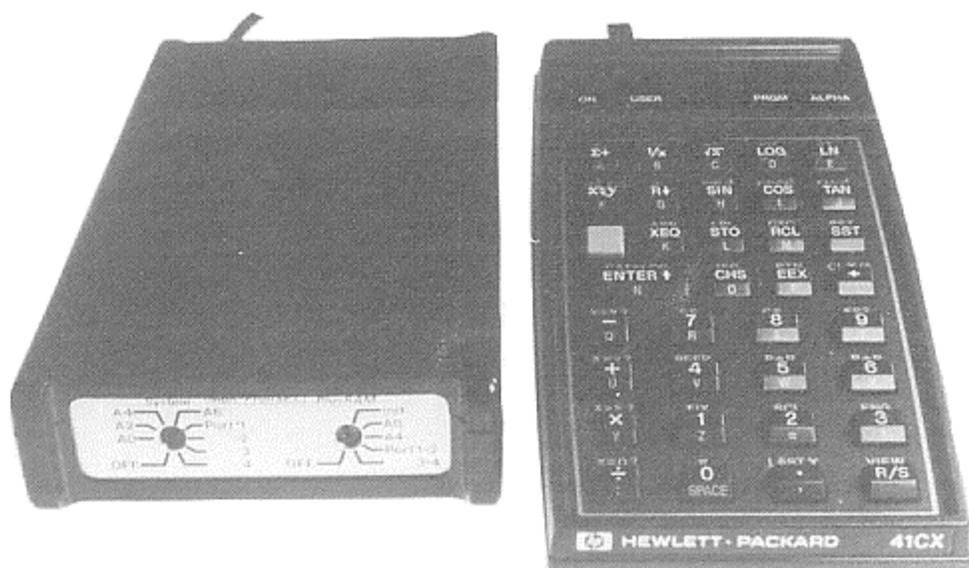


Mathematisches Büro Köln

MBK-ProfiSET

Benutzerhandbuch

für den 16K-RAM zum HP-41



Dieses Exemplar des Benutzerhandbuchs zum
MBK-ProfiSET besitzt die Nummer _____.
Das zugehörige ProfiSET trägt dieselbe INIT-Nummer.

Copyright 1984 by Mathematisches Büro Köln
Dipl.-Math. Achim B. Gemein

Verfasser: Dipl.-Ing. W. Maschke
Dipl.-Math. A.B. Gemein

printed in Germany

M B K - P R O F I S E T

BENUTZERHANDBUCH
ARBEITSANLEITUNG

DEZEMBER 1984

	Seite
EIN WORT VORAB	5
 <u>TEIL 1 : ARBEITEN MIT DEM PROFiSET</u>	
Kapitel 1 Einstellung des ProfiSETs	9
Kapitel 2 Die Basis-/Management-Funktionen	13
Kapitel 3 Die Informations-Funktionen	19
Kapitel 4 Anwenderprogramme im ProfiSET	23
Kapitel 5 Daten im ProfiSET	29
Kapitel 6 Die Massenspeicher-Funktionen	37
Kapitel 7 Die ProfiSET-PSUB-Funktionen	39
Kapitel 8 Programmierung mit ProfiSET-Funktionen	43
 <u>TEIL 2 : ARBEITEN MIT DEM MBK-EPROMER</u>	
Vorbemerkung	49
Kapitel 1 Handhabung des Programmiergeräts	51
Kapitel 2 Standardfälle der EPROM-Programmierung	53
Kapitel 3 Fortgeschrittene EPROM-Programmierung	55
 <u>TEIL 3 : ANHÄNGE</u>	
Anhang Ø Kurzbeschreibung der ProfiSET-Funktionen	65
Anhang 1 ID-Nummern bekannter Module/ROMs/EPROM-Sätze ..	69
Anhang 2 ProfiSET-Fehlermeldungen	71
Anhang 3 Aufbau des ROM-Bereichs beim HP-41	75
Anhang 4 Filetypen im ProfiSET	77
Anhang 5 Formatierung durch das ProfiSET	79
Anhang 6 Vereinbarkeit mit anderen Erweiterungen zum HP-41 ..	83
 Gewährleistung und Service	 85
Begriffserklärungen	87
Funktionsindex	91

Mit dem Kauf dieser 16KB-RAM-Erweiterung zum beliebten HP-41-System haben Sie einen wesentlichen Schritt zur intensiven weiteren Benutzung des Rechners getan. Viele Engpässe eines 'immer zu kleinen' Taschencomputers sind jetzt für geraume Zeit behoben, ohne daß Ihr System Einbußen in seiner Mobilität hinnehmen muß. Wir wünschen Ihnen viel Freude und ein zufriedenstellendes Arbeiten mit dem ProfISET.

Konzipiert wurde dieses Gerät vor allem für den beruflichen Einsatz, bei dem besonders die Zeitverzögerungen durch das Programmeladen vom Bandlaufwerk oder aus den X-Memory-Modulen als negativ empfunden wurden. Neben diesem mehr quantitativen Ziel wollen die Entwickler aber auch eine qualitative Komponente realisiert sehen: Die Software-Entwicklung auf dem HP-41 kann sich jetzt entscheidend verbessern; Unterroutinen - vor allem für einen sauberen Dialog und gute Outputgestaltung - brauchen nur ein einziges Mal vorhanden zu sein und können platz- und zeitsparend von allen Hauptprogrammen benutzt bzw. bei Software-Neuentwicklungen einfach programmiert werden. Auch wird jetzt mit Sicherheit manches Anwendungsgebiet behandelt werden, von dem bislang aus Kapazitätsgründen des Rechners gesagt wurde: 'Lohnt nicht, wird schwierig bei nur 2.233 Bytes!' und 'Wohin mit den vielen benötigten Daten?'

Software-Herstellern, die ihre Programme kommerziell vertreiben, bietet das ProfISET die Möglichkeit, umfangreiche, ein Arbeitsgebiet abdeckende Programmsammlungen zusammenzustellen. Ob diese den Anwendern im ProfISET oder in einer der MBK-EPROM-Boxen übergeben werden, muß im Einzelfall entschieden werden. Das ProfISET bietet für diese Zielgruppe auf jeden Fall mehr Vorteile als bisherige software-development-systems.

Das ProfISET verfügt über einige technische Leistungsmerkmale, die der normale Benutzer des HP-41 zunächst nicht oder nur geringfügig braucht. Ziel war es aber, von vornherein diese Möglichkeiten mitzugeben, damit sie eventuell später ohne großen Mehraufwand genutzt werden können. Erwähnt werden sollen an dieser Stelle lediglich die Anwendungen

- der EPROM-Programmierung für Software-Pakete
- der mobilen Datenerfassung und des Datentransfers über geeignete Schnittstellen
- der Maschinensprache-Programmierung

Das Mathematische Büro Köln wird Sie ggfs. bei der Realisierung derartiger Anwendungen unterstützen. Wir bitten Sie eindringlich, Anschlußversuche ohne genaue Kenntnisse über die an der Rückfront angebrachte Schnittstelle zu unterlassen; die CMOS-Bauteile im ProfISET und Ihr Rechner werden es Ihnen danken.

Der Platz im ProfISET und die neuen programmierbaren Funktionen erlauben es - auch im Zusammenspiel mit dem Digitalkassettenlaufwerk -, künftig deutlich schneller und komfortabler zu arbeiten; das gilt vor allem für die neuen Datenfile-Funktionen. Allerdings werden Sie bald merken, daß Sie vom 'Einzelprogramm-Denken' immer mehr zum 'Block-Denken' kommen - eine Folge des jetzt wesentlich größeren RAM-Bereichs.

TEIL 1
ARBEITEN MIT DEM PROF1SET

ACHTUNG! Umschaltungen am ProfiSET sollten Sie nur bei ausgeschaltetem Rechner durchführen.

Damit Sie Ihr HP-41-System optimal nutzen können, bietet das MBK-ProfiSET verschiedene Einstellmöglichkeiten. Da das ProfiSET ein ROM-Simulator ist (es werden HP-Einsteckmodule simuliert), belegt es 3 der 4 Steckplätze Ihres HP-41. Diese Belegung ist unabhängig davon, in welchen Port des HP-41 Sie den Stecker des ProfiSETs tatsächlich eingesteckt haben.

Die Einstellung des ProfiSETs kann an der Stellung der beiden Drehschalter auf der Frontplatte abgelesen werden (Abb. 1). Das Betriebssystem des ProfiSETs belegt einen Steckplatz (Port) und wird über den linken Schalter auf der Frontplatte, der mit 'System' bezeichnet ist, eingestellt. In Zusammenhang mit dem System ProfiSET sollten Sie nur die Einstellungen Port 1 bis 4 und OFF benutzen. Die Einstellungen A0 bis A6 sind für spätere Erweiterungen des ProfiSETs vorgesehen und können mit dem normalen ProfiSET-Betriebssystem zu Fehlfunktionen des Rechners führen.

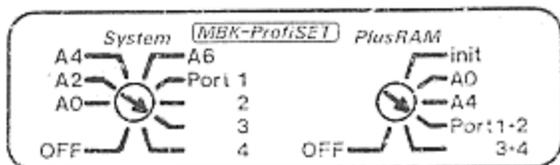


Abb. 1: Grundeinstellung ProfiSET (Frontplatte)

In den Stellungen Port 1 bis 4 belegt das Betriebssystem den jeweiligen Steckplatz; in der Stellung OFF ist das Betriebssystem des ProfiSETs abgeschaltet. In dieser Schalterstellung sind die Funktionen des Betriebssystems für den Rechner nicht vorhanden und können demzufolge auch nicht ausgeführt werden. Der rechte Schalter 'PlusRAM' dient zur Einstellung der beiden Steckplätze, die durch den PlusRAM belegt werden sollen. Hier dürfen vorläufig nur die Einstellungen Port 1+2, 3+4 und OFF genutzt werden (Ausnahme INIT, siehe Seite 18).

Die Abbildung 1 zeigt die Grundeinstellung des ProfiSETs. Das System befindet sich auf Steckplatz 3, der PlusRAM auf Steckplatz 1 und 2, Steckplatz 4 ist frei für ein beliebiges Erweiterungsmodul, den Kartenleser oder den Lesestift.

Wichtig! Das HP-IL-Modul, der Stecker des Einsteckdruckers HP 82143, das TIME-Modul und alle Arten von Memory-Modulen sind unabhängig vom Steckplatz und können somit auch auf die vom ProfiSET belegten Steckplätze gesteckt und gleichzeitig mit dem ProfiSET benutzt werden; d.h. konkret: PlusRAM geschaltet auf Port 1+2, Druckerstecker eingesteckt in Port 2, trotz Überschneidung stehen alle Druckerfunktionen zur Verfügung.

Auf der Rückseite des ProfiSETs befindet sich ein kleiner roter Schiebeshalter, der Prioritätsschalter. Dieser Schalter hat zwei Schaltstellungen: Ist der kleine Punkt sichtbar, hat das ProfiSET Priorität vor anderen HP-Geräten auf dem gleichen Steckplatz. Ist der Punkt nicht sichtbar, haben

EINSTELLUNG DES PROFISETS

Die Module und die HP-Peripherie stecken Sie dann wie folgt:

in Port 1	X-Functions-Modul (evtl. mit eingebautem TIME-Modul)
in Port 2	X-Memory-Modul (evtl. als Doppel-X-Memory)
in Port 4	Drucker bzw. IL-Modul

<i>X-Functions</i>	<i>X-Memory</i>
<i>ProfiSET</i>	<i>Drucker</i>

Beispiel 3

Ein HP-41CX mit CCD-Modul (8K-ROM) und Plotter-Modul (8K-ROM) wird mit weiterer IL-Peripherie (Plotter, Monitor, ThinkJet-Drucker, Cassettenlaufwerk) benötigt, um die im ProfiSET gesammelten umfangreichen Meßdaten mit den im PlusRAM befindlichen Verarbeitungsprogrammen im Dialog auszuwerten, darzustellen und auf Band zu sichern.

Es empfiehlt sich folgende Einstellung:

System	geschaltet auf	OFF	
PlusRAM	geschaltet auf	Port 1+2	eingesteckt in Port 1
CCD-Modul	eingesteckt in	Port 3	
Plotter-Modul	eingesteckt in	Port 4	
IL-Modul	eingesteckt in	Port 2	



<i>PlusRAM</i>	
FREI	FREI

<i>ProfiSET</i>	<i>IL</i>
<i>CCD</i>	<i>Plotter</i>

Der Benutzerspeicher des ProfiSETs, im folgenden PlusRAM genannt, hat eine Gesamtkapazität von 16384 Bytes. Aus Gründen, die im Betriebssystem des HP-41 liegen, wird dieser Speicherraum in 4 Teile zu je 4096 Bytes oder 4 KBytes aufgeteilt. Ein solcher Teil wird mit 'Block' bezeichnet; als Kürzel BL findet sich diese Bezeichnung in verschiedenen Funktionsnamen wieder. Die 4 Blöcke des PlusRAMs sind von 1 bis 4 durchnummeriert und erscheinen auch in dieser Reihenfolge im CAT2.

SLCT NEXTBL

Für die meisten Funktionen muß zunächst ein bestimmter Block angewählt werden. Zu diesem Zweck dient die Funktion SLCT, die die Eingabe der Blocknummer im X-Register erfordert, bevor sie ausgeführt wird:

1 SLCT

Wollen Sie den nächsten Block anwählen, können Sie dies auch mit der Funktion NEXTBL tun. Diese Funktion selektiert z.B. den Block 3, wenn vorher der Block 2 selektiert war. Der selektierte Block bleibt so lange angewählt, bis ein anderer Block selektiert wird. Das Entfernen des ProfiSETs vom Rechner oder ein MEMORY LOST im Rechner haben keinen Einfluß auf die Selektierung.

NEWBL

Bevor Sie nach der Selektierung in einem neuen (noch nicht formatierten) Block arbeiten können, muß dieser zunächst formatiert werden. Dazu müssen Sie entscheiden, wieviele Labels und Namen in dem Block vorkommen können. Die mögliche Anzahl der Labels liegt zwischen 1 und 64; jedes globale Label in einem Programm verbraucht - genau wie jeder Name oder jeder Datenfile - einen Eintrag. Wenn Sie im Umgang mit dem ProfiSET noch nicht geübt sind, wählen Sie am besten immer 64 Einträge. Zur Formatierung dient die Funktion NEWBL. Sie erfordert die Angabe der Anzahl der gewünschten Einträge in diesem Block im X-Register und formatiert den Block, indem sie den Raum für die Einträge (je Eintrag 2 Bytes) und für das Betriebssystem des HP-41 (16 Bytes) reserviert. NEWBL kann auch zum Löschen eines bereits belegten Blockes dienen.

64 NEWBL

IDBL

Des weiteren muß jeder Block eine Identität besitzen. Dies ist eine Zahl zwischen 1 und 31, bezeichnet mit ID. Die Identität darf dabei nur einmal in ihrer System-Konfiguration vorkommen und muß ebenfalls von Ihnen vergeben werden. Neben dem ProfiSET besitzen auch alle HP-Software- und Funktionsmodule sowie die ROMs der Peripheriegeräte eine Identität. Betrachten Sie bitte die Aufstellung im Anhang 1 über die ID-Nummern bekannter Module und Einheiten; kennzeichnen Sie die IDs der Module, die

DIE BASIS-/MANAGEMENT-FUNKTIONEN

Sie besitzen. Arbeiten Sie mit derartigen Modulen, können Sie deren Identitäten nie gleichzeitig in Ihrem ProfiSET verwenden; die anderen sind somit frei; Sie können sie für die einzelnen Blöcke des PlusRAMs vergeben. Für das ProfiSET als Peripherie zum HP-41 wurden die ID-Nummern 31 und 27 vergeben. Diese Identität können Sie also nicht mehr benutzen.

Für die ersten Übungsschritte geben Sie jetzt dem selektierten Block eine Identität; entfernen Sie ggfs. das Mathematik-Modul aus dem HP-41, da wir davon ausgehen, daß außer dem ProfiSET und eventuell einem Drucker nichts am Rechner angeschlossen ist.

1

IDBL

REVBL

Der Block 1 ist nun vollständig initialisiert, und er kann zum Arbeiten benutzt werden. Zur Unterscheidung verschiedener Versionen eines Blockes kann jederzeit noch ein normalerweise unsichtbarer Vermerk über die Funktion REVBL eingebracht werden. Die sogenannte ROM-Revision besteht aus maximal 4 Zeichen, benötigt jedoch keinen zusätzlichen Speicherraum in Ihrem PlusRAM. Üblicherweise wird die Revision in der Form AB1C angegeben, wobei AB ein Kürzel für den Namen des Blockes oder den Namen des Verfassers sein kann, 1 die Nummer des Blockes ist und C für die alphabetische Durchnummerierung der Revision steht. Die Revision des selektierten Blockes kann jederzeit eingetragen/überschrieben werden. Bei Aufruf der Funktion REVBL verlangt der Rechner Ihre Eingabe. Geben Sie mehr als 4 Zeichen ein, werden nur die ersten vier als Revision eingetragen.

Geben Sie jetzt zur Übung dem ersten Block die Revision TS1A .
REVBL "TS1A"

NAMEBL

Sie können jetzt oder später Ihren Block mit einem Namen versehen. Ein Blockname ist eine Zeichenkette mit 8 bis 11 Zeichen und dient zur Gliederung der Filenamen in Ihrem PlusRAM. Jeder Name verbraucht einen Eintrag im CAT2 und zusätzlich n+1 Bytes, wobei 'n' die Anzahl der Zeichen des Namens ist. Es können auch mehrere Namen in einem Block vorkommen; der erste Eintrag sollte jeweils an der ersten Stelle im CAT2 des selektierten Blockes erfolgen. Geben Sie daher zunächst einen 'Blindnamen' ein, sollte Ihnen bei der Blockeinrichtung noch nicht der endgültige Name eingefallen sein. Den Namen müssen Sie in das ALPHA-Register schreiben, wobei auf die richtige Anzahl der Zeichen zu achten ist. Alle im Display darstellbaren Zeichen außer '.', ',', ' ' und ':' sind zulässig. Anschließend wird die Funktion NAMEBL ausgeführt.

Geben Sie dem Block den Namen TEST-BLOCK1 :

TEST-BLOCK1
NAMEBL

Im Kapitel 8 finden Sie ein Programm, das im Dialog hilft, einen Block komplett einzurichten.

DIE BASIS-/MANAGEMENT-FUNKTIONEN

CLRN

Zum Löschen eines Namens dient die Funktion CLRN, die ebenfalls zuerst die Eingabe des Blocknamens in das ALPHA-Register erfordert. Diese Funktion ist unabhängig von der Selektierung und löscht immer den ersten Namen, der mit der Eingabe im ALPHA-Register übereinstimmt. Ist der gelöschte Name nicht der letzte Eintrag in einem Block, so wird an die Stelle des Namens ein Platzhalter gesetzt. Dieser Platzhalter gibt Ihnen die Möglichkeit, an die gleiche Stelle einen anderen Namen zu setzen, da bei einem Neueintrag immer zunächst die Platzhalter aufgefüllt werden, bevor neue Einträge am Ende angefügt werden.

Geben Sie zunächst einen weiteren Namen ins ALPHA-Reg. und führen Sie NAMEBL aus:
Das gleiche mit dem Namen TESTNAME-2:

```
TESTNAME-1      NAMEBL
TESTNAME-2      NAMEBL
CAT 2
```

Im CAT2 erscheinen neben den anderen Einträgen die Blocknamen:

```

:
TEST-BLOCK1
TESTNAME-1
TESTNAME-2
```

Schreiben Sie TESTNAME-1 ins ALPHA-Reg. und führen Sie CLRN aus:

```
TESTNAME-1
      CLRN
```

Führen Sie CAT2 aus und betrachten Sie die Einträge im Block 1. An Stelle des gelöschten Namens ist die Revision als Platzhalter getreten; hier würde ein neuer Name eingetragen werden können.

```

:
TEST-BLOCK1
TS1A
TESTNAME-2
```

Wenn Sie auch TESTNAME-2 löschen (letzter Eintrag), stehen keine Platzhalter mehr im CAT2.

```
TESTNAME-2
      CLRN
CAT 2
:
TEST-BLOCK1
```

CLEAR

Zum Löschen aller anderen Files im ProfiSET - gleichgültig ob Programme, Datenfiles oder Funktionen - dient die Funktion CLEAR. Sie fordert zur Eingabe des Filenamens auf und arbeitet wie CLRN, d.h. sie ist unabhängig von der Selektierung und löscht immer den ersten File im PlusRAM mit dem angegebenen Namen. Auch CLEAR setzt Platzhalter in den CAT2 ein. Programme können auch mit CLEAR alpha alpha (ohne

DIE BASIS-/MANAGEMENT-FUNKTIONEN

Eingabe eines Namens) gelöscht werden, wenn der Rechner auf das Programm im PlusRAM positioniert ist. Wird das Programm, auf das der Rechner gerade positioniert ist, mit CLEAR gelöscht, so wird der Rechner automatisch auf den Anfang des ersten Programmes im Hauptspeicher gesetzt. Besitzt ein Programm oder eine Funktion mehrere Namen, so kann jeder dieser Namen zur Eingabe benutzt werden; es wird immer das gesamte Programm oder die gesamte Funktion gelöscht. Beispiele für die Funktion CLEAR finden sich im Kapitel 'Programme im Profiset'.

PACKBL

Nach einem Löschen im PlusRAM ergibt sich eine Lücke, die unabhängig vom CAT2 wieder gefüllt wird, sobald ein File kleinerer oder gleicher Größe geladen wird. Auch ein dann noch verbleibender Platz wird auf die gleiche Weise genutzt; es ist jedoch wahrscheinlich, daß sich nach wiederholtem Laden und Löschen mehrere kleine Lücken in einem Block befinden. Erhält man vom Betriebssystem des Profisets beim Laden die Fehlermeldung ERR'NO ROOM, so kann es durchaus sein, daß durch Zusammenschieben der einzelnen Files eine ausreichende Lücke zum Laden entsteht. Zu diesem Zweck dient die Funktion PACKBL, die bei einem normal gefüllten Block ca. 20 Sekunden dauert und keine weiteren Eingaben erfordert; sie wirkt nur auf den selektierten Block.

PACKC

Mit der Funktion PACKC kann der CAT2 für den selektierten Block gepackt werden. Diese Funktion beseitigt alle durch Löschen entstandenen Platzhalter aus dem CAT2 und schiebt die übrigen Einträge zusammen; sie ändert jedoch nichts an der mit NEWBL vorgewählten Anzahl der Einträge, schafft also keinen zusätzlichen Speicherraum für das Laden von Files.

Achtung: Da sich die XROM-Nummern der einzelnen Files nach PACKC ändern können, sollte die Funktion nur im Ausnahmefall verwendet werden. Eine ausführliche Erklärung zu den XROM-Nummern befindet sich im Kapitel 'Programme im Profiset' und sollte vor der Ausführung von PACKC unbedingt gelesen werden.

COPYBL

Ganze 4KByte-Blöcke können mit der Funktion COPYBL in ca. 5,3 Sekunden in den selektierten PlusRAM-Block kopiert werden. Die Funktion COPYBL erfordert die Eingabe der Adresse des Quell-Blockes ins X-Register. Diese Adresse ist nicht identisch mit den Blocknummern im PlusRAM und kann der nachstehenden Tabelle entnommen werden; im Anhang 3 finden Sie eine vollständige Übersicht über die Adreßbereiche (Adressen) des HP-41.

DIE BASIS-/MANAGEMENT-FUNKTIONEN

<u>Adresse</u>	<u>Inhalt bei Grundeinstellung des ProfiSETs</u>
8	Port 1, 1. Hälfte (PlusRAM Block 1)
9	Port 1, 2. Hälfte (PlusRAM Block 2)
10	Port 2, 1. Hälfte (PlusRAM Block 3)
11	Port 2, 2. Hälfte (PlusRAM Block 4)
12	Port 3, 1. Hälfte (ProfiSET-Betriebssystem Teil 1)
13	Port 3, 2. Hälfte (ProfiSET-Betriebssystem Teil 2)
14	Port 4, 1. Hälfte
15	Port 4, 2. Hälfte

Achtung: Das Kopieren der Inhalte der Adressen 8 bis 15 ist mit dem normalen ProfiSET-Betriebssystem nicht erlaubt, da dies unweigerlich zu Fehlfunktionen des Rechners führt.

Soll ein Software-Modul bei der Grundeinstellung des ProfiSETs (wie in der Tabelle in Klammern angegeben) in das ProfiSET kopiert werden - z.B. des Mathematik-Modul -, so ist wie folgt vorzugehen:

Schalten Sie den Rechner aus und stecken Sie das zu kopierende Modul in den freien Port; in der Grundeinstellung ist das Port 4:

Selektieren Sie den Block, in den kopiert werden soll; geben Sie die Adresse ein, auf der das Modul steckt (Port 4, 1. Hälfte). Führen Sie COPYBL aus; entfernen Sie das Modul; seine Software ist im ProfiSET (CAT2!).	2 SLCT 14 COPYBL
---	---------------------------

Sie können auch einen PlusRAM-Block in einen anderen kopieren (sichern); z.B. 3 SLCT 9 COPYBL kopiert in der Grundeinstellung des ProfiSETs den zweiten PlusRAM-Block in den dritten. Da ein nicht selektierter Block total übernommen und das Selektierungskennzeichen im Zielblock überschrieben wurde erscheint der Hinweis: ERR'NO SLCT. Selektieren Sie noch einmal Ihren Block, damit Sie im PlusRAM weiter arbeiten können.

Es wird eindringlich darauf hingewiesen, daß insbesondere für die Funktion COYPBL das Urheberrecht zu beachten ist. Sie sollten in Zweifelsfragen sich vom jeweiligen Hersteller (Autor) der Software die Erlaubnis einholen, den 'Datenträger' für die Programme 'wechseln' zu dürfen.

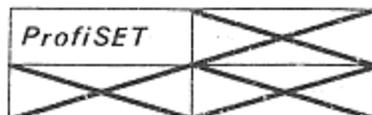
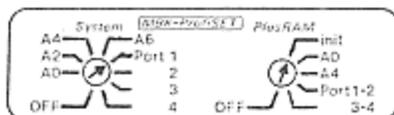
Es können auch Kopien von Blöcken mit PRIVATE-geschützten Programmen gemacht werden; der PRIVATE-Schutz bleibt jedoch auch in der Kopie erhalten. Bitte wundern Sie sich aber nicht, wenn Sie in Zukunft geschützte Blöcke immer seltener kopieren können (Meldung: ERR'NO COPY); dann wurden diese nach dem verbesserten Standard des MBK zum Software-Schutz erstellt.

DIE BASIS-/MANAGEMENT-FUNKTIONEN

INIT

Sollten Sie trotz Warnung einen der Blöcke 0 bis 7 kopiert haben, so kann Ihnen die Funktion INIT möglicherweise weiterhelfen. Die Funktion INIT löscht gleichzeitig alle PlusRAM-Blöcke des ProfiSETS. Zur Ausführung der Funktion INIT bei Funktionsstörungen des Rechners in Zusammenhang mit dem Inhalt des PlusRAMs gehen Sie wie folgt vor:

1. Entfernen Sie alle Geräte und Software-Module vom Rechner.
2. Schalten Sie den System-Schalter am ProfiSET auf Port 1.
3. Schalten Sie den PlusRAM-Schalter am ProfiSET auf INIT.



4. Prüfen Sie, ob der Rechner wieder auf die Tastatur reagiert. Falls dies nicht der Fall ist, gehen Sie nach dem Abschnitt 'Wartung' im Handbuch zum HP-41 vor, bis der Rechner wieder betriebsbereit ist.
5. Führen Sie die Funktion INIT aus. Die Funktion INIT verlangt die Eingabe einer 3-stelligen Nummer. Diese ist geräteabhängig. Sie sollten sich Ihre Nummer merken. Wird eine andere Nummer eingegeben, erhalten Sie DATA ERROR (ein kleiner Schutz vor unbefugter schneller Totallöschung).
6. Schalten Sie den Rechner aus; schalten Sie das ProfiSET wieder auf die normale Einstellung.

SUMBL

Die Funktion SUMBL dient zur Bildung einer Prüfziffer, die im selektierten Block ohne zusätzlichen Speicherbedarf abgelegt wird. Diese Prüfziffer wird dann gebraucht, wenn der Inhalt des PlusRAMs in EPROM-Boxen vervielfältigt werden soll. Sie sollte erst dann gebildet werden, wenn alle Arbeiten im Block erledigt sind; sie wird wie Prüfziffern in HP-Modulen berechnet und eingetragen. Nach Bildung der Prüfziffer muß Sorge getragen werden, daß Platzhalter im CAT2 nicht als Funktion aufgerufen werden. Beim Sichern von 4K-Blöcken auf Magnetband wird die Funktion SUMBL automatisch durchgeführt.

BLCAT

Die wichtigste Informations-Funktion des ProfiSETs ist der Block-Catalog (BLCAT). Er zeigt in einer Übersicht den Inhalt des jeweils selektierten Blockes. In der Kopfzeile sind folgende Angaben enthalten:

BL: Nummer des selektierten Blockes
 ID: die gewählte Identität
 ADR: die Adresse des Blockes

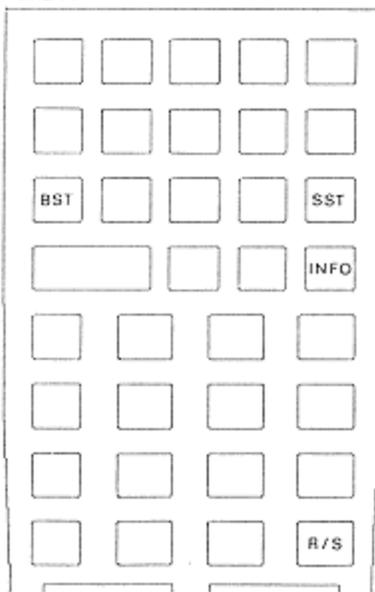
```
=====
BL:04 -- 10:12 -- ADR:11
PROFI-BLOCK
-----
```

Die Zeile danach enthält den ersten Eintrag im CAT2; dort sollte ein Blockname stehen; ist kein Name vergeben worden, wird der als erstes eingetragene Filename (LBL) ausgedruckt.

```
NAME      NR TYP BYTES
PROFI-BLOCK 00 N 0012
LADEP+    01 P 0031
CLEAR+    02 >LADEP+
AUTOBL    03 P 0377
??        04 >AUTOBL
VERS-1    05 DES 1574
KEYVERS   06 KEY 0092
=====
```

Darunter werden die einzelnen Files aufgezählt mit Namen, Nummer (Nachkommateil der XROM-Nummer), Typ (siehe auch Anhang 4) und Länge in Bytes. Typ und Länge werden jeweils nur beim ersten Eintrag eines Files angegeben; bei weiteren Einträgen wird auf den ersten Eintrag durch einen Pfeil und seinen Namen hingewiesen. Während der Funktion BLCAT wird ein spezielles Tastenfeld aktiv:

BREAK



Taste	Bedeutung
ON	Abbruch der BLCAT-Funktion (BREAK)
R/S	Stop und Start des Block-Catalogs, Aktivierung von BST, SST und INFO
SHIFT	rückwärts Einzelschritt (BST) mit Wiederholung
SST	vorwärts Einzelschritt (SST) mit Wiederholung
	INFO (nur aktiv, wenn kein Drucker vorhanden) zeigt Nummer, Typ und Länge an, solange Taste betätigt wird.

Als Länge eines Files wird immer der im PlusRAM benötigte Raum angegeben, für ein Programm also 2 Bytes mehr als im Hauptspeicher des Rechners. Die Funktion BLCAT hinterläßt die letzte Angabe im ALPHA-Reg.

DIE INFORMATIONEN-FUNKTIONEN

BLCATX TYP? LENG?

Einzelne Angaben aus dem Block-Catalog können abgefragt werden mit den Funktionen BLCATX, TYP? und LENG?. Die Funktion BLCATX benötigt die Eingabe der Nummer der Funktion (zweiter Teil der XROM-Nummer) in das X-Register. Ist eine Funktion, ein Blockname oder ein Programm mit dieser Nummer im selektierten Block vorhanden, so wird das ALPHA-Register mit diesem Filenamen überschrieben. Ist der File nicht vorhanden, bleibt das ALPHA-Register unverändert und das X-Register wird mit der Zahl -1 überschrieben. Die Funktionen TYP? und LENG? wirken nicht nur auf den selektierten Block, sondern auf den gesamten CAT2-Bereich des Rechners und fordern bei manuellem Aufruf zur Eingabe des Namens des Programmes oder der Funktion auf. Namenseinträge (Typ N) können nicht verarbeitet werden. TYP? schreibt die entsprechende Zeichenkette in das X-Register (siehe Anhang 4) und LENG? die Länge des Files in Bytes.

BLF? CATS? CATF?

Eine weitere Gruppe der Informations-Funktionen bilden BLF?, CATS? und CATF?. Sie benötigen keine Angabe und beziehen sich immer auf den selektierten Block. Das jeweilige Ergebnis wird in das X-Register geschrieben. BLF? gibt an, wieviele Bytes im Block frei sind. Dabei erfolgt diese Angabe jeweils über die größte Lücke, falls mehrere Lücken im Block vorhanden sind. Der freie Platz kann sich nach dem Packen mit PACKBL ändern. CATS? gibt die Anzahl der möglichen Einträge im Block wieder, die mit NEWBL vorgewählt wurde; CATF? die Anzahl der Einträge, die noch frei sind.

ADR>ID ID>ADR

Die Funktionen ADR>ID und ID>ADR sind unabhängig vom PlusRAM und der Selektierung. ADR>ID erfordert die Eingabe einer Adresse (siehe Anhang 3) ins X-Register und überschreibt dieses mit der Identität auf dieser Adresse. ID>ADR ist die Umkehrung dieser Funktion und gibt bei Angabe einer Identität die zugehörige Adresse wieder. Einer der beiden Blöcke des Profiset-Betriebssystems hat die Identität 31. Wenn Sie nun wissen möchten, auf welcher Adresse sich dieser Block befindet, führen Sie folgende Übung durch:

31 Eingabe der Identität des PSET-BASE-Blockes

ID>ADR Ausführung der Funktion; das Ergebnis steht im X-Register

Falls Sie noch die Grundeinstellung des Profisets haben, müssen Sie die Zahl 13 sehen; der Adreßbereich gehört zum Port 3 (vergl. Anhang 3). Stecken Sie irgendein Modul in einen Port und lassen sich mit der Funktion ADR>ID seine Identität anzeigen.

DIE INFORMATIONEN-FUNKTIONEN

BL? BLA? BL1A? ID? REV?

Die Funktionen BL? , BLA? , BL1A? , ID? und REV? erfordern keine Eingabe. BL? schreibt die Nummer des selektierten Blockes in das X-Register, BLA? die Adresse des selektierten Blockes und BL1A? die Adresse des ersten PlusRAM-Blockes. Mit ID? erhalten Sie die mit IDBL vergebene Identität des selektierten Blockes. REV? schreibt die mit REVBL vergebene Revision in das ALPHA-Register.

CHKBL

Die Funktion CHKBL dient zur Überprüfung der mit SUMBL gebildeten Prüfziffer. Bitte beachten Sie, daß sich die Prüfziffer mit jedem Laden, Löschen und Packen ändert und somit evtl. nur kurze Zeit nach SUMBL aktuell sein kann. CHKBL führt zur Fehlermeldung ERR'PZ , falls die Prüfziffer nicht mehr stimmt.

SRV

Falls Sie wissen möchten, wann die Batterien in Ihrem Profiset gewechselt werden müssen, rufen sie die Funktion SRV auf. Sie zeigt das Wechseldatum MM.JJ für Monat und Jahr bei normalem Gebrauch des Profisets an. Sie sollten spätestens in diesem Monat wegen des Batteriewechsels mit dem Mathematischen Büro Köln Kontakt aufnehmen.

Vorbemerkung

Der Hauptspeicher des Rechners (in diesem Handbuch auch mit CAT1-RAM bezeichnet) ist der einzige Bereich, in dem Programme editiert (geschrieben), gespeichert und 'gefährdet' werden können. Von hier können Programme auch auf andere Speicherformen übertragen werden, z.B. auf Magnetkarten, Kassette, Diskette, X-Memory, PlusRAM, Steckmodule, EPROMs etc. Diese unterscheiden sich voneinander nicht nur in Preis und Gesamtkapazität, sondern vor allem in ihrer Benutzerfreundlichkeit, will man ein Programm ablaufen lassen. Damit ist nicht nur die mehr oder weniger lange Zugriffs- und Ladezeit gemeint, sondern auch das gesamte Drumherum beim 'handling' wie: Suchen der richtigen Magnet-Karten/-Bänder, Einlesen, vollständiges oder teilweises Löschen des Hauptspeichers, Size-/Statuseinstellung, Key-Assignments, Überprüfen, ob alle benötigten Unterprogramme/Routinen geladen sind. Wünschenswert ist natürlich mit XEQ... der Direktzugriff und der sofortige Programmstart, wenn das Programm in einer der o.g. Speicherformen resident ist. Möglich ist das aber nur bei den Programmspeichern: PlusRAM, Modulen, EPROMs in HP-EPROM-Boxen. Dennoch gilt auch für diese 'Bequem-Speicher':

erst:	Programm editieren	im CAT1-RAM
dann:	Programm speichern	möglichst einfach, schnell und preiswert
dann:	Programm ablaufen lassen	ohne Zurückladen / Zeitverlust
evtl. später:	Programm pflegen/verändern	möglichst sofort, wenn erforderlich; ohne große Kosten

Für Ihr Profiset-System und den darin enthaltenen PlusRAM-Bereich gilt also auch:

Step 1:	Programme werden im Hauptspeicher geschrieben oder in den Hauptspeicher eingelesen.
Step 2:	Dann werden sie in den PlusRAM mit den Möglichkeiten des Direktzugriffs und der sofortigen Lauffähigkeit geladen (abgelegt/gespeichert).

Bei notwendigen Änderungen von Programmen, die sich im PlusRAM befinden, gehen Sie in folgenden Schritten vor:

1. Holen Sie mit COPY das Programm in den Hauptspeicher des Rechners.
2. Editieren Sie Ihre Programmänderungen.
3. Testen Sie die neue Version Ihres Programms.
(XEQ Label greift auf die CAT1-Version zu.)
4. Selektieren Sie den Block mit der Alt-Version.
5. Löschen Sie mit CLEAR .
6. Versuchen Sie, die neue Version an die alte Stelle zu laden. Ist der Platz nicht mehr ausreichend, müssen Sie in einen anderen Block laden, falls nach PACKBL ein erneuter Ladevorgang nicht durchführbar ist.

ANWENDERPROGRAMME IM PROFISSET

Das Laden eines ProgrammsLADEP

Voraussetzung zum Laden eines Programms ist, daß es im Hauptspeicher des Rechners steht und mindestens ein globales Label enthält. Mit der Funktion LADEP kann grundsätzlich jedes Programm aus dem HP-41 in den PlusRAM des ProfISets geladen werden, also auch Programme mit synthetischen Befehlen mit der Einschränkung, daß diese keine Textketten mit mehr als 9 Nullbytes enthalten.

Nach dem Aufruf der Funktion LADEP werden Sie zur Eingabe des globalen Labels aufgefordert. Steht der Rechner auf dem zu ladenden Programm, was unmittelbar nach Einlesen eines Magnetkartenprogramms oder vom Band der Fall ist, dann brauchen Sie nicht mehr das globale Label einzutippen. In diesem Fall reicht es aus, wenn Sie nur alpha alpha drücken.

Haben Sie in Ihrem Programm lokale XEQ- oder GTO-Befehle (z.B. XEQ 55 oder GTO A) programmiert, müssen diese Labels auch existieren, da sonst der Ladevorgang mit der Fehlermeldung ERR'NO COMP< abgebrochen wird. Lesen Sie bitte jetzt im Anhang 'Fehlermeldungen des ProfISets', was dann zu tun ist.

Kurzform-GTO-Befehle (das sind GTO 00 bis GTO 14) können im PlusRAM sogar Sprungweiten von 128 Bytes im Gegensatz zu nur 112 Bytes im CAT1-RAM des Rechners 'vormerken'; Kurzform-GTOs springen während des Programmlaufs direkt ohne Suchen des entsprechenden Labels und beschleunigen die Durchlaufzeit. Das Vormerken der Sprungweiten geschieht normalerweise beim ersten Programmlauf im Hauptspeicher des Rechners. Wegen der erweiterten Möglichkeiten des PlusRAMs werden bei jedem Programm - ob compiliert oder nicht - die Sprungweiten während des Ladevorgangs berechnet und bei dem jeweiligen Label eingetragen; währenddessen steht in der Anzeige KURZFORM-LBL und LANGFORM-LBL.

Globale GTO-Befehle (z.B. GTO ABCD) werden im PlusRAM wie im Hauptspeicher gehandhabt. Wissenswert ist dabei, daß der Rechner bei einem Befehl GTO ABCD immer zunächst seinen CAT1-RAM - beginnend mit dem letzten Label (also entgegengesetzt zum Erscheinen im Catalog 1) - und erst danach den PlusRAM bzw. die eingesteckten Software-Module in der Reihenfolge des Catalogs 2 nach dem Label ABCD durchsucht. Analoges gilt für XEQ 'globales Label', sofern dies nicht in einen XROM-Befehl umgewandelt wurde (siehe unten). Sie können Suchzeiten optimieren, wenn Sie häufig angesprungene Labels 'vorne' in Ihren PlusRAM laden. Bedenken Sie, daß bei einem 'prall gefüllten' System durchaus 250 und mehr Labels überprüft werden müssen.

Globales XEQ und XROM(Wichtig zu lesen!)

Zum Unterprogrammaufruf gibt es neben dem globalen XEQ-Befehl noch eine zweite Möglichkeit: den XROM-Befehl. Der Unterschied zwischen den beiden Befehlen wird am besten an einem kurzen Beispiel deutlich. Schreiben Sie dazu in den Hauptspeicher Ihres Rechners das folgende kurze Programm:

ANWENDERPROGRAMME IM PROF1SET

```

                                GTO ..
PACKING
01*LBL *TEST1*
02 TONE 9
03 RTN
04*LBL *TEST2*
05 TONE 5

```

Verlassen Sie den PRGM-Modus und führen Sie aus:

```

2
      SLCT
64
      NEWBL
2
      IDBL

```

Laden Sie jetzt durch XEQ LADEP das Programm in den PlusRAM:

```
LADEP *TEST1*
```

Programmieren Sie ein weiteres Programm:

```

                                GTO ..
PACKING
01*LBL *TEST3*
02 XEQ *TEST1*
03 XEQ *TEST2*

```

Löschen Sie das Programm TEST1 aus dem Hauptspeicher des Rechners.

```

      CLP *TEST1*
PACKING

```

Gehen Sie jetzt in das Programm TEST3 und versuchen Sie, die Zeilen 02 und 03 des Programms TEST3 als Zeilen 04 und 05 erneut einzugeben.

```

                                GTO .003
04 XROM *TEST1*
05 XROM *TEST2*

```

Die Eingabe XEQ ALPHA TEST1 ALPHA hat sich automatisch in den Befehl XROM TEST1 umgewandelt, da das durchzuführende Programm im PlusRAM gefunden wurde und im Hauptspeicher des Rechners nicht mehr vorhanden ist. Lassen Sie nun das Programm TEST3 ablaufen. Sie hören die Tonfolge TONE 9 TONE 5 TONE 9 TONE 5; beide Befehle, XEQ und XROM haben offensichtlich den gleichen Zweck erfüllt, sie haben die Programme TEST1 und TEST2 als Unterprogramm aufgerufen. Wer ein gutes Gehör hat, hat sicherlich bemerkt, daß der Abstand zwischen dem ersten und zweiten Ton bedeutend länger ist als zwischen den übrigen Tönen. (Wenn nicht, programmieren Sie noch als Zeile 06 ein GTO TEST3 und rufen das Programm erneut auf.) Ein XROM-Befehl ist also schneller als ein XEQ-Befehl.

ANWENDERPROGRAMME IM PROFISSET

Schalten Sie nun den Rechner aus und entfernen Sie den Anschlußstecker des ProfISETs vom Rechner. Betrachten Sie erneut das Programm TEST3:

```
01*LBL *TEST3*
02 XEQ *TEST1*
03 XEQ *TEST2*
04 XROM 02,00
05 XROM 02,01
06 END
```

Aus den Befehlen XROM*TEST1 und XROM*TEST2 sind die Befehle XROM 02,00 und XROM 02,01 geworden. Der Rechner merkt sich also beim XROM-Befehl nicht den Namen des Programms, sondern eine Nummer. Schließen Sie nun das ProfISET wieder an den Rechner an und führen Sie die Funktion BLCAT aus.

```
=====
BL:02 -- ID:02 -- ADR:09
TEST 1
=====
NAME      NR TYP BYTES
TEST1     00 P 0020
TEST2     01 >TEST1
=====
```

Der Block, in dem das Programm TEST1 steht, hat die Identität 02 (ID:02). Dies ist auch gleichzeitig der erste Teil (Vorkomma-Teil) der XROM-Nummer. Die Labels TEST1 und TEST2 haben die laufenden Nummern 00 und 01 in diesem Block. Ein XROM-Aufruf hat immer die Form XROM ID,NR .

Nun soll das Programm TEST1 geändert werden:

```
COPY *TEST1*      Kopieren Sie das Programm TEST1 mit der
GTO .003          Funktion COPY .
04*LBL *TEST4*    Fügen Sie nach der Zeile 03 das Label TEST4
05 BEEP           und den Befehl BEEP ein.

01*LBL *TEST1*    Betrachten Sie zur Kontrolle des neue Programm
TONE 9 RTN       TEST1.

04*LBL *TEST4*
BEEP

06*LBL *TEST2*
TONE 5 .END.

CLEAR *TEST1*    Löschen Sie das Programm TEST1 aus dem
LADEP *TEST1*    PlusRAM und laden Sie an dessen Stelle das
BLCAT            neue Programm TEST1 . Überprüfen Sie den
=====          Blockinhalt.
BL:02 -- ID:02 -- ADR:09
TEST 1
=====
NAME      NR TYP BYTES
TEST1     00 P 0030
TEST4     01 >TEST1
TEST2     02 >TEST1
=====
```

Das Label TEST4 steht nun zwischen TEST1 und TEST2 , das Label TEST2 hat nun die Nr. 02.

ANWENDERPROGRAMME IM PROF1SET

Das Label TEST4 hat also nun die XROM-Nummer XROM 02,01, die vorher noch zum Label TEST2 gehörte. Die XROM-Nummer von TEST2 hat sich verändert.

Wenn wir noch einmal unser Programm TEST3 betrachten, stellen wir eine nicht gewollte Veränderung fest:

```
01*LBL *TEST3*
02 XEQ *TEST1*
03 XEQ *TEST2*
04 XROM *TEST1*
05 XROM *TEST4*
06 END
```

Die Zeile 05, in der das Programm TEST2 aufgerufen werden sollte, hat sich geändert, weil sich auch die XROM-Nummern im PlusRAM geändert haben.

Der Unterschied zwischen XEQ- und XROM-Befehl besteht also darin, daß der XEQ-Befehl immer nach einem Namen sucht, während der XROM-Befehl die Identität des Blocks und die Nummer des Labels im Block beinhaltet. Ein XROM-Befehl sollte nur dann benutzt werden, wenn sichergestellt ist, daß sich die XROM-Nummer des aufgerufenen Labels nicht mehr ändert. Der XROM-Befehl ist jedoch schneller und verbraucht nur 2 Bytes im Gegensatz zum XEQ-Befehl, der je nach Länge des Namens 3 bis 9 Bytes verbraucht. Entfernen Sie jeweils den Anschlußstecker des Prof1SETS vom Rechner, wenn Sie beim Programmieren bestimmte XROM-Befehle vermeiden wollen, d.h. wenn Sie ein XEQ mit einem Label programmieren, das sich schon im PlusRAM befindet, von dem aber der endgültige Platz ID,NR noch nicht festliegt. Sie können auch später noch XEQ-Befehle in XROM-Befehle mit den Funktionen XROMP und XROMBL umwandeln.

```
XROMP    XROMBL
```

Die Funktion XROMP wandelt alle XEQ-Befehle in einem bestimmten Programm in XROM-Befehle um. Sie verlangt die Eingabe des Labels des Programms. Sollten bestimmte XEQ-Befehle nicht umgewandelt werden, muß das zugehörige Label als Einzeilenprogramm vor dem Aufruf von XROMP in den Hauptspeicher des Rechners geschrieben werden.

Die Funktion XROMBL arbeitet wie XROMP; sie führt die Umwandlung jedoch in allen Programmen des selektierten Blockes durch.

Bitte machen Sie sich noch einmal klar, was es heißt, ein Unterprogramm über seine XROM-Nummer aufzurufen. Beachten Sie die Auswirkungen, wenn Sie dieses Unterprogramm löschen und an eine andere Stelle des PlusRAMs schreiben. Im Zweifelsfall müssen Sie alle möglichen Aufrufprogramme durchgehen und überall den Aufruf XEQ... neu programmieren, sonst wird zwar schnell, aber falsch die Nummer angesprungen, durchgeführt und wahrscheinlich mit nicht beabsichtigten Ergebnissen zurückgesprungen.

Weitere Funktionen für Anwenderprogramme

PRVP PRVBL

Sie können Ihre Programme im PlusRAM auch PRIVATE-schützen, falls sie gegen Zugriff gesichert werden sollen. Es existiert jedoch keine Möglichkeit, ein einmal geschütztes Programm wieder in ein normales Programm zurückzuverwandeln. Ein PRIVATE-geschütztes Programm im Profiset kann weder mit COPY in den Hauptspeicher des Rechners zurückkopiert noch aufgelistet oder betrachtet werden. Mit CLEAR läßt es sich allerdings löschen.

Die Funktion PRVP versieht ein einzelnes Programm mit dem PRIVATE-Schutz. Die Funktion PRVBL versieht alle Programme im selektierten Block mit diesem Schutz. Sie benötigt zusätzlich 14 Bytes am Ende des Blocks. Beachten Sie dies, wenn Sie viele Programme zu einem 'vollen' Block zusammenstellen wollen.

Anmerkung: Falls Sie in einen Block die Autostart-Option geschrieben haben (vgl. Kapitel 7) oder aber ihn schon mit PRVBL geschützt haben, erhalten Sie bei Durchführung von PRVBL die Meldung ERR'NO ROOM , selbst wenn noch über BLF? viele freie Bytes angezeigt werden.

Vorbemerkung

Das ProfiSET ermöglicht Ihnen, neben Programmen und Funktionen auch Daten zu speichern. Das Datenfile-Konzept im ProfiSET weicht bewußt von dem normalen Registeraufbau im HP-41 ab, da Ihnen bei der Verwendung des ProfiSETs der gesamte Hauptspeicher des Rechners bereits für Daten zur Verfügung steht und mit SIZE 319 für sehr viele Anwendungen ausreichen dürfte. Wenn auf das ProfiSET als Datenspeicher zurückgegriffen wird, ist davon auszugehen, daß große Datenmengen programmgesteuert erfaßt, verarbeitet oder gespeichert werden müssen. Ein platzsparender Aufbau der Datenfiles ist daher neben einer einfachen Handhabung beim Abrufen der Daten sinnvoll.

Anlegen eines Datenfiles mit DATI. DATE. DATEN

Datenfiles müssen wie im X-Memory einen Filename besitzen, der 1 bis 7 Zeichen lang sein darf. Aus Gründen einer gewissen Programmierbequemlichkeit sollten Sie aber maximal 6 Zeichen wählen. Dann lassen sich solche Namen in ein Register des HP-41 ablegen (ASTO..) und nach CLA mit ARCL.. bequem in das ALPHA-Register zurückrufen. Auch wird dadurch der indirekte Aufruf eines Datenfiles möglich, beispielsweise durch XEQ IND Y mit dem Filename im Y-Register. Spätere Beispiele werden dies verdeutlichen.

Datenfiles im ProfiSET können bis zu 999 Werte in ihre Register aufnehmen; eine Ausnahme bilden die Datenfiles, in die Zahlen mit maximaler Genauigkeit, d.h. mit zehnstelliger Mantisse bei zweistelligem Exponenten, geschrieben werden sollen. Hier lassen sich maximal 801 Werte in einem Block unterbringen (bei 801 kann der Filename nur 5 Zeichen umfassen). Ein Datenfile muß in den selektierten Block - genauso wie ein Programm - hineinpassen; er kann nicht in den nächsten Block 'hineinragen'. In einem Block können aber mehrere Datenfiles gleichen oder unterschiedlichen Typs stehen. Es können nur numerische Werte, also keine Textketten, gespeichert werden. Dafür ist das X-Functions-Modul mit seinen ASCII-File-Möglichkeiten (insbesondere beim CX) geeignet: Hilfsweise lassen sich Textketten auch in den Registern des HP-41 (Size 319 ist möglich) unterbringen. Schreiben Sie ins X-Register, wieviel Zahlen Sie in einen Datenfile aufnehmen wollen, bevor Sie ihn mit einer Funktion anlegen. Entsprechend viele Register - beginnend mit der Reg.-Nr. 0 - werden vorgesehen.

Sowohl in technischen Anwendungen als auch im kaufmännischen Bereich läßt sich häufig abschätzen, welches die größte vorkommende Zahl ist oder welche Genauigkeit (Nachkomma-Stellen) zur weiteren Berechnung ausreicht. Dementsprechend wählt man den geeigneten Typ über die jeweilige Funktion zur Erzeugung des Datenfiles aus, um die Werte Bytes-sparend abzulegen. Aus der nachstehenden Übersicht gehen die verschiedenen Typen (Genauigkeiten der Zahlen) und deren Platzverbrauch hervor. Dazu ist wissenswert, daß im Hauptspeicher des Rechners Register eine zehnstellige Mantisse und einen zweistelligen Exponenten der Zahl enthalten, die dort gespeichert ist. Es können also im HP-41 Zahlen in einer Größenordnung von $9,999999999 \times 10^{99}$ bis $-9,999999999 \times 10^{99}$ dargestellt werden. Die Funktion DATEN erreicht die gleiche Genauigkeit; es werden

DATEN IM PROFISSET

allerdings nur 5 Bytes je Register im PlusRAM statt 7 Bytes im Hauptspeicher verbraucht, also bei größeren Datenmengen fast 30 % weniger an Platz. Mit der Funktion DAT12 erzeugte Datenfiles haben eine Genauigkeit von zwei Stellen je Register, d.h. es können Werte von 99 bis -99 abgelegt werden, wobei jedes Register nur ein Byte benötigt.

- Zusammenfassung:
- Ins X-Register wird die Anzahl der insgesamt abzulegenden Werte (max. 999) geschrieben;
 - mit XEQ DAT1. oder DATE. wird der Datenfile angelegt und für einen speziellen Zahlentyp (Genauigkeit) spezifiziert;
 - dabei wird bei manuellem Aufruf nach dem Filenamen gepromptet; programmgesteuert ist der Name des zu erzeugenden Files vorher ins ALPHA-Register zu schreiben;
 - nach der Erzeugung haben alle Register des Files den Wert Null; die Register werden mit Ø beginnend durchgezählt.

Funktion zur Erzeugung des Datenfiles	Genauigkeit der Werte Anzahl Stellen von		benötigte Bytes je		maximale Registerzahl (in einem Block)
	Mantisse	Exponent	File	Reg.	
DAT12	2	-	72 - 78	1	999
DAT14	4	-	72 - 78	2	999
DAT17	7	-	72 - 78	3	999
DAT19	9	-	72 - 78	4	999
DATE2	2	2	69 - 75	2	999
DATE5	5	2	69 - 75	3	999
DATE7	7	2	69 - 75	4	999
DATEN	10	2	69 - 75	5	801

Die Länge eines Integer-Datenfiles (DAT12 bis DAT19) kann mit folgender Formel berechnet werden:

$$71 + \text{Anz. der Zeichen des Namens} + (\text{Anz. der Bytes je Reg.} \times \text{Anz. Reg.})$$

Für einen 'Exponenten'-Datenfile berechnet sich die Länge in Bytes:

$$68 + \text{Anz. der Zeichen des Namens} + (\text{Anz. der Bytes je Reg.} \times \text{Anz. Reg.})$$

Beispiele zu den einzelnen File-Typen

- DAT12 - zweistellige Personal- oder Abteilungsnummer
 - zweistelliges Kennzeichen für Sorten/Typen mit dem jeweiligen Klartext über das Programm
- DAT14 - vierstellige Konto-/Belegnummern
 - Datumsangaben im Format TT,MM (x100)

- DAT17 - Datumsangaben im Format TT,MMJJ ($\times 10000$)
 - Zeitpunkte nach der Funktion TIME auf eine Zehntelsekunde genau, im X-Reg. steht HH,MMSSzh (evtl. runden, $\times 10^5$)
- DAT19 - positive und negative Beträge, deren absoluter Wert kleiner als 10 Mio. DM ist ($\times 100$, also in Pfennigen gespeichert)
- DATE2 - Mehrwertsteuersätze wie 14,0 % u. 7,5 %; nicht 6,25 %
- DATE5 - Kurse von Aktien/Pfandbriefen/Obligationen
 - Indexzahlen, statistische Reihen
 - technische Werte, z.B. 234,56; 1,2345; 0,0023456
 wenn es nur auf fünf signifikante Ziffern ankommt
 z.B. bei autom. Datenerfassung über Digital-Multimeter
- DATE7 - Sterbewahrscheinlichkeiten q_x (oft nur 5 Nachkomma-Stellen)
 - technische Daten mit hoher Genauigkeit

Eintragen von Zahlen in den Datenfile mit X>DATY

Nach der Erzeugung eines Datenfiles haben zunächst alle Register des Files den Wert Null. Zum Ablegen der Zahlen dient die Funktion X>DATY, die als Eingaben den Namen des Datenfiles, die Nummer des Registers im Y-Register und die abzuspeichernde Zahl im X-Register benötigt. Ist der Datenfile nicht vorhanden oder die Registernummer zu groß, erscheint die Fehlermeldung NONEXISTENT; überschreitet die abzuspeichernde Zahl den zulässigen Bereich (z.B. 100 bei einem DAT12-File), führt dies zur Fehlermeldung OUT OF RANGE. Ist die Genauigkeit der Zahl zu groß, wird sie nur in der Genauigkeit des Datenfiles abgelegt; 1,2345678 wird in einem DATE5-File als 1,2345 abgelegt, $234,5678901 = 2,345678901 \times 10^2$ als $2,3456 \times 10^2$, d.h. 234,56. Der Stack und das LastX-Register werden durch die Funktion X>DATY nicht verändert; der Inhalt des ALPHA-Registers bleibt ebenfalls erhalten.

Bei DATE.-Typen empfiehlt sich vor dem Ablegen in die Register des Datenfiles das Runden der Werte. Dazu eignet sich am besten das SCI - Format mit n-1 Nachkomma-Stellen, wenn n die Anzahl der signifikanten Ziffern bezeichnet; also SCI 4 bei gewähltem DATE5-Format, SCI 6 bei DATE7.

Rückruf der Daten

Zurückgerufen werden die einzelnen Registerinhalte durch Eingabe der Registernummer ins X-Register und durch Aufruf des File-Namens. Haben Sie z.B. einen DAT17-File mit dem Namen MASSE erzeugt und wollen den Inhalt des 15. Registers abrufen, dann geben Sie die Zahl 15 ein und rufen den Datenfile mit XEQ MASSE auf. Der Registerinhalt wird dann in das X-Register des Rechners geschrieben. Die Registernummer bleibt im LastX-Register erhalten.

Tips zur Bedienung und Programmierung

1. Die ProfiSET-Datenfiles sind für den Zugriff durch Programme konzipiert. Wenn Sie Daten zwischen den Registern des Hauptspeichers und einem Datenfile transferieren möchten, werden Ihnen die nachfolgenden Programme hilfreich sein. Alle Programme benötigen die Eingabe des Namens des Datenfiles im ALPHA-Register (für COPYRX max. 6 Zeichen) und eine Steuerzahl der Form sss,dddnnn im X-Register.

Das Programm LADERX kopiert nnn Register aus dem Hauptspeicher, beginnend mit dem Register sss, in das Register ddd und folgende des Datenfiles, der durch seinen Namen im ALPHA-Register gekennzeichnet ist. Das Programm COPYRX kopiert analog nnn Register, beginnend mit Nummer sss aus dem Datenfile ins Register ddd und folgende des Hauptspeichers. Sollte der Platz im Hauptspeicher oder im Datenfile nicht ausreichen, hält das Programm mit NONEXISTENT an.

01*LBL *LADERX*	10*LBL *COPYRX*
FRC LASTX INT ,999 +	INT LASTX FRC 1 E3 *
X(Y 1 E3 *	X(Y ,999 + ASTO 2
10*LBL 00	20*LBL 01
RCL IND Y X)DATY R0N	XEQ IND Z STO IND Y
ISG Y ISG X GTO 00	X(L ISG X ISG Y
RTN	GTO 01 END

Falls Sie diese Hilfsroutinen auf Dauer im ProfiSET speichern wollen, löschen Sie bitte noch 2 Bytes mittels EPACK nach dem Laden (vgl. Kapitel 7).

2. Sollen nur einige Werte über die Tastatur in einzelne, nicht unbedingt aufeinanderfolgende Register eines Datenfiles abgelegt werden, kann es als lästig empfunden werden, daß nach XEQ X)DATY die Funktion jedesmal nach dem Filenamen promptet. Es empfiehlt sich, ein 3-Zeilen-Programm zu schreiben und auf eine Taste zu legen, nachdem der Datenfile z.B. mit dem Namen BILANZ angelegt worden ist; im USER-Modus brauchen Sie nur die entsprechend belegte Taste zu drücken, wenn Reg.-Nr. ENTER Wert getippt wurden:

```
01*LBL *EEE*           LBL EEE  wird einer Taste zugeordnet
*BILANZ* X)DATY END
```

3. Sollten Sie in unregelmäßiger Reihenfolge manuell Werte in verschiedene Files zur späteren Auswertung ablegen müssen, die in den einzelnen Files in fortlaufenden Registern gespeichert werden können, kann vielleicht folgendes Beispiel von Nutzen sein:

10 Personen rufen dem Mann am HP-41 Meßwerte in unregelmäßigen Abständen und in ständig wechselnder Reihenfolge zu, die er je Maßstelle erfassen muß. Er hat sich dazu 10 DATE5-Files mit den Namen MST-1 bis MST-10 zu je 80 Registern angelegt und die Register 01 bis 10 im Hauptspeicher gelöscht, in denen während der Erfassung die Anzahl der eingegebenen Werte je Meßstelle abgelegt werden.

Er braucht nur noch die ihm zugerufenen Werte einzugeben und sie mit XEQ und einer der Tasten 01 bis 10 dem Meßstellenfile zuzuordnen.

```
01*LBL "AAA"
FIX 0 CF 29 1.01 00
```

```
06*LBL a
*MST-* ARCL Y DATES
ISC Y GTO a 10 0
```

```
14*LBL b
STO IND Y DSE Y GTO b
```

```
18*LBL c
*X : XEQ 01-10* PROMPT
```

```
21*LBL 01
1 GTO 00
```

```
24*LBL 02
2 GTO 00
```

```
27*LBL 03
3 GTO 00
```

```
30*LBL 04
4 GTO 00
```

```
33*LBL 05
5 GTO 00
```

```
36*LBL 06
6 GTO 00
```

```
39*LBL 07
7 GTO 00
```

```
42*LBL 08
8 GTO 00
```

```
45*LBL 09
9 GTO 00
```

```
48*LBL 10
10
```

```
50*LBL 00
*MST-* ARCL X
RCL IND X RCL Z X>DATY
1 ST+ IND T GTO c END
```

Modifiziert könnte auch bei jedem der Label 01 bis 10 statt einer Zahl ein Filename stehen; entsprechend sind im Label 00 der Anfang und das 'Registersteuern' zu verändern.

4. Beachten Sie: Datenfiles im Profiset sind für das System nichts anderes als Funktionen. Datenfiles können - so wie sie sind - sofort mit SAVEF auf Magnetband unter ihrem Namen abgelegt und mit GETF wieder in den selektierten Block zurückgelesen werden.

DATEN IM PROFISSET

Ein Beispiel soll das Arbeiten mit ProfISSET-Datenfiles verdeutlichen: Von bis zu 300 zylindrischen Körpern sollen Höhe und Durchmesser mit einem Lineal gemessen werden, ferner soll deren Masse mit einer Waage ermittelt und die drei Daten je Zylinder sollen mit einem HP-41 und einem MBK-ProfISSET erfaßt werden. Die Höhe der Körper beträgt maximal 300 mm; ein DAT14-File reicht aus, da nur ganze Millimeter abgelesen und eingegeben werden. Als Name bietet sich HOEHE an. Der Durchmesser beträgt maximal 99 mm und wird ebenfalls in ganzen Millimetern eingegeben. Wir wählen also einen DAT12-File mit den Namen DURCHM. Der Bereich für die Masse der Körper ist unbekannt; die Waage kann mit einer Genauigkeit von 5 Stellen abgelesen werden. Für diesen Zweck eignet sich ein DATE5-File mit dem Namen MASSE. Die Körper sind, von 1 beginnend, in aufsteigender Reihenfolge durchnummeriert. Wir verzichten zur Vereinfachung des Programms auf das erste Register (Nummer 0) der Datenfiles und setzen die Registernummer gleich der ersten Nummer des Körpers. Es ergibt sich folgendes Erfassungsprogramm:

```
01*LBL *ERF*
2 SLCT *HOEHE* SF 25
CLEAR CF 25 301 DAT14
*DURCHM* SF 25 CLEAR
CF 25 DAT12 *MASSE*
SF 25 CLEAR CF 25
DATES 1,3 STO 00
CF 29 FIX 0
```

Der zweite Block im PlusRAM wird selektiert.

Es wird versucht, den File HOEHE zu löschen, falls dieser bereits vorhanden ist. Ein neuer File HOEHE mit 301 Registern vom TYP DAT14 wird erzeugt. File DURCHM, File MASSE werden erzeugt. Eine Steuerzahl für die Registernummer (1,3 für Register 1 bis 300) wird im Register 0 des Hauptspeichers abgelegt.

```
24*LBL 01
*ZYLINDER* ARCL 00 1
PROMPT X=0? GTO 03
*HOEHE?* PROMPT RCL 00
X<Y *HOEHE* X>DATY
*DURCHMESSER?* PROMPT
RCL 00 X<Y *DURCHM*
X>DATY *MASSE?* PROMPT
RCL 00 X<Y *MASSE*
X>DATY ISG 00 GTO 01
GTO 03
```

Hier beginnt die Schleife zur Erfassung.

Die Nummer des Zylinders wird angezeigt; gibt der Benutzer eine Null ein, wird die Eingabe beendet.

Die Höhe des Zylinders wird abgefragt, seine Nummer zurückgerufen. Der Stack wird geordnet: X=Höhe, Y=Registernummer, in ALPHA Name des Datenfiles. Abspeichern der Höhe, Abfrage des Durchmessers, Abspeichern im Datenfile DURCHM. Abfrage und Speichern der Masse im File MASSE. Erhöhen und Prüfen der Registernummer: wenn kleiner oder gleich 300 zum Schleifenanfang, sonst zum Ende der Erfassung.

```
52*LBL *AENDER*
CF 29
```

Zur Änderung einer Eingabe dient der Programmteil AENDER :

```
54*LBL 04
0 *NR.*? PROMPT X=0?
GTO 03 STO 00
XEQ *HOEHE* *H* *
FIX 0 ARCL X *?*
PROMPT RCL 00 X<Y
*HOEHE* X>DATY X<Y
XEQ *DURCHM* *d* *
ARCL X *?* PROMPT
RCL 00 X<Y *DURCHM*
X>DATY X<Y
XEQ *MASSE* *M* *
FIX 3 ARCL X PROMPT
RCL 00 X<Y *MASSE*
X>DATY GTO 04
```

Schleifenbeginn zur Änderung einzelner Daten. Abfrage des zu ändernden Zylinders über seine Nummer. Eingabe=0, dann Ende der Änderungen. Ablegen der Zylinder-Nr. in Reg. 0. Rückruf des alten Eingabewerts für die Höhe, Aufforderung an den Benutzer, den Wert mit R/S zu bestätigen oder einen neuen Wert einzugeben. Ablegen des Höhen-Werts.

Analoges Verfahren für Durchmesser und Masse des Zylinders.

DATEN IM PROFISSET

Nach der Erfassung werden die Daten ausgewertet. Es soll die Dichte der einzelnen Zylinder berechnet und mit der Nummer des jeweiligen Zylinders ausgegeben werden. Dazu dient der Programmteil DICHTe :

```

92*LBL "DICHTe"
CF 29 1,3 STO 00

96*LBL 02
RCL 00 XEQ "MASSE"
X=0? GTO 03 LASTX
XEQ "HOEHE" LASTX
XEQ "DURCHM" X12 PI *
4 / * / "NR."
FIX 0 ARCL 00 "t : "
FIX 3 ARCL X PROMPT
ISG 00 GTO 02

121*LBL 03
* ** ENDE ** PROMPT
END

```

Schleifenbildung über Reg. 0.

Rückruf der Masse; wenn Wert=0, letzter Zylinder berechnet und zum Programmende. Im LastX-Reg. befindet sich noch die Reg.-Nr, Höhe und Durchmesser werden zurückgerufen. Berechnung der Dichte nach Formel:

$$\text{Dichte} = \frac{\text{Masse}}{\text{Grundfläche} \times \text{Höhe}}$$

Rückruf der Nummer für die Ausgabe. Erhöhen und Prüfen des Schleifenzählers.

Programmende für alle drei Programmteile.

Wenn Sie das Programm zur Übung eingeben möchten, können Sie es zusammen mit den Datenfiles in den Block 2 Ihres ProfISeTs laden. Es sind hier noch einmal die einzelnen Schritte aufgeführt, die Sie ausführen sollten:

```

2
SLCT          Der zweite Block wird selektiert.
20
NEWBL        Er wird gelöscht und für 20 Einträge initialisiert.
5            Er bekommt die Identität 5.
IDBL

-ZYLINDER-
NAMEBL      Der Block bekommt den Namen -ZYLINDER-.
LADep "ERF" Das Programm ERF wird in den PlusRAM geladen
CLP "ERF"   und anschließend im Hauptspeicher gelöscht.

PACKING
XROM "ERF"  RUF
ZYLINDER1   RUF
HOEHE?      200 RUF      Durchmesser = 50 mm
DURCHMESSER?
MASSE?      50 RUF      Masse = 1,455 kg
1,455 RUF
ZYLINDER2   RUF
HOEHE?      155 RUF
DURCHMESSER?
MASSE?      60 RUF
1,23 RUF    Nach dem zweiten Zylinder wird die Eingabe ab-
ZYLINDER3   0 RUF      gebrochen.
** ENDE **

```

DATEN IM PROF1SET

Wenn Sie das Berechnungsprogramm DICHTe durchführen, erhalten Sie folgende Ergebnisse:

```

XROM *DICHTe*
NR. 1 : 3.705E-6
                RUH
NR. 2 : 2.807E-6
                RUH
** ENDE **

```

Dichte für Nr. 1 = 3,705 mg/mm³
Dichte für Nr. 2 = 2,807 mg/mm³

Führen Sie eine XROM-Umwandlung für diesen Block durch, damit die Programmzeilen 61, 72 usw. nicht mehr XEQ"HOEHE", sondern HOEHE usw. lauten.

XROMBL Umwandlung der XEQ-Befehle in XROM-Befehle.
PACKBL Es empfiehlt sich das Block-Packen, da das
BLCAT Programm kürzer geworden ist.

```

=====
BL:02 -- 10:05 -- ADR:09
-ZYLINDER-
=====

```

Der Blockkatalog gibt eine Übersicht über den derzeitigen Inhalt.

```

=====
NAME        NR TYP BYTES
-ZYLINDER- 00 N    0011
ERF        01 P    0322
AENDER     02 XERF
DICHTe     03 XERF
HOEHE      04 D14 0678
DURCHM     05 D12 0378
MASSE      06 D05 0976
=====

```

BLF? Die Funktion BLF? zeigt, daß noch 1675
1675.000 *** Bytes in diesem Block frei sind.

Zur Erinnerung: Die Datenfiles HOEHE, DURCHM und MASSE haben jeweils 301 Register. Dies würde im normalen HP-41-Maß für die Register schon 6321 Bytes entsprechen. Das Beispiel zeigt die enorme Platzersparnis durch eine geeignete Wahl der Genauigkeit für die Datenfiles. Statt normalerweise 6321 Bytes werden in unserem Beispiel nur 2032 Bytes für die drei Datenfiles benötigt, also weniger als ein Drittel des normalen Speicherraums.

SAVEBL GETBL

Die Funktionen SAVEBL und GETBL erlauben Ihnen, einen ganzen Block des PlusRAMs auf dem Digital-Kassettenlaufwerk abzulegen bzw. wieder in das ProfISET einzulesen. Der Block kann auf der Kassette einen beliebigen Namen erhalten, der vor der Ausführung der Funktionen SAVEBL und GETBL in das ALPHA-Register eingegeben werden muß. Ist der eingegebene Name länger als 7 Zeichen, werden nur die ersten 7 Zeichen verwandt. Der erzeugte File auf der Kassette hat den Typ ??,A und eine Länge von 640 Registern. Sie können bis zu 25 Blöcke auf einer mit NEWM Ø25 neu initialisierten Kassette ablegen. Beim Ablegen wird die Prüfziffer gebildet (vergl. SUMBL) und beim Einlesen überprüft (vergl. CHECKBL).

Beispiel für SAVEBL und GETBL :

```

          1.000
          SLCT
TESTBLOCK SAVEBL

```

```

          3.000
          SLCT
TESTBLOCK GETBL

```

Beachten Sie, daß GETBL vorhandene Files im selektierten Block total überschreibt.

SAVEF GETF

Die Funktionen SAVEF und GETF dienen zum Abspeichern und Einlesen von ProfISET-formatierten Funktionen, Daten- und KEY-Files. Der Name der abzuspeichernden Funktion muß im ALPHA-Register angegeben werden und wird auch gleichzeitig zum Filenamen auf der Kassette. Der File kann bei Bedarf später auf der Kassette mit RENAME umbenannt werden.

Der erzeugte File auf dem Band wird vom DIR als TYP ?? ausgegeben und hat die Länge in Register:

$$\text{Anz. Reg.} = ((\text{Anzahl der Einträge im CAT2}) + (\text{Länge im BLCAT}) + 4) / 2$$

Zur Erweiterung Ihrer Programmpakete können Sie vom Mathematischen Büro Köln verschiedene Funktionen auf Band erhalten, die Sie mit GETF in Ihr ProfISET einlesen können.

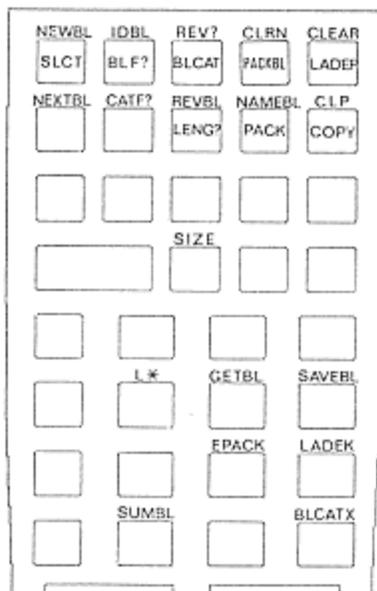
Diese Funktionen sollen Ihnen bei Programmierarbeiten das Leben etwas leichter machen. Als Subroutinen von Ihren Programmen aufgerufen oder bei Bedarf manuell durchgeführt (XEQ), haben sie für Sie hoffentlich ein wenig Nutzen.

PKEY

Nach der Ausführung der Funktion PKEY wird im USER-Modus das abgebildete ProfiSET-Tastenfeld wirksam. Alle anderen Tastenzuordnungen werden gelöscht. Das ProfiSET-Tastenfeld (vergl. auch das mitgelieferte Overlay) enthält die beim Umgang mit dem ProfiSET häufig benutzten Funktionen und benötigt im Hauptspeicher des Rechners 13 Register, die vor der Ausführung von PKEY frei sein müssen. Sie können jederzeit manuell weitere Tasten nach Ihrem Bedarf zuordnen (ASN).

USER KEYS:

- 11 SLCT
- 11 NEWBL
- 12 BLF?
- 12 IOBL
- 13 BLCAT
- 13 REV?
- 14 PACKBL
- 14 CLRN
- 15 LADEP
- 15 CLEAR
- 21 NEXTBL
- 22 CATF?
- 23 LENG?
- 23 REVBL
- 24 PACK
- 24 NAMEBL
- 25 COPY
- 25 CLP
- 42 SIZE
- 62 L*
- 63 GETBL
- 64 SAVEBL
- 73 EPACK
- 74 LADEK
- 82 SUMBL
- 84 BLCATX



DIE PROFISSET-PSUB-FUNKTIONEN

EPACK

Die Funktion EPACK fordert zur Eingabe des Namens eines Programmes im PlusRAM auf. Das Programm wird darauf durchsucht, ob Zahlen der Form 1 E.. auftreten. Werden solche Zahlen (z.B. 1 E-15) im Programm gefunden, wird die 1 vor dem E beseitigt, was den Platzbedarf des Programmes verringert und die Laufzeit an den entsprechenden Programmstellen etwas verkürzt (Beispiel: aus 1 E-15 wird E-15, aus 1 E wird E und aus 1 E99 wird E99). Der Rechner erkennt beide Schreibformen (1 E-15 = E-15) als gleiche Zahl.

LADEK

Mit der Funktion LADEK können Sie die aktuellen Tastenzuordnungen als File (Typ: KEY) im ProfISSET ablegen. Dabei werden analog zur Funktion WRTK des Bandlaufwerkes nur die Zuordnungen übertragen, die sich nicht auf Labels im Hauptspeicher des Rechners beziehen. LADEK erfordert die Eingabe eines Filenamens; es können also mehrere Tastenzuordnungen im PlusRAM abgelegt werden. Die Länge eines KEY-Files ist abhängig von der Länge seines Namens und der Anzahl der Tastenzuordnungen; er hat jedoch einen Grundbedarf von 85 Bytes. Für jedes Zeichen des Namens wird ein weiteres Byte benötigt, für jeweils 2 Tastenzuordnungen 5 weitere Bytes. Soll beispielsweise ein KEY-File mit 25 Zuordnungen unter dem Namen TASTE1 abgelegt werden, beträgt der Platzbedarf 85 Bytes Grundbedarf + 6 Bytes für die 6 Zeichen des Namens + 65 Bytes für 25 Tasten ($25/2 = 12,5$; $13 \times 5 = 65$), also insgesamt 156 Bytes. Wenn Sie zu einem späteren Zeitpunkt eine im PlusRAM befindliche Tastenzuordnung wieder aktivieren, d.h. in den Hauptspeicher des Rechners zurückladen wollen, müssen Sie den Namen des KEY-Files nur als Funktion (z.B. XEQ TASTE1) ausführen. Es werden dann alle Tastenzuordnungen gelöscht und die Tastatur durch den Inhalt des KEY-Files neu belegt. KEY-Files können mit dem Befehl SAVEF direkt aus dem PlusRAM unter Umgehung des Hauptspeichers auf Band abgelegt werden. Im PlusRAM werden Sie mit CLEAR gelöscht. Die Ausführung eines KEY-Files mit 26 Tastenzuordnungen benötigt z.B. ca. 4 Sekunden; es erscheint während dieser Ausführungszeit jedoch keine Meldung in der Anzeige. LADEK kann auch ausgeführt werden, wenn keine Tastenzuordnungen existieren; der so erzeugte File wirkt dann wie die Funktion CLKEYS aus dem X-Functions-Modul, d.h. bei der Ausführung eines KEY-Files, der keine Tastenzuordnungen enthält, werden alle Tastenzuordnungen gelöscht.

LADES CLRS

Die Funktion LADES veranlaßt den Rechner, bei jedem Einschalten den Befehl GTO*START auszuführen und das Programm START abzuarbeiten. Das von Ihnen geschriebene Programm mit dem Label START kann z.B. eine bestimmte Tastenzuordnung, eine Statuseinstellung oder ein Menu zur Benutzerführung enthalten. LADES benötigt im selektierten Block 33 Bytes, erscheint jedoch nicht im Block-Catalog. Da der gesamte

DIE PROFISSET-PSUB-FUNKTIONEN

Rechner nach einem Label START abgesucht wird, kann sich beim Einschalten des Rechners eine Verzögerung ergeben. Es empfiehlt sich, das Label START und die Autostart-Option im Block 1 unterzubringen. Mit CLRS kann der Autostart (nicht das Label START) wieder gelöscht werden; dazu muß aber der Block selektiert sein, in dem man früher mit LADES die Autostart-Option eingetragen hat.

Anwendungsbeispiel

```

01*LBL *START*
*PASSWORT ?* AON STOP
AOFF ASTO Y *PROFI*
ASTO X X=Y? GTO 00
CLA CLST OFF

14*LBL 00
*41-MANUELL* N? GTO 00
CLX FIX 2 CF 27 CF 28
SF 29 STOP

24*LBL 00
SF 27 *FINANZMATH.* J?
GTO *FIMA*
*KALKULATION* J?
GTO *KALK* *STEUERN*
J? GTO *STEU*
*BUCHHALTUNG* J?
GTO *BUHA* GTO 00 END

```

Nach dem Einschalten wird nach dem Passwort gefragt, das richtig eingegeben werden muß, sonst schaltet sich der Rechner wieder aus. Im Dialog wird der Benutzer gefragt, ob er eine manuelle Bedienung des HP-41 wünscht oder eines der Arbeitsgebiete des Menues haben will; dort könnte jeweils mit Untermenues zum einzelnen Programm verzweigt werden (J?/N? siehe S. 42).

Die folgenden Funktionen sollen Ihnen helfen, Ihre Programme in bezug auf Ein- bzw. Ausgabe etwas benutzerfreundlich zu gestalten. Falls Sie das Profiset als Entwicklungssystem für Software-Pakete benutzen, die später für EPROM-Boxen vervielfältigt werden sollen, beachten Sie bitte, daß diese Funktionen beim Benutzer nur dann vorhanden sind (also aufgerufen werden können), wenn auch das Betriebssystem des Profisets vorhanden ist. Falls Sie einige dieser oder weitere Tool-Funktionen in den PlusRAM laden möchten, können Sie vom Mathematischen Büro Köln die entsprechenden Funktionen auf Band erwerben und mit GETF in den PlusRAM einlesen.

L* L- L= PRM

L*, L- und L= erzeugen auf einem angeschlossenen Drucker eine Stern-, Strich- oder Doppelstrichlinie. Ist kein Drucker vorhanden, wird die Funktion ohne Fehlermeldung übergegangen (Ausnahme: Ist ein IL-Modul vorhanden und der Schalter am IL-Modul in der Stellung 'enable' und kein Drucker in der IL, wird die Fehlermeldung NO PRINTER ausgegeben). Ist Flag 12 gesetzt, werden 12 Zeichen gedruckt, sonst sind es 24.

DIE PROFISSET-PSUB-FUNKTIONEN

Flag 21 und 55 haben keinen Einfluß auf diese Funktionen. Der Befehl PRM bewirkt einen zentrierten (mittigen) Druck des ALPHA-Registers; er verhält sich in bezug auf das Vorhandensein eines Druckers und den Zustand der Flags 21 und 55 wie die Funktionen L*, L- und L=.

```

L*
*****
-----
L-
-----
L=
=====
PROFISSET
PRM
PROFISSET
SF 12

```

```

L*
*****
-----
L-
-----
L=
=====
PROFISSET
-PROFISSET
DAS PROFISSET
IST GUT

```

J? N?

Die Funktionen J? und N? führen einen neuen Benutzerdialog ein: die Ja/Nein-Entscheidung über die Tastatur. Steht in einem Programm die Funktion J?, wird das Programm an dieser Stelle angehalten und das ALPHA-Register angezeigt (wie bei der Funktion PROMPT). Hinter der Textkette (am rechten Rand der Anzeige) erscheint ein blinkendes Fragezeichen; alle Tasten außer ON, ENTER und R/S sind blockiert. Wird die Taste ON betätigt, schaltet sich der Rechner wie gewohnt aus. Die Taste ENTER (N) hat die Bedeutung NEIN, die Taste R/S die Bedeutung JA. Wird nun die JA-Taste betätigt, wird die Programmausführung mit dem nächsten Befehl fortgesetzt. Bei Betätigung der NEIN-Taste wird die Programmausführung auch fortgesetzt, jedoch mit dem übernächsten Befehl. Die Funktion N? ist die Umkehrung der Funktion J?, d.h. das Überspringen des nächsten Befehls erfolgt bei der Betätigung der JA-Taste. Wenn Sie das folgende Beispielprogramm eingeben und ausführen, wird Ihnen die Bedeutung der beiden Funktionen und deren Verzweigung nach Tastendruck klar:

```

@1*LBL *JA-NEIN*
SF 26 *MIT SIGNAL* N?
CF 26 *LANGES SIGH.*
FS? 26 J? BEEP TONE 9
END

```

Wird ca. 2 Minuten nach dem Beginn einer der Funktionen J?/N? keine Taste betätigt, schaltet sich der Rechner automatisch aus. Wird er anschließend wieder eingeschaltet, startet das Programm automatisch (unabhängig von Flag 11) wieder mit der Funktion J?/N?. Dies gilt nicht für das manuelle Ausschalten mit der ON-Taste.

Dieses Kapitel soll zeigen, wie Sie mit Hilfe kurzer Programme und der Verwendung von Profiset-Funktionen Ihre PlusRAM-Anwendungen vereinfachen können.

Manche Profiset-Funktionen fordern Sie zur Eingabe eines Textes auf, sie prompten. Es könnte sein, daß Sie durch die Bedienungshinweise des X-Functions-Moduls oder des Bandlaufwerks gewohnt sind, Ihre Eingaben vorher in das ALPHA-Register zu schreiben, und dies bei der manuellen Ausführung von Profiset-Funktionen auch tun möchten. Für diese Anpassung reicht es aus, wenn Sie vor die Profiset-Funktionen ein Label der Form schreiben:

```
01*LBL *LADEP+          04*LBL *CLEAR+
02 LADEP **            05 CLEAR **
03 RTN                 06 RTN
```

Beachten Sie jedoch bitte, daß dann die Eingabe eines Namens immer notwendig ist. Mit einem leeren ALPHA-Register würde bei der Ausführung von 'CLEAR+ Ihr Hilfsprogramm und nicht das Programm, auf dem Sie sich vorher befanden, gelöscht.

Eine weitere Besonderheit des Profisets ist, daß beim Laden eines Files nicht ein alter File mit gleichem Namen automatisch überschrieben wird. Dennoch läßt sich auch das automatische Überschreiben mit einem kurzen Hilfsprogramm erreichen:

```
01*LBL *LADEP+
02 SF 25
03 CLEAR **
04 CF 25
05 LADEP **
06 RTN
```

Natürlich kann dieses Programm für weitere Filetypen (z.B. Datenfiles) analog erweitert werden. Sie können aber auch auf Wunsch eine Fehlermeldung erhalten, falls ein File gleichen Namens bereits vorhanden ist:

```
01*LBL *LADEP+
02 SF 25
03 TYP? **
04 FC?C 25
05 GT0 00
06 LADEP **
07 RTN
:
:
13*LBL 00
14 *ERR DUP FILE*
15 PROMPT
16 END
```

PROGRAMMIERUNG MIT PROFISSET-FUNKTIONEN

Das folgende Programm BL-EIN hilft bei der vollständigen Blockeinrichtung. Es zwingt zur Eingabe der Blocknummer, fragt nach der Anzahl der Einträge mit dem Vorschlag 64 (dann reicht R/S) und nach der Identität des Blockes. Wenn keine ID-Nummer eingegeben wird, wird die Blocknummer genommen. Das Programm fragt weiter nach einem Blocknamen; wird keine Angabe gemacht, reserviert es Platz für einen späteren Eintrag. Nach der Frage nach der Revision erfolgt abschließend zum Überblick ein Ausdruck der Eingaben. Für ein eventuell folgendes Programm-Laden werden die freien Einträge im CAT2 gezeigt, die freien Bytes und und REGS. für Programme vom Band. Dieser Teil kann auch selbständig aufgerufen werden.

```

01*LBL "BL-EIN"
, *BLOCK-NR. =?
PROMPT SLCT 64
*ANZ. EINTR. =? PROMPT
NEWBL BL? *ID-NR. =?
PROMPT IDBL
*BLOCKNAME ?* ADN STOP
NAMEBL *REVISION ?*
STOP FS?C 23 REVBL
ROFF ADV BLCAT REV?
SF 12 PRM CF 12

29*LBL "??*
FIX 0 CF 29 CATF?
*CAT2 FREI * ARCL X
AVIEW BLF? *FREI *
ARCL X *+ BYTES * 2 -
? / INT ARCL X
*+ REGS* AVIEW L- END

```

Soll eine Funktion auf alle 4 Blöcke des PlusRAMs nacheinander angewandt werden, kann das Programm FKTSET helfen. Es macht Sinn für die Funktionen:

PACKBL	ID?
PACKC	REV?
XROMBL	CATF?
SUMBL	BLF?
	CATS?

Das Programm benutzt die Funktion POSA des X-F-Moduls; es kann bei Bedarf gekürzt werden. Läßt sich in einem Block kein Eintrag mehr im CAT2 vornehmen, bleibt das Programm mit der Meldung stehen. Sie können dann nach der Ausführung von NEXTBL wieder starten.

```

01*LBL *FKTSET*
FIX 0 CF 29 CF 12
CF 14

06*LBL 01
TONE 0 ADN
*FKT.-NAME ?* CF 23
STOP FC? 23 GTO 01
ROFF 63 POSA X?0?
SF 14 82 POSA X=0?
SF 12 FS? 12 CF 14
ASTO X 1 SLCT

28*LBL 00
XEQ IND Y TONE 9
FS? 12 AVIEW FS? 14
VIEW X FS? 14 RDN
SF 25 NEXTBL FS?C 25
GTO 00 CF 14 CF 12
CLX END

```

PROGRAMMIERUNG MIT PROFISSET-FUNKTIONEN

Wenn man alle Blöcke des PlusRAMs hintereinander auf Band ablegen möchte, sind für die Block-Files auf dem Band jeweils unterschiedliche Namen erforderlich. Das Programm SAVESET fordert zur Eingabe eines gemeinsamen Namens (max. 6 Zeichen) auf und legt die Block-Files mit diesem Namen und der angehängten Blocknummer nacheinander auf Band ab. Mit GETSET kann der gesamte PlusRAM-Inhalt wieder geladen werden.

```

01*LBL *SAVESET*
FIX 0 CF 29 *NAME ?*
AON STOP AOFF ASTO X
I

10*LBL 00
CLA ARCL Y ARCL X
SF 25 SLCT FC?C 25
RTH SAVEBL SEC 1 +
GTO 00

23*LBL *GETSET*
FIX 0 CF 29 *NAME ?*
AON STOP AOFF ASTO X
I

32*LBL 01
CLA ARCL Y ARCL X
SF 25 SLCT FC?C 25
RTH GETBL 1 + GTO 01
END

```

Mit dem Programm SETCAT werden die Block-Cataloge aller PlusRAM-Blöcke und zusätzlich noch die Angaben über die Anzahl der noch möglichen Einträge und über den verbleibenden Raum in den Blöcken ausgegeben. Sie erhalten also eine komplette Übersicht über den aktuellen Inhalt Ihres PlusRAMs.

```

01*LBL *SETCAT*
FIX 0 CF 29 1 SLCT

06*LBL 00
BLCAT *FREI : * 0
SF 25 CAT? CF 25
ARCL X *E EINTRAEGE*
PRA * * BLF?
ARCL X *+ BYTES* PRA
*REVISION: * ACA REV?
ACA PROUF L= SF 25
NEXTBL FS?C 25 GTO 00
ADV ADV ADV ADV END

```


T E I L 2

ARBEITEN MIT DEM EPROMER

Für die Software-Vervielfältigung oder die 'Hard-Copy' mit sofortigem Zugriff auf umfangreiche Programmsammlungen beim einzelnen Benutzer des HP-41 haben sich im Verlauf der letzten drei Jahre immer stärker EPROM-Boxen durchgesetzt. Sie sind als Alternative zum Steckmodul entwickelt worden, das nur von Hewlett-Packard als sogenanntes 'customer rom' produziert werden kann. Da eine solche Modulerstellung mit sehr hohen Fixkosten und recht langer Produktionsdauer (bis zu 6 Monate) verbunden ist, sind customer roms in der Regel nur für sehr hohe Auflagen von Software geeignet, von denen mehr oder weniger feststeht, daß in den Programmen keine Änderungen mehr vorgenommen werden müssen. Sollte dennoch eine neue Version der Software notwendig werden, ist eine Neuproduktion mit demselben Kosten- und Zeitaufwand wie bei der Erstauflage des Moduls erforderlich.

Hier sind EPROM-Boxen deutlich wirtschaftlicher und flexibler einsetzbar. Bleibt ein HP-ROM auf maximal 8K begrenzt, besitzen EPROM-Boxen Kapazitäten beginnend bei 4K bis zu 32K. Die gängigste Größe sind EPROM-Boxen mit 16K, mit denen aber auch nur 4K oder 8K Programmumfang betrieben werden können. Derzeit sind allein im deutschsprachigen Raum weit mehr als 3.000 EPROM-Boxen zum HP-41 im Gebrauch.

EPROM-Boxen haben als Programmspeicher für den einzelnen Anwender allerdings den Nachteil, daß er neben den einmaligen Anschaffungskosten für die Hardware (Box und EPROM-Material) weiteren Honorar-Aufwand für die EPROM-Programmierung in Kauf zu nehmen hat. Bislang war es nur bei wenigen Stellen in Deutschland möglich, mit speziellen Programmiergeräten und geeigneter Software HP-41-Programme auf EPROMs zu übertragen.

Das Mathematische Büro Köln hat als Ergänzung zum ProfiSET und den von ihm entwickelten EPROM-Boxen MBK-16 und MBK-32 ein EPROM-Programmiergerät am Markt eingeführt, das eine äußerst einfache und sichere Art der EPROM-Satz-Erstellung für den HP-41 ermöglicht. Mit dem MBK-EPROMer ist damit auch dem Einzelanwender und kleineren Firmen ein Gerät an die Hand gegeben worden, das auf Dauer gesehen die Verwendung von EPROM-Boxen noch wirtschaftlicher gestaltet. Mit der Betriebssoftware für das EPROM-Programmieren, die sich schon im ProfiSET befindet, kann jetzt jeder einzelne Anwender seine EPROM-Sätze selbst erstellen und sich somit das sonst fällige Honorar ersparen.

Der MBK-EPROMer muß nicht unbedingt gekauft werden; Sie können ihn auch vom Mathematischen Büro Köln gegen Gebühr ausleihen.

In den folgenden Kapiteln ist die Bedienungsweise von ProfiSET und MBK-EPROMer beschrieben. In der Regel werden Sie die Standardfälle der EPROM-Programmierung anwenden, die in Kapitel 2 beschrieben sind. Hierbei wird z.B. der Inhalt des PlusRAMs (4 Blöcke = 16K) auf EPROMs übernommen; d.h. im ProfiSET entwickelte und ausgetestete Versionen von umfangreichen Programmsammlungen werden auf Festwertspeicher übertragen und können dann mit der EPROM-Box MBK-16 betrieben oder einem anderen zur Benutzung übergeben werden, der vielleicht kein Bandlaufwerk besitzt.

Das Programmiergerät besitzt als einziges Bedienungselement einen Netzschalter (auf der Rückseite des Gerätes). Alle übrigen Einstellungen erfolgen programmgesteuert über das Betriebssystem des ProfiSETs. Zur Inbetriebnahme des Programmiergerätes muß zunächst am ProfiSET folgende Einstellung vorgenommen werden:

System auf Port 4	PlusRAM auf Port 1+2	schalten !
-------------------	----------------------	------------

Während der Benutzung des Programmiergerätes darf sich in Port 4 des Rechners nichts befinden, ausgenommen der Stecker des ProfiSETs oder ein Memory- oder X-Memory-Modul. Es darf beim Programmieren nur ein ProfiSET und keine zusätzliche EPROM-Box am Rechner angeschlossen sein. Der gleichzeitige Anschluß von mehreren Programmiergeräten ist nicht möglich. Die sinnvollste Systemkonfiguration für EPROM-Programmierarbeiten besteht aus: Rechner, ProfiSET, EPROMer und (falls vorhanden) Drucker bzw. IL mit Drucker und Bandlaufwerk.

Gehen Sie beim Anschluß des EPROMers wie folgt vor:

1. Schließen Sie den EPROMer am Stromnetz (220V) an.
2. Schalten Sie den EPROMer ein (gelbe Kontrollanzeige 'POWER' leuchtet).
3. Schließen Sie das ProfiSET an den ausgeschalteten Rechner an und überprüfen Sie die Einstellung des ProfiSETs (s.o.)
4. Schließen Sie jetzt den EPROMer an das ProfiSET an.
5. Schalten Sie den Rechner ein. Sollte der BAT-Indikator in der Anzeige des Rechners jetzt oder zu einem späteren Zeitpunkt erscheinen, müssen Sie frische Batterien in den Rechner einsetzen, die Programmierfunktionen brechen sonst mit der Fehlermeldung ERR'LOW BATT ab.
6. Beginnen Sie mit der Programmierung der EPROMs. Sie können mit angeschlossenen EPROMer wie gewohnt mit Ihrem ProfiSET arbeiten, also z.B. Blöcke vom Band in das ProfiSET laden.
7. Führen Sie die Punkte 5 bis 1 in umgekehrter Reihenfolge und im umgekehrten Sinn aus.

Folgende Punkte sollten Sie beim Umgang mit dem EPROMer unbedingt beachten:

- Schließen Sie nie den EPROMer an das ProfiSET an, wenn es nicht unmittelbar danach an den Rechner angeschlossen wird, denn sonst verkürzt sich die Lebensdauer der Batterien im ProfiSET.
- Unterbrechen Sie nie die Stromversorgung des EPROMers, wenn er mit dem ProfiSET verbunden ist.
- Lassen Sie die Finger vom gesamten System, wenn die gelbe Kontrollanzeige 'BUSY' am EPROMer leuchtet; Sie könnten das EPROM im EPROMer zerstören.
- 'Spielen' Sie nicht mit den einzelnen Funktionen des EPROMers, sonst könnten Sie 'spielend' ein EPROM, das nur 12,5V verträgt, mit 25V zu einem Schmuckstück ohne Gebrauchswert machen.

HANDHABUNG DES PROGRAMMIERGERÄTS

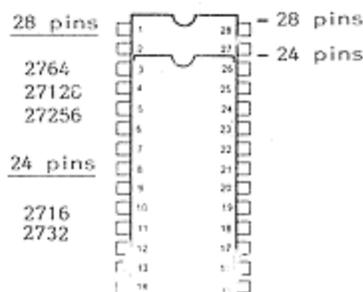
Die Kontrollanzeigen des EPROMers haben folgende Bedeutung:

POWER (gelb)	OK (grün)	BUSY (gelb)	BAD (rot)	Bedeutung
Aus	Aus	Aus	Aus	Die Stromversorgung ist unterbrochen.
Ein	?	?	?	Die Stromversorgung ist vorhanden.
Ein	Aus	Aus	Aus	Zustand nach dem Einschalten des EPROMers; das Gerät ist betriebsbereit.
Ein	Ein	Aus	Aus	Falls zuvor ein EPROM programmiert wurde, ist dieses in Ordnung. Das Gerät ist für den nächsten Programmiervorgang bereit.
Ein	Aus	Ein	Aus	Der EPROMer arbeitet und darf nicht gestört werden.
Ein	Aus	Aus	blinkt	Es ist ein Fehler aufgetreten; achten Sie auf die Anzeige des Rechners.
Ein	Aus	Aus	Ein	Es ist ein Fehler aufgetreten, während Sie nicht anwesend waren. Falls sich der Rechner noch nicht ausgeschaltet hat, sehen Sie die Meldung in der Anzeige. Sonst können Sie den Rechner einschalten und versuchen, den Programmiervorgang fortzusetzen; der Fehler wird wahrscheinlich erneut auftreten. Das Gerät ist bereit für den nächsten Programmiervorgang.

Einsetzen eines EPROMs in den EPROMer

Ein EPROM darf erst dann eingesetzt werden, wenn die BUSY-Kontrollanzeige am EPROMer nicht mehr leuchtet, alle notwendigen Verbindungen jedoch hergestellt sind. Wichtig ist des weiteren, daß das EPROM in der richtigen Lage eingesetzt wird. Das EPROM besitzt an einer seiner Schmalseiten eine kleine Kerbe; diese Seite muß mit der Markierung auf dem

EPROMer übereinstimmen. Werden EPROMs mit 24 'Beinen' eingesetzt, müssen sie an der Seite des Spannhelms der Fassung bündig eingesetzt werden. Der Spannhelm hat folgende Funktion: Steht er senkrecht, ist die Fassung geöffnet und zum Herausnehmen oder Einsetzen des EPROMs bereit; liegt er waagrecht, ist das EPROM an allen Anschlüssen mit dem Programmiergerät sicher verbunden.



Die EPROMer-Funktionen ergeben nur einen Sinn, wenn der MBK-EPROMer am ProfISET angeschlossen ist. Wenn nur EPROM-Sätze der Größen 4, 8 und 16K mit ein oder zwei L-EPROMs programmiert werden sollen, sind Kenntnisse über die einzelnen PSET-PROM-Funktionen nicht unbedingt erforderlich; das Programm ^TPROM führt den Benutzer im Dialog. Für fortgeschrittene Anwendungen wie 32K-Sätze oder Sätze, die in der Reihenfolge der Blöcke im CAT2 nicht mit dem Inhalt des PlusRAMs übereinstimmen, können jedoch auch die einzelnen EPROMer-Funktionen aufgerufen werden. Die Lektüre des Kapitels 'Der EPROM-String' ist dann unumgänglich. Sie sollten die Einzel-Funktionen in ein kurzes, für Ihre Zwecke geeignetes Programm schreiben, da eine manuelle Fehlbedienung (vor allem falscher EPROM-String) fatale Folgen haben kann.

Die meisten Fälle für die Erstellung von EPROM-Sätzen für den HP-41 werden durch das Programm ^TPROM abgedeckt. Sie können mit diesem Programm folgende EPROM-Sätze erstellen:

Satz	U-EPROM	L1-EPROM	L2-EPROM	Quellblöcke	erforderliche Eingaben ^T PROM
4K	2716	2732	-	1	X , Nein , Ja
8K	2716	2732	2732	1 und 2	X , Nein , Nein Ja , Ja
8K	2716	2764	-	1 und 2	X , Nein , Nein Ja , Nein
16K	2732	2764	2764	1 bis 4	X , Ja , Ja
16K	2732	27128	-	1 bis 4	X , Ja , Nein

Wenn Sie ^TPROM ausführen, hält das Programm zunächst mit der Eingabeaufforderung

ANZAHL

an. Geben sie die Anzahl \times der insgesamt zu erstellenden (zu programmierenden) EPROM-Sätze ein. Möchten Sie nur einen Satz programmieren, reicht auch die Betätigung der R/S-Taste ohne eine Eingabe aus. Als nächstes werden Sie zur Eingabe der Satz-Größe (4, 8 oder 16K) aufgefordert. Zu dieser Eingabe sind nur die Tasten NEIN (ENTER-Taste) und JA (R/S-Taste) aktiv. Das Programm läuft so lange in einer Schleife, bis Sie die JA-Taste betätigen und damit die Satz-Größe bestimmt haben. Die Bedienung für einen einzelnen 4K-Satz sieht z.B. wie folgt aus:

Anzeige	Eingabe	Kommentar
?	XEQ ^T PROM	Aufruf des Programmes ^T PROM
Anzahl	1 R/S	Eingabe: ein Satz
16K ?	ENTER(=NEIN)	Nein
4K ?	R/S(=JA)	Ja, die Satzgröße soll 4K betragen.

STANDARDFÄLLE DER EPROM-PROGRAMMIERUNG

Nach diesen Eingaben erscheint beim 8- und 16K-Satz noch die Frage:

2 x L ?

Falls Ihr EPROM-Satz aus zwei L-EPROMs bestehen soll, beantworten Sie diese Frage mit JA (R/S-Taste), sonst mit NEIN. Anschließend erscheint in der Anzeige:

PZ BILDEN

Es wird automatisch in den benötigten Blöcken die Prüfziffer gebildet und eingetragen (siehe Funktion SUMBL). Ist dies geschehen, werden Sie mit

U = 27..

zum Einsetzen des ersten EPROMs (U-EPROM) aufgefordert. Das Programm startet automatisch weiter, wenn Sie das EPROM in die Fassung des EPROMers eingesteckt haben und diese geschlossen wurde. Falls das EPROM nicht zügig genug eingesetzt wurde, hält das Programm mit der Fehlermeldung ERR'NO EPROM an, und nur in diesem Fall müssen Sie die R/S-Taste betätigen. Jetzt beginnt der eigentliche Programmiervorgang. Sobald das U-EPROM fertig ist, werden Sie durch ein Tonsignal aufgefordert, dieses zu entfernen und kurz darauf das nächste EPROM (L oder L1) einzusetzen. Das Programm läuft weiter, bis alle Sätze programmiert sind. Sollte während des Programmlaufes ein Fehler auftreten, versuchen Sie zunächst, durch Betätigung der Taste R/S das Programm erneut zu starten. Tritt die Fehlermeldung wieder auf, ist das eingesetzte EPROM wahrscheinlich defekt oder nicht komplett gelöscht. Schalten Sie in diesem Fall in den PRGM-Modus. Wenn Sie dort den Befehl EP=FF? sehen, dann brauchen Sie nur ein neues EPROM einzusetzen. Sollte dort ein anderer Befehl stehen, müssen Sie das Programm für die noch zu programmierenden Sätze erneut von Anfang an starten; teilweise programmierte Sätze sind dann zu löschen.

Bitte achten Sie darauf, daß Sie programmierte EPROMs an ihrem Fenster mit einem Aufkleber abdecken. Dieser sollte auch gleichzeitig mit U, L, L1 oder L2 beschriftet werden, um spätere Verwechslungen beim Einsetzen in eine EPROM-Box zu vermeiden. Ist das Fenster am EPROM nicht abgedeckt, können nach längerer Zeit Informationen auch durch normales Tages- oder Kunstlicht verlorengehen.

Zum Löschen von EPROMs ist eine spezielle UV-Lampe erforderlich.

Der EPROM-String

Im Betriebssystem des ProfISETs befinden sich die zum Betrieb des EPROM-Programmiergerätes erforderlichen Funktionen. Zur Ausführung dieser Funktionen muß in der Regel eine Textkette in das ALPHA-Register eingegeben werden, die im folgenden mit 'EPROM-String' bezeichnet ist. Darin enthalten sind die für den Programmiervorgang notwendigen Angaben: Typ des EPROMs, U- oder L-EPROM, Fast- oder Normal-Algorithmus und der Zielblock im EPROM. Diese Angaben teilen dem Programmiergerät u.a. mit, welche Programmiervorgang erforderlich ist. Da die Programmiervorgang bei den möglichen EPROMs in einem Bereich zwischen 12,5 und 25 Volt liegt, muß diese Eingabe sehr sorgfältig ausgeführt werden; eine Zerstörung des EPROMs ist durch falsche Eingaben schnell möglich. Ein typischer EPROM-String lautet z.B.:

2764LF2 für einen EPROM vom Typ 2764; L-EPROM, Fast-Programmieralgorithmus, Zielblock 2

oder

27128ALF4 EPROM vom Typ 27128A; L-EPROM, fast, Zielblock 4

Die verallgemeinerte Form für den EPROM-String lautet:

Typ a b n mit Typen von 2716 bis 27256 unter Einbeziehung der Typen 2732A bis 27128A

a = U oder L für U- oder L-EPROM

b = N oder F für Normal- oder Fast-Programmieralgorithmus

n = Nummer des Zielblockes im EPROM, je nach Typ von 1 bis maximal 8

Bemerkungen zum Typ

Die Bezeichnung des Typs (z.B. 2716, 2764, 2764A) enthält die Speicherkapazität und die Programmiervorgang des jeweiligen EPROMs. Sie ist angelehnt an die Typenbezeichnung durch die Hersteller, die sich auch auf dem EPROM befindet. Die einzelnen Hersteller fügen dieser Bezeichnung weitere Zeichen oder Ziffern hinzu, so daß das EPROM 2764 z.B. bei der Firma Hitachi die Aufschrift HN482764G trägt und kompatibel zum Typ D2764 der Firma Intel oder auch zum Typ TMS2764 der Firma Texas Instruments ist. Leider trifft dies nicht immer zu; so ist beispielsweise das EPROM TMS2716 kein 2716, sondern der Typ TMS2516 entspricht dem 2716 anderer Hersteller. Neben den 'normalen' EPROMs, die ein Fenster zum Löschen mit UV-Licht besitzen, können auch solche Speicherbausteine programmiert werden, die kompatibel zu den jeweiligen EPROMs sind und keine Fenster besitzen. Bei Intel heißen sie 'Production EPROM' (z.B. P2764); andere Hersteller wie Hitachi benutzen andere Namen, da man sich nicht einig zu sein scheint, ob EPROM nun für 'Electrically Programmable Read Only Memory' oder für 'U.V. Erasable and Programmable Read Only Memory' steht. Des Weiteren können auch CMOS-EPROMs programmiert werden, obwohl diese aufgrund ihres noch recht hohen Preises in der Regel selten für HP-41-EPROM-Boxen verwendet werden. Sie sind zu erkennen an dem Buchstaben C nach den Ziffern 27 (z.B. HN27C64G).

FORTGESCHRITTENE EPROM-PROGRAMMIERUNG

Nicht programmiert werden können EEPROMs oder E²PROMs. Die folgende Tabelle enthält einige Herstellerbezeichnungen und die hier verwendete Typenbezeichnung und Programmierspannung. Sollten die von Ihnen verwendeten EPROMs nicht aufgeführt sein, erkundigen Sie sich bitte bei Ihrem Händler über die Kompatibilität zu den genannten Typen

Typ	V _{pp}	Intel	Hitachi	Fujitsu	TI	NEC
2716	25V	D2716	HN462716G	MBM2716	TMS2516	D2716D
2732	25V	D2732	HN462732G	MBM2732	TMS2732	D2732D
2732A	21V	D2732A	HN482732AG	MBM2732A		D2732A
2764	21V	P2732A		MBM27C32A		
		D2764	HN482764G	MBM2764	TMS2764	D2764D
		P2764	HN482764P	MBM27C64		
2764A	12,5V		HN27C64G			
		D2764A				
		P2764A				
2712B	21V	D27C64				
		D2712B	HN4827128G	MBM2712B	TMS2712B	D27128D
27128A	12,5V	P2712B	HN4827128P			
		D27128A				
27256	12,5V	P27128A				
		D27256	HN27256G			

Hinter der Herstellerbezeichnung stehen oft noch weitere Zeichen, z.B. -25 oder -3, die sich auf die Zugriffszeit des EPROMs beziehen. Die Zugriffszeit ist jedoch nur bei schnelleren Rechnersystemen von Interesse; wählen Sie für den HP-41 den preiswertesten Typ.

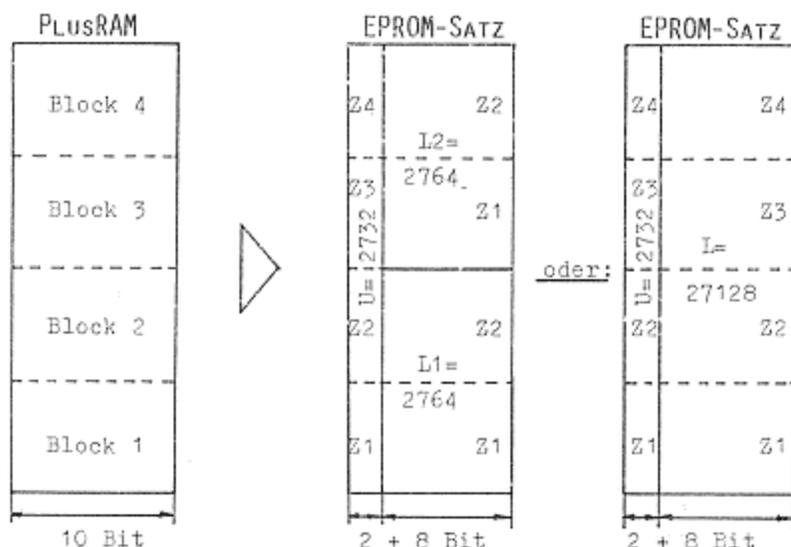
U- oder L-EPROM

Das nächste Zeichen hinter dem Typ ist im EPROM-String der Buchstabe U oder L für U- oder L-EPROM. Da der HP-41 im ROM-Bereich Worte mit einer Länge von 10 Bit verwendet, die handelsüblichen EPROMs jedoch in 8-Bit-Worten organisiert sind, muß das HP-41-Wort auf mehrere EPROMs aufgeteilt werden. Diese Aufteilung geschieht in der EPROM-Box nach folgendem Schema: Die niederwertigen 8 Bits (Bit 0 bis 7) eines Wortes befinden sich im L-EPROM, die höherwertigen 2 Bits (Bit 8 und 9) befinden sich zusammen mit den höherwertigen Bits von drei folgenden Worten im U-EPROM. Die Abkürzungen L und U haben sich eingebürgert und stehen für 'lower 8 bit' und 'upper 2 bit', den ursprünglichen englischsprachigen Bezeichnungen. Es sind also immer mindestens 2 EPROMs in einem Satz vorhanden, ein U- und ein L-EPROM, die nacheinander programmiert werden müssen. Zusätzlich spaltet man in manchen Fällen noch den L-Teil aus Kostengründen auf zwei EPROMs auf; dann wird aber nicht mehr das einzelne Wort weiter aufgeteilt, sondern die Zielblöcke.

FORTGESCHRITTENE EPROM-PROGRAMMIERUNG

Zielblock

Ein 16K-EPROM-Satz für den HP-41 kann z.B. aus einem EPROM 2732 und zwei EPROMs 2764 bestehen. Das U-EPROM 2732 enthält dann den U-Teil von allen 4 PlusRAM-Blöcken, also den U-Teil der Zielblöcke 1 bis 4. Das erste der beiden EPROMs 2764 (er wird auch L1-EPROM genannt) enthält den L-Teil der PlusRAM-Blöcke 1 und 2 in den Zielblöcken 1 und 2. Das zweite EPROM 2764 (L2) enthält den L-Teil der ProfiSET-Blöcke 3 und 4, jedoch ebenfalls in seinen Zielblöcken 1 und 2. Verwendet man statt des genannten EPROM-Satzes als L-EPROM einen 27128, enthält dieser den gesamten L-Teil der 16K in seinen Zielblöcken 1 bis 4. Die Grafik zeigt die Aufteilung der PlusRAM-Blöcke in L- und U-Teil und die Zielblöcke in den EPROMs:

Programmialgorithmus

Nach Definition des U/L-EPROMs folgt im EPROM-String die Bezeichnung des Programmialgorithmus mit den Zeichen N für Normal oder F für Fast (schnell). Beim Normal-Programmialgorithmus wird pro Byte eine feste Programmierzeit von 50 Millisekunden benötigt; dies führt zu einer Gesamtdauer von z.B. 4 Minuten für einen 2764 oder ca. 20 Minuten für einen 16K-Satz. Der Fast-Programmialgorithmus paßt die Programmierzeit jedem einzelnen Byte im EPROM an; die Gesamtdauer liegt in der Regel unter einer Minute für einen 2764. Leider ist der Fast-Algorithmus nicht auf alle EPROM-Typen anwendbar; die Typen 2716 und 2732 dürfen nicht Fast-programmiert werden. Bei den Typen 2764 und 27128 sind die

FORTGESCHRITTENE EPROM-PROGRAMMIERUNG

Herstellerangaben unterschiedlich; die allgemeine Tendenz geht jedoch in Richtung auf Zulassung dieser von der Firma Intel entwickelten schnellen Programmiermethode. Manche Typen 2764 und 27128 und alle 27256 müssen im Fast-Programmialgorithmus programmiert werden. In der folgenden Tabelle sind Informationen nach unserer derzeitigen Kenntnis zusammengefaßt:

Typ	Angaben zu Fast/Normal
2716	muß Normal programmiert werden.
2732	muß Normal programmiert werden.
2764	Intel: D2764 kann Fast, P2764 muß Fast, alle A- und CMOS-Typen müssen Fast programmiert werden. Hitachi: Alle 2764-Typen können Fast programmiert werden. Fujitsu: Alle 2764-Typen können Fast programmiert werden. AMD: Alle 2764-Typen können Fast programmiert werden.
27128	Intel: D27128 kann Fast, alle anderen 27128-Typen müssen Fast programmiert werden. Hitachi: wie 2764 Fujitsu: wie 2764 AMD: wie 2764
27256	alle Hersteller: muß Fast programmiert werden.

Im EPROM-String sind bei den Typen 2764 und 27128 beide Programmieralgorithmen zulässig; das Programm TPROM geht allerdings bei diesen Typen davon aus, daß auch Sie keine Zeit zu verschenken haben und die Freigabe des Fast-Algorithmus durch die anderen Hersteller nicht abwarten wollen.

FORTGESCHRITTENE EPROM-PROGRAMMIERUNG

In der nachfolgenden Tabelle mit den möglichen EPROM-Strings für die verschiedenen EPROM-Sätze ist für die Typen 2764 und 27128 der Fast-Algorithmus vorausgesetzt.

Satzgröße	U-EPROM	L1-EPROM	L2-EPROM	Ziel-Block	Anzahl L-EPROMs
4K	2716UN1	2732LN1	-	1	1
8K	2716UN1	2732LN1	-	1	2
	2716UN2	-	2732LN1	2	
8K	2716UN1	2764LF1	-	1	1
	2716UN2	2764LF2	-	2	
16K	2732UN1	2764LF1	-	1	2
	2732UN2	2764LF2	-	2	
	2732UN3	-	2764LF1	3	
	2732UN4	-	2764LF2	4	
16K	2732UN1	27128LF1	-	1	1
	2732UN2	27128LF2	-	2	
	2732UN3	27128LF3	-	3	
	2732UN4	27128LF4	-	4	
32K	2764UF1	27128LF1	-	1	2
	2764UF2	27128LF2	-	2	
	2764UF3	27128LF3	-	3	
	2764UF4	27128LF4	-	4	
	2764UF5	-	27128LF1	5	
	2764UF6	-	27128LF2	6	
	2764UF7	-	27128LF3	7	
	2764UF8	-	27128LF4	8	
32K	2764UF1	27256LF1	-	1	1
	2764UF2	27256LF2	-	2	
	2764UF3	27256LF3	-	3	
	2764UF4	27256LF4	-	4	
	2764UF5	27256LF5	-	5	
	2764UF6	27256LF6	-	6	
	2764UF7	27256LF7	-	7	
	2764UF8	27256LF8	-	8	

Die in der Spalte 'Zielblock' angegebene Nummer muß nicht identisch sein mit dem selektierten Block im ProfISET; sie bezeichnet die Reihenfolge der Blöcke in der EPROM-Box. Quellblock ist immer der jeweils selektierte Block im PlusRAM. Das Programmieren von 32K-Sätzen muß in zwei Hälften nacheinander durchgeführt werden, da als Quelle immer nur die 16K des PlusRAMs zur Verfügung stehen. Falls Sie weniger Blöcke, als die jeweilige Satzgröße umfaßt, programmieren wollen, müssen die restlichen mit leeren Blöcken im ProfISET programmiert werden. Beispiel: Wenn Sie 12K Software zur Verfügung haben und diese in einen 16K-EPROM-Satz programmieren möchten, dann müssen Sie einen Block im PlusRAM mit NEWBL formatieren und den nicht benutzten Zielblock im EPROM-Satz mit dem Inhalt des leeren Blockes programmieren. Geschieht dies nicht, wird sich der HP-41 beim Anschluß der EPROM-Box 'aufhängen'; der Inhalt des Hauptspeichers kann dabei verlorengehen, und der Batteriesatz muß möglicherweise vor dem erneuten Einschalten entfernt werden.

FORTGESCHRITTENE EPROM-PROGRAMMIERUNG

Zusammenfassung der zulässigen Parameter für den EPROM-String:

EPROM-TYP	U/L	N/F	Zielblöcke	Zielblöcke	TYP-Kürzel*
			U	L	
2716	U	N	1 bis 2	-	16
2732	U,L	N	1 bis 4	1	32
2732A	U,L	N	1 bis 4	1	2A
2764	U,L	N,F**	1 bis 8	1 bis 2	64
2764A	U,L	N,F**	1 bis 8	1 bis 2	4A
27128	L	N,F**	-	1 bis 4	28
27128A	L	N,F**	-	1 bis 4	8A
27256	L	F	-	1 bis 8	56

* Das Typ-Kürzel reicht zur Angabe des EPROM-Typs aus; statt 27128BALF3 kann also auch der EPROM-String 8ALF3 eingegeben werden.

** fast-programming aus Zeitgründen empfehlenswert.

Einzelfunktionen des EPROM-Programmervorganges, der bei dem im Kapitel 2 beschriebenen Standardfall automatisch abläuft, können bei Bedarf auch einzeln durchgeführt werden. Sie sind nachstehend erläutert.

EPROM?

Die Funktion EPROM? erfordert keine Eingaben; sie führt einen kurzen Selbsttest (Einstellung des ProfiSETs, Verbindung zum EPROMer und Funktion des EPROMers) durch und überprüft anschließend, ob ein EPROM im EPROMer eingesetzt ist. Nebenbei wird der EPROMer in einen Zustand versetzt, in dem er bereit ist zur Aufnahme eines EPROMs. Wenn alle Tests ohne Fehler abgelaufen sind, wird das Programm ohne Meldung fortgesetzt. Ist eine Fehlerbedingung aufgetaucht, blinkt die rote 'BAD'-Kontrollleuchte am EPROMer, vom Rechner ertönt ein Signal im gleichen Rhythmus, und in der Anzeige des Rechners erscheint eine Fehlermeldung (siehe Anhang 'Profiset-Fehlermeldungen'). Wird nun eine beliebige Taste am Rechner betätigt, hören das Blinken und der Ton auf, die Meldung bleibt jedoch in der Anzeige stehen, die 'BAD'-Leuchte brennt weiter. Dieser Zustand wird auch ohne die Betätigung einer Taste nach 40 Signalen erreicht. Wird keine weitere Taste betätigt, schaltet sich der Rechner nach ca. 10 Minuten aus; die rote Kontrollleuchte am EPROMer erlischt jedoch frühestens bei der Ausführung der nächsten EPROMer-Funktion.

FORTGESCHRITTENE EPROM-PROGRAMMIERUNG

EP=FF?

Wird ein neues oder gerade gelöscht ERPOM in das Programmiergerät eingesetzt, kann dieses mit der Funktion EP=FF? auf korrekte Löschung (Blank-Check) überprüft werden. Diese Funktion überprüft immer das gesamte EPROM; sollte also bereits ein Block im EPROM programmiert sein, führt dies zu einer Fehlermeldung. Als Eingabe ist nur der EPROM-String im ALPHA-Register notwendig, wobei in diesem Fall nur die Typ-Angabe von Belang ist. Es kann darum auch eine verkürzte Form des EPROM-Strings, die nur aus dem Typ besteht, zur Anwendung kommen (z.B. 2716 statt 2716UN1). Ansonsten verhält sich die Funktion EP=FF? wie die Funktion EPROM? ; sie schließt diese sogar, wie alle anderen EPROMer-Funktionen, mit ein.

EP=BL?

Die Funktion EP=BL? dient zum Vergleich eines Zielblockes auf einem EPROM mit dem Quellblock im PlusRAM. Nach einem Programmiervorgang braucht diese Funktion nicht aufgerufen zu werden, da sie bereits in der Programmierfunktion PROMBL enthalten ist. Eine sinnvolle Anwendung kann jedoch die Überprüfung eines fertigen EPROM-Satzes sein, der Quell-Block muß jedoch in unveränderter Form vorhanden sein. Die Funktion benötigt den kompletten EPROM-String, in dem jedoch die Angabe über den Programmieralgorithmus (N/F) ohne Belang ist. Verglichen (verifiziert) wird der durch den EPROM-String bezeichnete Zielblock mit dem selektierten Block im PlusRAM.

PROMBL

Die eigentliche Programmierfunktion ist die Funktion PROMBL. Sie beinhaltet die Funktion EPROM? (Selbsttest), das Programmieren eines Blocks und die Funktion EP=BL? (Verifizieren) und erfordert als Eingaben den EPROM-String und die Selektierung eines PlusRAM-Blockes. Je nach Programmieralgorithmus unterscheiden sich Anzeige und Bedienungsmöglichkeiten der Funktion: Im Fast-Algorithmus erscheint während der Programmierung nur die Anzeige PROGRAM FAST ; es ist keine Unterbrechung möglich. Im Normal-Algorithmus wird die jeweilige Speicheradresse im Block mit angezeigt (z.B. PROGRAM 4051), und die ON-Taste kann zur Unterbrechung des Vorganges benutzt werden. Wird im Normal-Algorithmus der Programmiervorgang durch einen Fehler oder manuell unterbrochen, erscheint die Nummer der Speicherstelle zusammen mit der Meldung in der Anzeige. Die Speicherstellen (Bytes) werden dezimal durchnumeriert, also von 0000 bis 4095 bei einem L-EPROM und von 0000 bis 1023 bei einem U-EPROM.

TEIL 3

ANHÄNGE

1. -PSET-BASE ProfiSET-Basis-Management-Funktionen
- CLEAR löscht ein Programm oder einen anderen File aus dem PlusRAM. Im CAT2 werden die Labels des Programms durch Platzhalter ersetzt. Diese Platzhalter werden bei erneuten Laden wieder überschrieben, die XROM-Nummern dadurch nicht verändert.
- CLRn löscht den im Alpha-Register stehenden Namen eines Blocks.
- COPYBL kopiert einen definierten Block (absolute Nummer seiner Adresse im X-Reg.) in den selektierten Block.
- IDBL gibt dem selektierten Block eine Identität durch den Vorkommateil der gewählten XROM-Nummer, die im X-Reg. steht.
- INIT initialisiert das ProfiSET bei der ersten Inbetriebnahme nach Datenverlust durch Überschreiten des Batteriewechselzeitraums oder nach Fehlbedienung bei der Microcodierung; löscht den gesamten PlusRAM.
- NAMEBL versieht den Block mit dem im Alpha-Register stehenden Namen; es können mehrere Namen in einen Block geschrieben werden, die als Überschriften des Blocks oder der nachstehenden Programme zu verstehen sind.
- NEWBL löscht den selektierten Block und reserviert die im X-Reg. angegebene Zahl von Einträgen in den CAT2 (Blockname, Labels, Funktionen); Blockformatierung.
- NEXTBL wählt den nächsten Block an; wie SLCT, jedoch ohne Eingabe ins X-Reg.
- PACKBL packt den selektierten Block.
- PACKC löscht alle Platzhalter aus dem CAT2. Achtung!: Die XROM-Nummern der anderen Programme verändern sich.
- REVB schreibt einen aus bis zu 4 Zeichen bestehenden String - die Revision - in den selektierten Block (zur Unterscheidung verschiedener Versionen fast gleicher Programme).
- SLCT selektiert den im X-Reg. angegebenen Block und macht ihn zum Arbeitsblock, bis dieser Status wieder durch Wahl eines anderen Blocks aufgehoben wird. Alle anderen Blöcke werden dadurch gegen unbeabsichtigtes Verändern gesichert.
- SCMBL bildet die Prüfziffer für einen Block mit Eintrag.
2. -PSET-INFO ProfiSET-Informations-Funktionen
- ADR>ID ersetzt die absolute Adresse im X-Reg. durch die Identität des Blocks auf dieser Adresse.
- BL? schreibt die Nummer des PlusRAM-Blocks, der selektiert ist, ins X-Reg.
- BLIA? schreibt die absolute Adresse des ersten PlusRAM-Blocks ins X-Reg.
- BLA? schreibt die absolute Adresse des selektierten Blocks ins X-Reg.
- BLCAT gibt eine Übersicht über den Inhalt des selektierten Blocks.
- BLCATX schreibt den Namen der x-ten (Inhalt des X-Reg.) Funktion ins Alpha-Register.
- BLF? zeigt an, wieviel freier Raum (freie Bytes) im selektierten Block verbleibt.
- CATF? zeigt die Anzahl der noch freien Einträge im CAT2 des Blocks an.
- CATS? gibt die Anzahl der insgesamt möglichen Einträge im CAT2 zurück (vgl. NEWBL).
- CHKBL vergleicht den Blockinhalt mit der Prüfziffer.
- ID>ADR schreibt bei gegebener Identität (im X-Reg.) die absolute Adresse des Blocks mit dieser Identität in das X-Reg.

ID?	schreibt die Identität des selektierten Blocks ins X-Reg.
LENG?	schreibt die Länge eines Programms oder eines anderen Files im PlusRAM in das X-Reg.
REV?	schreibt die ROM-Revision nach Alpha (vergl. REVBL).
SRV	zeigt im Display den Text 'SERVICE' mit MM.JJ (Monat.Jahr) für den Batterie- wechsel an.
TYP?	schreibt eine Textkette mit der Typenbezeichnung des Files, der Funktion oder des Programms nach X, z.B.:
P	Anwenderprogramm (USER-Code)
PPV	privatisiertes Anwenderprogramm (USER-Code)
Dly	Datenfile mit Integerwerten und y als Anzahl der Stellen der Mantisse
DFM	Datenfile mit normaler HP-41-interner Darstellung
KEY	Tastenzuordnungs-File
F?	Funktion, die nicht ProfISET-formatiert ist
fn	ProfISET-formatierte Funktion Typ n (vergl. Handbuch)

3. -PSET-PRGM ProfISET-Anwender-Programm-Funktionen

LADEP	lädt ein Programm in den PlusRAM und trägt die Labels in den CAT2 ein. Das Programm wird automatisch gepackt; die Sprungweiten werden berechnet und eingetragen.
PRVBL	privatisiert den selektierten Block und alle darin vorkommenden Programme.
PRVP	privatisiert ein einzelnes Programm im PlusRAM.
XROMBL	führt im Block bei allen Programmen die Umwandlung von XEQ nach XROM durch.
XROMP	wandelt alle XEQ-Aufrufe in einem Programm in XROM um, soweit das entsprechende Programm oder die Funktion im ROM oder im PlusRAM existiert und nicht im CAT1- Speicher des Rechners steht. Soll ein bestimmter Aufruf nicht umgewandelt werden, so muß sein Label vorher in den Programmspeicher des Rechners eingetragen werden.

4. -PSET-DATA ProfISET-Data-Management-Funktionen

Im PlusRAM können Datafiles mit verschiedenen Namen angelegt werden, die zur Aufnahme größerer
Datenmengen dienen. Sind Größenordnungen/Genauigkeiten der Werte bekannt, wählen Sie die dafür
spezifizierte Funktion, um möglichst platzsparend viele Zahlen unterzubringen. Den Namen des
Datafiles schreiben Sie ins Alpha-Reg. und rufen eine der Funktionen auf; in X steht die Anzahl
der voraussichtlich zu speichernden Zahlen (max. 999):

DATI _n	erzeugt Datafiles für Integerwerte mit n signifikanten Ziffern. Es gibt: DATI2 / DATI4 / DATI7 / DATI9
DATEN	erzeugt Datafiles für reelle Zahlen, die in der (exponentialschreibweise n signifikante Ziffern (Stellen) haben bei zweistelligen Exponenten: DATE2 / DATES / DATE7
DATEN	erzeugt Datafiles für normale im HP-41-Format dargestellte reelle Zahlen.
▷DATY	schreibt die Zahl vom X-Reg. in den durch das Y-Reg. bestimmten Platz (Register) des Datafiles; promptet für den Filenamen bei manuellem Aufruf; entnimmt den Filenamen bei Programmablauf dem ALPHA-Reg.

Alle Datafiles ersetzen bei Aufruf ihres Namens (XEQ NAME) das X-Register durch die
Zahl, die durch ihren Index im X-Reg. bestimmt wird; indirekter Datenrückruf.

5. -PSET-MASS ProfISET-Massenspeicher-Funktionen

- GETBL lädt einen mit SAVEBL auf Cassette aufgezeichneten Block in den selektierten Block des PlusRAMs und überprüft die Prüfziffer.
- GETF lädt eine für das ProfISET formatierte Microcode-Funktion vom Laufwerk in den selektierten Block und trägt sie in den CAT2 ein.
- SAVEBL zeichnet den selektierten Block als Profi-File von 640 Registern auf dem Digital-Cassetten-Laufwerk auf. Der Filename für das Directory wird vorher in das Alpha-Register geschrieben. Die Prüfziffer wird automatisch gebildet.
- SAVEF ist das Gegenstück zu GETF ; mit dieser Funktion können auch Datafiles und KEY-Files aus dem ProfISET abgespeichert werden.

6. -PSET-PSUB ProfISET-Programmier-Subroutines

- CLRS löscht im ProfISET die Autostart-Option (falls vorhanden).
- EPACK wandelt im angegebenen Programm alle Ziffernfolgen vom Typ 1 E.. in E.. um und packt das Programm (spart Bytes!).
- J? stoppt das laufende Programm mit Anzeige des Alpha-Reg. (wie PROMPT); zugleich blinkt am rechten Rand der Anzeige ein Fragezeichen; die Tastatur ist bis auf ON, NEIN (ENTER) und JA (R/S) blockiert; bei JA weiter mit dem nächsten Programmschritt, bei NEIN mit dem übernächsten.
- L* druckt in Abhängigkeit von Flag 12 eine Sternlinie mit 12 oder 24 Zeichen.
- L- druckt eine Strichlinie analog L*.
- L= druckt eine Doppellinie analog L*.
- LADEK lädt die aktuellen Tastenzuordnungen (außer Hauptspeicherprogramme) in den selektierten Block; promptet für einen Namen; wird der KEY-File mit XEQ aufgerufen, werden alle vorhandenen Zuordnungen gelöscht und durch die 'gespeicherten' ersetzt.
- LADES lädt in den selektierten Block die Autostart-Option; sie veranlaßt den Rechner beim Einschalten, den Befehl GTO START auszuführen. Sie können z.B. in einem LBL START ein Hauptmenue programmieren oder den Rechner auf einen bestimmten Beginn-Status einstellen.
- N? wie J?; springt jedoch bei JA und führt bei NEIN den nächsten Schritt aus.
- PKEY löscht alle Tastenfeldzuordnungen und belegt die Tastatur mit dem ProfISET-Keyboard (vergl. Overlay zum ProfISET).
- PRM druckt den Inhalt des Alpha-Reg. mittig auf einem 24-Zeilen-Drucker aus.

7. -PSET-PROM ProfISET-EPROM-Programmier-Funktionen (setzen MBK-EPROMer voraus)

- EP-BL? vergleicht (verifiziert) den Inhalt eines Zielblockes im EPROM mit dem selektierten Block im PlusRAM auf Übereinstimmung; erforderlich Eingabe des EPROM-Strings im ALPHA-Register
- EP-EF? Blank-Check des gesamten EPROMs; Typ-Eingabe im ALPHA-Reg.
- EPROM? Selbsttest EPROMer/ProfISET; EPROM vorhanden?
- PROMBL programmiert den selektierten 4K-Block aus dem ProfISET in das EPROM; erfordert Eingabe des kompletten EPROM-String im ALPHA-Reg.
- *PROM Dialog-Programm zur automatischen Programmierung von Standard-EPROM-Sätzen für HP-41-EPROM-Boxen

<u>Identität</u>	<u>Modul /Geräte-Bezeichnung</u>
ø1	Mathematik-Modul
ø2	Statistik-Modul
ø3	Surveying-Modul
ø4	Finanz-Modul
ø5	Standard-Modul / ZENROM
ø6	Circuit Analysis
ø7	Structural
ø8	Stress Analysis
ø9	Home Management / CCD-Modul
1ø	Games / Auto-Dup /PPC-ROM
11	Real Estate / CCD-Modul
12	Machine Design, PPC-EPROM2
13	Thermal & Transport
14	Navigation-Modul
15	Petrol-Modul / MC-EPROM
16	Petrol-Modul
17	Plotter-Modul / NFCROM
18	Plotter-Modul
19	Aviation / Clinical Lab / Securities / Structural / PPC-EPROM5
2ø	PPC-ROM
21	Assembler 3
22	HP-IL Development-Modul
23	Extended-I/O
24	HP-IL Development-Modul
25	Extended Functions (auch HP-41CX)
26	TIME-Modul (auch HP-41CX)
27	Lesestift / Profiset
28	HP-IL-Modul (Bandlaufwerk, Control-Functions)
29	Drucker (IL- und Einsteckdrucker)
3ø	Kartenleser
31	Profiset

Sollten Sie Module von Software-Häusern besitzen und deren ID-Nummern nicht kennen, können Sie das Modul in den nicht belegten Port des HP-41 stecken, dessen Adreßbereichs-Nummer (vergl. Anhang 3, untere 4K) eingeben und die Funktion `ADR>ID` aufrufen; im X-Reg. steht dann die ID. Bei 8K-Modulen erhöhen Sie die Adresse um 1 und erhalten die zweite ID. Ergänzen Sie die obige Tabelle für Ihre Zwecke.

Das Betriebssystem des MBK-ProfiSETs enthält einige eigene Fehlermeldungen zur besseren Benutzerführung. Allen ProfiSET-Fehlermeldungen ist gemeinsam, daß sie mit ERR'... im Klartext den Fehler erklären.

<u>Fehlermeldung</u>	<u>Erklärung/Hinweise</u>
ERR'CAT FULL	Der durch NEWBL vorgewählte Raum für Einträge im CAT2 reicht nicht. Wählen Sie einen anderen Block und führen Sie die Funktion erneut aus. Unter bestimmten Bedingungen kann diese Meldung auch auftreten, wenn nicht genügend Raum zum Laden eines Programmes vorhanden ist (statt ERR'NO ROOM).
ERR'DUP ID	Es wurde versucht, eine Identität zu vergeben, die bereits im System vorhanden ist. Wählen Sie eine andere Identität oder entfernen Sie den Block oder die Einsteck-einheit mit der gleichen Identität.
ERR'FUNCTION	Es wurde versucht, eine Funktion, die nicht auf einen Datenfile oder eine Funktion angewendet werden darf, mit dem Namen eines Datenfiles oder einer Funktion durchzuführen.
ERR'NAME	Der von Ihnen eingegebene Name entspricht nicht den vorgegebenen Vereinbarungen (z.B. weniger als 8 Zeichen bei der Funktion NAMEBL).
ERR'NO COMPK	Das geladene Programm enthält einen lokalen XEQ- oder GTO-Befehl, zu dem kein Label existiert. Betätigen Sie die  -Taste: Sie sehen die Programmzeile, die zu dieser Fehlermeldung geführt hat. Das Programm muß anschließend mit CLEAR gelöscht und im Hauptspeicher des Rechners berichtigt werden, bevor es erneut geladen werden kann.
ERR'NO COPY	Der zu kopierende Block besitzt den MBK-Kopierschutz (NO COPYRIGHT). Sie erhalten diese Meldung auch, wenn Sie versuchen, das ProfiSET-Betriebssystem zu kopieren.
ERR'NO DATA	Der angesprochene File ist kein Datenfile.
ERR'NO HPIL	Es ist keine HP-Interfaceschleife vorhanden.
ERR'NO ID	Der selektierte Block ist nicht vollständig initialisiert. Führen Sie die Funktionen NEWBL und IDBL aus.
ERR'NO LABEL	Es wurde versucht, ein Programm ohne globales Label zu laden.
ERR'NO PROF I	Der angesprochene File ist nicht ProfiSET-gerecht formatiert.
ERR'NO PS-1	Der erste Block des ProfiSET-Betriebssystems ist nicht vorhanden. Überprüfen Sie die Einstellung des ProfiSETs und die Portbelegung des Rechners.
ERR'NO +RAM	Es ist kein PlusRAM vorhanden, oder es wurde ein nicht existierender Block angewählt. Überprüfen Sie die Einstellung des ProfiSETs und die Portbelegung des Rechners.

PROFISSET-FEHLERMELDUNGEN

ERR'NO ROOM	Es ist nicht genügend Raum im angewählten Block vorhanden. Selektieren Sie einen anderen Block, wenn nach EPACK, PACKBL o.ä. die gleiche Meldung erneut auftritt.
ERR'NO SLCT	Es ist kein PlusRAM-Block selektiert.
ERR'PROGRAM	Es wurde versucht, eine Funktion, die nicht auf ein Benutzerprogramm angewendet werden darf, mit dem Namen eines Benutzerprogrammes durchzuführen.
ERR'PZ	Die Kontrolle der Prüfziffer (z.B. mit CHKBL) hat einen Fehler ergeben. Es gibt verschiedene Ursachen für das Auftreten dieses Fehlers. Zum einen kann es sein, daß Sie nach der letzten Prüfziffernbildung in diesem Block Veränderungen vorgenommen oder in diesem Block noch keine Prüfziffer mit SUMBL gebildet haben. Tritt die Meldung nach dem Laden eines Blockes vom Bandlaufwerk auf, so liegt der Fehler möglicherweise im Bandlaufwerk oder in der IL. Beim Überschreiten des Service-Datums zeigt diese Meldung möglicherweise einen Datenverlust im PlusRAM an.
ERR'ROM	Der angesprochene File befindet sich nicht im PlusRAM.

In Verbindung mit dem MBK-EPROMer können bei EPROM-Programmierarbeiten zusätzlich folgende Fehlermeldungen auftreten:

ERR'AT / ERR'AT FAST	Der Programmiervorgang wurde unterbrochen, weil eine Speicherzelle des EPROMs nicht programmiert werden kann. Im Normal-Programmiermodus sehen Sie in der Anzeige die Nummer der Speicherzelle. Löschen Sie das EPROM und wiederholen Sie den Programmiervorgang. Tritt der Fehler erneut auf, ist das EPROM wahrscheinlich defekt; (Hinweis: Es kann jedoch möglicherweise noch zur Programmierung mit einem anderen Inhalt verwandt werden).
ERR'BAD CHIP	Das EPROM ist defekt oder wurde vor dem Programmieren nicht gelöscht.
ERR'LOW BATT	Die Leistung der Batterien des Rechners reicht nicht zur kompletten Durchführung des Programmiervorganges. Schalten Sie den Rechner aus, wechseln Sie die Batterien und starten Sie das Programm nach dem Einschalten des Rechners mit R/S erneut, ohne zwischenzeitlich die Verbindung zwischen Profiset und Programmiergerät bzw. Rechner und Profiset zu trennen.
ERR'NO EPROM	Es befindet sich kein oder ein defekter EPROM im Programmiergerät.
ERR'PSETPROM	Die PSET-PROM-Funktionen führen eine Überprüfung des EPROM-Programmiergerätes (Selbsttest) und der Einstellung des Profisets (System: Port 4, PlusRAM: Port 1+2) durch. Diese Meldung erscheint auch, wenn kein Programmiergerät am Profiset angeschlossen ist. Überprüfen Sie die

PROFISSET-FEHLERMELDUNGEN

Steckverbindung zwischen Profiset und Programmiergerät, die Einstellung des Profisets und die Stromversorgung des Programmiergerätes (ON-Anzeige).

- ERR'TEXT Eine PSET-PROM-Funktion kann die eingegebene Textkette nicht entschlüsseln. Überprüfen Sie den Inhalt des ALPHA-Registers auf seine Gültigkeit.
- STOP AT / Die Programmierfunktion wurde mit der ON-Taste abgebrochen und kann mit R/S wieder gestartet werden.
- STOP AT FAST

Andere Fehlermeldungen

Die Funktionen des Profisets rufen auch andere Fehlermeldungen des HP-41 auf, sofern diese zur Benutzerführung ausreichen:

- ALPHA DATA Ein Argument ist nicht numerisch.
- DATA ERROR Das Argument ist außerhalb des zulässigen Bereichs.
- DIR FULL siehe Bedienungshandbuch zum HP-IL-Modul
- DRIVE ERR dito
- DUP FL NAME dito
- FL NOT FOUND dito
- FL SECURED dito
- FL TYPE ERR dito
- MEDM ERR dito
- MEDM FULL dito
- NAME ERR dito
- NO DRIVE Die HPIL enthält kein Bandlaufwerk.
- NO MEDM siehe Bedienungshandbuch zum HP-IL-Modul
- NONEXISTENT Der angesprochene File ist nicht vorhanden, oder das Argument liegt außerhalb des zulässigen Bereichs.
- OUT OF RANGE Bei der Ausführung der Funktion X>DATY wurde der zulässige Bereich für den Integer-Datenfile überschritten.
- PRIVATE Das angesprochene Programm ist geschützt.
- RAM Das angesprochene Programm befindet sich im Hauptspeicher des Rechners.
- ROM Sie versuchen, unmittelbar im ROM-Bereich zu arbeiten.

Der ROM-Bereich des HP-41 ist in 16 Blöcke zu je 4096 Worten (4K) eingeteilt. Die Tabelle gibt eine Übersicht über den Inhalt der einzelnen Blöcke mit ihren Adreß-Bereichen.

<u>Adresse</u>	<u>Inhalt</u>
ø	HP-41-Betriebssystem, Teil 1
1	HP-41-Betriebssystem, Teil 2
2	HP-41-Betriebssystem, Teil 3
3	X-Functions beim CX, sonst frei
4	Service-Modul / IL-Drucker, wenn Schalter auf disable
5	Time-Modul und CX-Funktionen*
6	Drucker / IL-Drucker, wenn Schalter auf enable
7	IL-Modul (Mass.Stor.-, IL-Functions)
8	Port 1, erste Hälfte (untere 4K)
9	Port 1, zweite Hälfte (obere 4K)
10	Port 2, erste Hälfte (untere 4K)
11	Port 2, zweite Hälfte (obere 4K)
12	Port 3, erste Hälfte (untere 4K)
13	Port 3, zweite Hälfte (obere 4K)
14	Port 4, erste Hälfte (untere 4K) / Kartenleser
15	Port 4, zweite Hälfte (obere 4K)

* Der Block 5 ist beim HP-41CX doppelt belegt. In der zweiten Belegung enthält er einen Teil der CX-Funktionen; dieser zweite Teil kann mit normalen Mitteln nicht erreicht werden (z.B. durch COPYBL).

Die Funktionen TYP? und BLCAT geben Ihnen zu den verschiedenen Files im Prof1SET den Typ. Die folgende Tabelle gibt Aufschluß über die einzelnen File-Typen:

<u>Typ</u>	<u>Bezeichnung</u>	<u>erzeugen</u>	<u>löschen</u>	<u>ablegen</u>
N	Überschrift im CAT2 (Name)	NAMEBL	CLRN	--
P	Programm	LADEP	CLEAR	COPY,WRTP
PPV	PRIVATE-geschütztes Programm	PRVP	CLEAR	--
KEY	Tastenzuordnungen	LADEK	CLEAR	SAVEF
DE2	Datenfile, 2stell. Mantisse+Exp.	DATE2	CLEAR	SAVEF
DE5	Datenfile, 5stell. Mantisse+Exp.	DATE5	CLEAR	SAVEF
DE7	Datenfile, 7stell. Mantisse+Exp.	DATE7	CLEAR	SAVEF
DEN	Datenfile, 10stell. Mantisse+Exp.	DATEN	CLEAR	SAVEF
D12	Datenfile, 2stell. Mantisse	DAT12	CLEAR	SAVEF
D14	Datenfile, 4stell. Mantisse	DAT14	CLEAR	SAVEF
D17	Datenfile, 7stell. Mantisse	DAT17	CLEAR	SAVEF
D19	Datenfile, 9stell. Mantisse	DAT19	CLEAR	SAVEF
FØ	Funktion, verschiebbar	--	CLEAR	SAVEF
F1	Funktion, nicht verschiebbar*	--	CLEAR	--
F?	Funktion, kein Prof1SET-Format*	--	--	--
	Automatischer Start**	LADES	CLRS	--
	Zusätzlicher PRIVATE-Schutz**	PRVBL	--	--

* Diese File-Typen werden bei der Ausführung von PACKBL nicht verschoben.

** Diese File-Typen besitzen keinen Namen und erscheinen weder im CAT2 noch im BLCAT, belegen jedoch einen Teil des Blocks (PRIVATE: 14 Bytes, Start: 33 Bytes); es kann immer nur einer dieser File-Typen vorhanden sein.

Dieser Abschnitt soll dem MCode-Programmierer zusätzliche Informationen über den Blockaufbau durch das ProfiSET-Betriebssystem geben, um alle Möglichkeiten seiner Nutzung auszuschöpfen.

Selektierung

Die SLCT-Information steht im Bit 8 auf der Adresse xFFE, die gleichzeitig das erste Zeichen der Revision enthält. Ist beispielsweise die Revision gleich ABCD, so steht auf der Adresse das Datenwort $\emptyset\emptyset 1$ ($\emptyset 1$ = Zeichen A), wenn der Block selektiert ist. Wird ein anderer Block selektiert, so wird Bit 8 im nicht selektierten Block gesetzt; in unserem Beispiel stände auf xFFE nun $1\emptyset 1$. Sollten mehrere Blöcke gleichzeitig selektiert sein (mit dem ProfiSET-Betriebssystem nur durch INIT möglich), so wird immer der erste PlusRAM-Block (niedrigste Adresse) als selektierter Block erkannt.

Reservierung der CAT2-Einträge (FAT) durch NEWBL

Die Funktion NEWBL überschreibt den zu reservierenden Raum ab Adresse x $\emptyset\emptyset 4$ mit dem Datenwort 3FF (= Befehl JC- $\emptyset 1$). Auf xFFB wird $\emptyset 8\emptyset$ und auf xFF4 wird 3A \emptyset (= Befehl ?NC RTN) geschrieben; alle übrigen Worte werden mit $\emptyset\emptyset\emptyset$ überschrieben.

Platzhalter im CAT2

Wird eine File im ProfiSET gelöscht, so werden die Einträge in der FAT mit der Adresse xFFF überschrieben, falls hinter dem jeweiligen Eintrag weitere Einträge folgen. Als Platzhalter steht also nun die Revision als Funktion im CAT2. Aus diesem Grund wird nach jeder Funktion, die mit *OK* in der Anzeige endet (z.B. COPYBL), überprüft, ob die Revision einen zulässigen Funktionsnamen darstellt (Bit 7 in Wort xFFB wird automatisch gesetzt). Dies führt beim Kopieren des ZENROMs in das ProfiSET zu Schwierigkeiten, da dort entgegen aller Konventionen über die ROM-Blockstruktur ein Programmteil (!) im Platz für die Revision steht und dieser automatisch vom ProfiSET verändert wird. Benutzen Sie also das ZENROM nur als Einsteckmodul.

File-Typen im ProfiSET

Um das Verarbeiten von MCode-Funktionen im ProfiSET zu ermöglichen, wurde eine spezielle Codierung für File-Typ und -Länge eingeführt. Sie besteht aus drei Datenworten vor dem ersten Namen einer Funktion:

abX	ab = Typ-Kennzeichnung
$\emptyset YZ$	XYZ = Länge der Funktion
$\emptyset\emptyset F$	$\emptyset\emptyset F$ = ProfiSET-Formatierungskennung (Befehl JC $\emptyset 1$)
E	
M	
A	
N	

DIE FORMATIERUNG DURCH DAS PROFISSET

Die unterschiedlichen File-Typenkennzeichnungen ab sind wie folgt festgelegt:

<u>ab</u>	<u>Typ</u>	<u>Erklärung</u>
ØØ	FØ	MCode-Funktion ohne relocatable Sprungbefehle
Ø1	F1	andere MCode-Funktionen
Ø2	F2	wie F1
Ø3	F3	wie F1
Ø4	F4	wie F1
Ø5	F5	wie F1
Ø6	F6	wie F1
Ø7	F7	wie F1
Ø8	ON	wie F1
1Ø	D12	Data-Funktion
11	D14	Data-Funktion
12	D17	Data-Funktion
13	D19	Data-Funktion
14	KEY	Tastenzuordnungs-Funktion
3Ø	DE2	Data-Funktion
31	DE5	Data-Funktion
32	DE7	Data-Funktion
33	DEN	Data-Funktion
--	N	Steht ein RTN hinter einem Funktionsnamen, so wird dieser als Typ N erkannt.
--	F?	alle Funktionen ohne Kennung (außer N)
--	P	alle User-Code-Programme, nicht private
--	PPV	User-Code-Programme, private

Für eigene MCode-Funktionen können Sie alle F-Typen verwenden. Die Länge der Funktion muß dabei richtig angegeben werden:

XYZ = Adresse des letzten Worts der Funktion
 - Adresse des ersten Worts der Kennung
 + 1

Innerhalb der angegebenen Länge können beliebig viele Funktionsnamen stehen. FØ-Funktionen müssen so geschrieben sein, daß sie auf jeder beliebigen Adresse lauffähig sind; sie werden beim Packen durch PACKBL verschoben und können mit SAVEF auf Band abgelegt werden. F1-Funktionen bleiben beim Packen auf den alten Adressen stehen. Befindet sich eine F1-Funktion in der Mitte eines Blockes, so können demnach beim Packen zwei Lücken entstehen. FØ- und F1-Funktionen können mit CLEAR gelöscht werden.

Such-Programme

Alle Lade-Funktionen innerhalb des Profiset-Betriebssystems suchen die erste Lücke innerhalb des selektierten Blockes, die genügend Raum für den zu ladenden File bietet. Als Lücke ist im Profiset eine Folge von ØØØ-Worten definiert. Die kleinste verwendbare Lücke beträgt 9 Worte (Name mit 8 Zeichen + RTN); dabei wird keine Rücksicht darauf genommen, ob sich die Lücke innerhalb eines Files befindet. Es dürfen also in MCode-Funktionen keine Lücken mit mehr als 8 Worten vorkommen, da diese gegebenenfalls mit anderen Files gefüllt werden. Die Funktion BLF? gibt immer die größte Lücke innerhalb eines Blockes an. Befindet sich diese am Ende des Blockes, so wird nur bis zur Adresse xFF3 gezählt.

DIE FORMATIERUNG DURCH DAS PROFiSET

ProfISET-Files auf dem Bandlaufwerk

Die Funktion SAVEBL erzeugt auf dem Bandlaufwerk (oder auch Diskettenlaufwerk) einen File vom Typ 7 (HP-41) mit 20 Records. Dieser Typ wird nach unserer Kenntnis bisher von keinem anderen IL-Controller benutzt. Die Daten werden im folgenden Format abgelegt:

Wort:	\emptyset 1 2 3	
Bit :	$\underbrace{9876543210987654321098765432109876543210}_{\text{---}} $	im PlusRAM
Byte:	4 3 2 1 \emptyset	--- File auf Band

Die Funktion SAVEF erzeugt einen File vom Typ B (HP-41); auch dieser File-Typ auf dem Bandlaufwerk wird bisher nur vom ProfISET genutzt. Dieser File enthält neben der abgelegten Funktion deren Länge, die Anzahl und die Position der CAT2-Einträge. Es werden zwei Bytes auf dem Band für ein 10-Bit-Wort verwendet.

Sie können grundsätzlich alle von der Firma Hewlett-Packard produzierten Geräte und alle IL-Geräte gleichzeitig mit dem MBK-ProfiSET an Ihrem HP-41C, CV und CX betreiben, wenn Sie die Einschränkungen im Abschnitt 'Einstellung des ProfiSETs' beachten.

Portextender dürfen grundsätzlich nicht gleichzeitig mit dem ProfiSET an den Rechner angeschlossen werden, da dies zu einer Überlastung des Rechners bzw. zur Beschädigung des ProfiSETs führen kann.

EPRoM-Boxen können bei gleichzeitigem Anschluß mit dem ProfiSET ebenfalls zu einer Überlastung des Rechners führen. Die EPRoM-Box MBK-16 kann problemlos gleichzeitig mit dem ProfiSET angeschlossen werden, wenn der Prioritätsschalter auf ProfiSET (Punkt sichtbar) geschaltet wird.

Auch andere RAM-Erweiterungen können zusammen mit dem ProfiSET zur Überlastung des Rechners führen. Zwei MBK-ProfiSETs können gleichzeitig an einem HP-41 betrieben werden, wenn an beiden der Schalter 'System' auf OFF geschaltet wird.

Rechner mit erhöhter Taktfrequenz (Speed-Up)

Falls Ihr HP-41 auf erhöhte Taktfrequenz umschaltbar ist, betreiben Sie das ProfiSET bitte nur mit der normalen Taktfrequenz des Rechners. In der Regel wird das ProfiSET auch noch mit dem 1,5-fachen der normalen Taktfrequenz einwandfrei arbeiten; eine Garantie dafür wird nicht gegeben.

Achtung! Das EPRoM-Programmiergerät (MBK-EPRoMer) darf auf keinen Fall mit erhöhter Taktfrequenz betrieben werden.

Rechner mit eingebauten Modulen

Auch die in Ihren Rechner eingebauten Module belegen Steckplätze. Dies ist je nach Typ bei der Einstellung und Benutzung des ProfiSETs zu beachten.

Zusammenspiel mit Lesestift

Da eine Hälfte des ProfiSET-Systems dieselbe Identität (ID 27) wie der Lesestift hat, können einzelne Funktionen des Lesestifts nur aufgerufen werden, wenn er in einen Port mit niedrigerem Adreßbereich als das System eingesteckt ist. Die normale Anwendung des Lesestifts (Lesen von Barcodes ohne Funktionsaufruf) ist jedoch immer möglich. Umgekehrt können die EPRoMer-Funktionen nur aufgerufen werden, wenn kein Lesestift angeschlossen ist oder er sich in einem Port mit höherem Adreßbereich als das System des ProfiSETs befindet.

Gewährleistung

Das Mathematische Büro Köln gewährleistet, daß das MBK-ProfiSET frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn das Gerät direkt oder über Ihren Händler an folgende Adresse eingesandt wird:

Mathematisches Büro Köln
Dipl.-Math. A.B. Gemein
An der Flora 9
D-5000 Köln 60

Die Gewährleistungsfrist beträgt 12 Monate ab Verkaufsdatum. Schäden, die durch unsachgemäße Bedienung oder Gewalteinwirkung entstanden bzw. auf Reparaturen oder Veränderungen durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt. Die Gewährleistung erstreckt sich nicht auf die im PlusRAM gespeicherte Software. Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Gewährleistungsregelung unberührt.

Das MBK-ProfiSET wird auf der Basis der zum Zeitpunkt der Herstellung gegebenen technischen Spezifikation verkauft. Das Mathematische Büro Köln übernimmt keine Verpflichtung zur nachträglichen Anpassung oder Modifikation einmal gekaufter Produkte.

Die in diesem Handbuch dargestellten Programme dienen der beispielhaften Illustration. Der Benutzer trägt das Risiko hinsichtlich der Güte und Anwendbarkeit der Programme oder Teile hiervon. Das Mathematische Büro Köln übernimmt keine Haftung oder Gewährleistung aus der Lieferung und Benutzung dieser Programme, auch nicht für Neben- oder Folgeschäden.

Service

Das Mathematische Büro Köln steht Ihnen jederzeit für eine eventuelle Reparatur Ihres ProfiSETs zur Verfügung, auch wenn die Gewährleistungsfrist bereits abgelaufen sein sollte. Nach Ablauf der Garantiezeit sind Reparaturen kostenpflichtig. Normalerweise erfolgt die Instandsetzung eingesandter Geräte innerhalb von 5 Werktagen. Im Einzelfall kann diese Frist überschritten werden; dann erhalten Sie entsprechende Mitteilung.

Jedes neue ProfiSET enthält frische Lithium-Zellen zur RAM-Pufferung. Die Kapazität dieser Batterien reicht, aus die Informationen in Ihrem PlusRAM mindestens 4 Jahre aufrechtzuerhalten, selbst wenn in dieser Zeit das ProfiSET niemals am Rechner angeschlossen wird. Je häufiger Sie mit dem in den Rechner eingesteckten ProfiSET arbeiten, um so weniger Strom wird aus den Lithium-Zellen verbraucht. Es sind nur ganz geringfügige Selbstentladungen zu erwarten. Aus Sicherheitsgründen empfehlen wir, dennoch nach Ablauf von 5 Jahren das ProfiSET an das Mathematische Büro Köln zum Batteriewechsel einzusenden. Nach heutigem Preisniveau würden hierfür Kosten von DM 40,- entstehen.

Lieferumfang / Zubehör

Zur Grundausstattung des MBK-ProfiSETS gehören:

- Gehäuse mit Verbindungs-Kabel und -Stecker zum HP-41
- Drehschlüssel zur Einstellung von System und PlusRAM
- Tastenfeldschablone
- Übersichtskarte zu den ProfiSET-Funktionen
- dieses Benutzerhandbuch

Bei Verlust von Zubehörteilen können Sie diese gegen Kostenerstattung auch einzeln nachbestellen.

BEGRIFFSERKLÄRUNGEN

Adresse	bezeichnet eine bestimmte Speicherzelle oder einen Speicherbereich. Die im ProfISET benötigten Adressen bezeichnen immer einen 4-Block und liegen, da der HP-41 64K oder 16 Blöcke zu 4K adressieren kann, im Bereich zwischen 0 und 15.
Betriebssystem	ist der Oberbegriff der zum Betreiben eines Computers notwendigen Software. Der HP-41 besitzt ein 12K-Betriebssystem; es enthält z.B. die notwendigen CPU-Instruktionen zum Addieren zweier Zahlen mit dem Rechnerbefehl '+'. Das Betriebssystem des ProfISETs ermöglicht es, bequem, sicher und schnell in PlusRAM zu arbeiten.
Block	ist die Bezeichnung für einen Teil des Speicherbereichs des HP-41. Der PlusRAM besteht aus 4 Blöcken, von denen jeder einzeln über die Selektierung angesprochen werden kann (siehe auch: Adresse).
Byte	ist eine Einheit für die Speichergröße. Ein Byte umfaßt 8 Bit; ein Bit ist die kleinste Einheit in diesem Zusammenhang, es kann die Werte 0 oder 1 annehmen. In bezug auf den HP-41 wird die Einheit Byte oft mißbräuchlich benutzt; selbst Hewlett-Packard wirbt für den HP-41 mit der Angabe von 12KByte Betriebssystem, obwohl ein Wort im Betriebssystem 10 Bit umfaßt und somit die richtige Bezeichnung 12K x 10Bit lauten müßte.
CAT1-RAM	ist eine andere Bezeichnung für den Teil des Hauptspeichers, in dem sich Programme befinden.
CAT2	ist jener Speicherbereich, der vom Betriebssystem über die Adressen 3 ff erreicht wird; im engeren Sinne wird erst ab Adresse 8 gezählt, d.h. über die Ports und eingesteckte Peripherie erreichbare Programme, Files etc. Mit CAT2 ist zugleich auch die Funktion zum Anzeigen/Druck des Inhalts dieses Speicherbereichs gemeint.
EPROM	ist ein 'Ultraviolet erasable and electrically programmable read-only-memory', ein Speicher-IC, der mit UV-Licht gelöscht und mit einem speziellen Programmiergerät elektrisch programmiert werden kann.
EPROMer	ist das EPROM-Programmiergerät zum ProfISET.
EPROM-Box	ist eine Hardware-Erweiterung zum HP-41. In einer EPROM-Box können Softwarepakete größeren Umfangs preisgünstig untergebracht werden. Es gibt für den HP-41 EPROM-Boxen mit einem Speicherumfang bis zu 32K. Zum Laden von eigenen Programmen in eine EPROM-Box ist ein Programmiergerät wie der EPROMer erforderlich.
EPROM-Satz	besteht für den HP-41 je nach Kapazität immer aus 2 oder 3 EPROMs, um die 10Bit-Worte unterzubringen; ein EPROM-Satz hat je nach Wahl der EPROM-Typen die Kapazität von 4, 8, 16 oder 32K.

BEGRIFFSERKLÄRUNGEN

EPROM-String	Textkette zur Steuerung des MBK-EPROMers
FAT	Abkürzung für function adress table
File	bezeichnet einen abgegrenzten Teil im PlusRAM, z.B. ein Programm, einen Datenfile oder einen KEY-File.
Formatieren	Ein PlusRAM-Block wird vorbereitet zur Aufnahme mehrerer Files und einer Liste der Startadressen dieser Files.
Funktion	Ein Beispiel für eine Funktion ist der Rechnerbefehl SIN. In diesem Handbuch werden alle MCode-Programme mit Funktion bezeichnet.
Global	Ein globales Label in einem HP-41-Programm kann von jedem anderen Programm erreicht werden. Es wird auch oft mit Text-Label im Gegensatz zum numerischen Label bezeichnet. LBL'ABC ist ein globales Label. LBL a und LBL 1Ø sind lokale Labels.
Hauptspeicher	ist der im HP-41 eingebaute Speicher; er umfaßt 319 Register. Er ist der einzige Bereich im HP-41, in dem Programme editiert werden können.
Identität	siehe XROM-Nummer
INIT-Nummer	ist die geräteabhängige dreistellige Nummer des ProfISETs, die bei Durchführung der Funktion INIT eingegeben werden muß (siehe S. 18).
Kopieren	bezeichnet beim ProfISET das Bewegen von Files aus dem PlusRAM in den Hauptspeicher, aber auch das Bewegen von ROM-Blöcken in einen PlusRAM-Block.
Laden	steht für das Bewegen von Files in das ProfISET.
Lokal	siehe Global
MBK	Mathematisches Büro Köln
MCode	bezeichnet die Maschinensprache des HP-41, die auch Microcode genannt wird.
Name	ist ein Eintrag im CAT2 mit mehr als 7 Zeichen.
PlusRAM	ist der Benutzerspeicher des ProfISETs; ein RAM, der im ROM-Bereich des HP-41 liegt.
Port	ist einer der vier Steckplätze des HP-41; über einen Port erreicht man 2 Adreßbereiche mit je 4K.
Prioritätsschalter	ist ein auf der Rückfront des ProfISETs angebrachter Schiebeschalter zur Umschaltung vom PlusRAM auf Steckmodul im gleichen Adreßbereich.
Prompten	bezeichnet das Stehenbleiben mit Anzeige während eines Programmablaufs zur Eingabe von Zahlen oder Zeichen.
PZ	ist die Abkürzung für Prüfziffer.
RAM	ist ein 'random access memory'; ein Speicher, auf den der Benutzer vollen Zugriff hat.

BEGRIFFSERKLÄRUNGEN

Revision	ist eine Zeichenkette von maximal 4 Zeichen in jedem Block des PlusRAMs oder in jedem HP-Einsteckmodul.
ROM	ist ein 'read-only-memory'; er kann nur mit zusätzlichen Hilfsmitteln beschrieben, jedoch jederzeit gelesen werden.
Selektierung	Auswahl eines bestimmten PlusRAM-Blocks; die Information, welcher PlusRAM-Block selektiert ist, bleibt im PlusRAM erhalten.
Sichern	ist ein Begriff für das Ablegen von ProfiSET-Files oder -Blöcken auf Band.
System	siehe Betriebssystem
Vpp	ist die Programmierspannung für EPROMs; sie liegt beim EPROMer zwischen 12,5 und 25V.
XROM-Nummer	ist eine Zahl im Format ID,NR mit ID = Identität und NR = lfd. Nummer, über die ein LBL und die Programmausführung erreicht wird, statt XEQ (siehe S.13, 24 ff.).

FUNKTIONSINDEX

In alphabetischer Reihenfolge sind die 56 ProfiSET-Funktionen wiedergegeben; ihre ausführliche Beschreibung finden Sie auf der in der nächsten Spalte angegebenen Seite des Handbuchs. Die dritte Spalte enthält den Kurznamen der sieben Gruppen, in die die Funktionen gegliedert sind. Damit ist ein schnelles Auffinden der Kurzbeschreibung der ProfiSET-Funktion (Anhang β , Seite 65) gewährleistet.

<u>Funktion</u>	<u>Seite</u>	<u>Gruppe</u>	<u>Funktion</u>	<u>Seite</u>	<u>Gruppe</u>
ADR>ID	20	-INFO	LADES	40	-PSUB
BL?	20	-INFO	LENG?	20	-INFO
BL1A?	21	-INFO	L*	41	-PSUB
BLA?	21	-INFO	L-	41	-PSUB
BLCAT	19	-INFO	L=	41	-PSUB
BLCATX	20	-INFO	NAMEBL	14	-BASE
BLF?	20	-INFO	NEWBL	13	-BASE
CATF?	20	-INFO	NEXTBL	13	-BASE
CATS?	20	-INFO	N?	42	-PSUB
CHKBL	21	-INFO	PACKBL	16	-BASE
CLEAR	15	-BASE	PACKC	16	-BASE
CLRN	15	-BASE	PKEY	39	-PSUB
CLRS	40	-PSUB	PRM	41	-PSUB
COPYBL	16	-BASE	†PROM	53	-PROM
DAT..	30	-DATA	PROMBL	61	-PROM
EPACK	40	-PSUB	PRVBL	28	-PRGM
EPROM?	60	-PROM	PRVP	28	-PRGM
EP=BL?	61	-PROM	REVBL	14	-BASE
EP=FF?	61	-PROM	REV?	21	-INFO
GETBL	37	-MASS	SAVEBL	37	-MASS
GETF	37	-MASS	SAVEF	37	-MASS
IDBL	13	-BASE	SLCT	13	-BASE
ID?	21	-INFO	SUMBL	18	-BASE
ID>ADR	20	-INFO	SRV	21	-INFO
INIT	18	-BASE	TYP?	20	-INFO
J?	42	-PSUB	XROMBL	27	-PRGM
LADEK	40	-PSUB	XROMP	27	-PRGM
LADEP	24	-PRGM	X>DATY	31	-DATA

Aus dem Angebot des MBK zum HP-41-System

Hardware

- MBK-ProfiSET der 16K-RAM zum HP-41
- MBK-ProfiSET III 16K-RAM und 8K-EPROM-Box in einem
- MBK-16 die 16K-EPROM-Box
- MBK-32 die 32K-EPROM-Box
- MBK-EPROMer das EPROM-Programmiergerät zum HP-41
- alle lieferbaren Systemteile (HP-Produkte)
- Sonderzubehör: Spezialtragetaschen für den mobilen Einsatz
preiswertes, schwarz druckendes Thermopapier

Software

- verschiedene Programme aus der Finanzmathematik (eigenes Copyright)
 - für das Kreditgeschäft
 - für die Anlage-/Spar-Produkte
 - für das Leasing-Geschäft
 - für die Baufinanzierung
- diverse Programmpakete aus der Technik (16K oder 32K-EPROM) in Verbindung mit befreundeten Software-Partnern; z.B.:
 - ECKARDT-CONVAL control valves, Regelventilberechnungen bis ins Detail
 - BASIC ENGINEERING Programme rund um die Rohrleitung u.a.
 - ZAHNRAD-GEOMETRIE / -KALKULATION
- Programmierung von kundenspezifischen Funktionen in der schnellen Maschinensprache des HP-41

Dienstleistungen

- Programmierunterstützung / HP-41-Firmenseminare
- Beratung über die Einsatzmöglichkeiten/Wirtschaftlichkeit von HP-41-Systemen bei Anwenderproblemstellungen
- Komplett-Service bei der EPROM-Programmierung/-Vervielfältigung
- Entwicklung spezieller Schnittstellen von HP-41/ProfiSET für den Datentransfer zu Drittgeräten
- Erstellung kundenspezifischer Tastenfeldschablonen

MBK Die Spezialisten für den HP-41
Ein Begriff, den man sich merken sollte



HEWLETT
PACKARD

HP-41CX

Der beste Taschencomputer der Welt?



Urteilen Sie selbst...

- ... wer auf so kleinem Raum
einen leistungsstärkeren
Computer realisiert hat -
- ... wer ein umfangreicheres
Software-Angebot für Ihre
Aufgabenstellungen anbietet -
- ... wer einen vielseitigeren System-
ausbau mit Peripheriegeräten
ermöglicht.