der Blockmarkierung sind in Kapitel 4 im Abschnitt "Blockfunktionen" enthalten.

In der DOS-Shell stehen weitere Druckoptionen zur Verfügung. Einzelheiten können dem DOS-Handbuch entnommen werden.

Drucken im Hintergrund

Für diesen Vorgang muß der DOS-Drucker-Spooler PRINT.COM vor dem Aufruf der BASIC-Entwicklungssoftware geladen sein. Außerdem muß eine Pfadangabe auf den File PRINT.COM weisen. Das Drucken im Hintergrund ist möglich, wenn:

1. der File auf das Laufwerk ausgegeben wird,
2. wenn er über die DOS-Verzeichnis-Shell (siehe Anhang B) gewählt wird,
3. die Funktionstaste [F8] gedrückt wird.

Übersetzen eines Programms

Dieses Kapitel soll Sie mit den folgenden Aufgabenbereichen vertraut machen:

- Übersetzen eines Programms
- Eingrenzung und Korrektur von Programmierfehlern
- Konfiguration des Übersetzers

Übersetzen eines Programms

Die Übersetzungs-/Kompilierfunktion der BASIC-Entwicklungssoftware wird abhängig von der Erweiterung des Filenamens ausgeführt. Details zur Konfiguration der Filenamenerweiterung sind in Kapitel 3 enthalten. Um die korrekte Programmübersetzung zu gewährleisten, müssen die folgenden Filenamenerweiterungen verwendet werden:

Tabelle 8.A
Filnamenerweiterungen

Filenamenerweiterung	Beschreibung
Name.BDL	Der in der BASIC-Entwicklungssoftware geschriebene Quellfile. Der zu übersetzende File.
Name.BAS	Der vom 1747-PBASE-Übersetzer erstellte BASIC-File. Der an das BASIC-Modul herunterzuladende File.
Name.MAP	Der vom 1747-PBASE-Übersetzer erstellte File. Er dient zur Suche nach Variablen bei der Fehlersuche mit der Entwicklungssoftware (siehe Kapitel 9).

Wichtig: Der 1747-PBASE-Übersetzer (**BTRAN.EXE**) verarbeitet nur diese Fileerweiterungen und kann nicht neu konfiguriert werden.

Laden Sie den File vor Ausführung einer Programm-Übersetzung/-Kompilierung in ein Editierfenster. Wurde der File bereits geladen, muß sichergestellt werden, daß er sich im derzeit aktivierten Fenster befindet. Anschließend wird das Programm mit einem der folgenden Pfade übersetzt/kompiliert:

Path: Toplevel Menu → Translate/compile

Path: Toplevel Menu → Edit mode → Other →

Translate/compile

QuickStroke: Toplevel Menu → Edit mode → [CTRL F8]

Die Software überprüft die Fileerweiterung im aktivierten Editierfenster und fordert Sie auf, einen Übersetzer/Kompilierer zu wählen.

Wichtig: Der gewählte File muß die Erweiterung .BDL besitzen. Anderenfalls meldet der 1747-PBASE-Übersetzer einen Fehler.

Der File im aktivierten Editierfenster wird vor der Übersetzung immer gesichert. Andere offene, modifizierte Files werden gesichert, wenn der Übersetzer für das Sichern aller Files konfiguriert wurde (siehe weiter unten in diesem Kapitel).

Nachdem ein Übersetzer/Kompilierer gewählt wurde, übersetzt dieser den in der BASIC-Entwicklungssprache geschriebenen Quellcode. Der Übersetzer:

- entfernt Kommentare
- fügt BASIC-Zeilennummern ein
- ersetzt Label mit Zeilennummern
- fügt einzuschließende Files ein
- erweitert Programmiermakros
- sucht nach Syntaxfehlern

Im folgenden Programmbeispiel wird veranschaulicht, wie der Übersetzer bestimmte Aufgaben ausführt:

Zweites Programm vor der Übersetzung (SECOND.BDL)

```

REM SECOND PROGRAM
      {will print screen below then check operator number}
{$I Include.bdl}
      {screen setup}
background_Black()
High_Intensity()
foreground_Yellow()
cls()
      {print statements}
print
P.tab(10)," "
P.tab(10)," "
P.tab(10)," "
P.tab(10)," "
hit_any_key()
      {pause and wait for keystroke}
cls()
{$L loop1}
Print "Enter Your Operator Number"
      {pause and wait for input}
get_key(op)
      {check for input of 49 or 50 which are ASCII decimal
      for 1 or 2}
if op=49 then goto op1setup
if op=50 then goto op2setup
      {print error and loop back for another entry}
cls()
print"Invalid Entry .. Operator number not recognized"
goto loop1
{$L op1setup}
get_time(d1,m1,y1)
      {setup for operator #1}
goto acknowledge
{$L op2setup}
get_time(d2,m2,y2)
      {setup for operator #2}
{$L acknowledge}
cls()
cursor_down(11)
print_time()
print"Operator #",op-48," acknowledged"
      {second program complete}
end

```

<p>WARNING Use of this machine is restricted</p>
--

Treten während der Übersetzung keine Syntaxfehler auf, dann erstellt der Übersetzer ein BASIC-Programm (mit der Erweiterung .BAS), das zum Herunterladen an das BASIC-Modul bereitsteht:

Zweites Programm nach der Übersetzung (SECOND.BAS)

```
0 REM ...
10 REM SECOND PROGRAM
20 P. CHR(27),"[40m",
30 P. CHR(27),"[1m",
40 P. CHR(27),"[33m",
50 P. CHR(27),"[2J",
60 P.
70 P. TAB(10),"
80 P. TAB(10),"
90 P. TAB(10),"
100 P. TAB(10),"
110 P.
120 P. "Hit any key to continue....",
130 IF (EOF) THEN GOTO 130
140 PUSH A
150 A=GET
160 POP A
170 P.
180 P. CHR(27),"[2J",
190 P. "Enter Your Operator Number"
200 IF (EOF) THEN GOTO 200
210 OP=GET
220 IF OP=49 THEN GOTO 270
230 IF OP=50 THEN GOTO 320
240 P. CHR(27),"[2J",
250 P. "Invalid Entry .. Operator number not recognized"
260 GOTO 190
270 CALL 46
280 POP D1
290 POP M1
300 POP Y1
310 GOTO 360
320 CALL 46
330 POP D2
340 POP M2
350 POP Y2
360 P. CHR(27),"[2J",
370 P. CHR(27),"[11B",
380 PUSH A
390 CALL 46
400 POP A
410 P. A,":",
420 POP A
430 P. A,":",
440 POP A
450 P. A," ",
460 POP A
470 P. "Operator #",OP-48," acknowledged"
480 END
```

WARNING Use of this machine is restricted	" "
--	-----

Ferner erstellt er einen File mit der Erweiterung .MAP, der bei der Ausführung des Debugging-Programms eingesetzt wird (siehe Kapitel 9).

Wird ein Fehler im .BDL-File vorgefunden, bricht der Übersetzer/Kompilierer die Übersetzung ab und markiert den Fehler, damit er vom Anwender korrigiert werden kann. Beim Auffinden von Übersetzungsfehlern müssen diese, wie im folgenden Abschnitt beschrieben, korrigiert werden.

Beseitigen von Übersetzungsfehlern

Werden bei der Übersetzung des .BDL-Files Fehler festgestellt, so wird die Übersetzung gestoppt und ein Fehlerfenster geöffnet. Dieses enthält eine Auflistung aller eingetretenen Fehler sowie die entsprechenden Programmzeilennummern.

Wird während der Übersetzung ein Fehler in einem Include-File festgestellt, so wird ein weiteres Fenster geöffnet, das diesen File enthält. Der Cursor wird in diesem File an der Fehlerstelle positioniert.

Suche nach dem nächsten Kompilierfehler

Mit dieser Funktion wird der nächste Fehler im Quellfenster gefunden, der beim letzten Kompilervorgang eingetreten ist. Ist das Quellfenster nicht aktiviert, wählen Sie die Option `find next error`, um dieses Fenster aufzurufen.

Path: Toplevel Menu → Edit mode → Other → `find next error`

QuickStroke: Toplevel Menu → Edit mode → [Ctrl F2]

Bei Wahl der Option `find next error` wird der Cursor auf die Zeile positioniert, die den Übersetzungsfehler verursacht hat. In der Meldungszeile wird eine kurze Beschreibung des Fehlers angezeigt.

Konfiguration des Übersetzers

Der 1747-PBASE-Übersetzer ist so konfiguriert, daß .BDL-Files in .BAS-Files umgewandelt werden. Die Konfiguration des Übersetzers kann im Dialogfeld `COMPILER/PROGRAM SETUP`, das wie folgt aufgerufen wird, geändert werden:

Path: Toplevel Menu → Configuration menu → `Filename extensions` → BDL → `Compiler-Program setup` → 1747-PBASE Translator

```

L:3 C:1 1747-PBASE V x.xx [Text Edit] ↑ Ins 175k 06-12-91 1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
+-----SETUP AND CONF+-----EXTENSION SETUP-----+
+|DOS=3.30 CPU=80386 | Create<Ins> Delete<Del> |=====No-File?===+
||Video Card = VGA Colo+---COMPILER/PROGRAM SETUP-----+ | | |
||PBASE Path = C:\ABBAS | Create<Ins> Delete<Del> | |
||Exp+-----COMPILER/PROGRAM SETUP-----+ |
||Ser| Description: 1747-PBASE Translator | |
||---| Command line: BTRAN <FILE> /I\ABBASIC\BAS..... | |
||Edi| Program type: [1747-PBASE ] | |
||Dis| Memory required in k bytes(0=all): 0 | |
||Mou| | |
||Fil| Swap method: Options: | |
||Pri| (.) Use default [ ] Clear screen | |
||sWa| ( ) No swap [X] Save all files | |
||Bak| ( ) To disk only [ ] Reload file | |
||Sea| ( ) To EMS only [ ] Do NOT process compiler errors | |
||Ter| ( ) To EMS then to disk | |
||saV| | |
|| | OK<ENTER> Cancel<ESC> Help<F1> | |
+-----+-----+
|
|
+-----+-----+
STRING INPUT: Next field<TAB>, Previous<ShftTAB>, History list <↓>

```

In der folgenden Tabelle sind die Konfigurationsoptionen des Übersetzers aufgeführt, die im Dialogfeld **COMPILER/PROGRAM SETUP** zur Verfügung stehen. Dieses Feld enthält u.a. eine Beschreibung verschiedener Funktionen des 1747-PBASE-Übersetzers.

Tabelle 8.B
Konfigurationsoptionen des Übersetzers

Optionen	Beschreibung
Befehlszeile	Ermöglicht die Wahl verschiedener Parameter für den Übersetzeraufruf. Die vorgegebene Befehlszeile lautet BTRAN <FILE> /I\ABBASIC\BAS. Die Optionen für diese Zeile lauten BTRAN Filename[.ext] [/Ipath] [/Tpath], wobei gilt: .ext – wird durch .BDL ersetzt Ipath – ist der Pfad, der die Include-Files enthält. Falls dieser Parameter hier nicht spezifiziert wird, ist er das aktuelle Arbeitsverzeichnis. Tpath – ist der Pfad, über den der Übersetzer den übersetzten BASIC-File speichert. Falls dieser Parameter nicht spezifiziert wird, wird der File im selben Verzeichnis wie der Quellfile abgelegt. Im folgenden Beispiel wird der File im aktuellen Arbeitsverzeichnis übersetzt, wobei Include-Files aus dem Verzeichnis INCLUDE des Laufwerks D: verwendet werden. Der übersetzte File wird auf Laufwerk A: gesichert: BTRAN <FILE> /ID:\INCLUDE /TA:
Programmtyp	Mit dieser Option wird für jede Kompilierer-/Programmierschnittstelle ein Programm- bzw. Kompilierertyp gewählt.
Speicheranforderungen	Geben Sie den vom Kompilierer benötigten Speicherplatz in KByte (0 = gesamter Speicher) ein. Bei Verwendung der Speicherauslagerung läßt sich mit diesem Parameter der Umfang der Auslagerung bestimmen.

Optionen	Beschreibung
Auslagerungsmethode	<p>Mit den hier beschriebenen Optionen wird festgelegt, wo ggf. die Speicherauslagerung stattfindet. Sie werden je nach Größe des erforderlichen Speichers konfiguriert. In der Regel sollte hier die Option "Use default" gewählt werden. Die Optionen lauten:</p> <p>Use default – Die unter "Swapping/expanded memory" konfigurierte Auslagerungsmethode wird verwendet.</p> <p>No swap – Vor Aufruf des Kompilers/Programms wird kein zusätzlicher Speicherplatz geschaffen.</p> <p>To disk only – Mit dieser Option wird die Software bei Aufruf des Kompilers/Programms auf die Festplatte ausgelagert. Verwenden Sie diese Option, wenn zusätzlicher Speicherplatz benötigt wird, jedoch kein Erweiterungsspeicher im Personalcomputer installiert ist.</p> <p>To EMS only – Mit dieser Option wird die Software bei jedem Aufruf des Kompilers/Programms in den Erweiterungsspeicher ausgelagert.</p> <p>To EMS then to disk – Der gesamte verfügbare Erweiterungsspeicher wird belegt, bevor die Auslagerung auf eine Festplatte erfolgt.</p>
Optionen	<p>Es stehen eine Reihe verschiedener Optionen zur Verfügung:</p> <p>Clear screen – Die BASIC-Entwicklungssoftware wird während einer Übersetzung/Kompilierung ausgeblendet. Die Übersetzung kann somit von der DOS-Ebene aus auf Fehler überwacht werden. Diese Option sollte normalerweise nicht gewählt werden.</p> <p>Save all files – Wenn diese Option aktiviert ist, werden alle derzeit geöffneten Files, die geändert wurden, zwangsweise gesichert, bevor die Übersetzung ausgeführt wird. Ist diese Option nicht aktiviert, wird vor Ausführung der Übersetzung nur der File im derzeit aktivierten Fenster gesichert.</p> <p>Reload file – Der soeben kompilierte File wird in das aktivierte Fenster geladen. Dies ist dann von Vorteil, wenn der Originalfile vom Compiler verändert wird. Diese Option sollte normalerweise nicht gewählt werden.</p> <p>Do NOT process compiler errors – Bei Wahl dieser Option werden Compilerfehler nicht verarbeitet. Diese Option sollte normalerweise nicht gewählt werden.</p>

Kommunikation mit dem BASIC-Modul

Nachdem der Anwendungsfile in ein für das BASIC-Modul verständliches Format übersetzt wurde (Kapitel 8), muß die Kommunikation zwischen dem BASIC-Modul und der BASIC-Entwicklungssoftware hergestellt werden. Einzelheiten über den Anschluß des Personalcomputers an das BASIC-Modul sind in Kapitel 1 dieses Handbuchs und im Design- und Integrationshandbuch des BASIC-Modul SLC 500 (Publikationsnummer 1746–6.1DE) sowie im Benutzerhandbuch des BASIC-Moduls (Publikationsnummer 1771–6.5.113DE) enthalten.

Dieses Kapitel umfaßt die folgenden Themen:

- Wechseln in den Terminal-Modus
- Herunterladen des übersetzten Files an das BASIC-Modul
- Hochladen von Files aus dem BASIC-Modul
- interaktives Programm-Debugging im BASIC-Modul
- Einrichten der Terminal-Kommunikationsparameter

Wechseln in den Terminal-Modus

Für die Kommunikation zwischen dem Personalcomputer und dem BASIC-Modul muß der Terminal-Modus aufgerufen werden. Der Kommunikationsmodus des BASIC-Moduls wird durch die Einstellung seiner Brücken bestimmt. (Siehe Design- und Integrationshandbuch des BASIC-Modul SLC 500, Publikationsnummer 1746–6.1DE oder Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771–6.5.113DE). Der Kommunikationsmodus des Personalcomputers wird durch die BASIC-Entwicklungssoftware über das Konfigurationsmenü definiert (Terminal-Option in Kapitel 3 dieses Handbuchs). Stellen Sie sicher, daß die Kommunikationsmodi vor Aufruf des Terminal-Modus übereinstimmen.

Wenn die Herstellung der Kommunikation mit dem BASIC-Modul beim ersten Aufruf des Terminal-Modus nicht erfolgreich ist, stehen mehrere Menüoptionen zur Verfügung, um die Kommunikationsparameter (wie Kommunikationsport, Baudrate, Parität) in Übereinstimmung zu bringen. Hinweise zur Modifizierung der Kommunikationsparameter des Personalcomputers sind in den Abschnitten "Einrichtung der RS-232-Kommunikation" und "Einrichtung der DH-485-Kommunikation" am Ende dieses Kapitels enthalten.

Path: Toplevel Menu → teRminal

Path: Main Menu → Other → teRminal

QuickStroke: Edit mode → [F3]

Befindet sich das BASIC-Modul beim Aufruf des Terminal-Modus im RUN-Modus, wird die Ausgabe des derzeit ausgeführten Programms auf dem Bildschirm des Personalcomputers angezeigt. Befindet sich das BASIC-Modul nicht im RUN-Modus, wird bei Drücken der [ENTER]-Taste das Aufforderungszeichen > angezeigt.

In der folgenden Tabelle sind die BASIC-Befehle aufgelistet, die das BASIC-Modul im Terminal-Modus ausführen kann. Weitere Informationen über diese Befehle sind im Referenzhandbuch der BASIC-Programmiersprache (Publikationsnummer 1746-6.3DE) und im Benutzerhandbuch des BASIC-Moduls (Publikationsnummer 1771-6.5.113DE) enthalten.

Tabelle 9.A
Befehle des BASIC-Moduls

Befehl:	Funktion:	Beispiele:
CONT	Fortsetzung der Programmausführung nach einer STOP-Anweisung oder einem CONTROL-C-Befehl	CONT
[CTRL] C	Anhalten der aktuellen Programmausführung und Rückkehr in den Befehlsmodus	[CTRL] C
[CTRL] S	Unterbrechung eines LIST-Befehls	[CTRL] S
[CTRL] Q	Erneute Aktivierung eines LIST-Befehls nach einem CONTROL-S-Befehl	[CTRL] Q
DISABLING [CTRL] C	Deaktivierung der Unterbrechungsfunktion CONTROL C. Mit CALL 18 wird die Unterbrechungsfunktion CONTROL C deaktiviert und mit CALL 19 erneut aktiviert.	CALL 18 (Deaktivierung) CALL 19 (erneute Aktivierung)
ERASE	Löschen des Programms im ROM-Speicher	ERASE
LIST	Auflistung des aktuellen Programms bzw. der angegebenen Programmzeilen auf der Konsole	LIST, LIST 10-50
LIST#	Auflistung des aktuellen Programms bzw. der angegebenen Programmzeilen auf dem an PRT1 angeschlossenen Gerät	LIST#, LIST#50
LIST@	Auflistung des aktuellen Programms bzw. der angegebenen Programmzeilen auf dem an PRT2 angeschlossenen Gerät	LIST@, LIST@50
MODE	Einrichten der Portparameter	MODE(DH485,19200)
NEW	Löschen des Programms im RAM-Speicher	NEW
NULL	Zählen der Nullzeichen, die das BASIC-Modul nach einem Wagenrücklauf ausgibt	NULL, NULL4
PROG	Programmierung des EEPROM-Moduls mit dem aktuellen Programm	
PROG1	Programmierung des EEPROM-Moduls mit Portdaten aller drei Ports und Speichern von MTOP-Daten	

Befehl:	Funktion:	Beispiele:
PROG2	Ausführung des ersten im EEPROM gespeicherten Programms bei Einschalten des BASIC-Moduls	
RAM	Wahl des aktuellen Programms aus dem RAM-Speicher	RAM
ROM	Wahl des aktuellen Programms aus dem EEPROM-Speicher	ROM, ROM3
RUN	Ausführung des derzeit gewählten Programms	RUN
XFER	Übertragung eines Programms vom EEPROM- in den RAM-Speicher, anschließende Wahl des RAM-Modus	XFER

Herunterladen von Files

Nach der Erstellung und Übersetzung des Programms muß dieses an das BASIC-Modul heruntergeladen werden. Dieser Vorgang wird von der BASIC-Entwicklungssoftware ausgeführt, während der Terminal-Modus aktiviert ist. Der Terminal-Modus wird, wie im vorausgehenden Abschnitt beschrieben, aufgerufen.

Programme können vom Personalcomputer über eine RS-232- oder eine DH-485-Kommunikationsverbindung heruntergeladen werden. Für die DH-485-Kommunikationsverbindung ist ein Adapter lieferbar, der RS-232-Daten in DH-485-Daten umwandelt (siehe Design- und Integrationshandbuch des BASIC-Modul SLC 500, Publikationsnummer 1746-6.1DE oder das Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771-6.5.113DE).

1. Wählen Sie `Download from host to module` unter Verwendung des folgenden Pfads:

`Path: Toplevel Menu → teRminal → File → Download from host to module`

`Path: Main Menu → Other → teRminal → File → Download from host to module`

`QuickStroke: Toplevel Menu → teRminal → [Page Down]`

Sie werden aufgefordert, den Namen des herunterzuladenden Files einzugeben (siehe folgende Bildschirmanzeige).

Auf der Statuszeile der BASIC-Entwicklungssoftware blinkt die Meldung DOWNLOADING, und die Zeilen des Files rollen über den Bildschirm, während sie an das BASIC-Modul heruntergeladen werden.

Hochladen von Files

Die BASIC-Entwicklungssoftware enthält eine Hochladefunktion, mit der Programme vom BASIC-Modul in den Personalcomputer kopiert werden können. Diese Funktion wird hauptsächlich zur Sicherung von Files des BASIC-Moduls verwendet.

Programme können über eine RS-232- oder DH-485-Kommunikationsverbindung an den Personalcomputer hochgeladen werden. Für die DH-485-Kommunikationsverbindung ist ein Adapter lieferbar, der RS-232-Daten in DH-485-Daten umwandelt (siehe Design- und Integrationshandbuch des BASIC-Modul SLC 500, Publikationsnummer 1746-6.1DE, oder das Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771-6.5.113DE).

1. Wählen Sie über den unten dargestellten Pfad **upload from module to host**:

Path: Toplevel Menu → **teRminal** → **File** → **Upload from module to host**

Path: Main Menu → **Other** → **teRminal** → **File** → **Upload from module to host**

QuickStroke: Toplevel Menu → **teRminal** → [Page Up]

Sie werden aufgefordert, den Namen des hochzuladenden Files einzugeben (siehe folgende Bildschirmanzeige).

```
STRING INPUT: Next field<TAB>, Previous<ShftTAB>, History list<↓>
File Setup Display deBug Exit          ↑↓→← to Edit
PBASE RS-232 TERMINAL MODE Version x.xx
```

```
+-----FILE TO UPLOAD-----+
|Filename: BUGFIX.BAS          |
|                               |
|+=====Directory List=====+
||select this field to get     | OK<ENTER>
||a directory listing.        |
||                               | Cancel<ESC>
||                               |
||                               |
||                               | Help<F1>
||                               | Sort
||                               |
|+-----+
|
+-----+
```

2. Geben Sie Pfad, Filenamen und Fileerweiterung zum Speichern des hochgeladenen Files ein. Verwenden Sie hierzu eine der folgenden Methoden:
 - Der bei diesem Vorgang vorgegebene Filename ist der während dieser Bearbeitung zuletzt erfolgreich übersetzte File. Die Erweiterung des vorgegebenen Files lautet .UPL. Soll der vorgegebene File hochgeladen werden, fahren Sie mit Schritt 3 fort.
 - Drücken Sie entweder die [TAB]-Taste oder wählen Sie die Verzeichnisliste (Directory List), um auf den Verzeichnisbildschirm zuzugreifen und einen File zu wählen. Nach Aufruf dieses Feldes können verschiedene Verzeichnisse über ..\ angewählt bzw. verlassen werden. Es steht eine Sortierfunktion zur Verfügung, mit der das aktuelle Verzeichnis sortiert werden kann. Bei Wahl der Option `sort` werden Sie aufgefordert, einen Sortierschlüssel einzugeben. Geben Sie den bzw. die Buchstaben der gewünschten Sortierfunktion in das Sortierfenster ein.
 - Geben Sie Pfad, Filenamen und Fileerweiterung (z.B. \abbasic\bas\myfile.bas) ein.

Wichtig: Wenn ein File hochgeladen wird, ohne daß ein Filename angegeben wird, erhält dieser File den Namen des zuletzt übersetzten Files und die Fileerweiterung .UPL.

3. Nachdem der Filename eingegeben bzw. auf dem Verzeichnisbildschirm ausgewählt wurde, kann der Hochladevorgang durch Drücken der [ENTER]-Taste begonnen werden.

Auf der Statuszeile der BASIC-Entwicklungssoftware blinkt die Meldung UPLOADING, und die Zeilen des Files rollen über den Bildschirm, während sie vom BASIC-Modul hochgeladen werden.

Wichtig: Ist der spezifizierte File bereits im Personalcomputer vorhanden (dies ist bei der Wahl eines Files des Verzeichnisbildschirms immer der Fall), wird ein Bestätigungsfeld angezeigt. In diesem Feld werden Sie gefragt, ob der vorhandene File überschrieben oder der Hochladevorgang abgebrochen werden soll.

Hex ist die Abkürzung für das hexadezimale Datenformat. Dieses Datenformat wird generell in elektronischen Speichern verwendet, wie auch im optionalen EEPROM- bzw. UVPROM-Speichermodul des BASIC-Moduls. Mit dieser Art von Datenübertragungen werden hexadezimale Daten an den/aus dem EEPROM- bzw.

UVPROM-Speicher des BASIC-Moduls hoch- und heruntergeladen. Die Übertragungen von Hexadezimalfiles wird hauptsächlich dazu verwendet, den gesamten Inhalt des EEPROM-Speichers eines BASIC-Moduls an den EEPROM-Speicher eines zweiten BASIC-Moduls zu übertragen. Außerdem dienen sie dazu, den Inhalt des EEPROM-Speichers über ein PROM-Programmiergerät in den UVPROM-Speicher zu schreiben.

Übertragungen von Hexadezimalfiles

Hochladen von hexadezimalen Files

Das Hochladen von hexadezimalen Files gleicht dem Hochladen von Programmfiles, wobei einige Ausnahmen gelten. Stellen Sie sicher, daß der Terminal-Modus aktiviert ist und die Kommunikation mit dem BASIC-Modul, wie vorstehend beschrieben, hergestellt ist. Gehen Sie wie folgt vor:

1. Wählen Sie über den folgenden Pfad die Option `upload hex file from module to host`:

`Path: Toplevel Menu → teRminal → File → uPload hex file from module to host`

`Path: Main Menu → Other → teRminal → File → uPload hex file from module to host`

`QuickStroke: none`

Ebenso wie beim Hochladen eines Programmfiles werden Sie aufgefordert, einen Namen für den hochzuladenden File einzugeben.

2. Geben Sie, ebenso wie bei einem Programmfile, den Pfad, Filenamen und die Fileerweiterung zum Speichern des hochgeladenen Files ein. Ist der spezifizizierte Filename bereits vorhanden, werden Sie gefragt, ob dieser überschrieben werden soll.
3. Ein Dialogfeld wird angezeigt, in dem Sie die Anfangs- und Endadresse zum Hochladen des hexadezimalen Files eingeben müssen. Geben Sie als Anfangsadresse stets `8000H` ein. Drücken Sie die `[TAB]`-Taste oder positionieren Sie den Mauscursor auf dem Feld "Ending Adress".
4. Ist im BASIC-Modul ein Speichermodul 1747-M1 installiert, geben Sie als Endadresse `9FFFH` ein, und bei einem Speichermodul 1747-M2 geben Sie `FFFFH` ein. Informationen zum Typ des installierten Speichermoduls (M1 oder M2) können Sie dem Design- und Integrationshandbuch des BASIC-Modul SLC 500, Publikationsnummer 1746-6.1DE, und dem Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771-6.5.113DE, entnehmen. Für das Modul 1771-DB/B steht das 8 KByte große EEPROM-Speichermodul 1771-DBMEM1 (ähnlich dem SLC M1) und das 32-KByte große EEPROM-Speichermodul 1771-DBMEM2 (ähnlich dem SLC M2) zur Verfügung.

Wichtig: Die Adressen müssen genau wie oben beschrieben eingegeben werden, um beim Herunterladen dieses Files in andere EEPROM- bzw. UVPROM-Speicher einen Prüfsummenfehler zu vermeiden.

5. Drücken Sie nach Eingabe der Endadresse die `[ENTER]`-Taste, um das Hochladen des hexadezimalen Files zu starten.

Herunterladen von hexadezimalen Files

Das Herunterladen von hexadezimalen Files erfolgt auf die gleiche Weise wie das Herunterladen von Programmfiles. Die zum Hochladen von hexadezimalen Files erforderlichen Adreßdaten sind im File eingebettet und müssen zum Herunterladen nicht eingegeben werden. Stellen Sie sicher, daß der Terminal-Modus aktiviert ist und die Kommunikation mit dem BASIC-Modul, wie vorstehend beschrieben, hergestellt ist. Gehen Sie wie folgt vor:

1. Wählen sie über den folgenden Pfad die Option **Download hex file from host to module**:

Path: Toplevel Menu → teRminal → File → dOwnload hex file from host to module

Path: Main Menu → Other → teRminal → File → dOwnload hex file from host to module

QuickStroke: none

Ebenso wie beim Herunterladen eines Programmfiles werden Sie aufgefordert, einen Namen für den herunterzuladenden File einzugeben.

2. Geben Sie, ebenso wie bei einem Programmfile, den Pfad, Filenamen und die Fileerweiterung zum Speichern des hochgeladenen Files ein.
3. Drücken Sie nach Eingabe des Filenamens die **[ENTER]**-Taste, um das Herunterladen des hexadezimalen Files zu starten.

Neben den EEPROM-Speicheroptionen verfügt das BASIC-Modul auch über zwei UVPROM-Speicheroptionen. Näheres hierzu ist im Design- und Integrationshandbuch des BASIC-Modul SLC 500, Publikationsnummer 1746-6.1DE, und im Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771-6.5.113DE, enthalten. Es empfiehlt sich, beim Herunterladen von hexadezimalen Files zur Programmierung von UVPROM-Speichern die Dienstprogramme des PROM-Programmiergerätes zu verwenden.

Sie können jedes PROM-Programmiergerät verwenden, das:

- an den Personalcomputer angeschlossen werden kann,
- das Fileformat INTEL HEX unterstützt,
- die im BASIC-Modul verwendeten UVPROM-Speicher unterstützt.

Einzelheiten zu Konfiguration, Verbindung und Protokoll des PROM-Programmiergerätes sind in dessen Dokumentation enthalten.

AB Spares

Sichern des Modulabbildes

Bei diesem Vorgang kann der gesamte Inhalt des Modulspeichers, einschließlich aller im RAM und Anwender-PROM enthaltenen Programme, der batteriegepufferten Variablen und aller Modulkonfigurationen auf ein Laufwerk gesichert werden. Diese Funktion wird sowohl vom Modul 1771-DB, Serie B, als auch vom Modul 1746-BAS unterstützt. Gehen Sie folgendermaßen vor:

1. Wählen Sie unter Verwendung des folgenden Pfades die Option **Backup module image**:

Path: Toplevel Menu→ **teRminal**→ **File**→ **Backup module image**

Path: Main Menu→ **Other**→ **teRminal**→ **File**→ **Backup module image** **QuickStroke: none**

Sie werden aufgefordert, den Namen des zu sichernden Files einzugeben.

2. Geben Sie Pfad, Filenamen und Fileerweiterung des zu sichernden Files ein. Der vorgegebene Filename lautet **MODULE.IMG**.
3. Drücken Sie nach Eingabe des Filenamens die **[ENTER]**-Taste, und das Sichern des Modulabbildes beginnt.

Die folgenden Daten werden vom Modul hochgeladen:

- das im RAM-Speicher enthaltene BASIC-Programm, das im Intel-HEX-Format in einem File mit der Bezeichnung **<FILENAME>.RAM** gespeichert ist
- alle im Anwender-PROM enthaltenen Programme und Konfigurationsdaten, die im Intel-HEX-Format in einem File mit der Bezeichnung **<FILENAME>.ROM** gespeichert sind
- batteriegepufferte Parameter (Baudraten, Portkonfigurationen, MTOP usw.) und alle gespeicherten Programme und Konfigurationsdaten des Anwender-PROM, die im Intel-HEX-Format in einem File mit der Bezeichnung **<FILENAME>.PRM** enthalten sind
- vom Anwender definierten, batteriegepufferte Variablen, die im Intel-HEX-Format in einem File mit der Bezeichnung **<FILENAME>.USR** gespeichert sind.

Es wird ein weiterer File, **<FILENAME>.IMG**, erstellt, in dem die während des Sicherungsvorgangs aufgezeichneten Files aufgelistet sind.

Wichtig: Wenn der spezifizierte File im Personalcomputer bereits vorhanden ist, wird ein Bestätigungsfeld angezeigt, in dem Sie gefragt werden, ob der vorhandene File überschrieben werden soll, oder ob der Sicherungsvorgang abgebrochen werden soll.

Wiederherstellen des Modulabbildes

Bei diesem Vorgang wird die Konfiguration eines Moduls vollständig wiederhergestellt, so daß sie genau mit dem gesicherten Modulstatus übereinstimmt. Diese Funktion wird sowohl vom Modul 1771-DB, Serie B, als auch vom Modul 1746-BAS unterstützt. Gehen Sie folgendermaßen vor:

1. Wählen Sie über den folgenden Pfad die Option **Restore module image**:

Path: Toplevel Menu→ **teRminal**→ **File**→ **Restore module image**

Path: Main Menu→ **Other**→ **teRminal**→ **File**→ **Restore module image**

QuickStroke: none

Sie werden aufgefordert, den Namen des wiederherzustellenden Files einzugeben.

2. Geben Sie Pfad, Filenamen und Fileerweiterung des wieder herzustellenden Files ein. Der vorgegebene Filename lautet **MODULE.IMG**.
3. Drücken Sie nach Eingabe des Filenamens die **[ENTER]**-Taste, und das Wiederherstellen des Modulabbildes beginnt.
4. Wählen Sie eine der folgenden Optionen:
 - **Restore both RAM and User PROM** (RAM und Anwender-PROM wiederherstellen)
 - **Restore RAM only** (nur RAM wiederherstellen)
 - **Restore User PROM only** (nur Anwender-PROM wiederherstellen)

Sie können die im RAM oder im Anwender-PROM enthaltenen Programme oder alle in diesen beiden Speichern enthaltenen Programme wiederherstellen.

Alle im RAM oder Anwender-PROM gespeicherten Modulkonfigurationsdaten (z.B. Baudrate, Portkonfiguration) werden wiederhergestellt, wenn die entsprechende Menüwahl getroffen wird.

Auch können alle anwenderdefinierten batteriegepufferten Variablen wiederhergestellt werden. Werden diese Daten nicht wiederhergestellt, sollten sie vom Programm neu initialisiert werden.

AB Spares

Programm-Debugging

Der BASIC-Debugger dient zur Fehlersuche im BASIC-Modul auf Quellebene. Er dient ausschließlich zur Fehlersuche in BASIC-Programmen (mit der Filenamenerweiterung .BAS). Das Programm kann zeilenweise ausgeführt werden, um Variablen zu überprüfen und zu modifizieren und die Ausführung an Fixpunkten anzuhalten. Diese Funktion kann im Terminal-Modus online mit dem BASIC-Modul ausgeführt werden.

Zum Debugging muß zuerst das Programm an das BASIC-Modul, wie vorstehend in diesem Kapitel beschrieben, heruntergeladen werden. Aktivieren Sie den Debugger über den folgenden Pfad:

Path: Toplevel Menu → teRminal → deBug → Run Debugger

Nachdem die Debugging-Funktion aktiviert ist, sucht sie zuerst nach einem File mit der Erweiterung .MAP mit demselben Programmnamen wie das in das BASIC-Modul heruntergeladene Programm (dieser .MAP-File wird während der Übersetzung, wie in Kapitel 8 beschrieben, generiert). Ist kein .MAP-File vorhanden (z.B., wenn keine Übersetzung ausgeführt wird), wird der ursprüngliche BASIC-File (der zuvor heruntergeladene File) in das Debugging-Fenster geladen.

Ohne .MAP-File wird die Fehlersuche in den ursprünglichen Quellcodes u.U. erschwert. Der Bildschirm DEBUGGING INFORMATION, der den Namen des in das Debugging-Fenster zu ladenden Files enthält, wird angezeigt. Drücken Sie von diesem Bildschirm aus eine beliebige Taste, um den Quellfile-Debugger aufzurufen.

Während der Fehlersuche stehen über den Debugger die folgenden Funktionen zur Verfügung:

Tabelle 9.B
Funktionen des Debuggers

Funktions-taste	Funktion
[F1] Help	Aufruf des Hilfebildschirms
[F2] Inquire	Untersuchung und Modifizierung einer Variablen
[F3] Watch	Untersuchung des Wertes einer ausgeführten Variablen im Überwachungsfenster
[F4] Gotill	Ausführung des Programms bis zur Zeile, in der sich der Cursor befindet
[F5] Screen	Anzeige des aktuellen Status der Terminal-Bildschirmanzeige
[F6] Search	Suche nach einer Zeichenkette im aktuellen File
[F7] Trace	zeilenweise Ausführung des Programms, in dem die Fehlersuche stattfindet
[F8] Layout	Erstellung von Fenstern, in denen .BDL-, .BAS- und Überwachungsfenster angezeigt werden können
[F9] Run	Ausführung des BASIC-Programms von der gegenwärtigen Cursorposition aus
[F10] Quit	Abbruch, Beendigung der Fehlersuche
[Tab] Field	abwechselnde Aktivierung der offenen Debugging-Fenster

Wichtig: Mit der Anweisung `{$B}` können Fixpunkte auch manuell in den Quellcode eingefügt werden (siehe Kapitel 6). Diese Anweisung ist nicht für die Verwendung mit dem hier beschriebenen Debugger vorgesehen.

Debugging-Beispiel

Anhand des folgenden Programmbeispiels sollen die Funktionen der Debugging-Funktion veranschaulicht werden. Es handelt sich hier um ein einfaches Programm, damit Sie die Ausführung jeder Zeile während der Fehlersuche nachvollziehen können.

```
REM THIS IS A DEBUGGING EXAMPLE
{SORT EVEN NUMBERED POSITIVE ENTRIES}
{$L START}
INPUT "ENTER A NUMBER: ",Y
FOR X=1 TO Y
IF X*2=Y THEN GOTO DONE
NEXT X
PRINT "NUMBER IS ODD OR LESS THAN 1, RESTARTING"
GOTO START
{$L DONE}
PRINT "NUMBER IS EVEN"
END
```

Sichern Sie das Programm nach der Eingabe unter dem Namen `DEBUG.BDL`, und übersetzen/kompilieren Sie es anschließend. Treten keine Fehler auf, wechseln Sie in den Terminal-Modus über, und laden Sie den übersetzten File (`DEBUG.BAS`) an das BASIC-Modul herunter. Befolgen Sie bei diesem Vorgang die folgenden Schritte.

1. Geben Sie nach dem Herunterladen den Befehl `run` ein, um das Programm auszuführen. Sie werden aufgefordert, eine Zahl einzugeben. Anderenfalls liegt möglicherweise ein Fehler im Programm vor, der ggf. mit Hilfe des Debuggers beseitigt werden kann.
2. Geben Sie eine ungerade Zahl (unter 100) ein, und drücken Sie die `[ENTER]`-Taste. Es wird eine Meldung angezeigt, die bestätigt, daß die Zahl ungerade ist. Sie werden erneut aufgefordert, eine Zahl einzugeben.
3. Geben Sie eine gerade Zahl (unter 100) ein, und drücken Sie die `[ENTER]`-Taste. Es wird eine Meldung angezeigt, die bestätigt, daß die Zahl gerade ist, und das Programm wird beendet.
4. Wählen Sie aus dem Menü am oberen Bildschirmrand die Option `deBug` und anschließend die Option `run Debugger`. Wurde ein zuvor übersetzter File geladen, für den ein `.MAP`-File vorhanden ist (wie in diesem Beispiel der Fall ist), wird eine Meldung angezeigt, die bestätigt, daß der `.MAP`-File verwendet wird.

11. Drücken Sie [F2], um den Wert der Variablen zu überprüfen. Dieser Wert ist die Hälfte des in Schritt 3 eingegebenen Wertes. Dies entspricht der Art und Weise, auf die gerade Nummern vom Programm sortiert werden.
12. Drücken Sie die [Esc]-Taste, um den Debugger-Bildschirm wieder aufzurufen, ohne den Wert der Variablen zu ändern.
13. Drücken Sie [F3], während der Cursor noch auf der Variablen x positioniert ist. Die Variable x und deren Wert werden nun im Überwachungsfenster angezeigt. Derselbe Vorgang kann für y durchgeführt werden.
14. Verschieben Sie den Cursor nach unten an eine beliebige Stelle der Zeile `next x`, und drücken Sie anschließend [F4]. Hierdurch wird das Programm ausgeführt, bis es die markierte Zeile erreicht (in diesem Beispiel `next x`). Die Bildschirmanzeige des Terminal-Modus wird eingeblendet, und das ausführende Programm fordert Sie auf, eine Zahl einzugeben.
15. Geben Sie den Wert 3 ein, und drücken Sie die [ENTER]-Taste. Der Debugger-Bildschirm wird angezeigt, auf dem die Zeile `next x` hervorgehoben ist. Der Wert x im Überwachungsfenster beträgt 1, und der Wert y beträgt 3. Die Programmausführung wird beim ersten Auffinden der Zeile `next x` unterbrochen.
16. Drücken sie [F4] erneut. Jetzt nimmt x den Wert 2 an. Drücken Sie [F4] so oft, bis der Durchlauf der For-Next-Schleife abgebrochen wird, die Meldung einer ungeraden Zahl angezeigt wird und Sie aufgefordert werden, eine weitere Zahl einzugeben.
17. Geben Sie den Wert 6 ein, und drücken Sie die [ENTER]-Taste. Der Debugger-Bildschirm wird wieder eingeblendet, der Wert x im Überwachungsfenster beträgt nun 1, und der Wert y beträgt 6.
18. Drücken Sie [F7]. Sie werden zeilenweise durch das Programm geführt, wobei während der Ausführung kurz die Anzeige des Terminal-Modus eingeblendet und anschließend wieder der Debugger-Bildschirm angezeigt wird. Hier sehen Sie, wie sich der Wert x bei jedem Durchlauf der For-Next-Schleife ändert.
19. Drücken Sie [F7] so oft, bis Sie am Ende des Programms angelangt sind. Die Debugging-Funktion wird beendet, und der Terminal-Modus wird wieder aufgerufen.
20. Wiederholen Sie die Schritte 4 und 5, um zum Debugger zurückzukehren.
21. Drücken Sie [F5], um den Terminal-Modus vorübergehend aufzurufen. Drücken Sie eine beliebige Taste, um den Debugger-Bildschirm wieder einzublenden.

22. Drücken Sie [F6]. Die nun eingeblendete Suchmaske ist identisch mit der der BASIC-Entwicklungssoftware und dient zur Suche nach Zeichen im Programm, z.B. nach Variablen, die im Überwachungsfenster angezeigt werden sollen. Drücken Sie [Esc], um die Suche zu beenden.
23. Drücken Sie [F8]. Der Bildschirm wird nun in drei Abschnitte unterteilt. Das übersetzte BASIC-Programm befindet sich im oberen, der .BDL-File im mittleren und das Überwachungsfenster im unteren Abschnitt.
24. Drücken Sie die [Tab]-Taste, um die drei Fenster abwechselnd zu aktivieren. Sie können nun Variablen des angezeigten .BAS-Files bzw. .BDL-Files wählen, um sie dann zu untersuchen ([F2]) oder zu überwachen ([F3]).
25. Drücken Sie [F8] erneut. Jetzt befinden sich nur noch das BASIC-Programm und das Überwachungsfenster auf der Bildschirmanzeige. Drücken Sie [F8], um den ursprünglichen Debugger-Bildschirm wieder aufzurufen.
26. Drücken Sie [F9], um das Programm vollständig auszuführen und den Debugger-Bildschirm zu verlassen. Durch Drücken von [Esc] oder [F10] kann der Debugger-Bildschirm jederzeit verlassen werden.

Programmkorrekturen

Beim Programmieren mit der BASIC-Entwicklungssoftware wird normalerweise diese Reihenfolge beachtet:

1. Schreiben eines .BDL-Files mit der BASIC-Entwicklungssoftware
2. Übersetzen/Kompilieren des .BDL-Files, um einen .BAS-File zu erstellen
3. Herunterladen des Files an das BASIC-Modul

Nach dem Herunterladen sollte das Programm ausgetestet und auf Fehler untersucht werden. Aus dieser Prüfung resultierende Änderungen und Korrekturen sollten nur mit äußerster Sorgfalt durchgeführt werden. Es liegen derzeit drei Versionen des Files vor: der .BDL- und der .BAS-File im Personalcomputer sowie der an das BASIC-Modul heruntergeladene File.

Obwohl es schneller ist, einfache Korrekturen im Terminal-Modus durch das zeilenweise Editieren des Programms im BASIC-Modul vorzunehmen (eine Beschreibung dieses Vorgangs ist im Design- und Integrationshandbuch des BASIC-Modul SLC 500, Publikationsnummer 1746–6.1DE, und im Benutzerhandbuch des BASIC-Moduls, Publikationsnummer 1771–6.5.113DE, enthalten), sind diese Korrekturen dann nicht in den PC-Versionen des Programms enthalten.

Werden dennoch geringfügige Änderungen am Programm im BASIC-Modul vorgenommen, empfiehlt es sich, diese Änderungen auch im ursprünglichen .BDL-File nachzuvollziehen und das Programm anschließend zu übersetzen/kompilieren. Laden Sie das Programm anschließend an das BASIC-Modul herunter, und prüfen Sie, ob es ordnungsgemäß funktioniert. Damit wird sichergestellt, daß sich alle Versionen inhaltlich entsprechen und daß für das Programm im BASIC-Modul eine Sicherungskopie vorhanden ist.

Muß ein Programm umfassend editiert werden, sollten die erforderlichen Änderungen am ursprünglichen .BDL-File vorgenommen werden. Obwohl es möglich ist, einen editierten File vom BASIC-Modul an den Personalcomputer hochzuladen, würde es sich hierbei um eine .BAS-Version des Files handeln, die *nicht* in das .BDL-Format zurückübersetzt werden kann.

Einrichtung der RS-232-Kommunikation

Vor der Konfiguration der Kommunikationsparameter des Terminals muß als Kommunikationsverbund zunächst RS-232 oder DH-485 gewählt werden (die Option "General Purpose Communications" – Allzweckkommunikation ist nicht für den Einsatz mit dem BASIC-Modul vorgesehen). Hierzu rufen sie über das Menü "Setup and Configuration" das Menü "Terminal Selection" auf. Einzelheiten zur Auswahl des Terminals sind in Kapitel 3 erläutert.

Zur Konfiguration der Kommunikationsparameter für den jeweils gewählten Kommunikationsverbund rufen Sie über den folgenden Pfad das Menü "Terminal Setup" auf:

Path: Toplevel Menu → teRminal [RS-232] → Setup

Path: Main Menu → Other → teRminal [RS-232] → Setup

AB Spares

```
HELP<F1>|MENU<F2>|EXIT<F3>| COM1 1200 8 1 N|LOG CLOSED|ANSI | 06-12-91 1:15pm
File Setup Display deBug Exit          ↑↓→← to Edit
+-----SETUP-----+
|Autobaud                               |=====+
|General (file names, etc.)...          |
|Terminal...                            |
|com Port settings (baud rate, etc.)... <AltP>|
|Colors                                  |
|saVe setup                              |
+-----Cancel<ESC>-----+
|
|
|
|
+-----+

```

Autobaud

Über diese Menüauswahl wird die Baudrate des BASIC-Moduls, das an den derzeit konfigurierten RS-232-Port des Personalcomputers angeschlossen ist, automatisch festgestellt. Mit Autobaud wird zwar nach der Baudrate gesucht, Datenbits, Stoppbits, Parität und Portnummer werden jedoch nicht eingestellt. Bevor durch Autobaud die passende Baudrate eingestellt werden kann, müssen diese Parameter korrekt konfiguriert werden, so daß sie mit denen des BASIC-Moduls übereinstimmen.

Allgemeine Einrichtungsparameter

Bei Wahl dieser Menüoption wird ein Dialogfeld mit den folgenden allgemeinen Einrichtungsparametern eingeblendet:

- **Upload path** — Definition des DOS-Pfades im PC, über den die vom BASIC-Modul hochgeladenen Files gespeichert werden.
- **Download path** — Definition des DOS-Pfades im PC, über den die an das BASIC-Modul heruntergeladenen Files gespeichert werden.
- **Log File** — Name des Protokollfiles. Vor dem Öffnen eines Protokollfiles kann dessen Name geändert werden.
- **Beep on downloads and UPLOADs** — Tonsignal, das ertönt, wenn das Hoch- bzw. Herunterladen eines Files beendet ist oder abgebrochen wird.

- **word wrap in file viewer** — Die Wortumbruchfunktion beim Einsehen eines Files wird aktiviert.
- **right margin for file viewer** — Der rechte Rand für den Wortumbruch beim Einsehen eines Files.

Terminal

Durch Wahl der Option "Terminal" wird das Dialogfeld **TERMINAL SETUP** eingeblendet, in dem zwischen der ASCII- und ANSI-Terminalemulation gewählt werden kann.

Einstellung des COM-Ports

Mit dieser Menüoption wird das Dialogfeld **PORT SETUP** angezeigt, über das die Parameter des Kommunikationsports konfiguriert werden können. Die Vorgabeeinstellungen der BASIC-Entwicklungssoftware und des RS-232-Ports des BASIC-Moduls sind wie folgt aufeinander abgestimmt:

Tabelle 9.C
Vorgabeeinstellungen

Einstellung des COM-Ports	Vorgabe
aktiver COM-Port	Com1
Baudrate	1200
Parität	keine
Datenbits	8
Stoppbits	1
Handshake-Funktion	aktiviert

Die folgenden Parameter stehen zur Verfügung:

- **active com port** — Wahl des PC-Ports, der für die Kommunikation mit dem BASIC-Modul verwendet werden soll.

Wichtig: Es muß sichergestellt werden, daß der für die Maus gewählte Port nicht mit dem für die serielle Kommunikation mit dem BASIC-Modul gewählten Port konfliktiert.

- **Baud rate** — Wahl der Baudrate. Verfügbare Geschwindigkeiten: 300, 600, 1200, 2400, 4800, 9600 und 19200.
- **Parity** — Wahl zwischen gerader, ungerader oder keiner Parität.
- **Data bits** — Wahl der Datenbitanzahl jeder Zeichenübertragung an den bzw. vom Programmierport des BASIC-Moduls. Gültige Optionen: 7 oder 8.

- **sTop bits** — Wahl der Stoppbitanzahl. Gültige Optionen: 1 oder 2.
- **Handshaking** — Mit dieser Option wird entschieden, ob die Software-Handshake-Quittierung zwischen dem Personalcomputer und dem Gerät, zu dem eine Kommunikationsverbindung besteht, stattfindet. Dieses Menüelement sollte immer aktiviert werden, wenn Programme an das bzw. aus dem BASIC-Modul herunter- und hochgeladen werden.

Farben

Mit dieser Menüoption werden die Vordergrund- und Hintergrundfarben des Bildschirms eingestellt. Die Vordergrundfarbe wird mit den Pfeil-Auf- und Pfeil-Ab-Tasten und die Hintergrundfarbe mit den Pfeil-Links- und Pfeil-Rechts-Tasten eingestellt.

Sichern der Einstellung

Durch Wahl der Option **save setup** werden alle aktuellen Einstellungen der Parameter des RS-232-Terminal-Modus gesichert.

Einrichtung der DH-485-Kommunikation

Vor der Konfiguration der Kommunikationsparameter des Terminals muß als Kommunikationsverbund zunächst RS-232 oder DH-485 gewählt werden (die Option "General Purpose Communications" – Allzweckkommunikation ist nicht für den Einsatz mit dem BASIC-Modul vorgesehen). Hierzu rufen sie über das Menü "Setup and Configuration" das Menü "Terminal Selection" auf. Einzelheiten zur Auswahl des Terminals sind in Kapitel 3 erläutert.

Zur Konfiguration der Kommunikationsparameter für den jeweils gewählten Kommunikationsverbund rufen Sie über den folgenden Pfad das Menü "Terminal Setup" auf:

Path: Toplevel Menu → teRminal [DH-485] → Setup
Path: Main Menu → Other → teRminal [DH-485] → Setup


```

HELP<F1>|MENU<F2>|EXIT<F3>| COM1 19200 DH-485          |ANSI | 06-12-91  1:15pm
File Setup Display deBug Exit  Terminal Address:0 Basic Address:1 ↑↓→← to Edit
+-----SETUP-----+
|Attach                |=====+
|General (file names, etc.)... |
|Terminal...          |
|com Port settings (baud rate, etc.)... <AltP>|
|Who ACTIVE...        |
|wHo LISTEN           |
|Colors               |
|saVe setup           |
+-----Cancel<ESC>-----+
|
|
|
|
+-----+

```

Anschließen

Dieser Parameter dient dazu, im Falle eines Kommunikationsausfalls die Verbindung mit dem BASIC-Modul wieder herzustellen. Wenn diese Funktion aktiviert ist, versucht der PC, die Kommunikation mit dem BASIC-Modul an der konfigurierten Moduladresse des DH-485-Netzwerkes wieder herzustellen.

Allgemeine Einrichtungsparameter

Bei Wahl dieser Menüoption wird ein Dialogfeld mit den folgenden allgemeinen Einrichtungsparametern eingeblendet:

- **Upload path** — Definition des DOS-Pfades im PC, über den die vom BASIC-Modul hochgeladenen Files gespeichert werden.
- **Download path** — Definition des DOS-Pfades im PC, über den die an das BASIC-Modul heruntergeladenen Files gespeichert werden.
- **Log File** — Name des Protokollfiles. Vor dem Öffnen eines Protokollfiles kann dessen Name geändert werden.
- **Beep on downloads and uploads** — Tonsignal, das ertönt, wenn das Hoch- bzw. Herunterladen eines Files beendet ist oder abgebrochen wird.

- **word wrap in file viewer** — Die Wortumbruchfunktion beim Einsehen eines Files wird aktiviert.
- **right margin for file viewer** — Der rechte Rand für den Wortumbruch beim Einsehen eines Files.

Terminal

Durch Wahl der Option “Terminal” wird das Dialogfeld **TERMINAL SETUP** eingeblendet, in dem zwischen der ASCII- und ANSI-Terminalemulation gewählt werden kann.

Einstellung des COM-Ports

Mit dieser Menüoption wird das Dialogfeld **PORT SETUP** angezeigt, über das die Parameter des Kommunikationsports konfiguriert werden können. Die Vorgabeeinstellungen der BASIC-Entwicklungssoftware und des DH485-Ports des BASIC-Moduls sind wie folgt aufeinander abgestimmt:

Die folgenden Parameter stehen zur Verfügung:

- **active com port** — Wahl des PC-Ports, der für die Kommunikation mit dem BASIC-Modul verwendet werden soll. Wählen Sie zwischen COM1 und COM2.

Wichtig: Es muß sichergestellt werden, daß der für die Maus gewählte Port nicht mit dem für die serielle Kommunikation mit dem BASIC-Modul gewählten Port konfliktiert.

- **Baud rate** — Wahl der Baudrate. Verfügbare Geschwindigkeiten: 1200, 2400, 9600 und 19200.
- **Terminal Address** — Wahl der Terminaladresse im DH-485-Netzwerk.
- **Module Address** — Wahl der Adresse des BASIC-Moduls im DH-485-Netzwerk.
- **Maximum Node Address** — Wahl der maximalen Netzknotenadresse im DH-485-Verbund. Gültiger Bereich: 1 bis 31.

WHO ACTIVE (Aktive Netzknoten mit Terminal)

Auf dem Bildschirm WHO ACTIVE können die aktiven Netzknoten des DH-485-Netzwerks eingesehen werden. Der Personalcomputer (BASIC SOFT) ist in dieser Auflistung unter der Adresse aufgeführt, die für den Parameter des Kommunikationsports “Terminal Address” konfiguriert wurde.

```

HELP<F1>|MENU<F2>|EXIT<F3>| COM1 19200 DH-485          |ANSI | 06-12-91 1:15pm
File Setup Display deBug Exit Terminal Address:0 Basic Address:1 ↑↓→← to Edit
+-----SETUP-----+
|Attach |=====|
|General (file names, etc.)... | |
|-----WHO ACTIVE - Active Station Identification-----|
|   0 BASIC SOFT (31) 12 24 |
|   1 BASIC MOD (31) 13 25 |
|   2 5/02 (31) 14 26 |
|   3 15 27 |
|   4 16 28 |
|   5 17 29 |
|   6 18 30 |
|   7 19 31 |
|   8 20 |
|   9 21 |
|  10 22 |
|  11 23 |
|-----Port Settings<F2> Cancel<Esc> Help <F1>-----|
+-----+
    
```

WHO LISTEN (Aktive Netzknoten ohne Terminal)

Auf dem Bildschirm WHO LISTEN werden die aktiven Netzknoten angezeigt, ohne daß das Terminal in das Netzwerk eingebunden wird. Somit können Sie vor dem Wechseln in den Online-Betrieb sicherstellen, daß keine doppelten Netzknotenadressen vorhanden sind.

AB Spares

```
HELP<F1>|MENU<F2>|EXIT<F3>| COM1 19200 DH-485 |ANSI | 06-12-91 1:15pm
File Setup Display deBug Exit Terminal Address:0 Basic Address:1 ↑↓→← to Edit
+-----SETUP-----+
|Attach |=====|
|General (file names, etc.)... | |
|-----WHO LISTEN - Active Node Table-----|
| 0 12 24 |
| 1 * 13 25 |
| 2 * 14 26 |
| 3 15 27 |
| 4 16 28 |
| 5 17 29 |
| 6 18 30 |
| 7 19 31 |
| 8 20 |
| 9 21 |
| 10 22 |
| 11 23 |
|-----Port Settings<F2> Cancel<Esc> Help <F1>-----|
+-----+
```

Farben

Mit dieser Menüoption werden die Vordergrund- und Hintergrundfarben des Bildschirms eingestellt. Die Vordergrundfarbe wird mit den Pfeil-Auf- und Pfeil-Ab-Tasten und die Hintergrundfarbe mit den Pfeil-Links- und Pfeil-Rechts-Tasten eingestellt.

Sichern der Einstellung

Durch Wahl der Option **save setup** werden alle aktuellen Einstellungen der Parameter des DH-485-Terminal-Modus gesichert.

Bibliothek der BASIC-Makros

In diesem Anhang sind wichtige, generelle Informationen enthalten, die bei der Verwendung von Programmiermakros von Bedeutung sind und spezifische Informationen in vorausgehenden Kapiteln dieses Handbuchs ergänzen.

Die Themen:

- Bildschirmfunktionen
- Tastaturfunktionen
- Backplane-Funktionen
- Uhrfunktionen
- Batteriepufferung
- Programmsteuerung
- Zeichenkettenroutinen
- [Ctrl-C]-Funktionen
- Portsteuerungsfunktionen
- DH-485-Funktionen

Die Bibliothek im Überblick

In dieser Bibliothek ist eine Auflistung der Programmiermakros, d.h. Makroaufrufe enthalten, die bei der Programmierung mit der BASIC-Entwicklungssoftware verwendet werden können. Auch ist eine Beschreibung der Funktionsweise des Makroaufrufs sowie eine Auflistung der mit dem Makro verbundenen Ein- und Ausgaben enthalten.

Die Programmiermakros sind in Kapitel 6 dieses Handbuchs beschrieben. Diese Makros finden im Programmbeispiel in diesem Anhang (SECOND.BDL) Anwendung.

Wichtig: Der File INCLUDE.BDL ist Bestandteil der BASIC-Entwicklungssoftware und enthält alle in dieser Bibliothek aufgeführten BASIC-Programmiermakros sowie die zur Übersetzung erforderlichen Daten. Sehen Sie den Inhalt dieses Makrofiles nochmals durch, um sich mit seinen Funktionen und Interaktionen mit der jeweiligen Anwendung vertraut zu machen. Die Anweisung `{ $\$$ INCLUDE.BDL}` sollte stets am Beginn eines Programms stehen, in dem diese Programmiermakros angewandt werden sollen.

Bildschirmfunktionen

In der nachfolgenden Tabelle sind die Makroaufrufe beschrieben, die zur Durchführung von Funktionen auf dem Bedienerbildschirm verwendet werden.

Wichtig: Bitte beachten Sie, daß sich die Bildschirm-Makroaufrufe nur auf den derzeit konfigurierten Programmierport beziehen. Ein Versuch, die Ausgabe an einen anderen Port zu senden, wird bei der Programmübersetzung ignoriert.

Tabelle A.1
Funktionsmakros

Bildschirm-Makroaufrufe	Makro	Eingaben	Ausgaben
cls()	Löscht den Bildschirminhalt eines standardmäßigen ANSI-Terminals.	keine	keine
home()	Plaziert den Cursor in die linke obere Bildschirmcke eines standardmäßigen ANSI-Terminals.	keine	keine
clr_to_eol()	Löscht die Zeile von der Cursorposition bis zum Ende.	keine	keine
go_to_xy(%1,%2)	Positioniert den Cursor. Die Spalte %1 entspricht der X-Koordinate im Bereich 1 bis max_col. Die Reihe %2 entspricht der Y-Koordinate im Bereich 1 bis max_row. Max_row und max_col werden je nach Anzeigemodus geändert.	%1 = Spalte (0 ≤ col ≤ 99) %2 = Reihe (0 ≤ row ≤ 99)	keine
cursor_up(%1)	Bewegt den Cursor um %1 Reihen nach oben.	%1 = Anzahl der Reihen, um die der Cursor nach oben verschoben wird	keine
cursor_down(%1)	Bewegt den Cursor um %1 Reihen nach unten.	%1 = Anzahl der Reihen, um die der Cursor nach unten verschoben wird	keine
cursor_right(%1)	Bewegt den Cursor um %1 Spalten nach rechts.	%1 = Anzahl der Spalten, um die der Cursor nach rechts verschoben wird	keine
cursor_left(%1)	Bewegt den Cursor um %1 Spalten nach links.	%1 = Anzahl der Spalten, um die der Cursor nach links verschoben wird	keine
cursor_save()	Speichert die aktuelle Cursorposition.	keine	keine
cursor_restore()	Plaziert den Cursor auf der gespeicherten Position.	keine	keine
high_intensity()	Druckt Zeichen in verstärkter Darstellung.	keine	keine
low_intensity()	Druckt Zeichen in schwacher Darstellung.	keine	keine

Bildschirm-Makroaufrufe	Makro	Eingaben	Ausgaben
blink()	Druckt blinkende Zeichen.	keine	keine
reverse()	Druckt Zeichen in inverser Darstellung.	keine	keine
conceal_text()	Blendet die Textanzeige aus (keine Anzeige der Zeichen).	keine	keine
foreground_black()	Druckt Text mit schwarzem Vordergrund.	keine	keine
foreground_red()	Druckt Text mit rotem Vordergrund.	keine	keine
foreground_green()	Druckt Text mit grünem Vordergrund.	keine	keine
foreground_yellow()	Druckt Text mit gelbem Vordergrund.	keine	keine
foreground_blue()	Druckt Text mit blauem Vordergrund.	keine	keine
foreground_magenta()	Druckt Text mit magentarotem Vordergrund.	keine	keine
foreground_cyan()	Druckt Text mit türkisfarbenem Vordergrund.	keine	keine
foreground_white()	Druckt Text mit weißem Vordergrund.	keine	keine
background_black()	Druckt Text mit schwarzem Hintergrund.	keine	keine
background_red()	Druckt Text mit rotem Hintergrund.	keine	keine
background_green()	Druckt Text mit grünem Hintergrund.	keine	keine
background_yellow()	Druckt Text mit gelbem Hintergrund.	keine	keine
background_blue()	Druckt Text mit blauem Hintergrund.	keine	keine
background_magenta()	Druckt Text mit magentarotem Hintergrund.	keine	keine
background_cyan()	Druckt Text mit türkisfarbenem Hintergrund.	keine	keine
background_white()	Druckt Text mit weißem Hintergrund.	keine	keine
blink_off()	Verfügt über keine besonderen Attribute.	keine	keine
lines()	Zeigt dem Anwender den Aufbau einer komplexen Bildschirmanzeige.	keine	keine

Tastenfunktionen

In der folgenden Tabelle sind die Makroaufrufe zur Ausführung von Tastenfunktionen beschrieben.

Tabelle A.2
Tastentmakros

Tasten-Makroaufrufe	Makro	Eingaben	Ausgaben
hit_any_key()	Druckt die Meldung Hit any key to continue und wartet, bis eine Taste des angeschlossenen Programmiergerätes gedrückt wird.	keine	keine
get_key(%1)	Veranlaßt die Eingabe des nächsten Tastenanschlags und speichert dessen ASCII-Dezimalwert in %1.	keine	%1 = nächster Tastenanschlag
get1_key(%1)	Empfängt das nächste Zeichen von PRT1.	keine	%1 = nächstes Zeichen
get2_key(%1)	Empfängt das nächste Zeichen von PRT2.	keine	%1 = nächstes Zeichen

SLC-Backplane-Funktionen

In der folgenden Tabelle sind die Makroaufrufe für die Kommunikation mit der Backplane einer SLC 500-Steuerung sowie die Datenübertragung über diese Backplane beschrieben. Die Backplane-Makros des Moduls 1771-DB/B sind im File INCLUDBB.BDL enthalten.

Tabelle A.3
Kommunikationsmakros

Backplane-Makroaufrufe	Makro	Eingaben	Ausgaben
write_m1_image_unsigned(%1,%2)	Wandelt eine BASIC-Variable oder einen konstanten Wert in eine Ganzzahl ohne Vorzeichen (0 bis 65535) um und schreibt diesen Wert in den M1-File. Die Offset-Position (0 bis 63) im M1-File wird durch die erste Eingangsvariable definiert.	%1 = Offset-Position (0 bis 63) im M1-File %2 = umzuwandelnde BASIC-Variable oder Konstante	keine
read_m0_image_unsigned(%1,%2)	Liest eine Ganzzahl ohne Vorzeichen (0 bis 65535) aus dem M0-File. Die Offset-Position (0 bis 63) im M0-File wird durch die erste Eingangsvariable definiert. Die zweite Eingangsvariable empfängt die Daten.	%1 = umzuwandelnde Offset-Position (0 bis 63) im M0-File	%2 = Variable
write_m1_image_signed(%1,%2)	Wandelt eine BASIC-Variable oder einen konstanten Wert in einen Binärwert (-32768 bis 32767) um und schreibt das Ergebnis in den M1-File. Die Offset-Position (0 bis 63) im M1-File wird durch die erste Eingangsvariable definiert.	%1 = Offset-Position (0 bis 63) im M1-File %2 = umzuwandelnde BASIC-Variable oder Konstante	keine
read_m0_image_signed(%1,%2)	Liest einen Binärwert (-32768 bis 32767) aus dem M0-Abbildfile. Die Offset-Position (0 bis 63) im M0-File wird durch die erste Eingangsvariable definiert. Die zweite Eingangsvariable empfängt die Daten.	%1 = umzuwandelnde Offset-Position (0 bis 63) im M0-File	%2 = Variable
copy_image_to_m1(%1,%2)	Ermöglicht, daß der SLC-Prozessor die M1-Daten des BASIC-Moduls abliest.	%1 = Anzahl der zu kopierenden Worte	%2 = Erfolg der Übertragung 0 --> Run-Modus 1 --> Program-Modus 2 --> Test-Modus 10--> ungültige Länge 11--> nicht unterstützt
copy_bas_out_to_slc_in(%1)	Ermöglicht, daß der SLC-Prozessor die Ausgangsdaten des BASIC-Moduls abliest und in seiner Eingangsdatentafel ablegt.	keine	%1 = Erfolg der Übertragung 0 --> Run-Modus 1 --> nicht Run-Modus 73--> keine Antwort
copy_m0_to_image(%1,%2)	Ermöglicht, daß das BASIC-Modul die M0-Daten des SLC-Prozessors annimmt.	%1 = Anzahl der zu kopierenden Worte	%2 = Erfolg der Übertragung 0 --> Run-Modus 1 --> Program-Modus 2 --> Test-Modus 10--> ungültige Länge 11--> nicht unterstützt
copy_slc_out_to_bas_in(%1)	Ermöglicht, daß das BASIC-Modul die Ausgangsdatentafel des SLC-Prozessors abliest.	keine	%1 = Erfolg der Übertragung 0 --> Run-Modus 1 --> nicht Run-Modus 73--> keine Antwort
wait_m1()	Wartet, bis der M1-File gelesen wird.	keine	keine

Backplane-Makroaufrufe	Makro	Eingaben	Ausgaben
wait_m0()	Wartet, bis der M0-File aktualisiert wird.	keine	keine
wait_input_image(%1)	Wartet, bis der Eingangsabbildfile gelesen wird.	keine	%1 = Erfolg der Übertragung: 0 = noch nicht gelesen 1 = gelesen 2 = nicht unterstützt
wait_output_image(%1)	Wartet, bis der SLC-Ausgangsabbildfile aktualisiert wird.	keine	%1 = Erfolg der Übertragung: 0 = noch nicht gelesen 1 = gelesen 2 = nicht unterstützt

Uhrfunktionen

In der folgenden Tabelle sind die Makroaufrufe, die zum Einstellen und Lesen der Uhr verwendet werden, beschrieben.

Tabelle A.4
Uhrmakros

Uhr-Makroaufrufe	Makro	Eingaben	Ausgaben
input_time()	Fordert den Bediener auf, die korrekte Uhrzeit einzugeben, und stellt anschließend die Uhr. Keine Eingangs- und Ausgangsvariablen.	keine	keine
set_time(%1,%2,%3)	Stellt die Uhr auf die durch %1, %2 und %3 festgelegte Zeit ein.	%1 = Stunden (0 bis 23) %2 = Minuten (0 bis 59) %3 = Sekunden (0 bis 59)	keine
set_date(%1,%2,%3)	Stellt die Uhr auf das durch %1, %2 und %3 festgelegte Datum ein.	%1 = Tage (1 bis 31) %2 = Monate (1 bis 12) %3 = Jahre (0 bis 99)	keine
input_date()	Stellt das Datum der Systemuhr ein. Dieser Makro fordert den Bediener auf, das Datum einzugeben und stellt anschließend die Uhr ein.	keine	keine
set_day()	Fordert den Bediener auf, den Wochentag einzugeben und stellt anschließend den Wochentag der Systemuhr ein.	keine	keine
get_time(%1,%2,%3)	Liest die Uhrzeit ab und gibt sie in den Variablen %1, %2 und %3 zurück.	keine	%1 = Stunden %2 = Minuten %3 = Sekunden
print_time()	Liest die Uhrzeit und druckt diese aus. Der Cursor verweilt am Ende der Zeitanzeige (d.h., es findet kein Wagenrücklauf/Zeilenvorschub statt).	keine	keine
print_date()	Liest das Datum der Uhr ab und druckt dieses aus. Der Cursor verweilt am Ende der Datumsanzeige (d.h., es findet kein Wagenrücklauf/Zeilenvorschub statt).	keine	keine
print_day()	Liest den Wochentag der Uhr ab und druckt diesen aus. Der Cursor verweilt am Ende der Wochentaganzeige (d.h., es findet kein Wagenrücklauf/Zeilenvorschub statt).	keine	keine

Batteriepufferung

In der folgenden Tabelle sind die zur Steuerung der Batteriepufferung verwendeten Makroaufrufe beschrieben.

Tabelle A.5
Makros der Batteriepufferung

Makroaufrufe der Batteriepufferung	Makro	Eingaben	Ausgaben
battery_on()	Aktiviert die Batteriepufferung.	keine	keine
battery_off()	Deaktiviert die Batteriepufferung.	keine	keine
alloc_bat_var(%1)	Weist batteriegepufferten Speicherplatz für nichtflüchtige Daten zu. Ein Parameter ist erforderlich. %1 ist die Anzahl der zu speichernden Variablen. Wenn MTOP nicht mit 5FFFh spezifiziert wird, wird diese Funktion übersprungen.	%1 = Anzahl der Variablen	keine
read_bat_var(%1,%2)	Ruft die batteriegepufferten Variablen ab.	%2 = Speichernummer	%1 = abzurufende Variable
write_bat_var(%1,%2)	Speichert die batteriegepufferten Variablen.	%2 = Speichernummer	%1 = zu speichernde Variable

Programmsteuerung

In der folgenden Tabelle sind die zur Programmsteuerung verwendeten Makroaufrufe beschrieben.

Tabelle A.6
Programm-Makros

Makroaufrufe der Programmsteuerung	Makro	Eingaben	Ausgaben
CALL_ram()	Überträgt die Steuerung des BASIC-Programms auf die zweite Zeile des Programms im RAM-Speicher.	keine	keine
CALL_rom(%1)	Überträgt die Steuerung des BASIC-Programms auf die zweite Zeile des Programms im EEPROM-Speicher.	%1 = Nummer des ROM-Programms	keine
ret()	Überträgt die Steuerung wieder an die Stelle des Hauptprogramms, an der sie sich vor dem Aufruf befand.	keine	keine

Zeichenkettenroutinen

In der folgenden Tabelle sind die für Zeichenkettenroutinen verwendeten Makroaufrufe beschrieben.

Tabelle A.7
Makros der Zeichenkettenroutinen

Makroaufrufe der Zeichenkettenroutinen	Makro	Eingaben	Ausgaben
string_repeat(%1,%2)	Ermöglicht, daß ein Zeichen %1 mal wiederholt und in \$(%2) kopiert wird.	%1 = Anzahl der Zeichenwiederholung %2 = Nummer der Zeichenkette, die das Zeichen enthält	\$(%2) = die neue Zeichenkette
string_append(%1,%2)	Verknüpft zwei Zeichenketten (%1 und %2) und speichert das Ergebnis in %2.	%1 = Nummer der anzuhängenden Zeichenkette %2 = Nummer der Basiszeichenkette	\$(%2) = die verknüpfte Zeichenkette
num_to_str(%1,%2)	Konvertiert %1 von einem numerischen Wert in eine Zeichenkette und speichert das Ergebnis in \$(%2).	%1 = umzuwandelnder Wert %2 = Nummer der Basiszeichenkette	\$(%2) = Zeichenkette der konvertierten Daten
str_to_num(%1,%2,%3)	Konvertiert (%1) von einer Zeichenkette in einen numerischen Wert und speichert das Ergebnis in (%3). %2 ist ein Gültigkeitscode.	%1 = Nummer der umzuwandelnden Zeichenkette	%2 = Gültigkeit des Ausgangswertes: 0 = ungültiger Wert 1 ≤ gültiger Wert ≤ 255 %3 = konvertierter Wert
find_str_in_str(%1,%2,%3)	Sucht nach einer Zeichenkette innerhalb einer Zeichenkette und findet das erste Vorkommen (erste Position) der Zeichenkette.	%1 = zu suchende Zeichenkette %2 = zu durchsuchende Zeichenkette	%3 = Ergebnis 0 = Zeichenkette nicht gefunden 0 <> Zeichenkette gefunden

Makroaufrufe der Zeichenkettenroutinen	Makro	Eingaben	Ausgaben
replace_str_in_str(%1,%2,%3)	Ersetzt eine Zeichenkette innerhalb einer zweiten Zeichenkette.	%1 = Zeichenkette, die %2 ersetzt. %2 = zu ersetzende Zeichenkette. %3 = Nummer der Basiszeichenkette	\$(%1) = Basiszeichenkette
insert_str_in_str(%1,%2,%3)	Fügt eine Zeichenkette in eine zweite Zeichenkette ein.	%1 = Position, an der die Einfügung beginnt %2 = Anzahl der einzufügenden Zeichen %3 = Nummer der Basiszeichenkette	\$(%1) = Basiszeichenkette
delete_str_in_str(%1,%2,%3)	Löscht eine Zeichenkette aus einer Zeichenkette.	%1 = Nummer der Basiszeichenkette %2 = Nummer der zu löschenden Zeichenkette	\$(%1) = Basiszeichenkette
get_string_length(%1,%2)	Bestimmt die Länge einer Zeichenkette.	%1 = Nummer der Basiszeichenkette	%2 = Länge der Zeichenkette

[Ctrl C]-Funktionen

In der folgenden Tabelle sind die für [Ctrl C]-Funktionen verwendeten Makroaufrufe beschrieben.

Tabelle A.8
Makros der Control-C-Funktion

Makroaufrufe der Control-C-Funktion	Makro	Eingaben	Ausgaben
disable_c()	Deaktiviert die [CTRL C]-Funktion.	keine	keine
enable_c()	Aktiviert die [CTRL C]-Funktion.	keine	keine

Portsteuerungsfunktionen

In der folgenden Tabelle sind die zur Portsteuerung verwendeten Makros beschrieben.

Tabelle A.9
Portsteuerungsmakros

Makroaufrufe der Portsteuerung	Makro	Eingaben	Ausgaben
clear_prt1_buffers()	Löscht den Eingangs- und den Ausgangspuffer des PRT1-Ports.	keine	keine
clear_prt2_buffers()	Löscht den Eingangs- und den Ausgangspuffer des PRT2-Ports.	keine	keine

DH-485-Funktionen

In der folgenden Tabelle werden die während der Kommunikation über den DH485-Port verwendeten Makroaufrufe beschrieben.

Tabelle A.10
DH-485-Makros

DH-485-Makroaufrufe	Makro	Eingaben	Ausgaben
wait_common_input()	Veranlaßt, daß auf die Aktualisierung des gemeinsamen DH-485-Schnittstelleneingangsfiles gewartet wird.	keine	keine
copy_common_to_basic_input(%1,%2)	Kopiert den gemeinsamen DH-485-Schnittstelleneingangsfile in den Eingangspuffer des BASIC-Moduls.	%1 = File-Offset %2 = Anzahl der zu kopierenden Worte	keine

1771-DB/B-Backplane-Funktionen

Der File INCLUDBB.BDL ist in der BASIC-Entwicklungssoftware enthalten. Er enthält alle in dieser Bibliothek aufgeführten BASIC-Programmiermakros sowie die zur Übersetzung erforderlichen Daten. Sehen Sie den Inhalt dieses Makrofiles nochmals durch, um sich mit seinen Funktionen und Interaktionen mit der jeweiligen Anwendung vertraut zu machen. Die Anweisung `{ $I INCLUDBB.BDL }` sollte stets am Beginn eines Programms stehen, in dem diese Programmiermakros angewandt werden sollen. In dieser Bibliothek sind nur backplane-bezogene und die Zahlenumwandlung betreffende Makros des Moduls 1771-DB/B enthalten. Wenn allgemeine BASIC-Makros programmiert werden sollen, muß am Beginn des Programms ferner die Anweisung `{ $I INCLUDE.BDL }` stehen.

Tabelle A.11
Makros der 1771-DB/B-Backplane

Aufrufe der numerischen Umwandlung	Makro	Eingaben	Ausgaben
db_to_plc_16(%1,%2)	Wandelt einen BASIC-Fließkommawert in einen 16-Bit-Binärwert (PLC) ohne Vorzeichen (vierstelligen Hexadezimalwert) um.	%1=umzuwandelnder Wert %2=BTR-Speicherwort des Wertes	keine
db_to_plc_bcd_3(%1,%2)	Wandelt einen BASIC-Fließkommawert in einen dreistelligen BCD-Dezimalwert (PLC) mit Vorzeichen +/- XXX um.	%1=umzuwandelnder Wert %2=BTR-Speicherwort des Wertes	keine
db_to_plc_bcd_3_3(%1,%2)	Wandelt einen BASIC-Fließkommawert in einen 3.3-stelligen BCD-Dezimalwert (PLC) mit Vorzeichen +/- XXX.XXX um.	%1=umzuwandelnder Wert %2=BTR-Speicherwort des Wertes	keine
db_to_plc_bcd_4(%1,%2)	Wandelt einen BASIC-Fließkommawert in einen vierstelligen BCD-Wert (PLC) BCD XXXX um.	%1=umzuwandelnder Wert %2=BTR-Speicherwort des Wertes	keine
db_to_plc_bcd_6(%1,%2)	Wandelt einen BASIC-Fließkommawert in einen sechsstelligen BCD-Dezimalwert (PLC) mit Vorzeichen +/- XXXXXX um.	%1=umzuwandelnder Wert %2=BTR-Speicherwort des Wertes	keine

Aufrufe der numerischen Umwandlung	Makro	Eingaben	Ausgaben
db_to_plc_oct(%1,%2)	Wandelt einen BASIC-Fließkommawert in einen vierstelligen PLC-Oktalwert mit Vorzeichen +/-XXXX um.	%1=umzuwandelnder Wert %2=BTR-Speicherwort des Wertes	keine
db_to_plc5_float(%1,%2)	Wandelt einen BASIC-Fließkommawert in einen PLC-5-Fließkommawert um.	%2=BTR-Speicherwort des Wertes	keine
plc_16_to_db(%1,%2)	Wandelt einen 16-Bit-Binärwert (PLC) ohne Vorzeichen (vierstelligen Hexadezimalwert) in einen BASIC-Fließkommawert um.	%1=BTW-Wort, das den umzuwandelnden Wert enthält	%2=umgewandelter Wert
plc_bcd_3_to_db(%1,%2)	Wandelt einen dreistelligen BCD-Dezimalwert (PLC) mit Vorzeichen +/- XXX in einen BASIC-Fließkommawert um.	%1=BTW-Wort, das den umzuwandelnden Wert enthält	%2=umgewandelter Wert
plc_bcd_3_3_to_db(%1,%2)	Wandelt einen 3.3-stelligen BCD-Dezimalwert (PLC) mit Vorzeichen +/- XXX.XXX in einen BASIC-Fließkommawert um.	%1=BTW-Wort, das den umzuwandelnden Wert enthält	%2=umgewandelter Wert
plc_bcd_4_to_db(%1,%2)	Wandelt einen vierstelligen BCD-Wert (PLC) XXXX in einen BASIC-Fließkommawert um.	%1=BTW-Wort das den umzuwandelnden Wert enthält	%2=umgewandelter Wert
plc_bcd_6_to_db(%1,%2)	Wandelt einen sechsstelligen BCD-Dezimalwert (PLC) mit Vorzeichen +/- XXXXXX in einen BASIC-Fließkommawert um.	%1=BTW-Wort das den umzuwandelnden Wert enthält	%2=umgewandelter Wert
plc_oct_to_db(%1,%2)	Wandelt einen vierstelligen PLC-Oktalwert mit Vorzeichen +/-XXXX in einen BASIC-Fließkommawert um.	%1=BTW-Wort das den umzuwandelnden Wert enthält	%2=umgewandelter Wert
plc5_float_to_db(%1,%2)	Wandelt einen PLC-5-Fließkommawert in einen BASIC-Fließkommawert um.	%1=BTW-Wort das den umzuwandelnden Wert enthält	%2=umgewandelter Wert

Tabelle A.12
Aufrufe der 1771-DB/B-Backplane

Backplane-Aufrufe	Makro	Eingaben	Ausgaben
btr()	Führt einen Blocklesetransfer aus.	keine	keine
btw()	Führt einen Blockschreibtransfer aus.	keine	keine
btr_timed(%1)	Führt einen Blocklesetransfer mit einem Zeitablauf von 2 s aus.	keine	Status des Blocktransfers
btw_timed(%1)	Führt einen Blockschreibtransfer mit einem Zeitablauf von 2 s aus.	keine	Status des Blocktransfers
set_btr_length(%1)	Konfiguriert die Länge des Blocklesetransfers (1 bis 64).	%1=Länge des Blocklesetransferpuffers	keine
set_btw_length(%1)	Konfiguriert die Länge des Blockschreibtransfers (1 bis 64).	%1=Länge des Blockschreibtransferpuffers	keine

Die DOS-Verzeichnis-Shell

Dieser Anhang ergänzt Informationen über die DOS-Verzeichnis-Shell, die in vorausgehenden Kapiteln dieses Handbuchs enthalten sind.

Behandelte Themen:

- Shell-Funktion
- File-Funktionen von der DOS-Verzeichnis-Shell aus

Verwendung der DOS-Verzeichnis-Shell

Über die DOS-Verzeichnis-Shell wird ermöglicht, eine große Anzahl an DOS-bezogenen Aufgaben auszuführen, während die BASIC-Entwicklungssoftware abläuft. Es werden nicht nur DOS-Verzeichnisse angezeigt und Verzeichnisbäume erstellt, aus denen Files gewählt werden können, sondern es steht auch ein vollständig integrierter File/-Festplatten-Manager mit den folgenden Funktionen zur Verfügung:

- Anzeige von bis zu vier Verzeichnissen bei vollständiger oder komprimierter Verzeichnisanzeige
- Erstellung eines graphischen Verzeichnisbaums, durch den der Cursor bewegt und Verzeichnisse gewählt werden können
- Markierung mehrerer Files zum Kopieren, Löschen, Drucken und Laden oder Markierung einer Bedieneraufforderung, um einen DOS-Befehl bei allen markierten Files zu wiederholen. Beispielsweise können Sie mehrere .BDL-Files markieren, am Aufforderungszeichen den Kompilierbefehl eingeben und die [ENTER]-Taste drücken, um alle markierten Files zu kompilieren.
- Betrachten oder Ändern von Fileattributen
- Betrachten eines Files zur schnellen Durchsicht in einem Popup-Fenster (nur lesen)
- Sortieren von Verzeichnissen nach Name, Erweiterung, Größe, Datum und Uhrzeit (in steigender oder fallender Reihenfolge). Mehrfache Sortierfunktion werden unterstützt. Das sortierte Verzeichnis kann *nicht* auf der Festplatte gespeichert werden, sondern steht nur für die aktuelle Bearbeitung zur Verfügung. Beim Verlassen der BASIC-Entwicklungssoftware erscheint das Verzeichnis bei der nächsten Anzeige wieder wie vor dem Sortieren.
- Ausführung eines DOS-Befehls durch einen Tastenanschlag

- Überprüfung des Inhalts eines .ZIP- oder .ARC-Files durch einfaches Anwählen des Files
- Abarbeiten eines .EXE-, .COM- oder .BAT-Files lediglich durch Anwählen des Files

Funktionen der DOS-Verzeichnis-Shell

Der Zugriff auf die DOS-Verzeichnis-Shell erfolgt auf zweierlei Weisen. Ziehen Sie entweder vom Hauptmenü aus das File-Menü herunter und wählen anschließend "DOS directory shell" oder drücken Sie im Edit-Modus die Funktionstaste [Ctrl-F4]. Die DOS-Verzeichnis-Shell kann nicht vom Toplevel-Menü oder Terminal-Modus aus und nicht während der Verwendung der Shell-to-DOS-Funktion aufgerufen werden.

Path: Toplevel Menu → Main Menu → File → DOS directory shell

QuickStroke: Main Menu → [Ctrl F4]

Wichtig: Beachten Sie bitte, daß es sich bei Shell-to-DOS und der DOS-Verzeichnis-Shell um zwei verschiedene Funktionen handelt. Bei Shell-to-DOS verlassen Sie die BASIC-Entwicklungssoftware vorübergehend und kehren zur DOS-Ebene zurück, um von dort aus Programme abzuarbeiten oder DOS-Befehle auszuführen.

In der folgenden Abbildung ist die DOS-Verzeichnis-Shell mit dem Hauptmenü im Hintergrund dargestellt. Rechts befindet sich eine Auflistung der für bestimmte DOS-Befehle zu betätigenden Tasten. Links befindet sich eine Auflistung des derzeit angewählten DOS-Verzeichnisses (bei den hier aufgeführten Files und Verzeichnissen handelt es sich um Beispiele).


```

L:3   C:1   1747-PBASE V x.xx   [DOS Shell]   Ins 175k   06-12-91   1:15pm
File Window Block cUrsor Search Text Layout Print Macro Config Other Exit
"C:\ABBASIC\BAS\SECOND.BAS" loaded.           +-----1747-PBASE DOS Shell-----+
+-----E:\*. *-----+|Use cursor keys to move. |
|A: B: C: D: E: F: G: H: I: | |<TAB> Toggle display mode |
|E:4056K-4 files-----| |<Space> Mark/Unmark file |
| .. <DIR> 06-06-91 11:51a | |<CtrlBackSpace> Parent dir |
| HELP <DIR> 06-06-91 12:22a | |<F2> Change Directory |
| OE14214B 264096 06-06-91 11:53a | |<F3> Delete File(s) |
| BASE#1.BDL 139 05-28-91 02:44p | |<F4> Copy File(s) |
| BASE#2.BDL 240 05-29-91 02:16p | |<F5> Rename File(s) |
| DOSPROMP.BAT 480 04-16-91 11:14a | |<F6> Get New Dir Listing |
| EMPTY.BDS 0 06-06-91 12:21p | |<F7> Do any DOS Command |
| STATUS.BDS 1522 06-06-91 11:53a | |<F8> Print file with PRINT.COM |
| | |<F9> Load File into window |
| | |<F10> Sort directory |
| | |<ShftF3> Delete Marked files |
| | |<ShftF4> Copy Marked files |
| | |<ShftF8> Print Marked files |
| | |<ShftF9> Load Marked files |
| | |<ShftF10> Directory Tree |
| | |<AltF1> Create Window |
| | |<AltF2> Delete Window |
+-----Select<ENTER>=Done<ESC>-----+-----Press <F1> for more HELP====
1Help 2ChDir 3Delete 4Copy 5Rename 6Dir Of 7CMD 8Print 9Load 0Sort

```

Mit der Tastenkombination [**shift-F10**] wird ein Verzeichnisbaum erstellt, der die Verzeichnisliste ersetzt. Der Verzeichnisbaum zeigt alle Verzeichnisse und Unterverzeichnisse des aktuellen Laufwerks an. Mit der Einblendung der ersten Anzeige erscheint oben auf dem Bildschirm ein Wählerbalken, der innerhalb der Baumstruktur auf- und abwärts bewegt werden kann. Drücken Sie zur Wahl eines neuen Verzeichnisses die [**ENTER**]-Taste, wenn das gewünschte Verzeichnis hervorgehoben ist.

Mit der Tastenkombination [**Alt-F1**] kann ein neues Verzeichnisfenster erstellt werden. Auf dem Bildschirm können gleichzeitig bis zu vier Verzeichnisfenster angezeigt werden. Diese Fenster sind unabhängig voneinander und können unterschiedliche Verzeichnisse enthalten.

Die folgende Abbildung enthält ein Beispiel von zwei gleichzeitig auf dem Bildschirm angezeigten Verzeichnisfenstern. Das untere Fenster enthält lediglich eine Kurzliste, während das obere eine vollständige Fileauflistung mit allen Attributen darstellt.

```

L:3 C:1 1747-PBASE V x.xx [DOS Shell] Ins 175k 06-12-91 1:15pm
File Window Block Cursor Search Text Layout Print Macro Config Other Exit
+-----1747-PBASE DOS Shell-----+
+=====D:\SCREEN\*.*=====+|Use cursor keys to move. |
|A: B: C: D: E: F: G: H: I: |<TAB> Toggle display mode | |
|D:4982K-6 files-----|<Space> Mark/Unmark file |
| .. <DIR> 06-06-91 03:14p ||<CtrlBackSpace> Parent dir |
| MCOM 000 2150 06-08-91 03:13p A ||<F2> Change Directory |
| YPNLV 000 2146 06-06-91 06:58p A ||<F3> Delete File(s) |
| MCOM 001 2150 06-08-91 03:14p A ||<F4> Copy File(s) |
| YPNLV 001 2146 06-06-91 07:08p A ||<F5> Rename File(s) |
| MCOM 002 2146 06-08-91 03:20p A ||<F6> Get New Dir Listing |
| ||<F7> Do any DOS Command |
| +=====D:\ABBASIC\BDS\*.*=====+||<F8> Print file with PRINT.COM |
+=====|A: B: C: D: E: F: G: H: I: |<F9> Load File into window |
|D:4982K-67 files-----|<F10> Sort directory | | |
| | ..\ HELP\ ||<ShftF3> Delete Marked files |
| | 0E14214B 12072009 ||<ShftF4> Copy Marked files |
| | BASE#1 >BASE#2 ||<ShftF8> Print Marked files |
| | >BASE#1.BAS DOSPROMP.BAT ||<ShftF9> Load Marked files |
| | >BASE#1.BDL EMPTY.BDS ||<ShftF10> Directory Tree |
| | STATUS.BDS 232COM.DB ||<AltF1> Create Window |
| | KEYMAC.DB KEYMAP.DB ||<AltF2> Delete Window |
+=====+====Select<ENTER> Done<ESC>=====+====Press <F1> for more HELP=====
1Help 2ChDir 3Delete 4Copy 5Rename 6Dir Of 7CMD 8Print 9Load 0Sort

```

Es ist jeweils immer nur ein Verzeichnis aktiviert, wobei dieses durch den Wählerbalken gekennzeichnet ist (in der vorausgehenden Abbildung nicht sichtbar).

Verzeichnisfenster können in beliebiger Reihenfolge erstellt, neu dimensioniert, verschoben oder gelöscht werden. Weitere Hinweise sind in den Hilfetexten der DOS-Verzeichnis-Shell enthalten.

File-Funktionen der DOS-Verzeichnis-Shell

Einzelne Files können von der DOS-Verzeichnis-Shell aus kopiert, gelöscht, umbenannt, betrachtet, sortiert, gedruckt und in ein Fenster geladen werden. Einige dieser Funktionen können an mehreren Files gleichzeitig ausgeführt werden, sofern diese zuvor markiert wurden.

Zum Markieren von Files drücken Sie die [Leertaste], wenn der Filename hervorgehoben ist. Markierte Files sind durch das Zeichen ">" gekennzeichnet. In der vorausgehenden Abbildung mit mehreren Verzeichnissen sind einige Files des unteren Verzeichnisses markiert.

Verlassen Sie die DOS-Verzeichnis-Shell durch Wahl der Option Done<ESC> mit der Maus oder durch Drücken der [ESC]-Taste.

Symbols

Empty, [2-7](#)

Numbers

1747-PBASE-Übersetzer, [8-1](#)

1747-AIC isolierter Verbindungskoppler,
[1-3](#)

1747-PIC Schnittstellenwandler, [1-3](#)

3.5-Zoll-Disketten, [1-6](#)

5.25-Zoll-Disketten, [1-5](#)

A

Abbildungsfile, [8-1](#), [9-11](#)

Abkürzungen und Begriffe, [V-5](#)

Allen-Bradley, Kontaktaufnahme für
technische Unterstützung, [V-7](#)

Anmerkungen, [6-2](#)

Anschleiß-Parameter, [9-20](#)

Anschluß an das BASIC-Modul, [2-14](#)

ANSI-Terminal, [9-18](#), [9-21](#)

anwenderdefinierte Programme

Anfangsverzeichnis, [5-8](#)

Benennung, [5-5](#)

COMMAND.COM-Option, [5-8](#)

Speicheranforderungen, [5-8](#)

Speicherzuordnung, [5-6](#)

Überblick, [5-4](#)

Wahl der Speicherauslagerung, [5-6](#)

Wahl eines Filenamens, [5-6](#)

Anwendermenü, [5-3](#)

Anzeige

Länge und Breite, [3-3](#)

Optionen, [3-7](#)

Schneeunterdrückung, [3-3](#)

schwarzweiß, [3-3](#)

ASCII

Tabelle, [4-10](#)

Terminal, [9-18](#), [9-21](#)

ASCII-Tabelle, [4-10](#)

ASCII-Terminal, [9-18](#), [9-21](#)

Aufheben, [4-6](#)

Aufrufen von Programmen, [6-9](#)

Ausführung eines Programms, [6-16](#)

Ausschneiden und Einfügen, [4-6](#)

Autobaud, [9-17](#)

Autosave, [3-13](#)

B

Backplane-Aufrufe, 1771-DB\B, [A-9](#)

Backplane-Funktionen, 1771-DB\B, [A-9](#)

Backplane-Makros, 1771-DB\B, [A-9](#)

BASIC-Entwicklungssoftware

Leitfaden, [1-1](#)

Sprache, [6-1](#)

starten, [2-1](#)

BASIC-Entwicklungssprache

Erstellen eines Programms, [2-11](#),
[6-11](#)

Makros, [6-8](#)

Überblick, [2-11](#), [6-1](#)

Übersetzungsanweisungen, [6-2](#)

BASIC-Modul

Anschluß an das, [2-14](#)

Befehle, [9-2](#)

Debugging von Programmen, [9-11](#)

DH-485-Kommunikation, [9-19](#)

Dokumentationssatz, [V-3](#)

Herunterladen von Files, [9-3](#)

Herunterladen von hexadezimalen
Files, [9-8](#)

Hochladen von Files, [9-5](#)

Hochladen von hexadezimalen Files,
[9-7](#)

Kommunikation, [9-1](#)

PC-Schnittstelle, [1-3](#)

RS-232-Kommunikation, [9-16](#)

Sichern des Modulabbildes, [9-9](#)

Terminal-Modus, [9-1](#)

Wählen von Files, [9-17](#)

Wiederherstellen des Modulabbildes,
[9-10](#)

Baudrate

Autobaud, [9-17](#)

Einstellungsmöglichkeiten, [9-18](#),
[9-21](#)

Bearbeiten von Files, [5-1](#)

Befehlszeile, Übersetzer, [8-6](#)

Begriffe und Abkürzungen, [V-5](#)

Benennung eines Programms, [5-5](#)

Blockfunktionen, [4-4](#)

C

CALL 70, [6-10](#)

CALL 71, [6-10](#)
 COM-Port, Einstellung
 DH-485, [9-21](#)
 RS-232, [9-18](#)
 COMMAND.COM-Option, [5-8](#)
 Configuration-Menü, [3-2](#)
 Cursor
 Bewegung, [4-1](#)
 Positionsmarkierungen, [4-17](#)

D

Debugging
 Beispiel, [9-12](#)
 Korrekturen, [9-15](#)
 von Programmen, [9-11](#)
 Debugging-Anweisung, [6-7](#)
 Definitionen, [V-5](#)
 DH-485
 Anschließen, [9-20](#)
 COM-Port, Einstellung, [9-21](#)
 Kommunikation, Einrichtung, [9-19](#)
 Schnittstellenkarte, [3-3](#)
 WHO ACTIVE, [9-21](#)
 WHO LISTEN, [9-22](#)
 Dialogfeld, [V-5](#), [2-7](#)
 Dokumentationssatz, [V-3](#)
 DOS
 Funktionsmerkmale, [1-2](#)
 Verzeichnis-Shell, [B-1](#)
 Verzeichnisbaum, [B-3](#)
 Drucken
 eines Programms, [7-6](#)
 Hintergrund, [7-6](#)
 Überblick, [7-1](#)
 Drucker
 Einstellung, [7-1](#), [7-3](#)
 Gerät, [7-3](#)
 Optionen, [3-11](#)
 seriell, [7-4](#)
 serieller, [7-4](#)
 Steuerungen, [7-5](#)
 Wahl, [7-2](#)

E

Edit-Modus, [2-2](#), [4-1](#)
 Editieren eines Programms
 generelles Editieren, [2-11](#)
 Herunterladen, [2-15](#)
 Sichern, [2-12](#)
 Übersetzen, [2-13](#)
 Wechseln in den Terminal-Modus,
 [2-14](#)

Editieren von Programmen
 allgemeines, [6-11](#)
 Blockfunktionen, [4-4](#)
 Debugging, [9-11](#)
 einfache Methoden, [4-1](#)
 erweiterte Funktionen, [4-8](#)
 Herunterladen, [6-16](#), [9-3](#)
 Korrekturen, [9-15](#)
 Suchfunktion, [4-2](#)
 Übersetzen, [6-14](#), [8-1](#)
 Wechseln in den Terminal-Modus, [9-1](#)
 Editieroptionen, [3-4](#)
 EEPROM, [9-6](#)
 Einfügen, von Files, [5-2](#)
 Einfügen von Programmen, [6-9](#)
 Einfügen/Überschreiben, [3-5](#)
 EMS, [V-5](#), [1-2](#), [3-12](#), [5-7](#)
 Ersetzen, Funktionen, [4-2](#)
 Erstellen eines Programms, [2-11](#), [6-11](#)
 erweiterte Editierfunktionen
 ASCII-Tabelle, [4-10](#)
 Cursormarkierungen, [4-17](#)
 Fenster, [4-10](#)
 Linienzeichnen, [4-9](#)
 Taschenrechnerfunktion, [4-10](#)
 Tastenmakros, [4-12](#)
 Erweiterungen, Filenamen, [3-7](#)

F

Farben, Terminal-Modus,
 Bildschirmanzeigen, [9-19](#), [9-23](#)
 Fenster, [1-1](#), [2-8](#), [4-10](#)
 File laden, [3-4](#)
 File-Menü, [5-1](#)
 Filenamenerweiterungen, [3-7](#)
 Files
 Autosave, [3-13](#)
 einfügen, [5-2](#)
 herunterladen, [9-3](#)
 hexadezimale, [9-6](#)
 hochladen, [9-5](#)
 sichern, [3-13](#), [5-2](#)
 Wahl, [5-3](#)
 Zwischenfiles, [3-13](#)
 Fixpunkt-Anweisung, [6-6](#)
 Funktionstasten, [2-5](#)

H

Handbuch, Konventionen, [V-6](#)
 Handbücher, themenverwandte, [V-4](#)

Hardware, Anforderungen an die, [1-3](#)

Hauptmenü

- Fenster, [2-8](#)
- Funktionstasten, [2-5](#)
- Hilfe, [2-6](#), [2-7](#)
- Meldungszeile, [2-5](#)
- Pfad, [2-4](#)
- Statuszeile, [2-4](#)

Herunterladen

- eines Programms, [2-15](#), [6-16](#)
- vom Terminal-Modus aus, [9-3](#)
- von Files, [9-3](#)
- von hexadezimalen Files, [9-8](#)

hexadezimale Files

- herunterladen, [9-8](#)
- hochladen, [9-7](#)
- Überblick, [9-6](#)

Hilfe, [2-6](#), [2-7](#)

Hochladen

- von Files, [9-5](#)
- von hexadezimalen Files, [9-7](#)

Hypertext-Hilfe, [V-5](#), [1-2](#)

I

INCLUDE.BDL, [6-4](#), [6-8](#)

Include-Anweisung, [6-3](#)

Inhalt dieses Handbuchs, [V-2](#)

Inkrement-Anweisung, [6-6](#)

Insert-Modus, [3-6](#)

Installation der Software, [1-4](#)

K

Kommentare, [6-2](#)

Kommunikation mit dem BASIC-Modul,
[9-1](#)

Kompilieren, [8-1](#)

Konfiguration der Software

- Anzeigeoptionen, [3-7](#)
- Autosave, [3-13](#)
- Druckeroptionen, [3-11](#)
- Editieroptionen, [3-4](#)
- Mausoptionen, [3-7](#)
- Sichern, [3-15](#)
- Sicherungsfiles, [3-13](#)
- Speicheroptionen, [3-11](#)
- standardmäßige
 - Filenamenerweiterungen, [3-7](#)
- Suchen und Ersetzen, [3-14](#)
- Tastaturoptionen, [3-7](#)
- Terminal-Modus, [3-14](#)
- Zwischenfiles, [3-13](#)

Kontaktaufnahme mit Allen-Bradley für
technische Unterstützung, [V-7](#)

Konventionen in diesem Handbuch, [V-6](#)

Kopieren von Text, [4-6](#)

L

Label-Anweisung, [6-4](#)

Linienfunktion, Merkmale, [1-2](#)

Linienzeichen

- Anwendungen, [4-9](#)
- Beispiel, [6-13](#)

Literaturhinweis, [V-4](#)

M

Makroanweisung, [6-8](#)

Makros

- Batteriepufferung, [A-6](#)
- Bibliothek, [A-1](#)
- Ctrl C, [A-8](#)
- DH-485, [A-9](#)
- Funktion, [A-2](#)
- Kommunikation, [A-4](#)
- Portsteuerung, [A-8](#)
- Programm, [A-7](#)
- programmieren, [6-8](#), [6-10](#)
- Tastatur, [A-3](#)
- Tasten, [1-1](#)
- Tastenmakros, [4-12](#), [6-8](#)
- Überblick, [6-8](#)
- Uhr, [A-5](#)
- Zeichenkettenroutinen, [A-7](#)

Markierungen

- Cursorpositionen, [4-17](#)
- text, [4-5](#)

Maus

- Befehlszeile, Zusatz, [3-3](#)
- Optionen, [3-7](#)
- Treiber, [2-1](#)
- Unterstützung, [1-2](#)

Meldungszeile, [2-5](#)

Menüs

- Block, [4-4](#)
- File, [5-1](#)
- Hauptmenü, [2-4](#)
- Print, [7-1](#)
- Text, [4-6](#)
- Toplevel, [2-2](#)
- User, [5-5](#)

Modulabbild

- sichern, [9-9](#)
- wiederherstellen, [9-10](#)

O

Overwrite-Modus, [3-6](#)

P

produktspezifische Unterstützung, [V-7](#)

Programm

- drucken, [7-6](#)
- editieren, [2-11](#)
- herunterladen, [2-15](#)
- sichern, [2-12](#)
- übersetzen, [2-13](#)
- verlassen, [2-16](#)

Programme, anwenderdefinierte, [5-4](#)

Programmieren

- BASIC-Entwicklungssprache, [6-1](#)
- Debugging, [9-11](#)
- Debugging-Beispiel, [9-12](#)
- Editierfunktionen, [4-8](#)
- Editiermethoden, [4-1](#)
- eines zweiten Programms, [6-11](#)
- Korrekturen, [1-1](#), [9-15](#)
- Makros, [6-8](#)
- Schreiben von Programmen, [6-1](#)
- Sichern, [2-12](#)
- Suchfunktionen, [4-2](#)
- Übersetzen, [2-13](#), [8-1](#)
- Unterprogramm CALL 70 und CALL 71, [6-10](#)
- von Makros, [6-10](#)

Programmiermakros, [6-8](#), [6-10](#)

Publikationen, themenverwandte, [V-4](#)

Pulldown-Menü, [2-7](#)

Q

QuickStroke

- Tastenmakro, [4-16](#)
- Überblick, [V-6](#), [1-1](#)

R

RS-232

- allgemeine Einrichtungsparameter, [9-17](#)
- Autobaud, [9-17](#)
- COM-Port, Einstellung, [9-18](#)
- Kommunikationseinrichtung, [9-16](#)

Rückgängigmachen, [4-6](#)

S

Schreiben von Programmen, [6-1](#)

Setup and Configuration, Menü, [3-2](#)

Sichern

- des Modulabbildes, [9-9](#)
- eines Programms, [2-12](#)
- Files und die Autosave-Funktion, [3-13](#)
- Konfiguration, [3-15](#)
- von Files, [5-2](#)
- vor dem Übersetzen, [2-12](#), [8-6](#)

Sichern der Software, [1-7](#)

Sicherungsfiles, [3-13](#)

SLC 500 BASIC-Modul,
Dokumentationssatz, [V-3](#)

Software

- Disketteninhalt, [1-4](#)
- Installation, [1-4](#)
- 3.5-Zoll-Disketten, [1-6](#)
- 5.25-Zoll-Disketten, [1-5](#)
- Leitfaden, [1-1](#)
- sichern, [1-7](#)
- starten, [2-1](#)
- Verzeichnisstruktur, [1-7](#)

Spaltenblock verschieben, [3-6](#)

Speicher

- Anforderungen, [1-5](#)
- anwenderdefinierte Programme, [5-8](#)
- Auslagerung, [5-6](#)
- Befehlszeile, Zusatz, [3-4](#)
- erweiterter, [V-5](#)
- Optionen, [3-11](#)
- Übersetzer, [8-6](#)

Sprache, Typ, [3-9](#)

Standardeinstellungen

- Filenamenerweiterungen, [3-7](#)
- wieder herstellen, [3-15](#)

Starten der Software, [2-1](#)

Statuszeile, [2-4](#)

Störungssuche, Kontaktaufnahme mit
Allen-Bradley, [V-7](#)

Suchen

- eines Files, [4-4](#)
- Funktionen, [4-2](#)

Suchfunktion, Standardeinstellung, [3-14](#)

T

Tabulator-Option, [3-6](#)

Taschenrechnerfunktion, [4-10](#)

Tastatur

- Befehlszeile, Zusatz, [3-3](#)
- Optionen, [3-7](#)

Tastenmakros

- editieren, [4-15](#)
- Überblick, [1-1](#), [4-12](#), [6-8](#)

Terminal-Modus
Herunterladen, [9-3](#)
Hochladen, [9-5](#)
Wahl des, [3-14](#)
Wechseln in den, [2-14](#), [9-1](#)
Terminal, Typ, [9-18](#), [9-21](#)
Text sortieren, [4-7](#)
Text wiederholen, [4-8](#)
Text-Anweisung, [6-5](#)
Textformatierung, [4-6](#)
Toplevel-Menü, [2-2](#)
typisches Pulldown-Menü, [2-7](#)

U

Übersetzen
eines Programms, [2-13](#)
Fehler, [8-5](#)
Funktion, [8-1](#)
Konfiguration, [8-5](#)
von Programmen, [8-1](#)
Übersetzen/Kompilieren, [2-2](#)
Übersetzer, [8-1](#)
Konfiguration, [8-5](#)
Übersetzungsanweisungen
Debugging, [6-7](#)
Fixpunkt, [6-6](#)
Include, [6-3](#)
Inkrement-Anweisung, [6-6](#)
Kommentar, [6-2](#)
Label, [6-4](#)
Makro, [6-8](#)
Text, [6-5](#)
Überblick, [6-2](#)
Zeilennummer, [6-5](#)
User-Menü, [5-5](#)
UVPROM, [9-8](#)

V

Verknüpfen, von Programmen, [6-11](#)
Verlassen eines Programms, [2-16](#)
Verschieben von Text, [4-6](#)
Verzeichnisstruktur, [1-7](#)

W

Wahl von Projektfile-Optionen, [5-3](#)
WHO ACTIVE, [9-21](#)
WHO LISTEN, [9-22](#)
Wiederherstellen
Befehlszeile, Zusatz, [3-3](#)
des Modulabbildes, [9-10](#)
Wiederherstellen von Vorgabewerten,
[3-15](#)
Window-Menü, [2-8](#)

Z

Zeichnen von Linien, [4-9](#)
Zeilennummer, Aufruf, [4-18](#)
Zeilennummeranweisung, [6-5](#)
Zeit/Datum, [4-8](#)
Zusätze der Befehlszeile, [3-2](#)
zweites Programm
ausführen, [6-16](#)
BAS-Auflistung, [6-15](#)
BAS-Liste, [8-4](#)
BDL-Liste, [8-3](#)
eingeben, [6-11](#)
herunterladen, [6-16](#)
Überblick, [6-11](#)
übersetzen, [6-14](#)
Zwischenfiles, [3-13](#)



Allen-Bradley • Sprecher+Schuh

Allen-Bradley und Sprecher+Schuh helfen ihren Kunden seit mehr als 90 Jahren bei ihrer Produktivitätssteigerung und Qualitätsverbesserung. Wir entwickeln, produzieren und unterstützen weltweit ein umfassendes Sortiment von Steuerungs- und Automatisierungsprodukten. Dazu zählen speicherprogrammierbare Steuerungen, Niederspannungsgeräte, Antriebs- und Achssteuerungen, intelligente Bediengeräte (MMI), Sensoren und zahlreiche Softwareprodukte. Allen-Bradley und Sprecher+Schuh gehören zu Rockwell International, einem der führenden Technologieunternehmen der Welt.



Unsere Niederlassungen finden Sie an wichtigen Standorten weltweit.

Ägypten • Algerien • Argentinien • Australien • Bahrain • Belgien • Brasilien • Bulgarien • Chile • Costa Rica • Dänemark • Deutschland • Ecuador • El Salvador • Finnland • Frankreich • Griechenland • Guatemala • Honduras • Hongkong • Indien • Indonesien • Irland • Island • Israel • Italien • Jamaika • Japan • Jordanien • Jugoslawien • Kanada • Kolumbien • Korea • Kroatien • Kuwait • Libanon • Malaysia • Mexiko • Myanmar • Neuseeland • Niederlande • Norwegen • Oman • Österreich • Pakistan • Peru • Philippinen • Polen • Portugal • Puerto Rico • Qatar • Rumänien • Rußland - GUS • Saudi Arabien • Schweiz • Singapur • Slowakei • Slowenien • Spanien • Südafrikanische Republik • Taiwan • Thailand • Tschechische Republik • Türkei • Ungarn • Uruguay • USA • Venezuela • Vereinigte Arabische Emirate • Vereinigtes Königreich • Vietnam • Volksrepublik China • Zypern

Hauptverwaltung: Allen-Bradley, 1201 South Second Street, Milwaukee, WI 53204 USA. Tel: (1) 414 382 2000, Fax: (1) 414 382 4444

Hauptverwaltung Europa: Allen-Bradley • Sprecher+Schuh, Avenue Herrmann Debroux, 46, 1160 Brüssel, Belgien. Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40

Deutschland: Allen-Bradley • Sprecher+Schuh, Düsseldorf Straße 15, 42781 Haan-Grüten. Tel: (49) 2104 9600, Fax: (49) 2104 960121

Schweiz: Rockwell Automation AG, Gewerbepark, Hintermättlistrasse 3, 5506 Mägenwil, Tel: (41) 62 889 7777, Fax: (41) 62 889 7766

Österreich: Allen-Bradley • Sprecher+Schuh, Bäckermühlweg 1, 4030 Linz. Tel: (4370) (0732) 38 909 0, Fax: (04370) (0732) 38 909 61

Vertriebsbüros Deutschland-

Düsseldorf: Tel: (49) 211 748350, Fax: (49) 211 7483511
Frankfurt: Tel: (49) 6103 37970, Fax: (49) 6103 379710
Hannover: Tel: (49) 511 674020, Fax: (49) 511 6740222
Stuttgart: Tel: (49) 711 77790, Fax: (49) 711 7779101
Hamburg: Tel: (49) 40 770171, Fax: (49) 40 7658843
München: Tel: (49) 89 4274430, Fax: (49) 42744323
Berlin: Tel: (49) 30 8913013, Fax: (49) 30 8913042
Mittweida: Tel: (49) 37 2792221, Fax: (49) 37 2798985

Vertriebsbüros Österreich-

Graz: Tel: (43) (0) 316 9153190, Fax: (43) (0) 316 9153195
Innsbruck: Tel: (43) (0) 512 34 13 62, Fax: (43) (0) 512 39 13 62
Linz: Tel: (4370) (0732) 38 909 0, Fax: (4370) (0732) 38 909 61
Wien: Tel: (431) (0222) 6966060, Fax: (431) (0222) 1 69660660

Vertriebsbüros Schweiz -

Bulle: Tel: (41) 292 0264, Fax: (41) 292 0267
Mägenwil: Tel: (41) 62 889 7777, Fax: (41) 62 889 7766
Aarau: Tel: (41) 62 837 2222, Fax: (41) 62 837 2907
Bern: Tel: (41) 31 9929800, Fax: (41) 31 9929803
Lamone: Tel: (41) 91 604 6262, Fax: (41) 91 604 6264
Renes: Tel: (41) 21 6313232, Fax: (41) 21 6313231
Wil: Tel: (41) 71 929 9225, Fax: (41) 71 929 9266