```
TTTTT RRRR    A   M   M                f
  T   R   R  A A  MM MM               f f  t
  T   R   R A   A M M M  sss   ooo    f   ttt
  T   RRRR  A   A M M M s     o   o  fff   t
  T   R R   AAAAA M   M  sss  o   o   f    t
  T   R  R  A   A M   M     s o   o   f    t t
  T   R   R A   A M   M ssss   ooo    f     t
```

Ambühler & Müller

```
*****************************************
*                                       *
*                                       *
*   A n l e i t u n g           z u m   *
*                                       *
*   3 2 k B - R A M / R O M - M o d u l *
*                                       *
*                                       *
*****************************************
```

01/12/86 TRAMsoft COPYRIGHT by TRAMsoft Ambühler & Müller
Vervielfältigungen, auch auszugsweise, verboten

Inhaltsverzeichnis
****

Allgemeines iii

Garantie-Bedingungen iv

Einsetzen des Moduls . . . . . MO- 1

Herausnehmen des Moduls . . . . . MO- 1

Installation der BANKING-Software . . . . . MO- 2

Memorymap des PC-1500/A nach dem Einsetzen des Moduls . . . . . MO- 3

Eingeben, Aufzeichnen und Einlesen von Programmen und Daten . . . MO- 4

BANKING-Software . . . . . MO- 5

Beispiel-Programme zur BANKING-Software . . . . . MO- 7

Memory-Management . . . . . MO- 9

Beispiel-Programm zum Memory-Management MO-10

A l l g e m e i n e s
****

Mit dem TRAMsoft 32 kB-RAM/ROM-Modul kann die Speicherkapazität eines SHARP PC-1500/PC-1500A um 32 kBytes erweitert werden. Dabei ist der Speicherplatz des Moduls in 2 Blöcke aufgeteilt. Die Umschaltung erfolgt mit Hilfe der Steuerleitung PU. Die beiden Speicherblöcke werden als BANK 0, bzw. BANK 1 bezeichnet. Das Umschalten zwischen den BANKs wird durch die mitgelieferte BANKING-Software ermöglicht. Auf der beiliegenden Cassette befinden sich zwei verschiedene Installations-Programme, welche die gewünschte BANKING-Software ins Modul kopieren. Alle Programme sind in Maschinensprache geschrieben. Bei der Installation der BANKING-Software wird das Modul auf seine Funktionstüchtigkeit geprüft.

Die grosse Speicherkapazität kann entweder dazu benützt werden, mehrere Programme gleichzeitig im Rechner zu speichern oder um sehr grosse Programme zu entwickeln. Befinden sich mehrere Programme im Speicher, so kann mit Hilfe der BANKING-Software manuell zwischen den BANKs umgeschaltet werden. Bei Programmen, welche die 16 kB-Grenze übersteigen, kann während dem Programmablauf zwischen den BANKs umgeschaltet werden. Die BANKING-Software ermöglicht auch den Aufruf von Subroutinen in der anderen BANK. Falls bei einem Programm der Platzbedarf für Daten grösser ist als der Bereich des Standard-RAMs des PC-1500/A, so können in einer oder in beiden BANKs sogenannte "lokale" Variablen angelegt werden. Bei einer solchen Anwendung müssen allerdings zwei Systemzeiger vom entsprechenden Programm verwaltet werden.

Das 32 kB-ROM-Modul eignet sich besonders zum Vertrieb von Anwender-Software. Da das Programm im Modul nicht mehr verändert werden kann, schützt es den Anwender gegen Verlust der Software bei Fehlbedienungen. Das Programm steht dem Kunden unmittelbar nach dem Einstecken des Moduls zur Verfügung und erspart damit zeitraubendes Einlesen des Programmes ab Band, zudem kann das Modul mit einem Listschutz versehen werden. Zur Herstellung von ROM-Moduln bietet TRAMsoft einen Programmier-Service an.

Ein für den SHARP PC-1500/A gekauftes Modul kann für den SHARP PC-1600 umkonfiguriert werden und umgekehrt. Im Bedarfsfall wenden Sie sich bitte an Ihren Händler oder direkt an TRAMsoft Ambühler & Müller in Winterthur. Die BANKING-Software wird beim PC-1600 nicht mehr benötigt, da dass System selbst die Verwaltung der verschiedenen Speicherbereiche übernimmt.
Die in dieser Anleitung beschriebene Methode der "lokalen Variablen" ist aber nur mit dem PC-1500/A möglich.

Falls Sie beabsichtigen, ein Anwendungspaket für ein ROM-Modul zu entwickeln, bitten wir Sie, sich vorher mit TRAMsoft in Verbindung zu setzen. Um ein einwandfreies Funktionieren im ROM-Modul zu gewährleisten, müssen beim Programmieren gewisse Bedingungen eingehalten werden.

G a r a n t i e - B e d i n g u n g e n
****

Garantie-Leistungen: Die Garantie erstreckt sich auf die Ausbesserung, bzw. den Ersatz defekter Bauteile sowie die Nachbesserung der Software im Rahmen der in dieser Anleitung beschriebenen Spezifikationen.

Garantie-Dauer: 6 Monate ab Rechnungsdatum

Einschränkungen: Der Garantie-Anspruch wird abgelehnt bei:

- Manipulationen am Modul
- unsachgemässem Betreiben des Moduls, insbesondere beim Einsetzen oder Herausnehmen des Moduls bei eingeschaltetem Rechner.

*** Die Haftung für sämtliche Folgeschäden, die durch das ***
*** Modul selbst oder durch dessen Anwendung verursacht ***
*** werden, wird ausdrücklich abgelehnt. ***

Einsetzen des Moduls
****

Das TRAMsoft 32 kB-RAM-Modul wird ins Modulfach auf der Rückseite des SHARP PC-1500/PC-1500A eingesetzt, siehe Bedienungsanleitung zum Rechner, Seite 142.
Vor dem Einsetzen des Moduls ist die Schutzkappe am Modul zu entfernen, dabei ist zu beachten, dass die Goldkontaktflächen nicht berührt werden. Eine Berührung dieser Kontaktflächen kann zu irreparablen Schäden am Modul führen. Ausserdem ist unbedingt darauf zu achten, dass der Rechner ausgeschaltet ist.
Zum Einsetzen Modul ins Modulfach legen, auf der dem Stecker abgewandten Seite leicht nach unten drücken und das Modul gleichzeitig Richtung Stecker schieben, bis der Deckel einschnappt.

Nach dem Einsetzen des Moduls zeigt der Rechner beim Einschalten die Meldung

NEW0? :CHECK

Durch diese Meldung zeigt der Rechner an, dass er das Modul erkannt hat.

ZUR BEACHTUNG

Auf Grund eines Software-Fehlers im Betriebssystem des PC-1500/A soll aber nicht NEW 0, sonder NEW &100 (entspricht NEW 256) angegeben werden. Dieser Fehler hat zur Folge, dass die Speicherlokation &FF (entspricht 255) unter gewissen Umständen verändert wird.

Herausnehmen des Moduls
****

*** ACHTUNG: Beim Herausnehmen des Moduls geht der Inhalt verloren. ***

Vor dem Herausnehmen des Moduls muss der Rechner unbedingt ausgeschaltet werden. Zum Herausnehmen den Moduldeckel bei der Aussparung etwas anheben und das Modul gleichzeitig herausschieben. Bitte sofort die Schutzkappe über die Kontaktflächen stecken, damit keine Beschädigung durch Berührung entsteht. Nach dem Einschalten zeigt der Rechner wiederum die Meldung

NEW0? :CHECK

Nach dem Ausführen der Anweisung NEW 0 ist der Rechner wieder betriebsbereit.

Installation der BANKING-Software
****

Auf der mitgelieferten Cassette befinden sich zwei verschiedene Installations-Programme:

- RAM-INST-Vxx.OBJ
- ROM-INST-Vxx.OBJ

Diese Installations-Programme prüfen das Modul auf seine Funktionstüchtigkeit und kopieren anschliessend die entsprechende BANKING-Software ins Modul.

Die Installations-Programme werden mit der BASIC-Anweisung CLOAD M eingelesen. Der Programmname kann, muss aber nicht eingegeben werden.
Unmittelbar nach dem Einlesen wird das Installations-Programm automatisch gestartet. Es testet das Modul und kopiert die BANKING-Software entsprechchend der vorangegangenen NEW Anweisung ins Modul. Die Anweisung NEW kann dazu verwendet werden, in den beiden BANKs zwischen den RESERVE-Speichern und dem BASIC-Programm Platz für Programme in Maschinensprache zu schaffen, siehe auch Bedienungsanleitung zum PC-1500/A. Die Anweisung NEW bezieht sich immer auf beide BANKs.

Wurde die BANKING-Software erfolgreich installiert, so wird kurz die Meldung "2*nnnnn Bytes free." angezeigt, danach erscheint das Bereitschaftszeichen. Der angezeigte Wert bezieht sich nur auf den freien Speicherplatz im Modul, das Standard-RAM ist bei dieser Anzeige nicht berücksichtigt.

Da die BANKING-Software nur mit Hilfe des Installations-Programmes in den Rechner geladen werden kann, wird das Installations-Programm nicht nur beim ersten Einsatz des Moduls benötigt, sondern immer dann, wenn der gesamte Speicherinhalt des Rechners mit NEW gelöscht wurde.

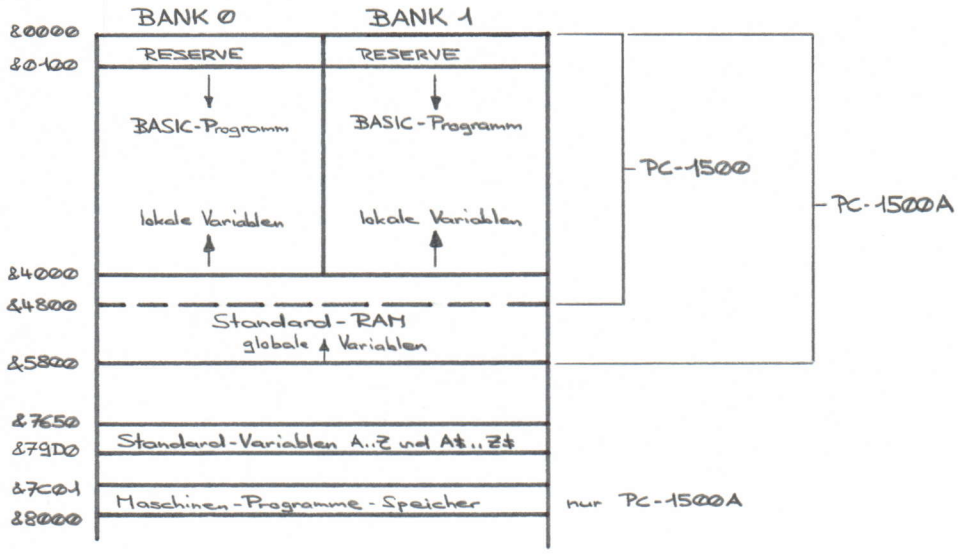
Beim Installieren der BANKING-Software können folgende Fehlermeldungen auftreten:

- ERROR 22: Im Modul steht zuwenig Speicherplatz für die BANKING-Software zur Verfügung, d.h. der Ausdruck bei der Anweisung NEW war zu gross.

- ERROR 177: Der Funktionstest des Moduls ist nicht erfolgreich durchgelaufen. Bitte prüfen, ob das Modul überhaupt eingesetzt wurde und ob nach dem Einsetzen die Anweisung NEW &100 (oder entsprechend) ausgeführt wurde. Wenn ja, muss das Modul an die Verkaufsstelle zur Reparatur zurückgegeben werden.

Memorymap des PC-1500/A nach dem Einsetzen des Moduls
****

Die nachfolgende Memorymap bezieht sich nur auf den PC-1500/A und das TRAMsoft 32 kB-RAM/ROM-Modul, weitere Peripheriegeräte und Ergänzungen sind nicht berücksichtigt.
Die 32 kB Memory des Moduls sind in den untersten 16 kBytes des Adressraums des PC-1500/A untergebracht. Mit Hilfe der Steuerleitung PU und der BANKING-Software kann zwischen den beiden BANKs umgeschaltet werden. Das Standard-RAM, sowie die Standard-Variablen des Rechners werden von dieser Steuerleitung nicht beeinflusst. Auch auf den Speicher des PC-1500A für Programme in Maschinensprache hat die Steuerleitung keine Wirkung.



Die Memorymap ist wie folgt zu interpretieren: Daten, die im Standard-RAM oder in den Standard-Variablen gespeichert sind, können aus beiden BANKs angesprochen werden, d.h. in diesen Variablen können Daten zwischen den BANKs ausgetauscht werden. Daten, die in einer BANK gespeichert sind, sogenannte "lokale" Variablen, können nur von Programmen in der gleichen BANK angesprochen werden. Dies bedeutet, dass in einer BANK nicht nur Daten gespeichert werden können, sondern auch der Programmteil, der diese Daten speichert und/oder liest, muss in der gleichen BANK gespeichert sein.

Beispiel

Eine Anwendung umfasse etwa 12 kB Programm und etwa 10 kB Daten. Das Programm wird so geschrieben, dass die gesamte Verarbeitung der Daten in der BANK 0 geschieht, während in der BANK 1 die 10 kB Daten als lokale Variablen gespeichert sind. Ebenfalls in der BANK 1 befindet sich der Programmteil, der die Daten aus den lokalen Variablen liest und in diese schreibt. Der Datenaustausch zwischen den Programmteilen erfolgt über Standard-Variablen. Siehe auch Programmbeispiel zu "Memory-Management".

Eingeben, Aufzeichen und Einlesen von Programmen und Daten
****

Manuelles Eingeben von Programmen

Wird nach dem Installieren der BANKING-Software ein Programm manuell eingegeben, so ist lediglich zu beachten, dass die Zeilennummern 1 ... 5 für die BANKING-Software reserviert sind und das die Zeilen 1 und 2 auf keinen Fall verändert werden dürfen. Falls die Zeilen 3 und 4 verändert werden, so ist darauf zu achten, dass deren Längen in beiden BANKs identisch sein müssen. Die BANKING-Software muss unbedingt an erster Stelle im Programmspeicher stehen. Die Eingabe eines Programmes kann wie gewohnt erfolgen.

Aufzeichnen von BASIC-Programmen und Daten

BASIC-Programme und Daten können wie üblich aufgezeichnet werden.

Einlesen von BASIC-Programmen und Daten

Durch den Einsatz des TRAMsoft 32 kB-RAM/ROM-Moduls gibt es beim Einlesen von Daten keine Aenderung. Sollen Daten in "lokale" Variablen eingelesen werden, ist jedoch darauf zu achten, dass die Variablen angelegt und die entsprechende BANK angewählt ist.
BASIC-Programme dagegen können nicht mehr wie gewohnt mit CLOAD eingelesen werden, sondern müssen mit MERGE eingelesen werden. TOOL2-Besitzer müssen die Anweisung FLOAD P an Stelle von FLOAD benützen. Besitzer von TRAMsoft Floppy-Interfaces müssen die Anweisung DLOAD P an Stelle von DLOAD verwenden. MERGE (bzw. FLOAD P oder DLOAD P) fügen das BASIC-Programm an die BANKING-Software an, CLOAD (bzw. FLOAD oder DLOAD) überschreiben die BANKING-Software. Wurde die BANKING-Software zusammen mit dem BASIC-Programm aufgezeichnet, so muss das BASIC-Programm trotzdem mit MERGE eingelesen werden. Die 5 Zeilen im soeben eingelesenen BASIC-Programm, welche die BANKING-Software enthalten, müssen anschliessend gelöscht werden. Die sich bereits im Rechner befindende BANKING-Software enthält aktuelle Systemzeiger und darf deshalb nicht überschrieben werden.

Zur Beachtung
****

Beim Eingeben, bzw. Einlesen von BASIC-Programmen ist darauf zu achten, dass STATUS 2 den Wert 16383 nicht überschreitet. Ab diesem Wert wird das Programm nicht mehr im Modul, sondern im Standard-RAM des Rechners gespeichert. Ueberschreitet das Programm nur in einer BANK diese Limite, so führt dies noch zu keinen Schwierigkeiten. Uebersteigt aber das Programm auch in der zweiten BANK diese Limite, wird das erste Programm zerstört. Anderseits soll STATUS 3 den Wert 16384 nicht unterschreiten, da sonst die DIM- und 2-Zeichen-Variablen nur noch aus der BANK angesprochen werden können, in welcher die Variablen angelegt wurden. Genügt die Speicherkapazität des Standard-RAMs nicht, so muss mit "lokalen Variablen" gearbeitet werden, siehe "Memory-Management".

BANKING-Software
****

Beim Installieren der BANKING-Software wird die Startadresse der BANKING-Software berechnet. Die Startadresse hängt vom numerischen Ausdruck ab, der bei der letzten NEW-Anweisung vor der Installation angegeben wurde. Die Startadresse ist als fünfstellige Zahl nach der Anweisung CALL am Anfang der Zeilen 3 und 4 abgespeichert.
Wurde zuletzt NEW &100 ausgeführt, so folgen den CALL-Anweisungen der Ausdruck 00285, d.h. die BANKING-Software hat in diesem Fall die Startadresse 285 (dezimal).

Die REM-Zeile am Anfang der BANKING-Software zeigt an, welche BANK zur Zeit angewählt ist. Diese Zeile darf niemals verändert werden, da in dieser Zeile gewisse Systemzeiger gespeichert sind. Eine Veränderung eines solchen Zeigers kann zum Verlust von Programmen und/oder Daten führen !

Manuelles Umschalten zwischen den BANKs

Das Umschalten in die andere BANK erfolgt mit der BASIC-Anweisung

CALL num. Ausdruck

Dabei ist num. Ausdruck die im vorangehenden Abschnitt beschriebene Startadresse. Wurde vor der Installation NEW &100 eingegeben, so lautet die Anweisung zum Umschalten in die andere Bank CALL 285.
Diese Anweisung gilt unabhängig von der zur Zeit aktuellen BANK, d.h. sie ist gleich zum Umschalten von BANK 0 nach BANK 1 und umgekehrt.

Programmkontrolliertes Umschalten

Soll während dem Programmablauf in die andere BANK umgeschaltet werden, so ist in der Variablen GB$ der Markenname zu speichern, auf den in der anderen BANK gesprungen werden soll, danach ist die Anweisung

GOTO "GOBA"

auszuführen. Die BANKING-Software schaltet dadurch in die andere BANK um und setzt die Programmausführung bei der Zeile mit dem angegebenen Markennamen fort.
Dabei ist zu beachten, dass wenn das Programm mit MERGE (bzw. FLOAD P oder DLOAD P) eingelesen wurde, ein Sprung GOTO 4 nicht möglich ist, sondern die BANKING-Software nur über Markennamen (Label) angesprochen werden kann.

Aufruf von Subroutinen in der anderen BANK

Subroutinen in der anderen BANK werden aufgerufen, indem in der Variablen GB$ der Markenname der aufzurufenden Subroutine gespeichert und anschliessend die Anweisung

GOSUB "GSBA"

ausgeführt wird. Mit dieser Anweisung schaltet die BANKING-Software in die andere BANK um, führt die entsprechende Subroutine aus und kehrt bei der zugehörigen RETURN Anweisung wieder in die ursprüngliche BANK zurück, wo die Programmausführung fortgesetzt wird.
Dabei sind weitere Subroutinen während der Ausführung der Subroutine in der anderen BANK möglich.

Wirkungsweise der BANKING-Software

Die BANKING-Software tauscht den Inhalt der beiden BANKs aus und führt einige Systemzeiger nach. Dieser Vorgang dauert etwa 1 Sekunde und kann nicht unterbrochen werden.
Wird ein Programm in EPROMs einprogrammiert, ist ein solcher Austausch nicht möglich. In diesem Fall wird eine andere BANKING-Software verwendet, welche mit dem Installations-Programm "ROM-INST-Vxx.OBJ" ins Modul kopiert wird. Diese Software sollte nur eingesetzt werden, wenn das Programm in EPROMs programmiert werden soll, oder wenn eine zeitkritische Anwendung vorliegt, die ein schnelleres Umschalten zwischen den BANKs nötig macht. Während der Entwicklungsphase soll aber unbedingt mit der RAM-Version gearbeitet werden, da diese ein viel besseren Schutz gegen Datenverlust durch Fehlmanipulationen bietet.

Wichtiger Hinweis für SHARP CE-158 Benutzer
****

Das Interface SHARP CE-158 benutzt die Steuerleitung PU ebenfalls. Dies hat zur Folge, dass nach dem Einschalten des Rechners das CE-158 in die andere BANK umschaltet. Es muss deshalb unmittelbar nach dem Einschalten unbedingt die BANKING-Software aufgerufen werden, da gewisse Systemzeiger falsche Werte enthalten, was zur sofortigen Zerstörung von Programmen und Daten führen würde.
Zudem müssen alle Anweisungen, die das Interface CE-158 betreffen, in beiden BANKs an der gleichen Stelle im Speicher stehen. Dies kann sichergestellt werden, indem ein solches Programm in beiden BANKs unmittelbar nach dem Installieren der BANKING-Software mit MERGE eingelesen wird.

Beispiel-Programme zur BANKING-Software
****

Beispiel 1

Das nachfolgende Programm schaltet zwischen den beiden BANKs hin und her und zeigt jeweils an, in welcher BANK es sich gerade befindet.

```
 1 ARUN :GOTO 5
 2 REM Bank Ø  ~~~~~~
 3 "GSBA"CALL ØØ285:GOSUB GB$:CALL ØØ285:RETURN
 4 "GOBA"CALL ØØ285:GOTO GB$
 5 END
10 "A"BEEP 1:WAIT Ø
20 PRINT "***     In BANK Ø     ***
30 GB$="B":GOTO "GOBA"
```

```
 1 ARUN :GOTO 5
 2 REM Bank 1  ~~~~~~
 3 "GSBA"CALL ØØ285:GOSUB GB$:CALL ØØ285:RETURN
 4 "GOBA"CALL ØØ285:GOTO GB$
 5 END
10 "B"BEEP 2:WAIT Ø
20 PRINT "***     In BANK 1     ***
30 GB$="A":GOTO "GOBA"
```

Beispiel 2

Das nachfolgende Programm berechnet die Lösungen einer Quadratischen Gleichung. Dabei wird die Eingabe, die Bestimmung der Art der Resultate (Reel, Doppellösung, Komplex) und die Anzeige der Resultate in der BANK 0 durchgeführt, die eigentlichen Berechnungen dagegen in der BANK 1 ausgeführt. Die Berechnungen sind als Subroutinen geschrieben.

```
  1 ARUN :GOTO 5
  2 REM Bank Ø  ~~~~~~
  3 "GSBA"CALL ØØ285:GOSUB GB$:CALL ØØ285:RETURN
  4 "GOBA"CALL ØØ285:GOTO GB$
  5 END
 1Ø " "CLEAR :CLS :WAIT 128
 2Ø DIM L$(Ø)*26
 3Ø PRINT "  A*x^2 + B*x + C = Ø"
 4Ø INPUT "A = ";A
 5Ø INPUT "B = ";B
 6Ø INPUT "C = ";C
 7Ø D=B*B-4*A*C
 8Ø IF D<ØLET GB$="K"
 9Ø IF D=ØLET GB$="D"
1ØØ IF D>ØLET GB$="R"
11Ø GOSUB "GSBA"
12Ø BEEP 3
13Ø PRINT L$(Ø)
14Ø WAIT
15Ø PRINT X,Y
16Ø END
```

```
  1 ARUN :GOTO 5
  2 REM Bank 1  ~~~~~~
  3 "GSBA"CALL ØØ285:GOSUB GB$:CALL ØØ285:RETURN
  4 "GOBA"CALL ØØ285:GOTO GB$
  5 END
1ØØ "K"X=-B/(2*A)
11Ø Y=SQR (-D)/(2*A)
12Ø L$(Ø)="2 komplexe Loesungen"
13Ø RETURN
2ØØ "D"X=-B/(2*A)
21Ø Y=X
22Ø L$(Ø)="Doppelloesung"
23Ø RETURN
3ØØ "R"X=(-B+SQR D)/(2*A)
31Ø Y=(-B-SQR D)/(2*A)
32Ø L$(Ø)="2 reelle Loesungen"
33Ø RETURN
```

Memory-Management
****

Um "lokale" Variablen anlegen zu können, müssen zwei Systemzeiger verändert werden. Es handelt sich dabei um die Speicherlokation &7864, welche das Ende des RAMs enthält und die Speicherlokationen &7899/&789A, welche den Beginn der DIM- und 2-Zeichen-Variablen enthält (STATUS 3).
Bevor diese Zeiger verändert werden können, müssen jedoch die alten Werte gespeichert werden.
Ausserdem ist zu beachten, dass sowohl für die Uebergabe des anzuspringenden Markennamens als auch für den Datenaustausch zwischen den BANKs nur Standard-Variablen verwendet werden können (A .. Z und A$ .. Z$).

Erläuterungen zum Beispiel-Programm auf der nächsten Seite:

- Die Subroutine "INIT" in der BANK 1 speichert die "globalen" Systemzeiger "RAM-Ende" und "Beginn der Variablen" und setzt diese anschliessend auf das Ende des Modul, d.h. setzt sie auf "lokal". In der BANK 1 werden dann die "lokalen Variablen" X(9,124), I1 und I2 dimensioniert. Danach werden die beiden Systemzeiger wieder auf "global" umgestellt.

- In der Subroutine "LOC" werden die "globalen" Systemzeiger den Variablen X,Y und Z zugewiesen, danach wird auf "lokal" umgeschaltet.

- In der Subroutine "GLO" werden die "lokalen" Systemzeiger den Variablen U,V und W zugewiesen, danach wird auf "global" umgeschaltet.

- In der Subroutine "STO" werden jeweils 10 Werte in den Standard-Variablen @(1) .. @(10) (entspricht den Variablen A .. J) aus der BANK 0 an die BANK 1 übergeben und in der "lokalen Variablen" X(9,124) gespeichert. Dabei ist zu beachten, dass die in der BANK 1 verwendeten Schleifenzähler I1 und I2 "lokale Variablen" sind und deshalb zur Laufzeit andere Werte beinhalten als die in der BANK 0 benutzten "globalen Variablen" I1 und I2.

- Die Subroutine "RCL" kopiert den Inhalt von 10 Elementen der "lokalen Variable" X(9,124) in die 10 Standard-Variablen A .. J um diese dann in der BANK 0 anzeigen zu können.

- Das Programm "A" in der BANK 0 generiert 10 mal 125 Zufallszahlen und speichert diese mit Hilfe der Subroutine "STO" in der "lokalen Variable" X(9,125) in der BANK 1. Die Programm-Ausführung dauert einige Minuten.

- Das Programm "B" fragt den Zeilenindex "T" ab und zeigt dann mit Hilfe der Subroutine "RCL" 10 Werte aus der in der BANK 1 gespeicherten "lokalen Variablen" X(9,124) an.

Bei diesem Beispiel ist zu beachten, dass eine DIM-Variable mit der Grösse von X(9,124) im Standard-RAM des PC-1500 oder PC-1500A gar nicht Platz finden würde (10*125*8+7 Bytes = 10007 Bytes).

Programm-Beispiel zum Memory-Management
****

```
  1 ARUN :GOTO 5
  2 REM Bank Ø   ~~~~~~
  3 "GSBA"CALL ØØ285:GOSUB G$:CALL ØØ285:RETURN
  4 "GOBA"CALL ØØ285:GOTO G$
  5 END
 1Ø "A"CLS :WAIT Ø:USING :G$="INIT":GOSUB "GSBA"
 2Ø FOR I1=ØTO 124
 3Ø FOR I2=1TO 1Ø
 4Ø @(I2)=RND (999)
 5Ø NEXT I2
 6Ø G$="STO":GOSUB "GSBA"
 7Ø PRINT "BANK Ø: I1 =";I1:NEXT I1
 8Ø END
1ØØ "B"INPUT "T = ";T
11Ø G$="RCL":GOSUB "GSBA"
12Ø FOR I1=1TO 1Ø
13Ø PAUSE I1,@(I1)
14Ø NEXT I1
15Ø END
```

```
  1 ARUN :GOTO 5
  2 REM Bank 1   ~~~~~~
  3 "GSBA"CALL ØØ285:GOSUB G$:CALL ØØ285:RETURN
  4 "GOBA"CALL ØØ285:GOTO G$
  5 END
 1Ø "LOC"X=PEEK &7899,Y=PEEK &789A,Z=PEEK &7864
 2Ø POKE &7899,U,V:POKE &7864,W
 3Ø RETURN
 5Ø "GLO"U=PEEK &7899,V=PEEK &789A,W=PEEK &7864
 6Ø POKE &7899,X,Y:POKE &7864,Z
 7Ø RETURN
1ØØ "INIT"U=&4Ø,V=Ø,W=&4Ø
11Ø GOSUB "LOC"
12Ø DIM X(9,124)
13Ø I1=Ø,I2=Ø
14Ø GOSUB "GLO"
15Ø RETURN
2ØØ "STO"GOSUB "LOC"
21Ø FOR I1=ØTO 9
22Ø X(I1,I2)=@(I1+1)
23Ø NEXT I1
24Ø I2=I2+1
25Ø GOSUB "GLO"
26Ø RETURN
3ØØ "RCL"GOSUB "LOC"
31Ø FOR I1=ØTO 9
32Ø @(I1+1)=X(I1,T)
33Ø NEXT I1
34Ø GOSUB "GLO"
35Ø RETURN
```