

Tastatur-Bildschirm-Ein-/Ausgabe

9 Tastatur-Bildschirm-Ein-/Ausgabe

Bildschirmarbeitsplätze sind über Leitungsanschluß mit der Zentraleinheit des Systems 8870 verbunden.

Der Bildschirmarbeitsplatz übernimmt folgende Funktionen:

- An- und Abmelden von Programmen (Tasks)
- Dialog zwischen Bediener und Programm
- Daten-Ein- und Ausgabe

Die Arbeitsplätze unterliegen nicht dem Kanalkonzept, d.h. sie brauchen nicht eröffnet und geschlossen werden. Jeder Arbeitsplatz ist immer dem Programm zugeordnet, das er aufgerufen und gestartet hat.

Die Programmierung der Daten-Ein- und -Ausgabe erfolgt mit den BASIC-Anweisungen:

- PRINT und MAT PRINT (Ausgabe) und
- INPUT und MAT INPUT (Eingabe),

die durch Bildschirm-Funktionen (<DisFkt>) zur Steuerung der Ein-/Ausgabemöglichkeiten unterstützt werden.

Eine Übersicht über die möglichen Bildschirmfunktionen finden Sie im Kapitel "Die BASIC-Anweisungen"; Abschnitt "Die Anweisungen zur Tastatur-Bildschirm-Ein-/Ausgabe".

Der Bildschirm verfügt über einen intern geführten Zeiger (Cursor), der nur bei Eingaben sichtbar wird. Ein- und Ausgaben werden grundsätzlich ab der aktuellen Cursorposition ausgeführt.

Außer durch Ein- und Ausgaben wird die Cursorposition durch die TAB-Funktion sowie einige andere Bildschirmfunktionen verändert.

Jeder Bildschirmarbeitsplatz hat einen Ein-/Ausgabepuffer in der Größe von 254 Byte, von denen 40 Byte (80 Byte beim DAP 4) als Voreingabepuffer reserviert sind. Damit ist schon vor dem Auftreten des INPUT-Modus die Eingabe von bis zu 40 (80 beim DAP 4) Zeichen und evtl. Auslösetasten möglich. Werden mehr Zeichen eingegeben, ertönt der akustische Signalgeber, und die zuviel eingegebenen Zeichen werden nicht übernommen. Der Voreingabepuffer wird gelöscht durch:

- Drücken der Taste "ESC"
- Ausgabe von $\leftarrow 207 \leftarrow$ per INPUT- oder PRINT-Anweisung.

© Microsoft, soweit Vervielfältigung dieser Unterlage, Verwertung und
Mithilfe ihres Inhalts nicht ausdrücklich gestattet. Alle Rechte für den Fall
Zwiderhandlungen vorbehalten. Alle Rechte für den Fall
der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.



Tastatur-Bildschirm-Ein-/Ausgabe

Eingegebene Zeichen werden, wenn sich das Programm im INPUT-mode befindet, direkt ab der aktuellen Cursorposition auf dem Bildschirm angezeigt. Voreingegebene Zeichen werden erst zum Zeitpunkt des Umschaltens auf INPUT-mode angezeigt.

Der Ein-/Ausgabepuffer wird nach einer Eingabe geleert, d.h. die eingegebenen Daten werden vom Programm übernommen, wenn die Eingabe mit einer Auslöstaste beendet wird. Die Leerung des Puffers nach einem PRINT-Befehl, d.h. das Anzeigen von Daten am Bildschirm erfolgt erst, wenn:

- ein gleichzeitig laufendes Programm seine Zeitscheibe erhält,
- der Ausgabepuffer voll ist,
- eine SIGNAL 3-Anweisung,
- eine INPUT-Anweisung,
- eine PRINT #-Anweisung,
- eine CHAIN-Anweisung,
- eine LINK-Anweisung ausgeführt wird,
- das Programmende erreicht ist.

9.1 Die INPUT-Anweisung

Syntax:

```

INPUT [ <SLit>, <DisFkt>, TAB(<NAusdr>[,<NAusdr>]), <NVar>, <SVar> ]n { <NVar> } { <SVar> }

```

Eingabe von Daten über die Tastatur in beliebige numerische und String-Variablen. Zusätzlich können Führungstexte und Bildschirmfunktionen ausgegeben werden. Eine genaue Beschreibung der INPUT-Anweisung erfolgt im Kapitel "BASIC-Anweisungen".

Tastatur-Bildschirm-Ein-/Ausgabe

9.2 Die MAT INPUT-Anweisung

Syntax:

```
MAT INPUT <MVar>[( <NAusdr>[, <NAusdr>] ) ]
[ , <MVar>[( <NAusdr>[, <NAusdr>] ) ] ]n
```

Mit dieser Anweisung wird die Eingabe aller Elemente einer Matrix (ohne Reihe und Spalte 0) über die Tastatur ermöglicht.

Eine genaue Beschreibung der MAT INPUT-Anweisung erfolgt im Kapitel "BASIC-Anweisungen".

9.3 Die PRINT-Anweisung

Syntax:

```
{ PRINT } [ USING <String> ; ] [ <NAusdr> ] [ { ; } { <NAusdr> } ]n [ ; ]
[ <String> ] [ { ; } { <String> } ]
[ <DisFkt> ] [ { ; } { <DisFkt> } ]
```

PRINT ermöglicht die Ausgabe

- des Inhalts von String-Variablen
- des Wertes von numerischen Ausdrücken
- von numerischen und alphanumerischen Konstanten
- von Bildschirm-Funktionen

in beliebiger Reihenfolge am Bildschirm.

Eine genaue Beschreibung der PRINT-Anweisung erfolgt im Kapitel "BASIC-Anweisungen".

9.4 Die MAT PRINT-Anweisung

Syntax:

```
MAT PRINT <MVar> [, | ; ]
```

Die Anweisung MAT PRINT erzeugt das Anzeigen einer gesamten Matrix (ohne Reihe und Spalte 0) am Bildschirm.

Eine genaue Beschreibung der MAT PRINT-Anweisung erfolgt im Kapitel "BASIC-Anweisungen".



© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten. In Form der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

Tastatur-Bildschirm-Ein-/Ausgabe

9.5

Die Bildschirmfunktionen

Bildschirmfunktionen werden durch die Ausgabe bestimmter ASCII-Zeichen ausgelöst. Da diese Zeichen jedoch mit der Tastatur nicht darstellbar sind, müssen sie entweder durch zwei oder drei symbolische Buchstaben in Hochkomma (' ') oder durch ihren Oktal-Code ersetzt werden. Einem Teil der Funktionen muß bei oktaler Darstellung der Leading-Code (←376←) vorangestellt werden.

Neben der symbolischen Darstellung ist in den folgenden Beschreibungen immer auch der Oktal-Code mit aufgeführt.

9.5.1

Die Funktion TAB

Spalte Zeile

Syntax: TAB (<NAusdr1>[,<NAusdr2>])

Die Bildschirmfunktion TAB ermöglicht eine beliebige Cursorpositionierung auf dem Bildschirm.

TAB (<NAusdr>)

Ab der aktuellen Cursorposition werden bis vor die mit dem Wert von <NAusdr> bezeichnete Spalte Blanks ausgegeben und der Cursor auf die mit <NAusdr> bezeichnete Spalte positioniert. Ein eventuell zwischen diesen Positionen stehender Text wird überschrieben.

Ist <NAusdr> kleiner als die aktuelle Cursorposition, bleibt die Funktion ohne Wirkung. Überschreitet <NAusdr> die Zeilenlänge, werden automatisch entsprechend viele Zeilenvorschübe durchgeführt.

TAB (<NAusdr1>,<NAusdr2>)

Der Cursor wird auf die mit dem Wert von <NAusdr1> bezeichnete Spalte in der mit dem Wert von <NAusdr2> bezeichneten Zeile positioniert. Eine Rückwärtspositionierung ist möglich. Der Bildschirminhalt bleibt vollständig erhalten.

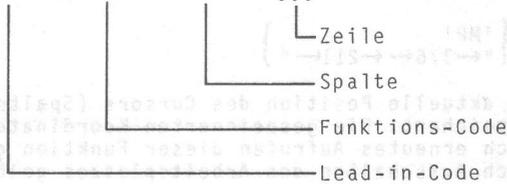
Die Spaltenangabe muß im Bereich von 0 bis 79, die Zeilenangabe im Bereich von 0 bis 24 liegen.

Bei oktaler Darstellung der TAB Funktion können Spalte und Zeile nur absolut angegeben werden (nicht als <NAusdr>):

Tastatur-Bildschirm-Ein-/Ausgabe

Oktaal-Code

"←376←←221←←xxx←←yyy←"



Spalte und Zeile müssen hier ebenfalls oktaal verschlüsselt werden.

• Beispiel:

Positionierung des Cursors auf Spalte 10 in Zeile 2.

```
50 PRINT TAB(10,2);...
```

Oktaal: 50 PRINT "←376←←221←←212←←202←";...

9.5.2 Die Funktion Backspace

Syntax:

```
{ 'BS'  
  "←210←" }
```

Rückpositionierung des Cursors ab der aktuellen Position um ein Zeichen. Ein auf dieser Position stehendes Zeichen wird gelöscht. Steht der Cursor auf Spalte 0 einer beliebigen Zeile, erfolgt eine Positionierung auf die Spalte 79 der darüberliegenden Zeile. Steht der Cursor auf Spalte 0 der Zeile 0, bleibt die Funktion ohne Wirkung.

Der Wechsel aus einem Vordergrundfeld in ein Hintergrundfeld ist nicht möglich: steht der Cursor unmittelbar hinter einem Hintergrundfeld, bleibt die Funktion ohne Wirkung, und der akustische Signalgeber ertönt.

• Beispiel: 50 PRINT 'BS';

Oktaal: 50 PRINT "←210←";

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlung wird strafrechtlich verfolgt.“ Dem Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Tastatur-Bildschirm-Ein/Ausgabe

9.5.3 Die Funktion Mark Cursorposition

Syntax:

```
{ 'MP'  
  "←376←←211←" }
```

Die aktuelle Position des Cursors (Spalte und Zeile) wird gespeichert. Die gespeicherten Koordinaten werden nur durch erneutes Aufrufen dieser Funktion geändert oder durch Ausschalten des Arbeitsplatzes gelöscht. Wurde ein Bildschirmfenster eingerichtet, werden auch dessen Koordinaten gemerkt. Die Funktion löscht außerdem ein bereits eingerichtetes Bildschirmfenster.

*
*

● Beispiel:

```
50 PRINT 'MP';
```

```
Okta1: 50 PRINT "←376←←211←";
```

9.5.4 Die Funktion Back to marked Cursorposition

Syntax:

```
{ 'BP'  
  "←376←←212←" }
```

Der Cursor wird auf die Position gesetzt, die beim letzten Aufruf der Funktion 'MP' (Mark Cursorposition) gespeichert wurde. Ein mit 'MP' gemerktes Bildschirmfenster wird wieder eingerichtet.

Nach Neueinschalten des Arbeitsplatzes wird beim Aufruf dieser Funktion der Cursor auf Spalte 0 in Zeile 0 gesetzt.

● Beispiel:

```
50 INPUT 'BP',X
```

Positionierung auf die gemerkte Bildschirmposition bei der Eingabe von X.

```
Okta1: 50 INPUT "←376←←212←",X
```

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.5 Die Funktion Carriage Return

Syntax:

```
{ 'CR'  
  " ← 215 ← " }
```

Positionieren des Cursors auf Spalte 0 der nächsten Zeile (Zeilenschaltung). Der Bildschirminhalt wird nicht verändert. Befindet sich der Cursor in Zeile 24, wird der gesamte Bildschirminhalt um eine Zeile nach oben geschoben. Die Informationen in Zeile 0 gehen verloren, und Zeile 24 wird frei. Anschließend steht der Cursor auf Spalte 0 in Zeile 24.

● Beispiel:

Ausgabe der String-Literale "FELD 1" und "FELD 2" in den Spalten 0 der Zeilen 5 und 6.

```
50 PRINT TAB(0,5);"FELD 1";'CR';"FELD 2";
```

9.5.6 Die Funktion Line Deletion

Syntax:

```
{ 'LD'  
  " ← 376 ← ← 223 ← " }
```

Alle Zeilen unterhalb der Zeile, in der der Cursor steht, werden um eine Zeile nach oben geschoben. Die Zeile, in der der Cursor steht, wird überschrieben, und Zeile 24 wird frei. Steht der Cursor in Zeile 24, wird nur diese eine Zeile gelöscht.

● Beispiel:

Löschen der Zeile 24 und Ausgabe des Textes "STATUS" ab der Spalte 0.

```
50 PRINT TAB(0,24);'LD';"STATUS :";
```



Tastatur-Bildschirm-Ein-/Ausgabe

9.5.7 Die Funktion Line Insertion

Syntax:

```
{ 'LI'  
  "←376←←232←" }
```

Einfügen einer Leerzeile ab der aktuellen Cursorposition. Die Zeile, in der der Cursor steht, sowie alle darunterliegenden werden um eine Zeile nach unten geschoben. Die Informationen in Zeile 24 gehen verloren, und der Cursor steht anschließend auf Spalte 0 der eingefügten Zeile.

● Beispiele:

- Verschieben des gesamten Bildschirminhaltes um eine Zeile nach unten:

```
50 PRINT TAB(0,0);'LI';
```

- Roll-up von Zeile 3 bis Zeile 20:

```
50 PRINT TAB(0,3);'LD';TAB(0,20);'LI';
```

- Roll-down von Zeile 3 bis Zeile 20:

```
50 PRINT TAB(0,20);'LD';TAB(0,3);'LI';
```

- Löschen von Zeile 11:

```
50 PRINT TAB(0,11);'LD';'LI';
```

9.5.8 Die Funktion Start Background

Syntax:

```
{ 'SB'  
  "←376←←231←" }
```

Alle folgenden Zeichen erscheinen auf dem Bildschirm als Hintergrundzeichen mit verminderter Leuchtkraft.

- Beispiel: 50 PRINT 'SB';

```
Okta1: 50 PRINT "←376←←231←";
```

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.9 Die Funktion Start Foreground

Syntax:

```
{ 'SF'  
  "←376←←237←" }
```

Alle folgenden Zeichen erscheinen auf dem Bildschirm als Vordergrundzeichen mit voller Leuchtkraft. Diese Funktion ist nur dann notwendig, wenn zuvor 'SB' (Start Background) abgesetzt wurde.

• Beispiel:

Nach Ausgabe des Textes "RABATT : " im Hintergrund umschalten in den Vordergrund und Eingabe in A.

```
50 INPUT 'SB',"RABATT : ", 'SF',A
```

9.5.10 Die Funktion Clear Screen

Syntax:

```
{ 'CS'  
  "←376←←234←" }
```

Der gesamte Bildschirm wird gelöscht. Der Cursor steht anschließend auf Spalte 0 in Zeile 0.

• Beispiel:

```
50 PRINT 'CS';
```

```
Okta1: 50 PRINT "←376←←234←";
```

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.11 Die Funktion Clear Foreground

Syntax:

```
{ 'CF'
  " ←376←←235←" }
```

Alle Vordergrundfelder werden gelöscht, und der Cursor steht anschließend auf dem ersten Zeichen des ersten Vordergrundfeldes. Ist kein Vordergrundfeld vorhanden, steht der Cursor auf Spalte 79 in Zeile 24.

• Beispiel:

```
50 PRINT 'CF';
```

```
Okta1: 50 PRINT " ←376←←235←";
```

9.5.12 Die Funktion Ring Keyboard Bell

Syntax:

```
{ 'BEL'
  " ←207←" }
```

Auslösen des akustischen Signalgebers der Tastatur. Die Funktion hat keinen Einfluß auf den Bildschirminhalt.

• Beispiel:

Ausgabe einer String-Variablen ab Spalte 25 in Zeile 24 und Setzen des akustischen Signalgebers.

```
50 PRINT TAB(25,24);A$;'BEL';
```

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.13 Die Funktion Tab to next Foregroundfield

Syntax:

```
{ 'TB'  
  " ← 211 ← " }
```

Cursorpositionierung auf das erste Zeichen des folgenden Vordergrundfeldes (Vordergrundfeld siehe Pkt. 9.5.14). Folgt kein Vordergrundfeld mehr, bleibt der Cursor auf der aktuellen Position stehen.

● Beispiel:

Tabulation auf das nächste Vordergrundfeld und Eingabe in das n-te Element eines Vektors.

```
50 INPUT 'TB',X(N)
```

```
Okta1: 50 INPUT " ← 211 ← ",X(N)
```

9.5.14 Die Funktion Clear Foregroundfield

Syntax:

```
{ 'CFF'  
  " ← 230 ← " }
```

Löschen des Vordergrundfeldes, in dem sich der Cursor befindet. Nachdem die Funktion ausgeführt ist, steht der Cursor am Zeilenanfang der betreffenden Zeile oder auf der ersten Position des entsprechenden Vordergrundfeldes.

Ein Vordergrundfeld besteht aus aufeinanderfolgenden Vordergrundzeichen und wird durch Hintergrundzeichen, Zeilenanfang oder Zeilenende begrenzt.

Soll nach der Benutzung von 'CFF' in der gleichen Zeile tabuliert werden, ist die TAB-Funktion unbedingt mit zwei Parametern zu verwenden.

● Beispiel:

```
50 PRINT TAB(0,14);'CFF';"UMSATZ:";TAB(30,14);
```

	Tastatur-Bildschirm-Ein-/Ausgabe	
--	----------------------------------	--

9.5.15 Die Funktion Cursor Home

Syntax:

{ 'CH'
" ← 376 ← 222 ← " }

Positionierung des Cursors in Zeile und Spalte 0, sofern dies ein Vordergrundfeld ist. Gehört diese Position zu einem Hintergrundfeld, erfolgt die Positionierung auf die erste Stelle des ersten Vordergrundfeldes.

Folgt kein Vordergrundfeld, bleibt der Cursor auf der aktuellen Position.

● Beispiel:

50 PRINT 'CH';

Okta: 50 PRINT " ← 376 ← 222 ← ";

9.5.16 Die Funktion Dark

Syntax:

'DRK'

Blindeingabe für genau eine Variable. Diese Eingabe über Tastatur wird nicht am Bildschirm angezeigt. Die Funktion steht in der INPUT-Anweisung und gilt für die folgende Eingabevariable.

● Beispiel: 50 INPUT TAB(0,10), "PASSWORD:", 'DRK', P\$

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.17 Die Funktion Hexadezimal-Ausgabe

Syntax:
'X'

Diese Funktion bewirkt eine hexadezimale Ausgabe des darauf folgenden Strings auf Display oder Drucker.

● Beispiel:

```
10 DIM A$(5)
20 LET A$="12345"
30 PRINT 'X';"AB";"HEXA:";A$;"ENDE";

Ergebnis: C1C2HEXA:B1B2B3B4B5ENDE
```

9.5.18 Die Funktion Duplicate Field

Übernahme eines Eingabefeldes vom Bildschirm in den Eingabepuffer.

Syntax:
{ 'DF' | "←376←←236←←200←" }, TAB(A,B),TAB(C,D)

A, B, C, und D sind <NAusdr>.
TAB(A,B) : Cursorposition nach Ausführung
TAB(C,D) : Anfang des Eingabefeldes

Mit Hilfe dieser Funktion wird ein Eingabefeld bis zu einem Hintergrundzeichen dupliziert. Ist das Eingabefeld nicht durch ein Hintergrundzeichen begrenzt, werden die Zeichen bis zum Bildschirmende oder bis zur maximalen Anzahl (254) in den Eingabepuffer übertragen. 'DF' darf nur in einer INPUT-Anweisung stehen.

Achtung:
Das Eingabefeld wird nur dupliziert, wenn mindestens ein Zeichen manipuliert wurde. Sonst wird ein Leerstring bzw. der Wert 0 übergeben. Als Manipulation gilt das Drücken einer Taste, mit der Daten eingegeben werden können, auch wenn dadurch nur ein Zeichen mit einem identischen Zeichen überschrieben wird.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.



	Tastatur-Bildschirm-Ein-/Ausgabe	
--	----------------------------------	--

9.5.19 Die Funktion Duplicate Characters Limited

Übernahme einer Anzahl Zeichen aus dem Bildwiederholpeicher in den Eingabepuffer mit Begrenzung der Feldlänge

Syntax:

```
{ 'DLC'
  " ← 376 ← ← 236 ← ← 201 ← " }, TAB(A,B), TAB(C,D), TAB(E,F)
```

A, B, C, D, E und F sind <NAusdr>.

- TAB(A,B) : Cursorkoordinaten nach Ausführung
- TAB(C,D) : Anfang des Duplizierauftrags
- TAB(E,F) : Ende des Duplizierauftrags

Diese Funktion ermöglicht es, eine vorgegebene Anzahl von Zeichen mit Begrenzung der Feldlänge aus einem Eingabefeld zu duplizieren. Der Duplizierauftrag wird beim Erreichen des Bildschirmendes, eines Hintergrundzeichens, der maximalen Anzahl Zeichen (254) oder der vorgegebenen Anzahl abgebrochen. Die Anzahl errechnet sich aus der Differenz der Positionen TAB(E,F) und TAB(C,D).

● Beispiel:

```
10 INPUT 'DLC', (TAB(10,9), TAB(10,9), TAB(19,9)), A$
```

Übernahme von 10 Zeichen ab Position 10 in Zeile 9. Danach wird der Cursor nach (10,9) gesetzt. Falls die übertragenen Daten über Tastatur geändert wurden, werden sie dem Anwenderprogramm in der Variablen A\$ übergeben. Diese Funktion darf nur in einer INPUT-Anweisung stehen.

Achtung:

Das Eingabefeld wird nur dupliziert, wenn mindestens ein Zeichen manipuliert wurde. Sonst wird ein Leerstring bzw. der Wert 0 übergeben. Als Manipulation gilt das Drücken einer Taste, mit der Daten eingegeben werden können, auch wenn dadurch nur ein Zeichen durch ein identisches Zeichen ersetzt wird.

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.20 Die Funktion Duplicate Characters

Übernahme einer Anzahl Zeichen aus dem Bildwiederholtspeicher in den Eingabepuffer ohne Begrenzung der Feldlänge

Syntax:

```
{ 'DC'  
  " ← 376 ← ← 236 ← ← 202 ← " }
```

A, B, C, D, E und F sind <NAusdr>.

TAB(A,B) : Cursorkoordinaten nach Ausführung

TAB(C,D) : Anfang des Duplizierauftrags

TAB(E,F) : Ende des Duplizierauftrags

Diese Funktion ermöglicht es, eine vorgegebene Anzahl von Zeichen aus einem Eingabefeld zu duplizieren. Der Duplizierauftrag wird beim Erreichen des Bildschirmendes, eines Hintergrundzeichens, der vorgegebenen Anzahl oder der Maximalzahl (254) von Zeichen abgebrochen.

Der Duplizierauftrag begrenzt die Länge des Eingabefeldes nicht. Die Feldlänge ist bis zu einem Hintergrundzeichen oder zur Manipulation von Zeichen (254) erweiterbar. Es werden unabhängig von der Cursorposition nur die eingegebenen und duplizierten Zeichen zur Zentraleinheit übermittelt.

Diese Funktion darf nur in einer INPUT-Anweisung stehen.

Achtung:

Das Eingabefeld wird nur dupliziert, wenn mindestens ein Zeichen manipuliert wurde. Sonst wird ein Leerstring bzw. der Wert 0 übergeben. Als Manipulation gilt das Drücken einer Taste, mit der Daten eingegeben werden können, auch wenn dadurch nur ein Zeichen mit einem identischen Zeichen überschrieben wird.

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.21 Die Funktion Display Memory

Syntax:

```
{ 'DM'
  " ← 376 ← ← 236 ← ← 203 ← " }, TAB(C,D), TAB(E,F)
```

C, D, E und F sind <NAusdr>.

TAB(C,D) : Anfangsposition der Übertragung

TAB(E,F) : Endposition der Übertragung

Mit Hilfe dieser Funktion kann die Zentraleinheit die Daten des Bildwiederholerspeichers von Arbeitsplatz anfordern. Es werden alle Zeichen (Vorder- und Hintergrundzeichen) beim nächsten INPUT zur Zentraleinheit übertragen. Mit einem Leseauftrag können maximal 254 Zeichen angefordert werden. Überschreitet ein Auftrag die Bildschirmgrenze, werden die Daten bis zum Bildschirmende übertragen.

● Beispiel:

```
10 INPUT 'DM', TAB(C,D), TAB(E,F), AS
```

Anmerkung:

Die Zeichen werden ohne das 8-te Bit übertragen. Deshalb können sich bei Stringvergleichen trotz offensichtlicher Gleichheit Unterschiede ergeben. In diesem Fall kann man mit dem CALL 20 gezielt eingreifen und die entsprechenden Bits wieder setzen.

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.22 Die Funktion Cursor Position

Syntax:

```
{ 'CP'
  " ← 376 ← ← 236 ← ← 204 ← " }
```

Diese Funktion ermöglicht es, die aktuelle Cursorposition (nächste Schreib-/Eingabeposition) vom Arbeitsplatz anzufordern. Die Cursorkoordinaten werden mit dem nächsten INPUT vom Arbeitsplatz zur Zentraleinheit übertragen. Die Cursorkoordinaten haben dabei die Darstellung SSZZ mit:

```
SS = Spaltenposition (00 - 79)
ZZ = Zeilenposition (00 - 24)
```

Die Übertragung kann in eine String-Variable oder eine numerische Variable erfolgen:

```
10 INPUT 'CP', A$
20 LET A = A$(1,2) /* Spalte
30 LET B = A$(3,4) /* Zeile

10 INPUT 'CP', E
20 LET A = INT(E/100) /* Spalte
30 LET B = FRA(E/100)*100 /* Zeile
```

9.5.23 Die Funktion Define Window

Syntax:

```
{ 'DW'
  " ← 376 ← ← 236 ← ← 205 ← " }, TAB(A,B) , TAB(C,D)
; ;
```

A, B, C und D sind <NAusdr>.

TAB(A,B) : Anfangskoordinate des Fensters
(Spalte=0 Anfangszeile)

TAB(C,D) : Endkoordinate des Fensters (Spalte=79 Endzeile)

	Tastatur-Bildschirm-Ein-/Ausgabe
--	----------------------------------

In einer INPUT-Anweisung muß als Trennzeichen das ";" benutzt werden, in einer PRINT-Anweisung das ";". Diese Funktion ermöglicht es, den Bildschirm (25 Zeilen x 80 Spalten) in Zeilenfenster einzuteilen. Die Spaltenangaben werden nicht ausgewertet, d.h. es gilt immer: A=0 und C=79. Der Wert für die Anfangszeile darf nicht größer als der Wert für die Endzeile des Fensters sein. Alle Funktionen des Arbeitsplatzes beziehen sich dann nur auf das aktuelle Fenster. Ein Fenster wird wie ein verkleinerter selbständiger Bildschirm behandelt.

Rückkehr in TAMOS oder SCOPE löscht das Fenster.

9.5.24 Die Funktion No Typewriter Keyboard

Syntax:

{ 'NTY' " ← 376 ← ← 236 ← ← 206 ← " }

Die Eingabeart der Tastatur für Groß- und Kleinschreibung in Abhängigkeit der Shifttaste kann mit dieser Funktion umgeschaltet werden. 'NTY' bewirkt, daß bei nicht gedrückter Shifttaste die Großschreibung eingeschaltet ist.

Mit dem Programm "TPCODEUTIL" kann definiert werden, welche Tasten von der Umschaltung betroffen sein sollen.

9.5.25 Die Funktion Typewriter Keyboard

Syntax:

{ 'TY' " ← 376 ← ← 236 ← ← 207 ← " }

Die Eingabeart der Tastatur für Groß- und Kleinschreibung in Abhängigkeit der Shifttaste kann mit dieser Funktion umgeschaltet werden. 'TY' bewirkt, daß bei nicht gedrückter Shifttaste die Kleinschreibung eingeschaltet ist.

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.26 Die Funktion 8-Bit-Übertragung einschalten

Syntax:

" ←376 ← ←236 ← ←210 ← "

Mit Hilfe des Bit 8 im Code eines Zeichens kann gesteuert werden, ob es hell oder dunkel am Bildschirm ausgegeben wird. Ist die 8-Bit-Übertragung eingeschaltet, so wird ein Zeichen mit gesetztem 8. Bit (Bit 8=1) hell, mit nicht gesetztem 8. Bit dunkel ausgegeben. Ist die 8-Bit-Übertragung ausgeschaltet, so wird Bit 8 nicht beachtet.

Mit der neuen Funktion DM können Bildschirminhalte unterschiedlicher Helligkeit gespeichert werden. Durch Ein- bzw. Ausschaltung der 8-Bit-Übertragung können sie später wahlweise hell oder mit der "eingelesenen" Helligkeit wieder am Bildschirm ausgegeben werden.

Die Funktionen SF und SB werden von dieser Funktion übersteuert.

9.5.27 Die Funktion 8-Bit-Übertragung ausschalten

Syntax:

" ←376 ← ←236 ← ←211 ← "

Mit dieser Funktion wird die Helligkeitssteuerung über das Bit 8 in einem Zeichen beendet.

• Beispiel:

```
10 DIM N$(20), A$(27)
20 INPUT 'CS', TAB(10,5), 'SB', "NAME : ", 'SF', N$
30 INPUT 'DM', TAB(10,5), TAB(36,5), A$
:
130 PRINT TAB(10,20); " ←376 ← ←236 ← ←210 ← "; A$
140 PRINT TAB(10,21); " ←376 ← ←236 ← ←211 ← "; A$
```

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung, insbesondere auch auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG. Alle Rechte vorbehalten.“

Tastatur-Bildschirm-Ein-/Ausgabe

9.5.28 Übersicht über die verfügbaren Bildschirm- und Gerätefunktionen

Symbol	Funktion	Lead-in Code	Oktal-Code
TAB (X,Y)	Cursorpositionierung X=Spalte, Y=Zeile	376	221 und Spalte und Zeile
'BS'	Cursor um eine Stelle nach links setzen	376	210
'MP'	Speichern der aktuellen Cursorposition	376	211
'BP'	Cursor auf die gespeicherte Position setzen	376	212
'CR'	Zeilenschaltung	-	215
'LD'	Eine Zeile löschen	376	223
'LI'	Eine Zeile einfügen	376	232
'SB'	Start Hintergrundausgabe	376	231
'SF'	Start Vordergrundausgabe	376	237
'CS'	Bildschirm löschen	376	234
'CF'	Vordergrundfelder löschen	376	235
'BEL'	Akustisches Signal ausgeben	-	207
'TB'	Tabulation auf das nächste Vordergrundfeld	-	211
'CFF'	Löschen des aktuellen Vordergrundfeldes	-	230
'CH'	Cursor in Zeile 0 und Spalte 0 setzen	376	222

Tastatur-Bildschirm-Ein-/Ausgabe

Symbol	Funktion	376 ← 236	Okta1 Code	TAB- Angabe
'DRK'	Blindeingabe	-	-	-
'X'	Hexadezimal-Ausgabe	-	-	-
'DF'	Eingabefeld duplizieren	x	200	x
'DLC'	Begrenztes Eingabefeld duplizieren	x	201	x
'DC'	Duplizieren einer An- zahl Zeichen	x	202	x
'DM'	Datenübertragung aus Bildwiederholtspeicher	x	203	x
'CP'	Übertragen der aktuel- len Cursorposition	x	204	-
'DW'	Bildschirmgliederung Fenster-Definition	x	205	x
'NTY'	Großschreibung	x	206	-
'TY'	Kleinschreibung	x	207	-
	8-Bit-Übertragung "EIN"	x	210	-
	8-Bit-Übertragung "AUS"	x	211	-

Tastatur-Bildschirm-Ein-/Ausgabe

Funktionen zur gezielten Normierung von SAS-Geräten

Symbol	Funktion	Funktionscode	Okta-Code	TAB-Angabe
-	Alle SAS-Geräte außer Drucker normieren	376 236 237	200	-
-	OCR-Leser normieren	" " "	201	-
-	BCR-Leser normieren	" " "	202	-
-	IDKG/SCR normieren	" " "	203	-
-	Kassenarbeitsplatz normieren	" " "	205	-

Folgende Gerätefunktionen (<DevFkt>) betreffen ebenfalls Peripheriegeräte, die am Bildschirm angeschlossen werden. Sie sind in den entsprechenden Abschnitten beschrieben:

Formularsteuerung: 'EOL', 'LFB', 'OPN', 'IN', 'FF', 'CR'

Einzelformularverarbeitung: MCL (<NAusdr>)

Papiercassettenauswahl: 'SC1', 'SC2'

Verbreiterung des Schriftbildes: 'ELO', 'ELE'

Zeichendichte: 'LC1', 'LC2'

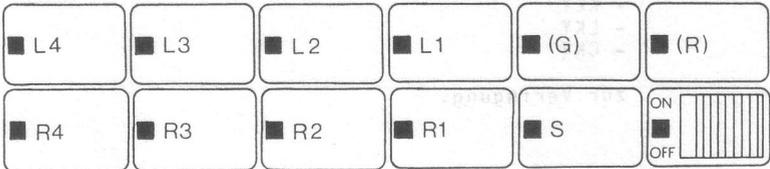
Weitere Funktionen finden Sie anschließend bei den einzelnen Gerätebeschreibungen.

Tastatur-Bildschirm-Ein-/Ausgabe

9.6.2

Die Funktion LKY

Die Funktion LKY liefert den Wert der gesetzten Rasttaste(n) (bei nicht gesetzter Rasttaste den Wert 0).



- LKY (0) liefert die Summe der Werte aller gesetzten Rasttasten.
- LKY (1) liefert den Wert der Rasttaste "R1".
- LKY (2) liefert den Wert der Rasttaste "R2".
- LKY (3) liefert den Wert der Rasttaste "R3".
- LKY (4) liefert den Wert der Rasttaste "R4".

Die Werte gesetzter Rasttasten sind:

- "R1" : 256 (2↑8)
- "R2" : 512 (2↑9)
- "R3" : 1024 (2↑10)
- "R4" : 2048 (2↑11)

• Beispiele:

```

190 IF LKY (0)=1536 GOTO 250
:
250 REM *** R2, R3 GESETZT ***

550 IF LKY (4)=2048 GOTO 660
:
660 REM *** R4 GESETZT ***
    
```

Tastatur-Bildschirm-Ein-/Ausgabe

9.6.3

Die Funktion CKY

Die Funktion CKY löscht alle gesetzten Rasttasten und setzt die Rasttastenwerte auf 0. Das Argument der Funktion muß 0 sein.

Die Funktion ist ausschließlich in den Anweisungen PRINT und INPUT erlaubt.

• Beispiele:

```
100 INPUT CKY (0),A  
190 PRINT CKY (0)
```

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und
Zuwendung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
Zuwendung verpflichtet zu Schadensersatz. Alle Rechte für den Fall
der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.“



Tastatur-Bildschirm-Ein-/Ausgabe

9.7 Eingabe über OCR-A/B-Handleser, BCR, SCR, IDKG

An die Tastatur-Schnittstelle des BA kann zusätzlich zur Tastatur ein OCR-A/B-Handleser, ein Strichcodeleser (Bar Code Reader, BCR), ein ID-Kartenleser (Swipe Card Reader, SCR) oder ein ID-Kartengerät (IDKG) angeschlossen werden.

Eingaben, die durch Programme mit entsprechenden INPUT-Anweisungen verlangt werden, können dann statt über die Tastatur auch mit diesen Geräten geschehen: OCR-A/B-Handleser und BCR sind optische Lesegeräte, die OCR-A- oder OCR-B-Schrift bzw. Balkencode lesen, SCR und IDKG sind Lesegeräte für ID-Karten mit Magnetstreifen. Über das IDKG kann der Magnetstreifen auch (durch eine entsprechende PRINT-Anweisung) beschrieben werden.

9.7.1 Ablauf einer Eingabe

Es können Eingaben in numerische oder String-Variablen mit einer INPUT-Anweisung sowohl über die Tastatur, als auch mit Hilfe der Geräte gelesen werden. Der Aufbau der überstellten Daten ist identisch. Beim Auftreten einer INPUT-Anweisung werden die Daten in ASCII-Code umgewandelt, dem Anwenderprogramm übergeben und ggfs. am BS angezeigt. Außer den eigentlichen Daten wird der Code einer Auslösetaste übertragen, der mit der Funktion KEY abgefragt werden kann. Die Daten werden in einem Puffer für das Gerät zwischengespeichert.

Läuft zu dem Zeitpunkt, wenn eine Eingabe mit einem der Geräte durchgeführt werden soll, eine Tastatureingabe, so werden beim SCR und IDKG die Eingaben zurückgewiesen, bis die Tastatureingabe beendet ist. Beim OCR-A/B- und BCR-Leser wird die Eingabe in den Zwischenpuffer übernommen. Weitere Eingaben werden ebenfalls zurückgewiesen. Ist die Tastatureingabe abgeschlossen, wird die Eingabe aus dem Zwischenpuffer übernommen und ggf. am Bildschirm angezeigt. Anschließend sind weitere Eingaben der Lesegeräte möglich.

Läuft eine Geräteeingabe, wenn eine Tastatureingabe erfolgen soll, so gehen die Tastatureingaben in den Voreingabepuffer, bis alle schon eingegebenen Daten des Gerätes übernommen wurden und eine neue INPUT-Anweisung ansteht. Während dieser Zeit werden die über Tastatur eingegebenen Zeichen nicht am Bildschirm angezeigt.

Tastatur-Bildschirm-Ein-/Ausgabe

Werden abwechselnd Eingaben über die Tastatur und Eingaben eines Lesegerätes vorgenommen, so werden diese auch in der vorgegebenen Reihenfolge verarbeitet. Sind jedoch mehrere Lesegeräte angeschlossen, und werden "durcheinander" Eingaben von verschiedenen Geräten getätigt, kann nicht gewährleistet werden, daß diese auch in der eingegebenen Reihenfolge verarbeitet werden.

Bei Erkennen einer Sonderfunktion (ESC, CTL/C) wird der dem Gerät zugeordnete Zwischenpuffer und der Voreingabepuffer der Tastatur gelöscht. Erkennt das Anwenderprogramm bei der Überprüfung oder Bearbeitung von Eingabedaten einen Fehler so kann es über den Steuercode "←376←←233←" den Platz in einen definierten Zustand überführen. Außer den genannten Maßnahmen bei Erkennen einer Sonderfunktion wird der akustische Signalgeber gesetzt. Durch Ausgabe des Oktal-Codes "←207←" beim OCR-A-Leser wird in diesem Fall nur der OCR-Leser-Puffer gelöscht und der akustische Signalgeber gesetzt. Damit stehen die fehlerhaften Daten im Voreingabepuffer für eine weitere Verarbeitung zur Verfügung. Erreicht der Cursor bei Ausgabe von Geräte-Daten ein Hintergrundzeichen, werden außer den bisher beschriebenen Maßnahmen die bisher ausgegebenen Zeichen gelöscht.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Tastatur-Bildschirm-Ein-/Ausgabe

9.7.2 Besonderheiten der einzelnen Geräte

Mit Hilfe eines Dienstprogrammes werden Codetabellen für die Geräte bearbeitet, über die die Übernahme von mit den Geräten eingelesenen Daten durch das Anwenderprogramm gesteuert werden.

9.7.2.1 Besonderheiten des OCR-A/B-Lesers

Der OCR-A/B-Leser liest bei jedem Lesevorgang eine Etikettenzeile, die mehrere Informationsfelder bis zu einer Gesamtlänge von 26 Byte enthalten kann und gibt die Daten am Bildschirm aus. Ein Informationsfeld ist wie folgt aufgebaut:

- Funktionscode (ein Byte).
- Numerische Daten. Die Länge wird vom Funktionscode beeinflusst.

Die numerische Information wird abhängig von einer Code-Tabelle in Vor- und Nachkommastellen aufgeteilt und durch den Dezimalpunkt bzw. das Dezimalkomma getrennt.

Der Steuercode wird über die Code-Tabelle auf Auslösetasten der Tastatur abgebildet und kann analog vom Anwender verarbeitet werden (Funktion KEY (0)).

Zusätzlich bestimmt der Steuercode die Länge des Eingabefeldes und den Status des Eingabefeldes:

- Standardfeld oder
- geschütztes Feld.

Geschützte Felder können im Gegensatz zu Standardfeldern nicht in der Länge modifiziert werden.

Grundsätzlich müssen die OCR-A-Zeichenfelder immer bis zu der im Steuercode angegebenen Länge mit Füllzeichen ergänzt sein, falls die Eingabedaten weniger Raum einnehmen.

Tastatur-Bildschirm-Ein-/Ausgabe

Beispiel einer Code-Tabelle für Steuer- und Füllzeichen:

Zeichen	Feldbeschreibung/Füllzeichen
A	Standard 13 Zeichen
C	Standard 3 Zeichen
D	Füllzeichen
M	Geschützt 7 Zeichen
N	Standard 9 Zeichen
P	Standard 8 Zeichen
R	Standard 10 Zeichen
U	Standard 6 Zeichen
X	Modifikation +2
Y	Modifikation -2
+	Geschützt 5 Zeichen
>	Füllzeichen
<Zi>/	Modifikation um <Zi>

Liste der Datenzeichen:

0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

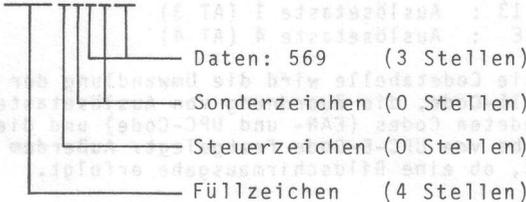
Liste der Sonderzeichen:

. | , | Ø

Die Sonderzeichen dienen lediglich der optischen Aufteilung der Felder und werden weder als Zeichen gezählt noch als Dezimalpunkt oder -komma interpretiert.

• Beispiel:

>>>DM 5.69



Weitere Beispiele für OCR-A-Zeichenfelder:

UY 02 146 20632033
DM 1690,90

M123 45.67
N3/12 34 56 78

„Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

Tastatur-Bildschirm-Ein-/Ausgabe

9.7.2.2 Besonderheiten des BCR

Die eingegebenen Daten werden in einen Zwischenpuffer abgelegt. Der Puffer ist so groß, daß bis zu vier 13-stellige oder bis zu fünf 8-stellige Etiketten voreingegeben werden können.

Bei jedem Eingabeauftrag (INPUT) wird ein Eingabefeld aus dem Puffer umcodiert und an das Anwenderprogramm übergeben. Dem Anwenderprogramm werden die Daten vom BCR grundsätzlich ohne Prüfziffer übergeben. Das heißt, daß von einem 13-stelligen Etikett nur die 12 Nutzdaten übergeben werden (bei 8-stelligen Etiketten 7 Nutzdaten). Ob die Daten vom BCR auf Bildschirm ausgegeben werden oder nicht, wird über die Tabelle der Betriebsparameter eingestellt.

Die Anzahl der übergebenen Nutzdaten ist von dem verwendeten Code abhängig.

- EAN-13 : 12 Nutzzeichen
- EAN-08 : 7 Nutzzeichen
- UPC-13 : 12 Nutzzeichen
- UPC-E : 6 Nutzzeichen

Jede Eingabe über BCR wird mit dem Code einer Auslösetaste abgeschlossen. Der Code dieser Taste wird dem Anwenderprogramm nicht in der Eingabevariablen übergeben, sondern kann mit der Funktion KEY ermittelt werden. Jedem lesbaren Code kann eine Auslösetaste zugeordnet werden. Standardmäßig sind folgende Tastencodes eingestellt:

- EAN-13 : Auslösetaste 3 (AT 1)
- EAN-08 : Auslösetaste 2 (AT 2)
- UPC-13 : Auslösetaste 1 (AT 3)
- UPC-E : Auslösetaste 4 (AT 4)

Über die Codetabelle wird die Umwandlung der BCD-Zeichen in ASCII-Code, die Zuordnung von Auslösetastencodes zu verwendeten Codes (EAN- und UPC-Code) und die Art der Übergabe von UPC-E-Code festgelegt. Außerdem wird eingestellt, ob eine Bildschirmausgabe erfolgt.

Tastatur-Bildschirm-Ein-/Ausgabe

9.7.2.3 Besonderheiten des SCR

Auf den ID-Karten können zwei Spuren (Spur 2 und 3) gelesen werden. Der Zwischenpuffer des SCR im BA kann den Inhalt beider Spuren aufnehmen. Über die Codetabelle wird eingestellt, ob nur Spur 2, nur Spur 3 oder beide gelesen werden. Dem Anwender stehen auf Spur 2 38, auf Spur 3 105 Zeichen zur Verfügung.

Ferner wird für jeden dieser Fälle der zu übergebende Auslösetastencode eingestellt.

Außerdem wird die Konvertierung der Datenzeichen in ASCII-Code und eine gewünschte Bildschirmausgabe festgelegt.

9.7.2.4 Besonderheiten des IDKG

Nach dem Einschalten des BA und der Initialisierung des IDKG befindet sich der Platz im "SCR-Modus". In diesem Zustand verhält sich das IDKG dem Anwenderprogramm gegenüber genauso wie der SCR.

Soll eine ID-Karte beschrieben werden, muß auf den IDKG-Modus umgeschaltet werden. Die Umschaltung erfolgt, gesteuert durch das Anwenderprogramm, durch Übergabe einer der Funktionen Test-Lesen, Schreiben, Kartenauswurf. Nach jedem Kartenauswurf befinden sich Platz und IDKG wieder im SCR-Modus.

Vom IDKG wird die gleiche Tabelle genutzt, die auch vom SCR genutzt wird.

Bei der Programmierung des IDKG ist darauf zu achten, daß keine Umschaltung in den IDKG-Modus erfolgt, wenn eine Karte gelesen wird. Eine Ausgabefunktion darf erst dann abgesetzt werden, wenn sichergestellt ist, daß eine vorhergehende Ausgabefunktion vollständig beendet ist.

Beim Lesen von ID-Karten verhält sich das IDKG wie der SCR.

Das Schreiben auf eine ID-Karte (mit einer PRINT-Anweisung) sollte grundsätzlich erst nach einem Test-Lesen erfolgen. Dabei wird Spur 2 und 3 gelesen. Die Daten werden bei der nächsten INPUT-Anweisung an das Anwenderprogramm übergeben. Die ID-Karte wird nicht ausgeworfen, die Ausgabe der Daten auf Bildschirm unterdrückt.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmusterantragung vorbehalten.



Tastatur-Bildschirm-Ein-/Ausgabe

● Beispiel : INPUT " ←376←←236←←231←", I\$

Der Okta1-Code ←376←←236←←231← be-
wirkt am IDKG "Lesen ohne Auswurf, ohne
Ausgabe am Bildschirm"

Treten Lesefehler auf, werden 2 Wiederholungen für die zu
lesende Spur gestartet. Bleiben auch diese Wiederholungen
erfolgreich, wird die Karte ausgeworfen.

Im Fehlerfall werden keine Daten an das Anwenderprogramm
übergeben, die Funktion Test-Lesen gilt jedoch als ausge-
führt. Dies bedeutet, daß die INPUT-Anweisung weiterhin
aktiv ist und über die Tastatur beendet werden muß, bevor
eine neue IDKG-Funktion gestartet werden kann.

Eine ID-Karte, die bei der Funktion Test-Lesen als feh-
lerhaft zurückgewiesen wurde, kann nur ohne vorhergehen-
des Test-Lesen beschrieben werden.

Das Schreiben auf ID-Karte geschieht mit einer PRINT-An-
weisung folgenden Aufbaus:

" ←376←←236←←232←←20*← D A T E N ←203←"

└─ Kennzeichen
für Datenende

└─ Aufzuzeichnende Daten

└─ Spurauswahl = 202 Spur 2
203 Spur 3
204 Spur 3 + 3

└─ Funktionscode

● Beispiel:

Schreiben des Strings 0\$ auf Spur 2 und 3 einer ID-Kar-
te.

PRINT " ←376←←236←←232←←204←";0\$;" ←203←";

Tastatur-Bildschirm-Ein-/Ausgabe

Soll eine ID-Karte ohne vorhergehendes Test-Lesen beschrieben werden (z.B. im Fehlerfall), ist sicherzustellen, daß direkt nach Absetzen der PRINT-Anweisung die zu beschreibende ID-Karte in das IDKG eingeführt wird.

Unmittelbar nach dem Schreiben einer Spur wird diese Spur gelesen und mit den Schreibdaten, die im Zwischenpuffer des BA stehen, verglichen. Sind Inhalt des Zwischenpuffers und der ID-Karte nicht identisch, folgen noch maximal 2 Schreibversuche. Sind auch diese Versuche erfolglos, wird die Karte als defekt zurückgewiesen.

Werden beide Spuren beschrieben, werden die Daten der Quellvariablen sequentiell, beginnend mit Spur 2 aufgeschrieben. Wird dabei Spur 3 nicht benutzt (bis zu 38 Zeichen), wird Spur 3 automatisch durch Ausgabe von Trennzeichen als leer gekennzeichnet.

Die Funktion Kartenauswurf dient dazu, ID-Karten, die sich nach Test-Lesen noch im IDKG befinden und nicht beschrieben werden sollen, wieder auszugeben.

Okta1-Code : " ←376 ←←236 ←←233 ←"

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung des Inhalts mit jeglichem Verfahren, insbesondere durch Nachdruck, Vervielfältigung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG. Alle Rechte vorbehalten.“



Tastatur-Bildschirm-Ein-/Ausgabe

9.8 ASCII-Code-Tabelle

Verwendet wird der 7-Bit-ASCII-Code. Das 8. Bit eines Bytes ist intern immer auf 1 gesetzt.

Zeichen	Oktal	Hexadezimal	Bit-Kombination	Taste
NUL	2.0.0	8.0	10 000 000	
SOH	2.0.1	8.1	10 000 001	
STX	2.0.2	8.2	10 000 010	
ETX	2.0.3	8.3	10 000 011	
EOT	2.0.4	8.4	10 000 100	
ENQ	2.0.5	8.5	10 000 101	CTL-E
ACK	2.0.6	8.6	10 000 110	
BEL	2.0.7	8.7	10 000 111	
BS	2.1.0	8.8	10 001 000	
TB	2.1.1	8.9	10 001 001	CTL-I
LF	2.1.2	8.A	10 001 010	
VT	2.1.3	8.B	10 001 011	
FF	2.1.4	8.C	10 001 100	CTL-L
CR	2.1.5	8.D	10 001 101	CTL-Z
SO	2.1.6	8.E	10 001 110	
SI	2.1.7	8.F	10 001 111	
DLE	2.2.0	9.0	10 010 000	
DC1	2.2.1	9.1	10 010 001	
DC2	2.2.2	9.2	10 010 010	
DC3	2.2.3	9.3	10 010 011	
DC4	2.2.4	9.4	10 010 100	
NAK	2.2.5	9.5	10 010 101	
SYN	2.2.6	9.6	10 010 110	
ETB	2.2.7	9.7	10 010 111	
CFF	2.3.0	9.8	10 011 000	
EM	2.3.1	9.9	10 011 001	
SUB	2.3.2	9.A	10 011 010	CTL-Y
ESC	2.3.3	9.B	10 011 011	
FS	2.3.4	9.C	10 011 100	
GS	2.3.5	9.D	10 011 101	
RS	2.3.6	9.E	10 011 110	
US	2.3.7	9.F	10 011 111	

Tastatur-Bildschirm-Ein-/Ausgabe

Zeichen	Oktal	Hexadezimal	Bit-Kombination	Taste
SP	2.4.0	A.0	10 100 000	Space
!	2.4.1	A.1	10 100 001	!
"	2.4.2	A.2	10 100 010	"
#	2.4.3	A.3	10 100 011	#
\$	2.4.4	A.4	10 100 100	\$
%	2.4.5	A.5	10 100 101	%
&	2.4.6	A.6	10 100 110	&
'	2.4.7	A.7	10 100 111	'
(2.5.0	A.8	10 101 000	(
)	2.5.1	A.9	10 101 001)
*	2.5.2	A.A	10 101 010	*
+	2.5.3	A.B	10 101 011	+
,	2.5.4	A.C	10 101 100	,
-	2.5.5	A.D	10 101 101	-
.	2.5.6	A.E	10 101 110	.
/	2.5.7	A.F	10 101 111	/
0	2.6.0	B.0	10 110 000	0
1	2.6.1	B.1	10 110 001	1
2	2.6.2	B.2	10 110 010	2
3	2.6.3	B.3	10 110 011	3
4	2.6.4	B.4	10 110 100	4
5	2.6.5	B.5	10 110 101	5
6	2.6.6	B.6	10 110 110	6
7	2.6.7	B.7	10 110 111	7
8	2.7.0	B.8	10 111 000	8
9	2.7.1	B.9	10 111 001	9
:	2.7.2	B.A	10 111 010	:
;	2.7.3	B.B	10 111 011	;
<	2.7.4	B.C	10 111 100	<
=	2.7.5	B.D	10 111 101	=
>	2.7.6	B.E	10 111 110	>
?	2.7.7	B.F	10 111 111	?
@	3.0.0	C.0	11 000 000	@
A	3.0.1	C.1	11 000 001	A
B	3.0.2	C.2	11 000 010	B
C	3.0.3	C.3	11 000 011	C
D	3.0.4	C.4	11 000 100	D
E	3.0.5	C.5	11 000 101	E
F	3.0.6	C.6	11 000 110	F
G	3.0.7	C.7	11 000 111	G
H	3.1.0	C.8	11 001 000	H
I	3.1.1	C.9	11 001 001	I
J	3.1.2	C.A	11 001 010	J
K	3.1.3	C.B	11 001 011	K
L	3.1.4	C.C	11 001 100	L
M	3.1.5	C.D	11 001 101	M
N	3.1.6	C.E	11 001 110	N
O	3.1.7	C.F	11 001 111	O

Tastatur-Bildschirm-Ein-/Ausgabe

Zeichen	Oktal	Hexadezimal	Bit-Kombination	Taste
P	3.2.0	D.0	11 010 000	P
Q	3.2.1	D.1	11 010 001	Q
R	3.2.2	D.2	11 010 010	R
S	3.2.3	D.3	11 010 011	S
T	3.2.4	D.4	11 010 100	T
U	3.2.5	D.5	11 010 101	U
V	3.2.6	D.6	11 010 110	V
W	3.2.7	D.7	11 010 111	W
X	3.3.0	D.8	11 011 000	X
Y	3.3.1	D.9	11 011 001	Y
Z	3.3.2	D.A	11 011 010	Z
Ä	3.3.3	D.B	11 011 011	Ä
Ö	3.3.4	D.C	11 011 100	Ö
Ü	3.3.5	D.D	11 011 101	Ü
↵	3.3.6	D.E	11 011 110	↵
β	3.3.7	D.F	11 011 111	β
a	3.4.0	E.0	11 100 000	a
b	3.4.1	E.1	11 100 001	b
c	3.4.2	E.2	11 100 010	c
d	3.4.3	E.3	11 100 011	d
e	3.4.4	E.4	11 100 100	e
f	3.4.5	E.5	11 100 101	f
g	3.4.6	E.6	11 100 110	g
h	3.4.7	E.7	11 100 111	h
i	3.5.0	E.8	11 101 000	i
j	3.5.1	E.9	11 101 001	j
k	3.5.2	E.A	11 101 010	k
l	3.5.3	E.B	11 101 011	l
m	3.5.4	E.C	11 101 100	m
n	3.5.5	E.D	11 101 101	n
o	3.5.6	E.E	11 101 110	o
p	3.5.7	E.F	11 101 111	p
q	3.6.0	F.0	11 110 000	q
r	3.6.1	F.1	11 110 001	r
s	3.6.2	F.2	11 110 010	s
t	3.6.3	F.3	11 110 011	t
u	3.6.4	F.4	11 110 100	u
v	3.6.5	F.5	11 110 101	v
w	3.6.6	F.6	11 110 110	w
x	3.6.7	F.7	11 110 111	x
y	3.7.0	F.8	11 111 000	y
z	3.7.1	F.9	11 111 001	z
ä	3.7.2	F.A	11 111 010	ä
ö	3.7.3	F.B	11 111 011	ö
ü	3.7.4	F.C	11 111 100	ü
LEAD-IN	3.7.5	F.D	11 111 101	
DEL	3.7.6	F.E	11 111 110	
	3.7.7	F.F	11 111 111	

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und
Verbreitung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
Sonderregelungen für den Fall
der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

Tastatur-Bildschirm-Ein-/Ausgabe

9.9

I/O-Makro

Das I/O-Makro ist ein Werkzeug zur felddweisen Verarbeitung von Ein- und Ausgaben am Bildschirm. Für die komplette Bearbeitung eines Ein-/Ausgabefeldes ist jeweils nur ein Aufruf des I/O-Makros notwendig, wobei Feldaufbereitung, Eingabebehandlung und Feldprüfung über Parameter gesteuert werden.

Die Feldbeschreibungparameter werden in einer Parameterdatei abgelegt. Die Parameterdatei wird mit der Systemfunktion "FB-GENERATOR" erstellt und gepflegt. Damit bietet das I/O-Makro eine einfache Möglichkeit zur Maskenerstellung und -Bearbeitung ohne Programmieraufwand.

Das I/O-Makro greift auf die Parameterdatei zu, ohne daß ein Kanal belegt wird. Die Anwendung übergibt lediglich den Dateinamen und die Satznummer des jeweiligen Parametersatzes. Obwohl das I/O-Makro zur Überprüfung einer ausgelösten Hardkeytaete auf die TF.PARAM zugreift, muß die TF.PARAM nicht offen gehalten werden.

Die Parameter aus dem Parametersatz können zum Teil durch Aufrufparameter des I/O-Makros übersteuert werden, um z.B. in verschiedenen Bildschirmzeilen dieselben Texte anzuzeigen (Ausgabedaten bleiben bspw. konstant, Positionierungsparameter werden jeweils im Programm neu gesetzt).

Im einzelnen führt das I/O-Makro die folgenden Funktionen aus, die durch Parameter in der Feldbeschreibungsdatei oder Aufrufparameter gesteuert werden können:

- Positionierung
- Prüfung und Anzeige einer Vorbelegung
- Prüfung, Formatierung und Anzeige einer Eingabe
- Platzspezifische Attributierung (Farbe, Unterstreichen, Blinken usw.)
- Anzeige von Fehlermeldungen in Form von Eingabebeschreibungen in der Nachrichtenzeile
- Manuelle Fragezeichenfunktion
- Automatische Fragezeichenfunktion in Verbindung mit Benutzerverwaltung oder HELP
- Temporäres Ausschalten der automatischen "?"-Funktion
- Bereitstellen eines Textes und Aufbereiten des Fensters für manuellen oder automatischen HELP-Aufruf, einstellbar in HELP oder über die Benutzerverwaltung
- Prüfung der benutzten Tasten
- Anzeige der zulässigen Softkeys.

Tastatur-Bildschirm-Ein-/Ausgabe

Eine Abprüfung auf Fehleingaben erfolgt innerhalb des I/O-Makros. Nur wenn eine korrekte Eingabe mit einer zulässigen Auslösetaste beendet worden ist, werden Rückgabeparameter an das Anwendungsprogramm geliefert.

Als Rückgabeparameter des I/O-Makros können abhängig vom Feldtyp die formatierte Eingabe, der Eingabewert, die Eingabelänge, die Auslösetaste bzw. eine Abbruchmeldung ausgewertet werden.

Einen Überblick über das Prinzip der Feldbehandlung bei den unterschiedlichen Ein-/Ausgaben und Auslösetasten finden Sie im Abschnitt 9.9.2.

Die einzelnen Feldbeschreibungparameter sind im Abschnitt 9.9.3 erläutert. Anschließend werden in zwei Abschnitten das Systemprogramm "FB-GENERATOR" zum Bearbeiten von Parameterdateien und der CALL 9 für die Übergabe von Hard- und Softkey-Strings an das I/O-Makro und an den BA beschrieben.

Copyright © 1986 Nixdorf Computer AG
Alle Rechte vorbehalten
Nixdorf Computer AG, Postfach 10 15 50, D-4100 Xanten 1

Die Parameter des I/O-Makros werden über den Aufrufparameter des I/O-Makros übergeben. Um z.B. in verschiedenen Bildschirmen dieselben Textausgaben (Ausgaben) zu realisieren, können die Parameter über den Aufrufparameter des I/O-Makros übergeben werden. Im einzelnen führt das I/O-Makro die folgenden Funktionen aus, die durch Parameter in der Feldbeschreibung über Aufrufparameter gesteuert werden können:

- Prüfung und Anzeige einer Vorbelegung
- Prüfung, Formatierung und Anzeige einer Eingabe
- Platzspezifische Attributierung (Farbe, Bildelemente, Blinzen usw.)
- Anzeige von Felderbindungen in Form von Feldbeschreibung in der Nachrichtenzeile
- Manuelle Freigeberfunktion
- Automatische Freigeberfunktion in Verbindung mit den Nutzerverwaltung oder HELP
- Temporäres Ausschalten der automatischen "F"-Funktion
- Bereitstellen eines Textes und Ausfüllen des Textes für manuelle oder automatische HELP-Ausgabe
- Prüfung der bedienten Tasten
- Anzeige der nächsten Softkeys.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.1 Aufruf des I/O-Makros

Syntax:

IOM <SVar1>, <NVar1>, <SVar2>, <SVar3>, <NVar2>, <NVar3>,
<SVar4>, <NVar4>, <NVar5>, <SVar5>

<SVar1>: Dateiname der Parameterdatei.
Bei mehreren I/O-Makro-Aufrufen mit derselben
Parameterdatei muß diese Variable nur beim ersten
Aufruf vorbesetzt werden.

<NVar1>: Satznummer der Feldbeschreibung aus der Parame-
terdatei (4% Variable).

<SVar2>: Übersteuerungsangaben (max. 13 Byte).
Die Bytes dieses Strings geben an, welche Über-
steuerungsparameter (Übersteuern eines Parameters
der Parameterdatei) für das I/O-Makro relevant
sind:

ASCII '1': Byte gesetzt
ASCII '0': Byte nicht gesetzt

Die Bedeutung der gesetzten Steuerbytes:

- | | | | |
|------|----|---|---|
| Byte | 1 | = | Spaltennummer aus <SVar3> |
| " | 2 | = | Zeilennummer aus <SVar3> |
| " | 3 | = | Skip-Status aus <SVar3> |
| " | 4 | = | max. Eingabelänge aus <SVar3> |
| " | 5 | = | min. Eingabelänge aus <SVar3> |
| " | 6 | = | Anzahl Nachkommastellen aus <SVar3> |
| " | 7 | = | Hardkeys aus <SVar3> |
| " | 8 | = | Softkeys aus <SVar3> |
| " | 9 | = | Attribute aus <SVar3> |
| " | 10 | = | Minimalwert für num. oder Datum-Eingabe
aus <NVar2> |
| " | 11 | = | Maximalwert für num. oder Datum-Eingabe
aus <NVar3> |
| " | 12 | = | Ausschalten der automatischen "?"-Funk-
tion für die entsprechende Eingabe |
| " | 13 | = | Vorbelegung aus <SVar4> oder <NVar4> |

Die Übersteuerungsangaben müssen nur bis zur ge-
wünschten Position (ohne Lücken) angegeben wer-
den.

Tastatur-Bildschirm-Ein-/Ausgabe

<SVar3>: Übersteuerungsstring (variable Länge).
Der Übersteuerungsstring enthält folgende Informationen:

Byte	1/2	= Spaltennummer	
"	3/4	= Zeilennummer	
"	5	= Skip-Status	
"	6/8	= max. Eingabelänge	
"	9/11	= min. Eingabelänge	
"	12/13	= Anzahl Nachkommastellen	
"	14/37	= Hardkey-Zulässigkeit	(0 = erlaubt 1 = nicht erlaubt)
"	38/53	= Softkey-Zulässigkeit	(0 = erlaubt 1 = nicht erlaubt)
"	54/n	= Attribute (pro Attribut nur die 2 letzten Bytes des oktalen Steuercodes)	

Die Anzahl für Hard- und Softkeys stimmt mit dem Aufbau des Parametersatzes überein und entspricht nicht der tatsächlich vorhandenen Anzahl Tasten.

Die Übersteuerungsangaben müssen bis zur gewünschten Position (ohne Lücken) angegeben werden.

<NVar2>: minimaler Eingabewert für numerische oder Datum-Eingabe (4% Variable).

<NVar3>: maximaler Eingabewert für numerische oder Datum-Eingabe (4% Variable).

<SVar4>: alphanumerische Vorbelegung / Rückgabe der alphanumerischen Eingabe bzw. aufbereiteten numerischen Eingabe. Diese Variable muß mit mindestens drei Bytes dimensioniert sein.

Tastatur-Bildschirm-Ein-/Ausgabe

<NVar4>: numerische oder Datum-Vorbelegung / Rückgabe der numerischen bzw. Datum-Eingabe / Länge der alpha-numerischen Eingabe (4% Variable).

<NVar5>: Statusvariable (1% Variable).
Ist als Aufrufwert hier eine "2" eingetragen, wird unabhängig vom aktuellen Skip-Status (= Art der Feldbehandlung) eine Eingabe erwartet. Rückgabewerte beziehen sich nur auf die Positionsnummer der Auslösetasten im durch CALL 9 übergebenen Hardkey- oder Softkey-String und können folgende Werte annehmen:

- 00 - 29 : benutzter Hardkey (Positionsnummer im Hardkey-String, "0" = "CR")
- 30 - 49 : benutzter Softkey (Positionsnummer im Softkey-String + 29)
- 60 : Abbruch der Eingabe durch zugeordneten virtuellen Port.

<SVar5>: I/O-Makro-Arbeitsbereich (min. Länge 1024 Bytes). Diese Variable darf im Programm nicht benutzt werden.

Hinweis:

Nicht übersteuerbare Parameter sind

- Ein-/Ausgabeart
- Art des Vorzeichens
- Ein-/Ausgabeformat für Datum
- Ein-/Ausgabeformat für numerische Ein-/Ausgabe
- Erlaubte Zeichen
- USING-Maske
- HELP-Text-Definition.

	Tastatur-Bildschirm-Ein-/Ausgabe	
--	----------------------------------	--

Fehlermöglichkeit: ERROR # 38

Bedeutung:

- Falscher Aufruf
- + <SVar5> kleiner als 1024 Bytes
- + falscher Variablentyp
- + falsche Dimensionierung
- + zu wenig Aufrufvariablen.
- Falsche Satznummer für die Parameterdatei.
- Parameterdatei nicht vorhanden.
- Probleme beim Lesen der Parameterdatei.
- Falsche Parameter
- + falscher Skip-Status
- + falsche Ein-/Ausgabeart.
- TF.PARAM nicht vorhanden.
- Probleme beim Lesen der TF.PARAM.
- Fehler in der USING-Maske.
- Fehler beim Lesen einer numerischen Variablen.
- Ein Parameter wird vom Programm übersteuert, aber die entsprechenden Angaben in <SVar3> fehlen.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.2 Prinzip der Feldbehandlung beim I/O-Makro

9.9.2.1 Behandlung von Vorbelegungen und Eingaben

• Alphanumerische Ein-/Ausgabe

Eine Vorbelegung wird linksbündig auf der vorgegebenen Position angezeigt und der Cursor auf den Feldanfang positioniert.

Bei einem Fehler nach Prüfung der Vorbelegung wird eine Fehlermeldung ausgegeben und die Vorbelegung angezeigt, die nun korrigiert werden muß.

Wird bei einer Vorbelegung ohne Veränderung eines Zeichens eine Auslösetaste gedrückt, gilt die Vorbelegung als Eingabe. Werden Zeichen verändert, wird wie bei "Duplicate Field" die Eingabe bis zum Feldende übernommen.

Mögliche Prüfungen bei einer alphanumerischen Eingabe sind:

- minimale Anzahl Bytes
- maximale Anzahl Bytes
- erlaubte Zeichen (siehe Feldbeschreibungparameter).

Bei einem Fehler in der Eingabe wird eine Fehlermeldung ausgegeben und die Eingabe muß korrigiert werden. Nach Abschluß der Eingabe werden der Eingabewert und die Eingabelänge an das Anwenderprogramm zurückgegeben.

Tastatur-Bildschirm-Ein-/Ausgabe

• Datum-Ein-/Ausgabe

Eine Datumsvorbelegung wird linksbündig ohne Trennzeichen und entsprechend der SYSMOD-Einstellung (Format YYMMDD bzw. YYYYMMDD) angezeigt. Eine Vorbelegung mit Null führt zu einer Fehlermeldung. Eine Eingabe ist mit und ohne Trennzeichen möglich.

Mögliche Prüfungen der Datum-Eingabe sind:

- max./min. Eingabewert
- zulässige Trennzeichen (b | , | - | / | : | ;)
- Eingabeformat, wenn das Datum mit Trennzeichen eingegeben wurde
- Gültigkeit (siehe CALL 32)
- min./max. Anzahl Ziffern.

Nach Abschluß der Eingabe wird das Datum im Ausgabeformat mit evtl. Trennzeichen und als numerischer Wert an das Anwendungsprogramm zurückgegeben.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

	Tastatur-Bildschirm-Ein-/Ausgabe	
--	----------------------------------	--

9.9.2.2 Behandlung von Auslösetasten

Eingaben können mit jeder Auslösetaste ("CR", Hard- und Softkeys) beendet werden, die im Parametersatz oder im entsprechenden Aufrufparameter als zulässig definiert ist.

Die benutzte Taste kann vom Programm abgefragt werden. So ist es beispielsweise möglich, durch Beenden einer Eingabe mit einer Softkey-Taste gleichzeitig einen neuen Programmzweig anzusteuern.

Nach einer Eingabe wird zuerst die benutzte Auslösetaste und erst dann die Eingabe geprüft.

Eingabefelder ohne Vorbelegung können ohne Eingabe mit jeder zulässigen Taste ungleich "CR" verlassen werden. In diesem Fall wird die Eingabe nicht geprüft und nur der Code der Auslösetaste an das Anwendungsprogramm zurückgegeben.

Wird mit Feldvorbelegung gearbeitet und mit "CR" ausgelöst, wird generell die Vorbelegung als Eingabe interpretiert und überprüft.

Tastatur-Bildschirm-Ein-/Ausgabe

• Hardkey-Auswertung

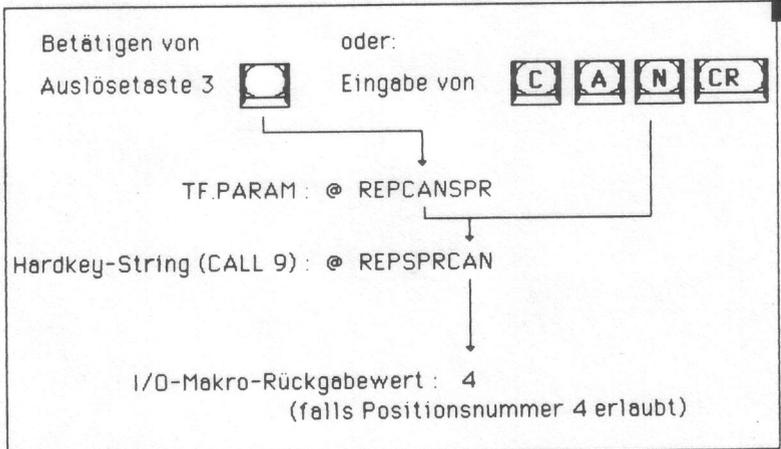
Eine durch ein Hardkeykürzel (wie bspw. CAN) gekennzeichnete Funktion kann sowohl durch Betätigen des entsprechenden Hardkeys bzw. durch explizite Eingabe über Tastatur (hier "C", "A", "N" und "CR") angestoßen werden.

Lediglich die Zuordnung von Auslösetasten zu Hardkey-Kürzeln wird dem Hardkey-String in der TF.PARAM entnommen.

Das I/O-Makro benutzt als Parameter nicht den Tastencode, sondern die Position eines Hardkey-Kürzels in einem Hardkey-String. Mit dem CALL 9, Funktion 4 (PRINT Hardkey) wird dieser Hardkey-String an das I/O-Makro übergeben.

Die Reihenfolge der Hardkey-Kürzel in diesem String muß nicht mit der Reihenfolge in der TF.PARAM übereinstimmen. Einem Anwendungsprogramm wird immer die einmal festgelegte Position im Hardkey-String zurückgegeben, so daß Auslösetasten unabhängig von der Belegung mit Hardkey-Kürzeln ausgewertet werden können.

Anwendungsprogramme können also über die Zuordnung Hardkey-Kürzel \leftrightarrow Position im Hardkey-String (CALL 9) mit Hardkeys arbeiten, ohne die aktuelle Tastenbelegung zu kennen.



Tastatur-Bildschirm-Ein-/Ausgabe

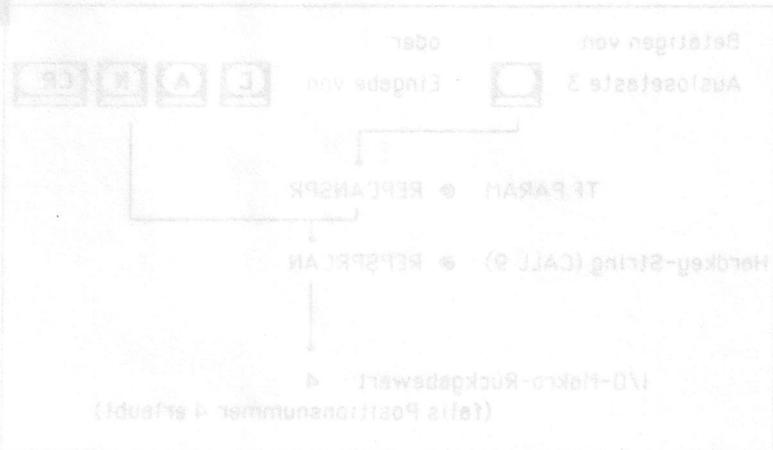
Anhand der Reihenfolge im Hardkey-String können Hardkeys in der Parameterdatei oder durch Aufrufparameter als erlaubt bzw. nicht erlaubt gekennzeichnet werden.

Wurde eine Hardkey-Taste gedrückt, entnimmt das I/O-Makro das entsprechende Kürzel der TF.PARAM und sucht dieses Kürzel im Hardkey-String.

Wenn in einem Eingabefeld ohne vorherige Eingabe und ohne Vorbelegung eine Hardkey-Taste gedrückt wurde, so wird das entsprechende Funktionskürzel im Eingabefeld angezeigt.

Wurde ein Funktionskürzel direkt über Tastatur eingegeben, wird dieses Kürzel sofort im Hardkey-String gesucht.

Anhand der Positionsnummer im String wird geprüft, ob der Hardkey zulässig ist. Im Fehlerfall werden vom I/O-Makro die zulässigen Hardkeykürzel angezeigt und eine erneute Eingabe verlangt. Die alte Eingabe bleibt erhalten und der Cursor steht am Feldanfang. Es muß jetzt lediglich eine korrekte Taste gedrückt werden.



Tastatur-Bildschirm-Ein-/Ausgabe

• Softkey-Auswertung

Die Softkeys sind Bestandteil der neuen Arbeitsplätze BA26 und BA34. Damit Anwendungsprogramme unabhängig vom angeschlossenen Bildschirmarbeitsplatz erstellt werden können, werden Softkeys je nach Arbeitsplatztyp unterschiedlich behandelt.

- Softkeys für BA26/BA34

Mit dem CALL 9 wird dem Bildschirmarbeitsplatz mitgeteilt, daß mit Softkeys (Funktion 0 "OPEN Softkey") und mit welchen Softkeys (Funktion 1 "PRINT Softkey") gearbeitet werden soll.

Die Funktion 1 bewirkt, daß der übergebene Softkey-String (maximal 9 Softkeys à 7 Zeichen) dem Bildschirm des BA26 zur Verfügung steht. Beim I/O-Makro-Aufruf werden die erlaubten Softkeys angezeigt.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung des Inhalts, insbesondere durch elektronische Medien, ist ohne schriftliche Genehmigung des Nixdorf-Verlages GmbH. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

Tastatur-Bildschirm-Ein-/Ausgabe

- Softkey-Emulation für den BA13

Das I/O-Makro emuliert die Softkeybehandlung auf dem BA13, wobei die 23. Zeile als Softkeyzeile genutzt wird. Dem Anwendungsprogramm steht damit eine Zeile weniger zur Verfügung.

CALL 9, Funktion 1 "PRINT Softkey" bewirkt die Übergabe der Softkeykürzel an das I/O-Makro. Beim I/O-Makro-Aufruf werden dann unzulässige Softkeys im Hintergrund und zulässige Softkeys im Vordergrund angezeigt.

Sollen für einige Softkeys keine Softkey-Texte definiert werden (7 Blanks), wird vom I/O-Makro an der entsprechenden Stelle ein fest definierter String (sieben mal "#") ausgegeben.

Am BA13 kann man keine speziellen Auslösetasten als Softkey zuordnen. Deshalb wird die Auslösetaste mit KEY(0)-Wert = "2" auf der rechten Seite der Tastatur als Softkey-Umschalttaste definiert. Sie bewirkt, daß das I/O-Makro auf Zeile 23 verzweigt (dort erscheint ein "?") und dort die Eingabe einer Softkey-Nummer (= Position des Softkeytextes) erwartet.

Die Eingabe einer Null führt zur Rückverzweigung zum letzten Eingabefeld. Bei Eingabe einer Nummer zwischen "1" und "9" (ohne Abschluß durch eine Auslösetaste) führt das I/O-Makro eine Zulässigkeitsprüfung durch.

Wurde eine erlaubte Softkeynummer eingegeben, wird die Ziffer anschließend in der Zeile 23 angezeigt. Es ertönt ein akustisches Signal und das I/O-Makro prüft anschließend die Eingabe des Feldes, in dem die Softkey-Umschaltung ausgelöst wurde bzw. führt die durch den Softkeytext gekennzeichnete Funktion direkt aus.

Wurde eine unzulässige Softkeynummer eingegeben, ertönt ebenfalls ein akustisches Signal. In der Nachrichtenzeile erscheint eine Fehlermeldung (unzulässiges Kommando), und es wird erneut eine Softkeynummer erwartet. Sie können also nur mit "0" oder einer der erlaubten Nummern zwischen "1" und "9" die Softkey-Emulation verlassen.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3 Felddeschreibungsparameter

In diesem Abschnitt werden die mit dem I/O-Makro steuerbaren Feldparameter beschrieben, die Sie zum I/O-Makro-Aufruf benötigen.

Beim Anlegen eines Felddeschreibungssatzes in einer Parameterdatei sind immer anzugeben:

- Spaltennummer
- Zeilennummer
- Skip-Status
- Ein-/Ausgabeart
- Maximale Eingabelänge (Anzahl Ziffern/Bytes)
- Minimale Eingabelänge (Anzahl Ziffern/Bytes)
- Anzahl Nachkommastellen
- Art des Vorzeichens
- Maske für Ausgabe.

Zusätzlich können noch angegeben werden:

- Hardkey-Zulässigkeit
- Softkey-Zulässigkeit
- HELP-Text-Definition
- Minimaler Eingabewert
- Maximaler Eingabewert/Erlaubte Zeichen
- USING-Maske (Ein-/Ausgabeformat für numerische Ein-/Ausgabe nach dem Using-Prinzip)
- Vorbelegung
- Attribute.

Die Parameter in der Parameterdatei können beim I/O-Makro-Aufruf übersteuert werden, mit Ausnahme der Werte von

- Ein-/Ausgabeart
- Art des Vorzeichens
- Maske für Ausgabe
- erlaubte Zeichen
- USING-Maske
- HELP-Text-Definition.

Zusätzlich kann beim I/O-Makro-Aufruf die automatische "?"-Funktion ausgeschaltet werden.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Vervielfältigung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patentierung oder Gebrauchsmustereintragung vorbehalten.

	Tastatur-Bildschirm-Ein-/Ausgabe
--	----------------------------------

9.9.3.1 Spaltennummer

Die Spaltennummer muß im Bereich von "0" bis "79" liegen.

9.9.3.2 Zeilennummer

Die Zeilennummer muß im Bereich von "0" bis "22" liegen.

9.9.3.3 Skip-Status

Mit dem Skip-Status wird die Art der Feldbehandlung nach den Kriterien

- Anzeige der Vorbelegung
- Prüfen der Vorbelegung nach Eingabekriterien
- Überschreiben der Vorbelegung (Eingabefeld)

definiert.

Kurzübersicht:

Skip-Status	! Anzeige Vorbelegung	! Prüfung Vorbelegung	! Eingabe
0	! ja	! nein	! nur im Fehlerfall
1	! nein	! ja	! nur im Fehlerfall
2	! ja	! ja	! nur im Fehlerfall
3	! nur Punkte	! nein	! ja
4	! ja	! nein	! ja
5	! nur Punkte	! ja	! ja
6	! ja	! ja	! ja
8	! Anzeige der Vorbelegung ohne Prüfung und Rückgabe (dient zur Anzeige von Texten)		

Ein Skip-Status "7" ist nicht implementiert.

Tastatur-Bildschirm-Ein-/Ausgabe

Bei einem Fehler, der bei Ausgabe einer Vorbelegung oder Prüfung der Vorbelegung, der Rückgabeparameter oder der Eingabe erkannt wird, läuft unabhängig vom Skip-Status folgende Fehlerroutine ab:

In Zeile 24 wird eine Fehlermeldung angezeigt. Der fehlerhafte Wert wird angezeigt und der Cursor auf den Feldanfang positioniert. Es wird eine Eingabe verlangt, die anschließend geprüft wird. Eine falsche Eingabe bleibt stehen, der Cursor wird auf Feldanfang positioniert und es wird erneut zur Eingabe verzweigt.

- Skip-Status = 0

Ein Feld wird folgendermaßen aufgebaut:

- Vorbelegung ausgeben
 - + wenn Länge der Vorbelegung kleiner als Feldlänge: löschen der restlichen Positionen innerhalb des Feldes
 - + wenn Länge der Vorbelegung größer als Feldlänge: die Vorbelegung in der Feldlänge ausgeben und zur Fehleroutine verzweigen.
- Punkte in voller Feldlänge ausgeben, wenn Vorbelegung nicht vorhanden.

Die Vorbelegung wird anschließend als Eingabe interpretiert und in die Rückgabeparameter überstellt. Tritt bei Prüfung der Rückgabeparameter ein Fehler auf, wird zur Fehleroutine verzweigt.

Schematische Darstellung des Ablaufs:

```
10 Anzeige Vorbelegung
20 Versorgen Rückgabeparameter
30 IF no error goto 100
40 Fehlermeldung
50 Positionieren Cursor auf Feldanfang
60 Start Eingabe
70 Prüfen Eingabe
80 IF no error goto 20
90 goto 40
100 Rücksprung
```



Tastatur-Bildschirm-Ein-/Ausgabe

Hinweis: Skip-Status 0 und Skip-Status 2 unterscheiden sich nur, wenn durch Übersteuerung im I/O-Makro eine Eingabe verlangt wird. Im Skip-Status 0 wird in diesem Fall eine fehlerhafte Vorbelegung ohne Fehlermeldung angezeigt. Nur wenn diese Vorbelegung als Eingabe bestätigt wird, erfolgt eine Prüfung mit anschließender Fehlermeldung.

- Skip-Status = 1

Es erfolgt keine Anzeige der Vorbelegung (das Feld bleibt unverändert). Die Vorbelegung wird geprüft. Im Fehlerfall wird die Vorbelegung ausgegeben und zur Fehlerroutine verzweigt.

Schematische Darstellung des Ablaufs:

```
10 Prüfen Vorbelegung
20 IF no error goto 110
30 Fehlermeldung
40 Ausgabe Vorbelegung
50 Positionieren Cursor auf Feldanfang
60 Start Eingabe
70 Prüfen Eingabe
80 IF no error goto 110
90 Fehlermeldung
100 goto 50
110 Versorgen Rückgabeparameter
120 Rücksprung
```

Schematische Darstellung des Ablaufs:

```
10 Anzeige Vorbelegung
20 Versorgen Rückgabeparameter
30 IF no error goto 100
40 Fehlermeldung
50 Positionieren Cursor auf Feldanfang
60 Start Eingabe
70 Prüfen Eingabe
80 IF no error goto 20
90 goto 40
100 Rücksprung
```

Tastatur-Bildschirm-Ein-/Ausgabe

● Skip-Status = 2

Die Vorbelegung wird angezeigt (siehe Skip-Status = 0) und geprüft. Im Fehlerfall wird in die Fehlerroutine verzweigt.

Schematische Darstellung des Ablaufs:

```
10 Anzeige Vorbelegung
20 Prüfen Vorbelegung
30 IF no error goto 100
40 Fehlermeldung
50 Positionieren Cursor auf Feldanfang
60 Start Eingabe
70 Prüfen Eingabe
80 IF no error goto 100
90 goto 40
100 Versorgen Rückgabeparameter
110 Rücksprung
```

● Skip-Status = 3

Statt einer Vorbelegung werden Punkte in der Länge des Eingabefeldes ausgegeben und eine Eingabe verlangt. Die Eingabe wird geprüft und im Fehlerfall in die Fehlerroutine verzweigt.

Schematische Darstellung des Ablaufs:

```
10 Anzeige Punkte
20 Positionieren Cursor auf Feldanfang
30 Start Eingabe
40 Prüfen Eingabe
50 IF no error goto 80
60 Fehlermeldung
70 goto 20
80 Versorgen Rückgabeparameter
90 Rücksprung
```

© Weitergabe sowie Vervielfältigung dieses Unterlages, Verwertung und
Mittel zur Erzielung ähnlicher Ergebnisse ist ohne schriftliche
Zustimmung des Herstellers nicht zulässig. Alle Rechte für den Fall
der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

	Tastatur-Bildschirm-Ein-/Ausgabe	
--	----------------------------------	--

● Skip-Status = 4

Die Vorbelegung wird angezeigt (siehe Skip-Status = 0) und eine Eingabe verlangt. Die Eingabe wird geprüft, und im Fehlerfall in die Fehlerroutine verzweigt.

Schematische Darstellung des Ablaufs:

```
10 Anzeige Vorbelegung
20 Positionieren Cursor auf Feldanfang
30 Start Eingabe
40 Prüfen Eingabe
50 IF no error goto 80
60 Fehlermeldung
70 goto 20
80 Versorgen Rückgabeparameter
90 Rücksprung
```

● Skip-Status = 5

Es werden Punkte in der Länge des Eingabefeldes ausgegeben. Die Vorbelegung wird geprüft und im Fehlerfall in die Fehlerroutine verzweigt. In jedem Fall wird der Cursor auf den Feldanfang positioniert und eine Eingabe verlangt. Die Eingabe wird anschließend geprüft. Im Fehlerfall wird in die Fehlerroutine verzweigt.

Schematische Darstellung des Ablaufs:

```
10 Anzeige Punkte
20 Prüfen Vorbelegung
30 IF no error goto 50
40 Fehlermeldung
50 Positionieren Cursor auf Feldanfang
60 Start Eingabe
70 Prüfen Eingabe
80 IF error goto 40
90 Versorgen Rückgabeparameter
100 Rücksprung
```

Tastatur-Bildschirm-Ein-/Ausgabe

- Skip-Status = 6

Die Vorbelegung wird angezeigt (siehe Skip-Status = 0) und geprüft. Im Fehlerfall wird eine Fehlermeldung angezeigt. In jedem Fall wird der Cursor auf den Feldanfang positioniert und eine Eingabe verlangt. Die Eingabe wird anschließend geprüft. Im Fehlerfall wird in die Fehler-routine verzweigt.

Schematische Darstellung des Ablaufs:

- 10 Anzeige Vorbelegung
- 20 Prüfen Vorbelegung
- 30 IF no error goto 50
- 40 Fehlermeldung
- 50 Positionieren Cursor auf Feldanfang
- 60 Start Eingabe
- 70 Prüfen Eingabe
- 80 IF error goto 40
- 90 Versorgen Rückgabeparameter
- 100 Rücksprung

- Skip-Status = 8

Dieser Skip-Status dient lediglich zur Anzeige von Texten. Es erfolgt keinerlei Prüfung und kein Setzen der Rückgabeparameter.

	Tastatur-Bildschirm-Ein-/Ausgabe
--	----------------------------------

9.9.3.4 Ein-/Ausgabeart

Es kann zwischen vier verschiedenen Ein-/Ausgabearten gewählt werden:

- 0 = Alphanumerisch mit Blankeliminierung
- 1 = Numerisch
- 2 = Datum
- 3 = Alphanumerisch mit Blankergänzung

Die Ein-/Ausgabeart kann beim I/O-Makro-Aufruf nicht übersteuert werden.

- Alphanumerische Ein-/Ausgabe

Es gibt zwei Möglichkeiten, eine alphanumerische Eingabe durchzuführen:

- Mit Blankeliminierung ("0")

Vor der Prüfung werden die rechts stehenden Blanks einer Vorbelegung/Eingabe gelöscht. Die an das Programm übergebene Eingabelänge bezieht sich dann auf den String, der nach der Blankeliminierung übrig geblieben ist.

- In voller Feldlänge ("3")

Die Vorbelegung/Eingabe wird im entsprechenden Rückgabeparameter auf die volle Länge des Eingabefeldes mit Blanks aufgefüllt. Wenn erlaubte Zeichen definiert werden, muß in diesem Fall auch das Blank eingetragen werden, da sonst ein Fehler ("Unzulässiges Zeichen") auftritt.

Hinweis: Eine Eingabe von "↑" wird in einem alphanumerischen Feld nur dann als Hardkey interpretiert, wenn Ein-/Ausgabeart = 3 definiert ist, da der Hochpfeil als "↑" im Hardkeystring abgestellt ist.

Tastatur-Bildschirm-Ein-/Ausgabe

● Numerische Ein-/Ausgabe

Eine Eingabe von Tausenderpunkten ist hier erlaubt. Die Ein-/Ausgabe von Tausenderpunkten und des Dezimalzeichens richtet sich nach der Einstellung per SYSMOD. Ein Vorzeichen kann nur als erstes Zeichen eingegeben werden und wird erst bei der Ausgabe entsprechend dem Parameter "Art des Vorzeichens" im Parametersatz formatiert.

● Datum - Ein-/Ausgabe

Die Prüfung des Datums wird analog zum CALL 32 durchgeführt. Datum = "0" ist als Eingabe nur dann zugelassen, wenn genau eine Null eingegeben wird. In diesem Fall erfolgt keine Eingabeprüfung. Sonst erfolgt die Eingabe nach dem Format, das per SYSMOD eingestellt ist. Es können folgende Trennzeichen eingegeben werden:

b , - / : ;

9.9.3.5 Maximale Eingabelänge (Anzahl Ziffern bzw. Bytes)

Dieser Parameter entspricht bei einer alphanumerischen Eingabe der tatsächlichen Feldlänge.

Bei numerischer Eingabe bezieht sich diese Angabe auf die Anzahl Ziffern (Vor- und Nachkommastellen), die eingegeben werden dürfen. Vorzeichen, Tausenderpunkte und Dezimalzeichen brauchen nicht berücksichtigt zu werden.

Bei Datum-Eingabe setzt sich die maximale Eingabelänge aus "Anzahl Ziffern + Anzahl Trennzeichen" zusammen, falls die Eingabe von Trennzeichen zugelassen wird.

Bei einer maximalen Eingabelänge kleiner als drei erfolgt eine Felderweiterung auf drei Zeichen.

	Tastatur-Bildschirm-Ein-/Ausgabe
--	----------------------------------

9.9.3.6 Minimale Eingabelänge (Anzahl Ziffern bzw. Bytes)

Bei numerischer Eingabe bezieht sich diese Angabe auf die Anzahl Ziffern (Vor- und Nachkommastellen), die mindestens eingegeben werden müssen. Vorzeichen, Tausenderpunkte und Dezimalzeichen brauchen nicht mitgerechnet zu werden.

Bei Datum-Eingabe muß die Anzahl der Trennzeichen hinzuge-rechnet werden, falls eine Trennzeichen-Eingabe erlaubt ist.

Bei einer maximalen Eingabelänge kleiner als drei erfolgt eine Felderweiterung auf drei Zeichen.

9.9.3.7 Anzahl Nachkommastellen

Die Anzahl der Nachkommastellen kann gleich der maximalen Anzahl Ziffern sein. "0" bedeutet, daß keine Nachkomma-stellen erlaubt sind.

9.9.3.8 Art des Vorzeichens

Dieser Parameter wird nur bei der Feld-Ausgabe ausgewertet und auch nur dann, wenn keine USING-Maske angegeben wurde:

- 0 = kein Vorzeichen erlaubt
- 1 = Vorzeichen vorne
- 2 = Vorzeichen hinten.

Die Art des Vorzeichens kann nicht übersteuert werden.

Bei der Eingabe muß das Vorzeichen immer als erstes Zei-chen eingegeben werden.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3.9 Maske für Ausgabe

Es können folgende Masken spezifiziert werden:

Numerisch

Datum

- | | |
|--|----------|
| 0 = Tausender-Punkte | YYMMDD |
| 1 = Vorlaufnullen | YYYYMMDD |
| 2 = Tausender-Punkte und Vorlaufnullen | YYWW |
| 3 = ohne Maske | YYYYWW |

• Ausgabemaske für numerische Ein-/Ausgabe

Dieser Parameter wird dann ausgewertet, wenn keine USING-Maske angegeben wurde, und bezieht sich nur auf die Ausgabe. Das I/O-Makro baut in diesem Fall selbst eine USING-Maske nach folgenden Kriterien auf:

- das Vorzeichen wird nur im negativen Fall ausgegeben
- für Maske = 0 (Tausenderpunkte) und Maske = 3 (ohne Maske) wird ein gleitendes Vorzeichen eingesetzt
- die Anzahl Vorkomma- (= max. Anzahl Ziffern - Anzahl Nachkommastellen) und Nachkommastellen wird berücksichtigt.

Der Vorteil dieses Parameters gegenüber der USING-Maske besteht darin, daß der Aufbau nicht fest ist, sondern von der Eingabelänge und den Parametern "Min./Max. Anzahl Ziffern " bzw. "Art des Vorzeichens" abhängt.

• Ausgabemaske für Datum-Ein-/Ausgabe

Diese Maske gilt nur für Rückgabeparameter. In der hier spezifizierten Form müssen auch die Vorbelegung und die minimalen und maximalen Eingabewerte angegeben werden.

	Tastatur-Bildschirm-Ein-/Ausgabe	
--	----------------------------------	--

9.9.3.10 Hardkey-Zulässigkeiten

In der Reihenfolge, die durch den mit CALL 9 übergebenen Hardkey-String festgelegt ist, können die Hardkeys als erlaubt oder nicht erlaubt spezifiziert werden.

Die Eingaben, die im FB-Generator die Erlaubnis festlegen, sind "Y" und "N", werden aber im Parametersatz als "0" und "1" abgestellt und auch so übersteuert.

9.9.3.11 Softkey-Zulässigkeiten

In der Reihenfolge, die durch den mit CALL 9 übergebenen Softkey-String definiert ist, können die Softkeys als erlaubt oder nicht erlaubt spezifiziert werden.

Die Eingaben, die im FB-Generator die Erlaubnis festlegen, sind "Y" und "N", werden aber im Parametersatz als "0" und "1" abgestellt und auch so übersteuert.

9.9.3.12 HELP-Text-Definition

Für jedes Eingabefeld kann ein HELP-Text inkl. der Fensterposition und -größe angegeben werden. Der HELP-Text wird spezifiziert durch Manual-ID und Textnummer. Das I/O-Makro übergibt diese Informationen an den HELP-Prozessor, so daß vor einer Eingabe mit der HELP-Taste oder innerhalb des Schulungsmodes eine eingabespezifische Information abgerufen werden kann.

Die HELP-Text-Definition kann nicht im I/O-Makro-Aufruf übersteuert werden.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3.9 Maske für Ausgabe

Es können folgende Masken spezifiziert werden:

Numerisch

Datum

- | | |
|--|----------|
| 0 = Tausender-Punkte | YYMMDD |
| 1 = Vorlaufnullen | YYYYMMDD |
| 2 = Tausender-Punkte und Vorlaufnullen | YYWW |
| 3 = ohne Maske | YYYYWW |

• Ausgabemaske für numerische Ein-/Ausgabe

Dieser Parameter wird dann ausgewertet, wenn keine USING-Maske angegeben wurde, und bezieht sich nur auf die Ausgabe. Das I/O-Makro baut in diesem Fall selbst eine USING-Maske nach folgenden Kriterien auf:

- das Vorzeichen wird nur im negativen Fall ausgegeben
- für Maske = 0 (Tausenderpunkte) und Maske = 3 (ohne Maske) wird ein gleitendes Vorzeichen eingesetzt
- die Anzahl Vorkomma- (= max. Anzahl Ziffern - Anzahl Nachkommastellen) und Nachkommastellen wird berücksichtigt.

Der Vorteil dieses Parameters gegenüber der USING-Maske besteht darin, daß der Aufbau nicht fest ist, sondern von der Eingabelänge und den Parametern "Min./Max. Anzahl Ziffern " bzw. "Art des Vorzeichens" abhängt.

• Ausgabemaske für Datum-Ein-/Ausgabe

Diese Maske gilt nur für Rückgabeparameter. In der hier spezifizierten Form müssen auch die Vorbelegung und die minimalen und maximalen Eingabewerte angegeben werden.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3.10 Hardkey-Zulässigkeiten

In der Reihenfolge, die durch den mit CALL 9 übergebenen Hardkey-String festgelegt ist, können die Hardkeys als erlaubt oder nicht erlaubt spezifiziert werden.

Die Eingaben, die im FB-Generator die Erlaubnis festlegen, sind "Y" und "N", werden aber im Parametersatz als "0" und "1" abgestellt und auch so übersteuert.

9.9.3.11 Softkey-Zulässigkeiten

In der Reihenfolge, die durch den mit CALL 9 übergebenen Softkey-String definiert ist, können die Softkeys als erlaubt oder nicht erlaubt spezifiziert werden.

Die Eingaben, die im FB-Generator die Erlaubnis festlegen, sind "Y" und "N", werden aber im Parametersatz als "0" und "1" abgestellt und auch so übersteuert.

9.9.3.12 HELP-Text-Definition

Für jedes Eingabefeld kann ein HELP-Text inkl. der Fensterposition und -größe angegeben werden. Der HELP-Text wird spezifiziert durch Manual-ID und Textnummer. Das I/O-Makro übergibt diese Informationen an den HELP-Prozessor, so daß vor einer Eingabe mit der HELP-Taste oder innerhalb des Schulungsmodes eine eingabespezifische Information abgerufen werden kann.

Die HELP-Text-Definition kann nicht im I/O-Makro-Aufruf übersteuert werden.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3.13 Minimaler Eingabewert

Gegen diesen Wert wird eine numerische oder Datum-Eingabe geprüft. Entsprechendes gilt bei der Prüfung von Vorbelegungen.

9.9.3.14 Maximaler Eingabewert/Erlaubte Zeichen

Gegen den max. Eingabewert wird eine numerische oder Datum-Eingabe geprüft. Entsprechendes gilt bei der Prüfung von Vorbelegungen.

Der Parameter "Erlaubte Zeichen" gilt nur für die alphanumerische Ein-/Ausgabe. Werden keine Angaben gemacht, sind alle Zeichen erlaubt. Die Prüfung der zulässigen Zeichen erfolgt wie beim CALL 62.

Die erlaubten Zeichen können einzeln eingegeben werden oder es kann ein Bereich von erlaubten Zeichen definiert werden, wobei das Zeichen " " als "bis" interpretiert wird.

Beispiele:

```
A B C D X Y Z
A   D X Y Z
0   5 . ,
```

Eine Eingabe von Oktalcodes ist nicht möglich bzw. führt zu Fehlern im I/O-Makro.

Soll keine Prüfung auf erlaubte Zeichen stattfinden, so muß "N" eingegeben werden.

Die "erlaubten Zeichen" können nicht im I/O-Makro-Aufruf übersteuert werden.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung ist ohne schriftliche Genehmigung des Nixdorf Computer AG. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3.15 USING-Maske

Eine hier angegebene USING-Maske entspricht einer Aufbereitungsmaske bei "PRINT USING".

Sobald dieser Parameter angegeben ist, wertet das I/O-Makro die Parameter "Art des Vorzeichens" und "Maske für Ausgabe" nicht mehr aus. Durch den starren Aufbau der USING-Maske kann es unter Umständen passieren, daß einige Vor- und Nachkommastellen nicht ausgegeben werden, wenn beim Erstellen der USING-Maske die Parameter "Max. Anzahl Ziffern/Bytes" und "Art des Vorzeichens" nicht berücksichtigt wurden.

Dieser Parameter kann nicht im I/O-Makro-Aufruf übersteuert werden.

9.9.3.16 Vorbelegung

Eine vorhandene Vorbelegung wird wie eine Eingabe behandelt.

Bei einer Datum-Vorbelegung muß der Wert im Format der Ausgabemaske angegeben werden.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3.17 Attribute

Bei Aufruf des I/O-Makros werden grundsätzlich alle eingestellten Attribute (z.B. intensiv, Unterstrich, blinken u.ä.) ausgeschaltet. Es werden 'SF' (Vordergrundausgabe) und die vorgegebenen Attribute, die auf dem angeschlossenen Arbeitsplatz möglich sind, eingeschaltet. Vor dem Rücksprung aus dem I/O-Makro werden diese Attribute wieder ausgeschaltet.

Es gibt Ausnahmefälle, bei denen die Ausgabe grundsätzlich ohne Attribute erfolgt:

- Beim Hintergrundblank als Feldbegrenzung
- bei Fehlermeldungen
- bei Informationen in der "?"-Funktion.

Beim Übersteuern der Attribute im I/O-Makro-Aufruf muß ein String mit oktalen Steuercodes angegeben werden.

Zeichenattribute	Steuercode
Blinken ein	← 201 ← ← 201 ←
Unterstreichen ein	← 202 ← ← 201 ←
Durchstreichen ein	← 203 ← ← 201 ←
Tiefstellen ein	← 204 ← ← 201 ←
Hochstellen ein	← 205 ← ← 201 ←
Inverse Darstellung ein	← 206 ← ← 201 ←
Doppelte Höhe obere Zeile ein	← 207 ← ← 201 ←
Teilhell/Teilintensiv ein	← 210 ← ← 201 ←
Doppelte Breite ein	← 211 ← ← 201 ←
Zeichen ignorieren ein	← 212 ← ← 201 ←
Geschütztes Feld ein	← 213 ← ← 201 ←
Doppelte Höhe untere Zeile ein	← 214 ← ← 201 ←



© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung ist ohne schriftliche Genehmigung des Nixdorf Computer AG. Alle Rechte vorbehalten.“
 In Zweifelsfällen sind die Herstellerangaben zu beachten. Alle Rechte vorbehalten.
 der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

Tastatur-Bildschirm-Ein-/Ausgabe

Farbsteuerungen Steuercode

Randfarbe normieren	←220← ←2xx←
Hintergrundfarbe normieren (Screen)	←221← ←2xx←
Vordergrundfarbe normieren (Screen)	←222← ←2xx←
Hintergrundfarbe einschalten	←223← ←2xx←
Vordergrundfarbe einschalten	←224← ←2xx←

Farbe Code

schwarz	200
rot	201
grün	202
gelb	203
blau	204
magenta	205
cyan	206
weiß	207
schwarz+teilintensiv	210
dunkelrot	211
dunkelgrün	212
oliv (dunkelgelb)	213
dunkelblau	214
purpur	215
dunkelcyan	216
grau	217

9.9.3.17
 Copyright © 1986 Nixdorf Computer AG
 Alle Rechte vorbehalten. Nachdruck, Vervielfältigung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG.

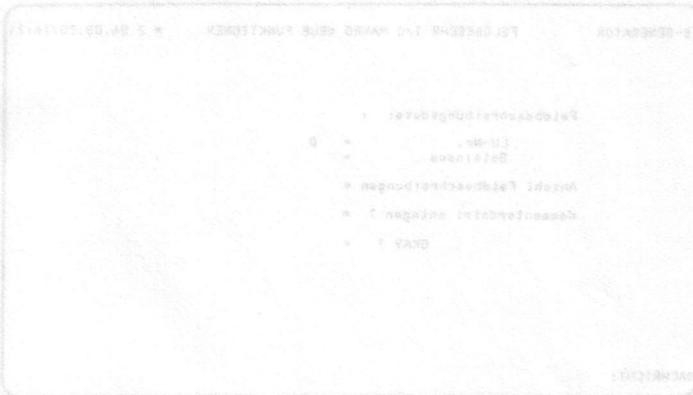
Tastatur-Bildschirm-Ein-/Ausgabe

9.9.3.18 Automatische "?"-Funktion

Die automatische "?"-Funktion wird im HELP-Parameter-Selektor oder in der Benutzerverwaltung eingeschaltet. Standardmäßig werden bei eingeschalteter automatischer "?"-Funktion die Elemente des Hardkeystings, die zulässig sind, vor einer Eingabe automatisch angezeigt.

Über eine Aufrufvariable des I/O-Makros ist es möglich, die automatische "?"-Funktion für die aktuelle Eingabe auszuschalten. Damit kann das Anwendungsprogramm eigene Fehlermeldungen ausgeben, ohne daß sie bei Aufruf des I/O-Makros sofort wieder durch die Anzeige der zulässigen Tasten überschrieben werden.

Ist die automatische "?"-Funktion nicht aktiviert, besteht die Möglichkeit, sich durch Eingabe von "?" die zulässigen Tasten trotzdem anzeigen zu lassen.



Tastatur-Bildschirm-Ein-/Ausgabe

9.9.4 Felddeschreibungs-(FB)-Generator

Mit der Systemfunktion "FB-GENERATOR" werden Felddeschreibungsdateien (Parameterdateien) angelegt und bearbeitet. Diese Funktion ist unter TAMOS im Selektor eingebunden.

Es ist möglich, jede Felddeschreibung mit einem erläuternden Kommentar zu versehen. Dieser Kommentar wird in einer eigenen Datei abgelegt. Der Name der Kommentardatei wird erzeugt, indem an den Namen der Felddeschreibungsdatei ein ".C" angehängt wird. Deshalb darf der Dateiname einer Felddeschreibungs-Datei nicht länger als 12 Zeichen sein.

Eine Beschreibung aller Felddeschreibungsparameter finden Sie im Abschnitt 9.9.2. Während der Parametereingabe können Sie Zusatzinformationen über HELP abrufen.

Nach dem Aufruf der Funktion "FB-GENERATOR" wird die folgende Maske angezeigt:

```
FB-GENERATOR          FELDBESCHR I/O MAKRO NEUE FUNKTIONEN          * 2 86.08.20/16:21

Felddeschreibungsdatei :
      LU-Nr.           = 0
      Dateiname       =
Anzahl Felddeschreibungen =
Kommentardatei anlegen ? =
      OKAY ?         =

NACHRICHT:
```

Tastatur-Bildschirm-Ein-/Ausgabe

LU-Nummer und Dateiname der Feldbeschreibungsdatei müssen eingegeben werden. Ist die angegebene Datei keine Parameterdatei, wird eine Fehlermeldung ausgegeben.

Bei einer neuen Datei muß zusätzlich angegeben werden:

- die Anzahl Feldbeschreibungen, die die Datei enthalten soll,
- ob eine Kommentardatei angelegt werden soll ("Y") oder nicht ("N").

Hinweis:

Die Parameterdatei wird bei Bedarf automatisch vergrößert. Die hier angegebene Anzahl Feldbeschreibungen wird nur zur Initialisierung der Datei benutzt.

Nach Eingabe von "OKAY = Y" wird die Parameterdatei angelegt und zur Funktionsauswahl auf der nächsten Maske verzweigt.



Tastatur-Bildschirm-Ein-/Ausgabe

FB-GENERATOR	FELDBESCHR	I/O	MAKRO	NEUE	FUNKTIONEN	* 2 86.08.20/16:12
Dateiname :		D/RE.PARAM		Kommentar =		
Spaltennummer	=			Zellennummer	=	
Skip-Status	=					
Ein-/Ausgabeart	=					
Max. Anz. Ziffern/ Max. Anz. Bytes	=			Min. Anz. Ziffern/ Min. Anz. Bytes	=	
Anz. Nachkommastellen	=					
Art d. Vorzeichens	=					
Maske f. Ausgabe	=			Weitere Parameter ?	=	
						OKAY ? =
Funktion =		Feldbeschreibungsnummer =				

NACHRICHT:

Folgende Funktionen können ausgewählt werden (Feld "Funktion ="):

- CRE Anlegen einer Feldbeschreibung
- AME Ändern einer Feldbeschreibung
- DEL Löschen einer Feldbeschreibung
- REV Anzeigen einer Feldbeschreibung
- LST Listen von Feldbeschreibungsnummern mit
 Kommentaren
- PRT Drucken von Feldbeschreibungsnummern mit
 Kommentaren
- REO Reorganisieren der Parameterdatei
- VGL Erstellen einer Protokolldatei mit den Unter-
 schieden zwischen zwei Parameterdateien als Vor-
 bereitung für Disk-Maintenance.

Die "Feldbeschreibungsnummer" ist die Satznummer der Parameterdatei, die im I/O-Makro-Aufruf angegeben wird.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.4.1 Anlegen einer Feldbeschreibung (CRE)

```

FB-GENERATOR          FELDBESCHR I/O MAKRO NEUE FUNKTIONEN          * 2 86.09.15/11:55
  Datename : D/RE.PARAM          Kommentar =
  Spaltennummer =                Zeilennummer =
  Skip-Status =
  Ein-/Ausgabeart =
  Max. Anz. Ziffern/          Min. Anz. Ziffern/
  Max. Anz. Bytes            Min. Anz. Bytes =
  Anz. Nachkommastellen =
  Art d. Vorzeichens =
  Maske f. Ausgabe =          Weitere Parameter ? =
                                   OKAY ? =
  Funktion = CRE          Feldbeschreibungsnummer =
  NACHRICHT:
  
```

Nach Eingabe der Feldbeschreibungszahl werden zunächst die Parameter erfaßt, die immer für eine Feldbeschreibung erforderlich sind:

- Kommentar (nur bei vorhandener Kommentardatei)
- Spaltennummer
- Zeilennummer
- Skip-Status
- Ein-/Ausgabeart
- Max. Anzahl Ziffern/Bytes
- Min. Anzahl Ziffern/Bytes
- Anzahl Nachkommastellen
- Art des Vorzeichens
- Maske für Ausgabe.

Durch Eingabe von "Y" bei "Weitere Parameter?" wird nach Eingabe von "OKAY = Y" zur nächsten Maske gesprungen, wo optionale Parameter erfaßt werden können. Wurde bei "Weitere Parameter?" ein "N" eingegeben, wird der Parametersatz angelegt und wieder zur Funktionsauswahl verzweigt (Feld "Funktion =").

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.“



Tastatur-Bildschirm-Ein-/Ausgabe

Maske für zusätzliche Parameter:

```

FB-GENERATOR          FELDBESCHR I/O MAKRO NEUE FUNKTIONEN          # 2 86.08.20/16:16
Dateiname : D/RE.PARAM          Kommentar :
Hardkey-String          =
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
Softkey-String          = +-----+-----+-----+-----+-----+-----+
Help-ID                 =          Manual-ID =          Text-Nr. =
Fensterparameter       :          erste Zeile =          Anz-Zeilen =
Min. Eingabewert       =
Max. Eingabewert       =
/erlaubte Zeichen      =
Ein-/Ausgabeformat     =          =          =
(USING-Maske)
Vorbelegung            =
Attribute               =          OKAY ? =

Funktion : CRE          Feldbeschreibungnummer : 103
NACHRICHT:

```

Als Zusatz-Parameter können folgende Angaben spezifiziert werden:

- erlaubte Hardkeys
- erlaubte Softkeys
- HELP-Text-Definition
- Min. Eingabewert
- Max. Eingabewert/erlaubte Zeichen
- Ein-/Ausgabeformat (USING-Maske)
- Vorbelegung.

Eine Eingabe von "N" bei "Hardkey-String" bzw. "Softkey-String" bewirkt eine Vorbelegung aller Hardkeys bzw. Softkeys mit "N", d.h. alle Hardkeys bzw. Softkeys sind unzulässig, sofern sie nicht im Aufruf übersteuert werden.

Durch Eingabe von "Y" bei "Attribute" oder bei Vorbelegung wird nach Eingabe von "OKAY = Y" zur nächsten Maske gesprungen. Wurde bei "Attribute" und bei Vorbelegung ein "N" eingegeben, wird der Parametersatz angelegt und wieder zur Funktionsauswahl gesprungen.

Tastatur-Bildschirm-Ein-/Ausgabe

Eingabemaske für Attribute:

```

FB-GENERATOR      FELDBESCHR I/O MAKRO NEUE FUNKTIONEN      # 2 86.08.20/16:19
Dateiname : D/RE.PARAM      Kommentar :
-----
Bildschirm-Steuerung = N      Farb-Steuerung = N      Farbe
-----
Blinken      =      Randfarbe normieren      =
Unterstreichen      =      Hintergrundfarbe normieren      =
Durchstreichen      =      Vordergrundfarbe normieren      =
Tiefstellen      =      Hintergrundfarbe setzen      =
Hochstellen      =      Vordergrundfarbe setzen      =
Invert      =
Doppelte Hoeh     =
Intensitaet      =
Doppelte Breite   =
Zeichen ignorieren      =
Geschuetztes Feld  =
-----
Funktion : CRE      Feldbeschreibungsnummer : 104
NACHRICHT:
    
```

Auf dieser Maske werden die Parameter zur Bildschirm- und Farbsteuerung angegeben.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“



Tastatur-Bildschirm-Ein-/Ausgabe

Nach Eingabe von "OKAY = Y" wird der Parametersatz angelegt und zur Funktionsauswahl gesprungen. Wurde in der vorherigen Eingabemaske im Feld "Vorbelegung" ein "Y" eingegeben, wird zuvor noch folgende Maske angezeigt:

FB-GENERATOR		FELDBESCHR I/O MAKRO NEUE FUNKTIONEN		# 2 86.08.20/16:19	
Dateiname : D/RE.PARAM		Kommentar :			
Bildschirm-Steuerung = N		Farb-Steuerung = N		Farbe	
Blinken	=	Randfarbe normieren	=		
Unterstreichen	=	Hintergrundfarbe normieren	=		
Durchstreichen	=	Vordergrundfarbe normieren	=		
Tiefstellen	=	Hintergrundfarbe setzen	=		
Hochstellen	=	Vordergrundfarbe setzen	=		
Invers	=				
Doppelte Hoehe	=				
Intensitaet	=				
Doppelte Breite	=				
Zeichen ignorieren	=				
Beschuetztes Feld	=				
Vorbelegung	=				
Funktion : CRE		Feldbeschreibungsnummer : 104			
NACHRICHT:					

Sie können nun die von Ihnen gewünschte Vorbelegung eingeben.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.4.2 Ändern einer Feldbeschreibung (AME)

Nach Eingabe der Parametersatz-Nummer werden die Feldbeschreibungs-Parameter angezeigt und können geändert werden. Ablauf und Bildschirmaufbau sind analog zur Funktion "CRE".

9.9.4.3 Löschen einer Feldbeschreibung (DEL)

Nach Eingabe der Parametersatz-Nummer werden die Feldbeschreibungs-Parameter angezeigt (vgl. Funktion "CRE").

Durch Quittieren mit "OKAY = Y" wird der Parametersatz gelöscht.

9.9.4.4 Anzeigen einer Feldbeschreibung (REV)

Nach Eingabe der Parametersatz-Nummer werden die Feldbeschreibungs-Parameter angezeigt (vgl. Funktion "CRE").

Die Funktion empfiehlt sich, falls ein vorhandener Satz im wesentlichen übernommen werden soll. Bei der anschließenden Anwahl von "CRE" und Angabe einer freien Satznummer werden Ihnen die mit "REV" angezeigten Parameter-Werte als Vorgabe angezeigt.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.4.5 Listen von Feldbeschreibungen in Kurzform (LST)

Nach Eingabe eines Bereichs von Parametersatz-Nummern ("Feldbeschreibungs-Nummer ... bis = ...") werden die Nummern mit den zugehörigen Kommentaren aufgelistet. Ist die Kommentardatei nicht vorhanden, so erfolgt eine entsprechende Meldung, und die Feldbeschreibungsnummern werden ohne Kommentar angezeigt. Ist ein Parametersatz nicht vorhanden, erscheint der Kommentar "Feldbeschreibung existiert nicht". Ist eine Bildschirmseite gefüllt, kann mit "CR" geblättert werden.

FB-GENERATOR	FELDBESCHR I/O MAKRO NEUE FUNKTIONEN	# 2 86.08.20/17:55
Dateiname : D/RE.PARAM		Kommentar =
Feldbeschreibung	Kommentar	
1		
Funktionsbeschreibung (REV)		
Nach Eingabe der Parametersatz-Nummer werden die Feldbeschreibungs-Parameter angezeigt (vgl. Funktion "CRE").		
Die Funktion empfiehlt sich, falls ein vorhandener Satz im wesentlichen übernommen werden soll. Bei der anschließenden Auswahl von "CRE" und Angabe einer freien Zeilennummer wird die Feldbeschreibung in der angegebenen Zeile eingegeben.		
Funktion : LST	Feldbeschreibungsnummer = 1	bis = 1
NACHRICHT: 310	:TASTE <CR> DRUECK	

9.9.4.6 Drucken von Feldbeschreibungen in Kurzform (PRT)

Diese Funktion ist analog zur Funktion "LST". Die Feldbeschreibungsliste wird auf dem mit Hilfe des Programms "GERAETEZUORDNUNG" für Ihren BA unter \$LPT eingetragenen Drucker ausgegeben. Es müssen neben den Parametern für den Feldbeschreibungsbereich die Zeilen pro Seite angegeben werden.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.4.7 Reorganisieren der Datei (REO)

Durch das Löschen oder Erweitern von Feldbeschreibungen kann es zu nicht genutzten/nicht mehr nutzbaren Bereichen in der Parameterdatei kommen. Die Reorganisation legt den Datenbereich wieder zusammenhängend an und aktualisiert den Verwaltungsbereich der Parameterdatei. Für die Reorganisation legt die Funktion eine Hilfsdatei auf der LU an, die für die Parameterdatei eingetragen ist. Die Hilfsdatei erhält den Namen 'UT.WORKpp' mit pp=Portnummer und hat die Länge der reorganisierten Parameterdatei. Nach erfolgreicher Reorganisation wird die Hilfsdatei wieder gelöscht.

9.9.4.8 Vergleichen zweier Feldbeschreibungsdateien (VGL)

Mit dieser Funktion wird ein Update von Feldbeschreibungsdateien per Disk-Maintenance vorbereitet. Zwei Feldbeschreibungsdateien werden verglichen (Satz für Satz) und die unterschiedlichen Parametersätze werden in eine Protokoll-Datei abgestellt. Diese Datei kann über die Disk-Maintenance in eine MAFI übernommen werden und dient im Zielsystem als Eingabe für den von der Disk-Maintenance aufgerufenen FB-Generator.

Nach Aufruf der Funktion werden in einer Folgemaske LU-Nummern und die Namen der Protokolldatei und Referenzdatei erfaßt.

Tastatur-Bildschirm-Ein-/Ausgabe

```
FB-GENERATOR          FELDBESCHR I/O MAKRO NEUE FUNKTIONEN      # 2 86.08.20/17:57
Dateiname : D/RE.PARAM
Protokoll-Datei :
LU-Nr.      = 0
Dateiname   =
Referenz-Datei :
LU-Nr.      =
Dateiname   =
OKAY ? =
Funktion : VGL
NACHRICHT:
```

Wenn mit "OKAY = Y" quittiert ist, wird die Funktion durchgeführt und anschließend zur Funktionseingabe zurückgesprungen.

Das Nixdorf-Business-BASIC ist ein eingetragenes Warenzeichen der Nixdorf Computer AG. Die Rechte vorbehalten. © Nixdorf Computer AG, 1986.

Tastatur-Bildschirm-Ein-/Ausgabe

9.9.5 CALL 9

Übergabe der aktuellen Hard-/Softkey-Belegung an das I/O-Makro.

Syntax:

CALL <NAusdr>, <NVar1>, <SVar1>, <SVar2>

<NAusdr>: Der Wert dieses numerischen Ausdrucks ist die Nummer des CALL-Unterprogramms (in diesem Fall 9).

<NVar1> : Funktionsnummer:

0: OPEN SOFTKEY

1: PRINT SOFTKEY

Beim BA13 wird der Softkeystring in den I/O-Makro-Arbeitsbereich übertragen. Beim BA26 wird der String zusätzlich an den Arbeitsplatz übergeben.

2: PRINT SOFTKEY-ZULÄSSIGKEIT

Diese Funktion ist nur beim BA26 wirksam. Sie ist bei Benutzung des I/O-Makros überflüssig, da das I/O-Makro die Softkeybehandlung für alle Arbeitsplatztypen beinhaltet.

3: CLOSE SOFTKEY

4: PRINT HARDKEY.

Für die Funktion 0 und 3 sind keine weiteren Parameter notwendig.

© . Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und
Zitierung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
Zurücksendung an den Herausgeber. In besonderen Fällen kann ein Fall
der Patentteilung oder Gebrauchsmustereintragung vorbehalten.



Tastatur-Bildschirm-Ein-/Ausgabe

<SVar1>: Bei den Funktionen 1, 2 und 4 muß diese String-Variable, wie folgt, vorbelegt werden:

- Funktion 1:
9 Softkeykürzel A, 7 Bytes.
Der String muß mit einer Länge von mindestens 63 Bytes dimensioniert sein.
- Funktion 2:
9 Zulässigkeitsangaben in der Form:
0: zulässiger Softkey
1: unzulässiger Softkey.
Der String muß mit einer Länge von mindestens 9 Bytes dimensioniert sein.
- Funktion 4:
max. 24 Hardkeykürzel A 3 Bytes.
Es müssen nicht alle Hardkeys angegeben werden, sondern nur die benötigten.

<SVar2>: Bei Funktion 1, 4: I/O-Makro-Arbeitsbereich (min. Länge 1024 Bytes).

Tastatur-Bildschirm-Ein-/Ausgabe

● Parameter

Die in dem Parametersatz abgelegten Werte sind in den folgenden Masken aufgeführt.

```

FB-GENERATOR          FELDBESCHR I/O MAKRO NEUE FUNKTIONEN          # 2 86.08.20/16:12
  Dateiname : 0/PB.PARAMETER          Kommentar =
  Spaltennummer      = 43              Zeilennummer      = 10
  Skip-Status        = 5
  Ein-/Ausgabeart    = 1
  Max. Anz. Ziffern/  = 008            Min. Anz. Ziffern/  = 002
  Max. Anz. Bytes
  Anz. Nachkommastellen = 2
  Art d. Vorzeichens = 1
  Maske f. Ausgabe    = 0
  Weitere Parameter ? = Y
  OKAY ?              = Y
  Funktion = REV      Feldbeschreibungnummer = 101
  NACHRICHT:
  
```

```

FB-GENERATOR          FELDBESCHR I/O MAKRO NEUE FUNKTIONEN          # 2 86.08.20/16:12
  Dateiname : 0/PB.PARAMETER          Kommentar :
  Hardkey-String      = N
  +-----+-----+-----+-----+-----+-----+-----+-----+
  Softkey-String      = N +-----+-----+-----+-----+-----+-----+
  Help-ID             = N              Manual-ID =          Text-Nr. =
  Fensterparameter : --               erste Zeile =          Anz-Zeilen =
  Min. Eingabewert   = -              1000,00
  Max. Eingabewert   =
  /erlaubte Zeichen  =                999000,00
  Ein-/Ausgabeformat = N
  (USING-Maske)
  Vorbelegung        = N
  Attribute           = N              OKAY ? = Y
  Funktion : REV      Feldbeschreibungnummer : 101
  NACHRICHT:
  
```

Tastatur-Bildschirm-Ein-/Ausgabe

• Beschreibung der I/O-Makro-Aufrufe

1. IOM-Aufruf (Zahl 1)

Die Daten im Parametersatz werden nicht übersteuert.

2. IOM-Aufruf (Zahl 2)

Es wird die Zeilenangabe sowie die Hard- und Softkey-Vorgabe übersteuert (<SVar2> = S2\$ im Programm). Es sind dann die beiden ersten Hard- und Softkeys zugelassen, d.h. im Beispiel END/CAN bzw. ESCAPE/HOCHPF. Sie sind durch Angaben in <SVar3> = S3\$ im Programm (Byte 14,15 bzw. 38/39) übersteuert.

3. IOM-Aufruf (Summe)

Es werden die Werte "Spalte", "Zeile", "Skip-Status" und "max. Eingabelänge" übersteuert und das Attribut "Inverse Darstellung" wird eingeschaltet (Byte 1,2,3,4,9 in <SVar2> = S2\$ im Programm).

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts, nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Tastatur-Bildschirm-Ein-/Ausgabe

```

10 REM
20 REM ***** VARIABLENDEFINITION FUER DEN IOM-AUFRUF
25 REM
30 GLOBAL S1$(17),S2$(13),S3$(80),S4$(50),S5$(1024)
40 GLOBAL 4%,N1,N2,N3,N4,1%,N5
50 REM
60 REM ***** HILFSFELDER
70 REM
80 DIM B$(72),C$(63),D$(22)
100 DIM 4%,Z
110 IF ERR 0 GOSUB 760
120 LET D$="-"
130 PRINT 'CS'
140 REM
150 REM ***** MASKE AUSGEBEN
160 REM
170 PRINT TAB (29,10);"1. ZAHL : ";
180 PRINT TAB (29,12);"2. ZAHL : ";
190 PRINT TAB (29,13);D$
200 PRINT TAB (29,15);"SUMME . : ";
210 REM
220 REM ***** DATEINAME UND SATZNUMMER
230 REM
240 LET S1$="00/PB.PARAMETER" /*NAME DER PARAMETER-DATEI
250 LET N1=101 /*SATZNUMMER
260 CLEAR Z
270 CLEAR S2$ /*UEBERSTEUERUNGSPARAMETER LOESCHEN
280 CLEAR S3$ /*UEBERSTEUERUNGSSTRING LOESCHEN
290 REM
300 REM ***** HARD-/SOFTKEY-STRING AUFBEREITEN
310 REM UND UEBERGEHEN
320 REM
330 LET B$="ENDCANSPRCREAMEPRTREV@ " /*HARDKEY-STRING
340 LET C$="ESCAPE HOCHPF.ANLEGENAENDERNANZEIGENLOESCHE"
350 REM SOFTKEY-STRING
360 CALL 9,0 /*OPEN SOFTKEY
370 CALL 9,1,C$,S5$ /*PRINT SOFTKEY
380 CALL 9,4,B$,S5$ /*PRINT HARDKEY
390 REM
400 REM ***** 1. IOM-AUFRUF
410 REM
420 IOM S1$,N1,S2$,S3$,N2,N3,S4$,N4,N5,S5$
430 REM
431 REM
432 REM

```

Tastatur-Bildschirm-Ein-/Ausgabe

```
440 REM ***** SVAR2/3 VORBESETZEN, 2. IOM-AUFRUF
450 REM
460 LET S2$="0100001100000" /*UEBERSTEUERN SVar2 :
470 REM ZEILE, HARD- UND SOFTKEY
480 LET S3$="0",S3$ /*SVar3 MIT "NULL" VORBESETZEN
490 LET S3$(3,4)="12" /*NEUE ZEILENUMMER = 12
500 LET S3$(16,37)="11111111111111111111111111111111"
505 REM /* BYTE NR. 14,15="00"
510 LET S3$(40,53)="11111111111111111111"
515 REM /* BYTE NR. 38,39="00"
520 LET S3$(54)=" " /*GRENZZEICHEN ABSTELLEN
530 IOM S1$,N1,S2$,S3$,N2,N3,S4$,N4,N5,S5$
540 REM
550 REM ***** PROG. STEUERUNG DURCH HARD-/SOFTKEY
560 REM
570 ON N5 GOTO 740, 260 /*SPRUNGVERTEILER HARDKEY
575 REM FUER "END/CAN"
580 ON N5-29 GOTO 740, 260 /*SPRUNGVERTEILER SOFTKEY
585 REM FUER "ESCAPE/HOCHPF."
590 REM
600 REM ***** HARD-/SOFTKEYS VORBESETZEN
610 REM
620 LET N4=N4+Z /*NUMERISCHE VORBELEGUNG
630 LET S2$="1111000100000" /*SPALTE,ZEILE,SKIP-STATUS
640 REM MAX. EINGABELAENGE,ATTRIBUT
650 LET S3$(1,8)="42158009"
660 LET S3$(54)="_206_201_" /*ATTRIBUT : INVERS
670 REM
680 REM ***** 3. IOM-AUFRUF
690 REM
700 IOM S1$,N1,S2$,S3$,N2,N3,S4$,N4,N5,S5$
710 CALL 9,3 /*CLOSE SOFTKEY
720 STOP
730 CHAIN ""
744 PRINT TAB (0,24);"ENDE"
750 CHAIN ""
760 REM FEHLERBEHANDLUNG
```

© Weitergabe sowie Vervielfältigung dieser Unterlage, Vervielfältigung und
Mittlung ihres Inhaltes nicht gestattet, soweit nicht ausdrücklich zugestanden.
Zuwendungsbedingungen verpflichten zu Schadensersatz. Alle Rechte für den Fall
der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

	Die Magnetplatte
--	------------------

10 Die Magnetplatte

10.1 Das Ein-/Ausgabekonzept

Für die Magnetplatte bildet ein logisches IOCS (Input-Output-Control-System) die Schnittstelle zwischen der Sprache Business-BASIC und dem Peripheriegerät. Das IOCS übernimmt die Verwaltung der Datenbestände. Die Daten werden vom IOCS als Dateien organisiert. Dem Programmierer wird der Zugriff auf die Daten ohne Rücksicht auf ihre physikalische Anordnung auf dem Datenträger nach rein logischen Gesichtspunkten ermöglicht.

Zur Bearbeitung von Dateien stellt Business-BASIC folgende Anweisungen zur Verfügung:

- BUILD #
- CLOSE #
- KILL
- MAT READ #
- MAT WRITE #
- OPEN #
- PRINT #
- READ #
- SEARCH #
- WRITE #

10.1.1 Dateiorganisation

Die Daten einer Datei werden in sogenannten Datensätzen untergliedert. Die Organisation einer Datei bestimmt, wie die Datei auf einem Datenträger angelegt wird und welche Möglichkeiten des Zugriffs auf die Datensätze bestehen. Auf der Magnetplatte unterstützt das IOCS folgende Arten der Dateiorganisation:

- formatierte Dateien
- relative Dateien
- Indexdateien
- Textdateien

Bei jeder dieser Organisationsarten ist sowohl ein sequentieller als auch ein direkter Zugriff auf die Daten möglich.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patentierung oder Gebrauchsmusteranmeldung vorbehalten.“



	Die Magnetplatte	
--	------------------	--

10.1.2 Record-Pointer

Für Magnetplattendateien wird vom IOCS ein Datensatzzeiger (Record-Pointer) geführt, sobald eine Datei eröffnet ist. Greifen mehrere Teilnehmer auf dieselbe Datei gleichzeitig zu, wird der Zeiger für jeden Teilnehmer individuell geführt.

Der Record-Pointer enthält die aktuelle Zugriffsposition in der Datei, d.h. die relative Satznummer des in der Datei zuletzt bearbeiteten Datensatzes. Der Inhalt des Record-Pointers kann mit der BASIC-Funktion CHF abgefragt werden.

Bei Textdateien wird die Satznummer des aktuellen Daten-Blocks (1 Block = 1 Sektor) und die Byte-Position innerhalb dieses Blocks, die auf das zuletzt bearbeitete Zeichen folgt, geführt.

10.1.3 Das Kanalkonzept

Die Bearbeitung der Dateien wird über Datenkanäle gesteuert. Jedem Programm stehen eigene Datenkanäle, die beginnend bei 0 durchnummeriert sind, zur Bearbeitung zur Verfügung.

Beim Einsatz von TAMOS sind die Kanäle 0 und 1 belegt.

Auf jedem Kanal kann genau eine Datei zur gleichen Zeit eröffnet werden. Magnetplattendateien können auf verschiedenen Kanälen des gleichen oder verschiedener Anwendungen gleichzeitig eröffnet werden (File-sharing).

10.1.4 Dateidefinitionen

Dateidefinitionen (Satzlänge, Organisationsarten, usw.) sind bei der Programmierung mit Business-BASIC nur bei der Dateierstellung erforderlich. Bei der Bearbeitung einer Datei werden die notwendigen Parameter aus den Dateikennsätzen übernommen.

Die Magnetplatte	Die Magnetplatte
------------------	------------------

10.1.5

Der Dateiname

Der Dateiname ist bei der Verarbeitung einer Datei nur einmal, und zwar bei der Dateieröffnung, anzugeben.

- Beispiel:

```
OPEN #2,"FAKTUR"
```

Diese Anweisung eröffnet auf Kanal 2 eine Plattendatei mit dem Namen FAKTUR.

Für alle weiteren Anweisungen, die der Bearbeitung der Datei dienen, ist nur noch die Angabe der Kanalnummer erforderlich.

Der Dateiname kann in einer String-Variablen oder als String-Literal in einer BASIC-Anweisung angegeben werden.

Bei Magnetplattendateien besteht der Dateiname aus 1 bis 14 Buchstaben, Ziffern oder Bindestrichen. Das erste Zeichen des Dateinamens muß ein Buchstabe sein.

```
<DatName> ::= <Bu> [ <Bu> | <Zi> | - ]13
```

- Beispiele:

```
CL1-S.0, TEST1, T
```

• Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und
 Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
 Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall
 der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.

Die Magnetplatte

10.1.6 Das Konzept der logischen Einheiten

Bei der Bearbeitung von Dateien auf Magnetplatten, speziell beim Anschluß mehrerer Laufwerke und der Bearbeitung mehrerer Problemkreise gleichzeitig, ist es unumgänglich, Dateien auf verschiedenen Magnetplatten gleichzeitig zu bearbeiten.

Der Inhalt von Magnetplatten wird deshalb durch logische Einheitennummern (LU-Nummern) unterschieden, die sowohl auf der Platte selbst als auch beim Arbeiten unter TAMOS auf der Systemplatte in einer sogenannten Archiv-Datei eingetragen sind. Die Systemplatte erhält grundsätzlich die logische Einheitennummer 0.

Wenn Ihr System nur eine Magnetplatte besitzt, wird auch diese in mindestens zwei "Physikalische Einheiten" aufgeteilt, wobei jede eine logische Einheit aufnehmen kann. Damit ist gewährleistet, daß Sie ein Arbeitsgebiet (eine bestimmte logische Einheit) gegen ein anderes austauschen können, während des Betriebssystems auf der Magnetplatte verbleibt.

Eingehender wird das Konzept der logischen Einheiten im "Bedienerhandbuch 8870" (modellabhängig) erläutert.

In BASIC-Programmen ist die LU-Nummer beim Eröffnen einer Datei mit der Anweisung OPEN #, und zwar in der Form:

<LU>/

bündig vor dem Dateinamen anzugeben. Wird eine Datei auf der logischen Einheit 0 (Systemplatte) angesprochen, kann diese Angabe entfallen.

● Beispiel:

Eröffnung der Datei FAKTUR auf Datenkanal 2. Die Datei liegt auf der logischen Einheit 3.

20 OPEN #2,"3/FAKTUR"

Die Magnetplatte

10.2 Aufbau von Magnetplattendateien

Alle auf einer Magnetplatte gespeicherten Daten sind in Dateien organisiert. Jede Plattendatei besteht aus einem oder mehreren Blöcken (Sektoren):

- einem Datei-Kennsatzblock (Header) und
- 0 bis n Datenblöcken.

Bei formatierten Dateien und bei Textdateien können zusätzlich zum Datei-Kennsatzblock Kennsatz-Erweiterungsblöcke vorhanden sein. Anwenderdaten sind in jeder von BASIC unterstützten Darstellungsart

- ASCII-Code
- BCD-Ganzzahl
- Gleitkommazahl

in Datendateien mit Ausnahme von Textdateien abzustellen. Textdateien erlauben nur Darstellung in ASCII-Code.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhaltes nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

10

	Die Magnetplatte	
--	------------------	--

10.3 File-Sharing

NIROS erlaubt das Bearbeiten von Magnetplattendateien durch mehrere Teilnehmer, auch wenn gleichzeitig auf denselben Datensatz zugegriffen wird. Dies erfordert das Vorhandensein von Schutzfunktionen gegen:

- Löschen von Dateien, während mehrere Teilnehmer diese benutzen.
- Unkontrolliertes, gleichzeitiges Verändern (updaten) von Datensätzen und damit verbundene Datenverfälschung.

Der Schutz gegen Löschen von Dateien, während andere Teilnehmer darauf zugreifen, ist im Punkt "KILL-Anweisung" beschrieben.

Der Schutz vor unkontrolliertem, gleichzeitigen Verändern ist durch die Möglichkeit des gezielten Sperrrens von Datensätzen gegeben.

Sperrren von Datensätzen

Ein Datensatz wird gesperrt, wenn eine der Anweisungen

- MAT READ #
- MAT WRITE #
- READ #
- WRITE #

ausgeführt wird und nach dem letzten Operanden kein ";" folgt. Außerdem setzt

SEARCH #

bei den Modi 1, 2, 3 und 5 Satzsperrren auf, wenn nach dem letzten Operanden ein ":" folgt.

Die Anweisung PRINT # setzt grundsätzlich Satzsperrre auf!

Werden mit einer der Anweisungen MAT READ #, MAT WRITE #, READ # oder WRITE # mehrere Datensätze gleichzeitig gelesen oder geschrieben, ist die Satzsperrre nur für den ersten gelesenen/geschriebenen Satz wirksam.

© Nixdorf Computer AG, 1985. Alle Rechte vorbehalten. Dieses Dokument ist ein Dokument der Nixdorf Computer AG. Es ist urheberrechtlich geschützt und darf nicht ohne schriftliche Genehmigung der Nixdorf Computer AG kopiert, verteilt oder in irgendeiner Weise öffentlich zugänglich gemacht werden.

Die Magnetplatte

Pro Datenkanal kann jeweils nur ein Satz gesperrt werden! Sollen in einer Datei gleichzeitig mehrere Sätze gesperrt werden, muß die Datei auf mehreren Kanälen gleichzeitig eröffnet und pro Kanal ein Satz gesperrt werden.

In Textdateien können keine Sätze gesperrt werden!

Wirkung der Satzsperrung

Ein gesperrter Datensatz kann im allgemeinen von anderen Teilnehmern nicht bearbeitet (gelesen oder geschrieben) werden. Wurde die Datei ohne den Parameter "R" eröffnet, bewirkt die Adressierung eines gesperrten Satzes einen Task-Wechsel. Bei jeder Aktivierung des Programmes wird versucht, den Satz zu lesen. Er wird erst dann zur Verfügung gestellt, wenn er vom sperrenden Programm wieder freigegeben ist. Es erfolgt keine Statusmeldung. Bei Eröffnung mit Parameter "R" wird bei Adressierung eines gesperrten Satzes BASIC-Fehler 123 erzeugt, auf den gezielt reagiert werden kann.

Aufheben der Satzsperrung

Das Aufheben der Satzsperrung kann nur durch das Programm erfolgen, das den Satz gesperrt hat.

Eine Satzsperrung wird aufgehoben, wenn:

- der gesperrte Satz auf dem Kanal, auf dem er gesperrt wurde, erneut gelesen oder geschrieben wird und in der ausführenden Anweisung auf den letzten Operanden ein ";" folgt.
- ein anderer Datensatz auf dem gleichen Kanal gelesen oder geschrieben wird.
- der Kanal, auf dem gesperrt wurde, geschlossen wird.
- das Programm durch eine der Bedingungen:
 - Auftreten einer END-Anweisung
 - durch "ESC" und "CTL-C" oder "CTL-Y", "ESC" und "CTL-C"beendet wird.

	Die Magnetplatte	
--	------------------	--

Dead lock

Beim Arbeiten mit Satzsperrung ist es möglich, daß sich zwei oder mehrere Programme gegenseitig blockieren. Bei der Organisation und auch bei der Programmierung sollte darauf geachtet werden, daß keine Konstellation von Zugriffen entsteht, die einen "dead lock" verursachen kann.

• **Beispiel:**

Programm A und Programm B arbeiten beide mit den Dateien X und Y.
Beide Programme sperren die Datensätze beim Lesen und geben sie beim Zurückschreiben bzw. beim Lesen des nächsten Satzes wieder frei.

Ablauf:

Zeitscheibe	Programm	
1	A	Lesen und sperren von Satz 15 in der Datei X.
2	B	Lesen und sperren von Satz 16 in der Datei Y.
3	A	Auswertung der gelesenen Daten und Lesen von Satz 16 in der Datei Y.
4	B	Auswertung der gelesenen Daten und Lesen von Satz 15 in der Datei X.

Beide Programme versuchen, den vom jeweils anderen Programm gesperrten Datensatz zu lesen. In diesem Fall muß zumindest eines der beiden Programme abgebrochen werden, um dem anderen das Weiterarbeiten zu ermöglichen.

Die Magnetplatte	Die Magnetplatte

10.4 Erstellung von Magnetplattendateien

Für das Erstellen von Plattendateien steht neben den Programmen, die im EXPERT-Selektor eingebunden sind, die BASIC-Anweisung

BUILD #

zur Verfügung, die im Kapitel "BASIC-Anweisungen" ausführlich beschrieben wird.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und
 Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
 Zuwiderhandlung führt zur Verurteilung. Die Rechte an dem Inhalt
 der Patenterteilung oder Gebrauchsmusterteilung vorbehalten.“

	Die Magnetplatte	Die Magnetplatte
--	------------------	------------------

10.5 Der Zugriff auf Magnetplattendateien 10.5

Für den Zugriff auf Datensätze in Magnetplattendateien stehen in Business-BASIC die Anweisungen:

- READ #
- WRITE #
- PRINT #
- MAT READ #
- MAT WRITE #

zur Verfügung.

Alle Magnetplattendateien erlauben sowohl direkten als auch sequentiellen Zugriff.

Zum Bearbeiten des Indexbereiches von Indexdateien steht die Anweisung:

SEARCH #

zur Verfügung.

Das Eröffnen und Schließen von Dateien erfolgt mit den Anweisungen:

- OPEN #
- CLOSE #

10.5.1 Die Dateieröffnung

Bevor auf Datensätze in einer Datei zugegriffen werden kann, muß die Datei mit der Anweisung

OPEN #

auf einem beliebigen freien Kanal eröffnet werden.

Eine Magnetplattendatei kann von einem Programm auf mehreren Kanälen gleichzeitig eröffnet werden. Pro Kanal kann jedoch nur jeweils eine Datei eröffnet werden.

Magnetplattendateien können von mehreren unabhängigen Programmen gleichzeitig eröffnet werden.

Die Magnetplatte

10.5.2 Der Dateizugriff

Der Dateizugriff ist in den einzelnen Dateiorganisationsarten unterschiedlich und erfolgt auch über unterschiedliche Anweisungen. Ausführlich ist der Dateizugriff deshalb bei den verschiedenen Dateiartern beschrieben.

10.5.3 Das Schließen von Dateien

Nach der Bearbeitung einer Datei ist der Kanal, auf dem diese Datei eröffnet wurde, mit der Anweisung:

CLOSE #

wieder zu schließen. Dadurch wird dieser Kanal für die Bearbeitung einer anderen Datei frei.

Kanäle werden auch dann geschlossen, wenn:

- eine END-Anweisung auftritt,
- mit einer CHAIN-Anweisung ein Prozessor aufgerufen wird,
- durch "ESC" und "CTL-C" oder "CTL-Y", "ESC" und "CTL-C" ein Programm abgebrochen wird.

Anmerkung:

Mit BUILD # eröffnete Kanäle werden nur durch die Anweisungen CLOSE # und END geschlossen!

Mit einer CLOSE #-Anweisung können beliebig viele Dateien geschlossen werden.

BASIC-Fehler 49 wird gemeldet, wenn ein Kanal geschlossen werden soll, auf dem keine Datei eröffnet ist. Mehrere Kanäle sollten wegen unzureichender Fehlerauswertung nicht mit einer Anweisung geschlossen werden.

10

Wichtiges: Die Weitergabe sowie die Vervielfältigung dieses Untertitels, die Weiterentwicklung und die Weitergabe des Inhalts, nicht nur in elektronischer Form, sondern auch in gedruckter Form, ist ohne schriftliche Genehmigung des Herstellers Nixdorf Computer AG. Die Weitergabe dieses Untertitels, die Weiterentwicklung und die Weitergabe des Inhalts, nicht nur in elektronischer Form, sondern auch in gedruckter Form, ist ohne schriftliche Genehmigung des Herstellers Nixdorf Computer AG. Die Weitergabe dieses Untertitels, die Weiterentwicklung und die Weitergabe des Inhalts, nicht nur in elektronischer Form, sondern auch in gedruckter Form, ist ohne schriftliche Genehmigung des Herstellers Nixdorf Computer AG.

	Die Magnetplatte	Die Magnetplatte
--	------------------	------------------

10.6 Das Löschen von Dateien

Für das Löschen von Dateien stellt BASIC die Anweisung

KILL

zur Verfügung. KILL löscht die ersten zwei Zeichen des Dateinamens im Inhaltsverzeichnis der Platte (INDEX). Die belegten Plattenblöcke werden in der Plattenbelegungsliste als frei eingetragen.

Anmerkung:

Ist die Datei noch von einem anderen Teilnehmer eröffnet, werden die belegten Blöcke erst beim nächsten IPL oder INSTALL freigegeben.

10.7 Die Funktion CHF

Die Funktion CHF ermöglicht dem Programmierer, die aktuelle Größe einer Datei oder die augenblickliche Zugriffsposition innerhalb einer Datei zu ermitteln. Das Argument der Funktion definiert die Kanalnummer und die durchzuführende Funktion. Für die verschiedenen Arten von Plattendateien werden durch CHF verschiedene Angaben übergeben bzw. bestehen verschiedene Funktionsmöglichkeiten:

Formatierte Dateien

<NAusdr>

Funktion

Kanalnummer

Nummer des höchsten geschriebenen Datensatzes +1.

Kanalnummer + 100

Nummer des Datensatzes, auf den zuletzt zugegriffen wurde.

Kanalnummer + 200

auf den zuletzt bearbeiteten Satz folgendes Byte-Displacement (relativ zum Blockanfang)

Kanalnummer + 300

Satzlänge in Worten

Die Magnetplatte	Die Magnetplatte
------------------	------------------

Relative Dateien

<NAusdr>	Funktion
Kanalnummer	Anzahl Datensätze, die die Datei maximal aufnehmen kann.
Kanalnummer + 100	Nummer des Datensatzes, auf den zuletzt zugegriffen wurde.
Kanalnummer + 200	Nummer des 1. Bytes des zuletzt bearbeiteten Datensatzes, relativ zum Blockanfang.
Kanalnummer + 300	Satzlänge in Worten.
Kanalnummer + 400	Nummer des höchsten geschriebenen Datensatzes +1, wenn der EOF-Zeiger definiert ist. Das ist er nur, wenn die Datei irgendwann vorher mit OPEN "C" eröffnet worden ist. Sonst ergibt sich die Anzahl der Datensätze, die die Datei aufnehmen kann.

Indexdateien

<NAusdr>	Funktion
Kanalnummer	Nummer des höchsten Datensatzes.
Kanalnummer + 100	Nummer des Datensatzes, auf den zuletzt zugegriffen wurde.
Kanalnummer + 200	Byteadresse des zuletzt bearbeiteten Datensatzes, relativ zum Blockanfang
Kanalnummer + 300	Satzlänge in Worten.

Für Indexdateien können weitere Informationen über den Dateistatus mit SEARCH # -Modus 1- abgefragt werden.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

Die Magnetplatte	Die Magnetplatte
------------------	------------------

Textdateien

Relative Dateien

<NAusdr>

Funktion

<NAusdr>

Kanalnummer	Nummer des höchsten geschriebenen Datenblocks +1. Bei erweiterten Textdateien muß allerdings die Anzahl der Kennsatzerweiterungsblöcke von diesem Wert abgezogen werden.
Kanalnummer + 100	Nummer des aktuellen Datenblocks.
Kanalnummer + 200	Nummer des Bytes, das hinter dem zuletzt übertragenen Byte liegt, relativ zum Anfang des aktuellen Blocks.

Nummer des höchsten geschriebenen Datensatzes +1, wenn der EOF-Zeiger des Datensatzes nicht auf den Anfang des Datensatzes vorwärts bewegt worden ist. Sonst ergibt sich die Anzahl der Datensätze, die das Datei aufnehmen kann.

Indexdateien

Funktion

<NAusdr>

Kanalnummer	Nummer des höchsten Datensatzes.
Kanalnummer + 100	Nummer des Datensatzes, auf den zuletzt zugegriffen wurde.
Kanalnummer + 200	Byteadresse des zuletzt bearbeiteten Datensatzes, relativ zum Blockanfang.
Kanalnummer + 300	Satzlänge in Worten.

Für Indexdateien können weitere Informationen über den Dateistatus mit SEARCH & MODUS I abgefragt werden.

	Dateibearbeitung	Dateibearbeitung
--	------------------	------------------

11 Dateibearbeitung

11.1 Dateiorganisationsarten

Das System 8870 bietet eine Reihe von Dateiorganisationsarten, damit für jeden Verwendungszweck die günstigste Organisationsart zur Verfügung steht.

Eingeschränkt werden die Organisationsmöglichkeiten durch den Aufbau des Datenträgers. Auf der Magnetplatte stehen die Organisationsarten

- formatierte Datei
- relative Datei
- Indexdatei
- Textdatei

zur Verfügung. Bei jeder dieser vier Dateiartern ist eine

- direkte oder
- sequentielle

Bearbeitung möglich.

Hingegen ist es auf den Peripheriegeräten

- Magnetband
- Magnetbandcassette
- Drucker
- Lochkartenleser
- Lochstreifenleser/-Stanzer

wegen der Beschaffenheit der Datenträger nur möglich, die Dateien

- sequentiell

zu organisieren und zu bearbeiten.

In den folgenden Abschnitten werden die verschiedenen Organisationsarten einschließlich ihrer Programmierung näher erläutert.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Dateibearbeitung

11.2 Formatierte Dateien

Eine formatierte Datei besteht aus einem Datei-Kennsatz (Header) und der entsprechenden Anzahl Datenblöcke, die erforderlich sind, um alle geschriebenen Datensätze aufzunehmen.

Sie belegt auf der Platte keinen zusammenhängenden Bereich und die Satzanzahl wird bei Erstellung nicht vorgegeben. Erweiterungen werden, wenn sie erforderlich sind, über die Liste der freien Plattenblöcke vorgenommen.

Bei Datensätzen mit Satznummern

> $128 * INT (256 / \text{Satzlänge in Worten})$

werden zusätzlich zum Dateikennsatz 1 bis 128 Kennsatzerweiterungsblöcke angelegt.

Ein Kennsatzerweiterungsblock ermöglicht die Verwaltung von maximal 256 Datenblöcken.

Alle Datensätze einer formatierten Datei haben die gleiche Struktur hinsichtlich Anordnung und Art der einzelnen Datenfelder. Ein Datensatz kann bis zu 64 verschiedene Datenfelder (ITEMs) aufnehmen. Die Formatbeschreibung wird im Datei-Kennsatz abgelegt.

Die maximale Anzahl von Datensätzen für eine formatierte Datei ist abhängig von der Satzlänge.

Die Satzanzahl darf jedoch nicht größer als 65535 werden.

Minimale Satzlänge: 1 Wort

Maximale Satzlänge: 256 Worte

Datenfelder, die mit Hilfe von Strings über WRITE # definiert werden, belegen außer dem Platz für den String ein weiteres Byte für ein Grenzzeichen. Zusätzlich wird, falls erforderlich, das Feld eines Strings auf ganze erforderlich, das Feld eines Strings Worte aufgefüllt (gilt sowohl für über WRITE # als auch über MAT WRITE # angelegte Datenfelder, s. Beispiele im nächsten Abschnitt).

Formatierte Dateien ermöglichen sowohl sequentiellen als auch direkten Zugriff auf die Datensätze.

Dateibearbeitung

11.2.1 Erstellen formatierter Dateien

Beim Erstellen einer formatierten Datei mit BUILD # wird nur der Datei-Kennsatz angelegt. Das Anlegen der Formatbeschreibungsliste im Kennsatz erfolgt, indem Satz 0 mit der WRITE #- oder MAT WRITE #-Anweisung direkt (<NAusdr2> = 0) geschrieben wird. Alle Datensätze der Datei haben das hiermit für Satz 0 festgelegte Format.

Werden Felder in Satz 0 mit MAT WRITE # geschrieben, können diese Felder in den anderen Sätzen auch nur mit MAT READ # und MAT WRITE # bearbeitet werden.

• Beispiele:

- 10 DIM A\$(19),B\$(10) /* A und B implizit 2% dimensioniert

:

150 BUILD #2,"TEST"

160 WRITE #2,0;A\$,A,B,B\$;

:

240 CLOSE #2

Anlegen einer formatierten Datei mit dem Namen TEST und folgenden Parametern auf Kanal 2:

- LU-Nummer : 0

- Satzstruktur: Feld 0 : String, 19 Byte+GZ

Feld 1 : Numerisch, 2 Worte

Feld 2 : Numerisch, 2 Worte

Feld 3 : String, 10 Byte+GZ+Füllbyte

- 10 DIM 3%,A,B,C\$(50)

20 BUILD #2,"FORDAT"

30 WRITE #2, 0;A,B;

40 MAT WRITE #2,0,2;C\$;

:

140 CLOSE #2

Erstellen einer formatierten Datei auf Kanal 2 mit folgender Datensatzstruktur:

Feld 0 : Numerisch, 3 Worte

Feld 1 : Numerisch, 3 Worte

Feld 2 : Matrix-Feld, 25 Worte (MAT WRITE # setzt kein Grenzzeichen - also auch kein Füllbyte erforderlich)

Auch in allen weiteren Sätzen dieser Datei können die bei den ersten Felder nur mit WRITE # und READ # und das dritte Feld mit MAT WRITE # und MAT READ # bearbeitet werden.

Dateibearbeitung	Dateibearbeitung

11.2.2 Zugriff auf formatierte Dateien

Bevor auf Datensätze in einer Datei zugegriffen werden kann, muß die Datei mit der Anweisung

OPEN #

auf einem beliebigen freien Kanal eröffnet werden.

Der Zugriff auf Datensätze in formatierten Dateien erfolgt mit den Anweisungen:

- READ #
- WRITE #
- PRINT #
- MAT READ #
- MAT WRITE #

Die Anweisungen MAT READ # und MAT WRITE # können jedoch nur für Felder benutzt werden, die in Satz 0 mit MAT WRITE # geschrieben wurde, also im Satzformat entsprechend definiert wurden. In diesem Fall können diese Felder nur mit MAT READ # und MAT WRITE # bearbeitet werden. Mit der Anweisung PRINT # können in formatierten Dateien nur Strings bearbeitet werden.

Alle Anweisungen ermöglichen sowohl sequentiellen als auch direkten Zugriff auf Datensätze und Datenfelder.

- Sequentieller Zugriff

Sequentieller Zugriff auf Datensätze erfolgt, wenn in der Anweisung keine relative Satznummer (<NAusdr2>) angegeben ist oder wenn als Satznummer -1 vorgegeben ist. Soll sequentiell auf Datensätze zugegriffen, aber nur bestimmte Felder der Datensätze übertragen werden, muß <NAusdr2> = -1 sein. Das Dateiende kann durch Auftreten von ERROR #52 erkannt werden.

- * Fehlende Sätze innerhalb der Datei werden ebenfalls
- * durch Error #52 angezeigt.

Dateibearbeitung	Dateibearbeitung
------------------	------------------

• Beispiele:

- Sequentielles Lesen und Übertragen der Datenfelder ab Feld 0 in die Variablen A, B, C und X\$.

```
1510 READ #C;A,B,C,X$;
```

- Sequentielles Schreiben und Übertragen der Variablen D, E, F und Y\$ in die Datensätze ab Feld 3.

```
1520 WRITE #C,-1,3;D,E,F,Y$;
```

- Zugriff auf den zuletzt bearbeiteten Satz

Ist als relative Satznummer der Wert -2 vorgegeben, wird auf den Satz zugegriffen, der zuletzt gelesen oder geschrieben wurde. Wird direkt nach der Dateieröffnung -2 vorgegeben, führt das zu BASIC-Fehler 51. Die Möglichkeit des Zugriffs auf den zuletzt gelesenen/geschriebenen Datensatz erleichtert das Verändern des vorhandenen Inhaltes von Datensätzen (updating).

• Beispiel:

Sequentielles Lesen in einer Datei, Verändern und Zurückschreiben der geänderten Datensätze.

```
150 READ #3;A,B,C /* SEQUENTIELL LESEN
160 GOSUB 2000 /* UPDATEN DER VARIABLEN
170 WRITE #3,-2;A,B,C; /* ZURÜCKSCHREIBEN
```

Nach dem Lesen ist der Datensatz vor dem Zugriff anderer Teilnehmer geschützt und wird nach dem Zurückschreiben wieder freigegeben.

- Direkter Zugriff

Ist als relative Satznummer ein Wert > 0 vorgegeben, wird auf diesen Satz direkt zugegriffen.

Es besteht jederzeit die Möglichkeit, die Zugriffsart zu wechseln.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung ist ohne schriftliche Genehmigung des Nixdorf-Verlages zulässig. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmusterehrtragung vorbehalten.“

Dateibearbeitung

Dateibearbeitung

● Beispiel:

Direktes Lesen von Satz 100, anschließend sequentiell weiterarbeiten.

```
150 READ #3,100;A$ /* RANDOM LESEN
160 GOSUB 2000 /* VERARBEITUNG
170 READ #3;A$ /* SEQUENTIELL LESEN
```

- Feldverarbeitung

Alle Datensätze einer formatierten Datei haben hinsichtlich Anordnung und Art der einzelnen Datenfelder die gleiche Struktur. Bei jedem Zugriff wird geprüft, ob die Variablentypen in der jeweiligen Anweisung identisch mit dem zugehörigen Eintrag im Dateikennsatz sind.

Besteht keine Übereinstimmung, wird BASIC-Fehler 54 gemeldet. Daten werden im Fehlerfall nicht übertragen.

Der gezielte Zugriff auf bestimmte Felder eines Datensatzes erfolgt durch Vorgabe der Feldnummer relativ zum Satzanfang (Feld 0).

Die relative Feldnummer wird unter <NAusdr3> angegeben; Bedingung ist, daß <NAusdr2> codiert ist.

Die Übertragung der Daten erfolgt feldweise.

Numerische Variablen werden logisch konvertiert, d.h. die Werte werden wie bei der LET-Anweisung konvertiert (z.B. können 1%-Variable in Felder für 2%-Variable übertragen werden und umgekehrt).

Bei Ausnutzung dieser Möglichkeit kann es zum BASIC-Fehler 15 kommen, wenn bei Übertragung in eine 1%-Variable ein Überlauf entsteht.

- Bei Strings wird die Übertragung beendet durch:

- Auftreten eines Grenzzeichens im Quell-String bei READ #, WRITE # und PRINT #.
- Erreichen von String-Ende im Quell- oder Ziel-String bei READ #, WRITE # und PRINT #.

Mit der Anweisung PRINT # darf kein String, der länger als sein Zielfeld ist, in eine formatierte Datei übertragen werden.

Die Anweisungen MAT READ # bzw. MAT WRITE # bearbeiten Variablen in ihrer Gesamtheit. Grenzzeichen beenden die Übertragung nicht, sondern nur das Erreichen von String-Ende im Quell- oder Ziel-String.

Dateibearbeitung

Es ist möglich eine formatierte Datei ohne Prüfung der Feldformate zu bearbeiten. (<NAusdr3>=-1 bei den Anweisungen MAT READ # und MAT WRITE #) Die Übertragung beginnt dann mit dem 1. Byte des adressierten Satzes und endet beim letzten Byte im Satz bzw. beim Erreichen der dimensionierten Länge der Ziel-Variablen.

- Dateiendebehandlung

Wird in einer Lese-Anweisung (READ # oder MAT READ #) ein Datensatz adressiert, der noch nicht geschrieben wurde, wird BASIC-Fehler 52 gemeldet.

Anmerkung:

Bei formatierten Dateien ist die Bearbeitung mehrerer Datensätze mit einer Anweisung nicht möglich.

Die Funktion CHF ermöglicht dem Programmierer, die aktuelle Größe der Datei oder die augenblickliche Zugriffsposition innerhalb der Datei zu ermitteln.

- Das Schließen der Datei

Nach der Bearbeitung der Datei ist der Kanal, auf dem diese Datei eröffnet wurde, mit der Anweisung

CLOSE #

wieder zu schließen.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patentierung oder Gebrauchsmustereintragung vorbehalten.

--	--

11.3 Relative Dateien

Eine relative Datei besteht aus dem Datei-Kennsatzblock (Header) und der entsprechenden Anzahl Datenblöcken, die erforderlich sind, um die bei der Erstellung festgelegte Anzahl von Datensätzen aufzunehmen.

Eine relative Datei belegt immer einen zusammenhängenden Bereich auf der Platte. Die Größe dieses Bereiches wird durch die Satzanzahl und die Satzlänge bestimmt.

Die maximale Anzahl von Datensätzen ist bei relativen Dateien auf 999 999 begrenzt.

Minimale Satzlänge: 1 Wort

Maximale Satzlänge: 64 K Worte

Relative Dateien ermöglichen sowohl sequentiellen als auch direkten Zugriff auf die Datensätze.

11.3.1 Erstellen relativer Dateien

Da relative Dateien einen zusammenhängenden Bereich auf einer Magnetplatte belegen, muß der gesamte Bereich zur Aufnahme der Datei bereits beim Erstellen reserviert werden.

Dies erfolgt mit dem entsprechenden Programm im EXPERT-Selektor oder in der BUILD #-Anweisung durch Angabe der Anzahl der Datensätze und der Satzlänge. Die Angaben

- Anzahl Datensätze und
- Satzlänge in Worten

werden durch ":" oder "?" voneinander getrennt in Klammern gesetzt und dem Dateinamen vorangestellt.

● Beispiel:

Anlegen einer relativen Datei mit dem Namen TEST auf Kanal 2:

- LU-Nummer : 1
- Anzahl Sätze : 1500
- Satzlänge : 50 Worte

50 BUILD #2,"(1500:50)1/TEST"

Dateibearbeitung

Beim Berechnen der Satzlänge ist zu beachten, daß String-Variablen (dimensionierte Länge + 1) Bytes belegen und bündig an eine eventuell vorhergehende Variable anschließen.

Numerische Variable beginnen immer in einem neuen Wort. Das kann, wenn String-Variable und numerische Variable aufeinander folgen, dazu führen, daß 1 Byte ungenutzt bleibt.

11.3.2 Zugriff auf relative Dateien

Bevor auf Datensätze in einer Datei zugegriffen werden kann, muß die Datei mit der Anweisung

OPEN #

auf einem beliebigen freien Kanal eröffnet werden.

Der Zugriff auf Datensätze in relativen Dateien erfolgt mit den Anweisungen:

- READ #
- WRITE #
- PRINT #
- MAT READ #
- MAT WRITE #

Alle Anweisungen ermöglichen sowohl sequentiellen als auch direkten Zugriff auf Datensätze.

- Sequentieller Zugriff

Sequentieller Zugriff auf einen Datensatz erfolgt, wenn in der Anweisung die relative Satznummer (<NAusdr2>) nicht angegeben ist oder als relative Satznummer -1 vorgegeben ist.

Soll sequentiell auf Datensätze zugegriffen werden, die Übertragung aber nicht mit Byte 0 beginnen, muß <NAusdr2> codiert und = -1 sein.

Dateibearbeitung	Dateibearbeitung
------------------	------------------

● Beispiel:

- Sequentielles Lesen und Übertragen ab Byte 5 in die Variable A\$. Der gelesene Datensatz wird gesperrt.

```
150 READ #3,-1,5;A$
```

- Sequentielles Schreiben der Variablen A, B und A\$ ab Byte 0. Die Daten werden als String auf Magnetplatte abgestellt, die numerischen Variablen werden mit "USING" über eine Maske aufbereitet.

```
150 PRINT #3;USING "#####";A,B,A$;
```

- Zugriff auf den zuletzt bearbeiteten Satz

Ist als relative Satznummer der Wert -2 vorgegeben, wird auf den Satz zugegriffen, der zuletzt gelesen oder geschrieben wurde.

Wird direkt nach der Dateieröffnung -2 vorgegeben, führt das zu dem BASIC-Fehler 51.

Die Möglichkeit des Zugriffs auf den zuletzt bearbeiteten Datensatz erleichtert das Verändern des Inhaltes von Datensätzen (updating).

● Beispiel:

Direktes Lesen eines Satzes, dessen relative Satznummer in der Variablen R vorgegeben ist. Der Inhalt des Satzes wird verändert und anschließend zurückgeschrieben.

```
150 READ #3,R;A,B,C /* DIREKT LESEN
160 GOSUB 2000 /* SATZINHALT VERÄNDERN
170 WRITE #3,-2;A,B,C; /* ZURÜCKSCHREIBEN
```

Dateibearbeitung

Nach dem Lesen ist der Datensatz vor dem Zugriff anderer Teilnehmer geschützt und wird erst nach dem Zurückschreiben wieder freigegeben.

- Direkter Zugriff

Ist als relative Satznummer ein Wert ≥ 0 vorgegeben, wird auf den adressierten Satz direkt zugegriffen. Es besteht jederzeit die Möglichkeit, die Zugriffsart zu wechseln.

• Beispiel:

Direktes Lesen des Satzes 120. Anschließend sequentielle Verarbeitung

```
150 READ #3,120;A$      /* DIREKT LESEN
160 GOSUB 2000
170 READ #3;A$          /* SEQUENTIELL LESEN
```

- Feldverarbeitung

Bei der Bearbeitung von Daten in relativen Dateien wird durch das IOCS keine Überprüfung der Datentypen wie bei formatierten Dateien vorgenommen.

Die Daten werden identisch übertragen. Der Programmierer muß dafür sorgen, daß Ziel- und Quellbereich den gleichen Feldtyp haben.

Durch die Möglichkeit der Vorgabe einer Byte-Adresse (Displacement) unter $\langle \text{NAusdr3} \rangle$ relativ zum Anfang des adressierten Datensatzes kann ab einem beliebigen Byte (erstes Byte: Byte 0) mit der Übertragung begonnen werden. Das mit $\langle \text{NAusdr3} \rangle$ adressierte Byte kann auch außerhalb des adressierten Datensatzes liegen.

Besteht ein Datensatz aus verschiedenen Feldtypen (Strings und numerische Variablen), muß bei der Ermittlung des Displacements berücksichtigt werden, daß:

- numerische Variable immer in einem neuen Wort (Wortgrenze) beginnen.
- Strings immer mit einem Grenzzeichen abgeschlossen werden (Ausnahmen siehe Besonderheiten).

© Mitgabgabe sowie Vervielfältigung dieser Unterlagen, Verwertung und
 Weiterverbreitung ist ohne schriftliche Genehmigung des Herstellers
 Mittelsurkunden verpflichten zu Schadensersatz. Alle Rechte für den Fall
 der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

	Dateibearbeitung	Dateibearbeitung
--	------------------	------------------

Besonderheiten:

- Die Anweisung WRITE # stellt Grenzzeichen ab:
 - nach Übertragung eines Literals,
 - nach Übertragung eines Strings ohne Indizes
 - nach Übertragung eines Strings mit Indizes in der ganzen dimensionierten Länge
 - nach Teilstrings, falls ein Grenzzeichen vor Teilstringende auftritt.
- Die Anweisung READ # erwartet beim Lesen eines Strings hinter dem letzten übertragenen Zeichen immer ein Grenzzeichen, d.h. das dem letzten übertragenen Zeichen eines Strings folgende Zeichen wird übersprungen.

Es kann also zu Problemen führen, wenn mit WRITE # Teilstrings geschrieben werden, da die entsprechende Anweisung READ # diese Daten nicht korrekt übertragen kann.

• Beispiel:

Die mit der Anweisung
50 WRITE #3;A\$(5,9),B\$;
geschriebenen Daten können mit der Anweisung
60 READ #3;A\$(5,9),B\$;
nicht korrekt gelesen werden.

Begründung:
Dimensionen : A\$(9),B\$(10)
Inhalt : A\$ = "ABCDEFGH I"
 B\$ = "1234567890"

Nach der Durchführung der Anweisung
50 WRITE #3;A\$(5,9),B\$;
hat der Datensatz den Inhalt:
EFGHI1234567890

Es wurde kein Grenzzeichen abgestellt. Nach Durchführung der Anweisungen

55 CLOSE # 3
57 OPEN # 3, "Datei"
60 READ # 3;A\$(5,9),B\$
haben die Zielvariablen den Inhalt:

A\$(5,9) = "EFGHI"
B\$ = "234567890"

Dateibearbeitung

Die "1" wurde nicht übertragen, da READ #, nachdem das Ende von A\$ erreicht war, ein Grenzzeichen im Datensatz erwartet hat und es übersprungen hat. Der Inhalt dieses Datensatzes kann nur dann korrekt übertragen werden, wenn mit einer Anweisung READ # oder MAT READ # in eine String-Variable gelesen wird, die groß genug ist, sämtliche Daten aufzunehmen.

- Dateiendebehandlung:

Wird ein Datensatz adressiert, der hinter dem angelegten Dateiende liegt, wird BASIC-Fehler 51 gemeldet.

Anmerkung:

Bei relativen Dateien ist die Bearbeitung mehrerer Datensätze mit einer Anweisung möglich.

● Beispiel:

Ein Datensatz besteht aus einem String mit einer Länge von 80 Byte. Es sollen 5 Datensätze mit einer Anweisung gelesen werden.

```
10 DIM A$(80),B$(80),C$(80),D$(80),E$(80)
:
50 READ #3,R;A$,B$,C$,D$,E$;
```

oder:

```
10 DIM A$(400)
:
50 READ #3,R;A$(1,80),A$(81,160),A$(161,240)...
```

oder:

```
10 DIM A$(400)
:
50 MAT READ #3,R;A$
```

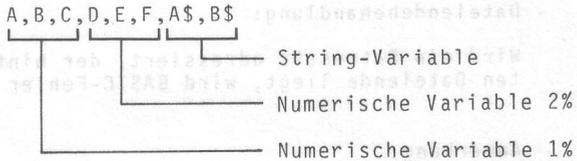
Es ist darauf zu achten, daß auch in solchen Fällen der Record-Pointer nur um 1 versetzt wird. Nach Durchführung einer dieser Anweisungen hätte der Record-Pointer den Inhalt R+1.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung des Nixdorf Computer AG. Die Nixdorf Computer AG ist an der Weitergabe und Verbreitung dieser Unterlage, Verwertung und Verbreitung, auch auszugsweise, nicht interessiert.“
Zusammenfassend verpflichtet zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

	Dateibearbeitung	Dateibearbeitung
--	------------------	------------------

- Regeln für das Arbeiten mit relativen Dateien
- Datensätze, die aus verschiedenen Feldtypen bestehen, sollten so aufgebaut sein, daß die Variablen eines Typs hintereinander stehen.

• Beispiel:



Dadurch besteht die Möglichkeit, daß Kapazität eingespart wird. Zusätzlich wird die Berechnung der Satzlänge und Displacements erleichtert.

- Nach Möglichkeit keine Sub-Strings verarbeiten.

Die Funktion CHF ermöglicht dem Programmierer, die aktuelle Größe der Datei oder die augenblickliche Zugriffspostion innerhalb der Datei zu ermitteln.

- Das Schließen der Datei

Nach der Bearbeitung der Datei ist der Kanal, auf dem diese Datei eröffnet wurde, mit der Anweisung

CLOSE #

wieder zu schließen.

	Dateibearbeitung	
--	------------------	--

11.3.3 Der EOF-Zeiger (End Of File)

Dieser Zeiger enthält die Satznummer des höchsten geschriebenen Satzes +1, wenn er zuvor mit OPEN #X;"C",... initialisiert worden ist. In einem BASIC-Programm kann mit der Funktion CHF (Kanalnummer + 400) darauf zugegriffen werden.

Wert	Bedeutung
-1	Datei ist leer (nach OPEN #X;"C",...)
0	Datei wurde noch nie mit OPEN #X;"C",... eröffnet
>0	Aktueller EOF-Zeiger

OPEN #X;"C",... initialisiert den EOF-Zeiger und setzt ihn auf -1.

BUILD # setzt den EOF-Zeiger auf 0. Ein Aktivieren des EOF-Zeigers ist nur durch Absetzen eines CLOSE und anschließendem OPEN mit "C" möglich.

WRITE aktualisiert EOF im Dateikennsatz, wenn EOF > 0 und die aktuelle Satznummer > EOF ist.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patentierung oder Gebrauchsmustereintragung vorbehalten.

	Dateibearbeitung	Dateibearbeitung
--	------------------	------------------

11.4 Indexdateien

Eine Indexdatei besteht aus dem Datei-Kennsatz (Header), 1 bis 15 Schlüsselverzeichnissen und dem Bereich zur Aufnahme der Datensätze. Jedes Schlüsselverzeichnis ist aufgeteilt in in Master-, Coarse- und Fine-Blöcke, um ein schnelleres Auffinden zu ermöglichen.

Eine Indexdatei belegt immer einen zusammenhängenden Bereich auf einer Magnetplatte. Die Größe des Bereiches wird durch die vorgegebene Anzahl Datensätze, die Satzlänge und die Länge der Ordnungsbegriffe bestimmt.

Berechnung des Plattenbedarfes:

A Anzahl Plattenblöcke für die Datensätze:
 $\text{Anzahl Sätze} * \text{Satzlänge} / 256$

B Anzahl Plattenblöcke für den Fine-Bereich:
 $\text{Anzahl Sätze} * 2 / \text{INT}(254/(\text{Schlüssellänge}+1))$

C Anzahl Plattenblöcke für den Coarse-Bereich:
 $\text{Anzahl Fine-Blöcke} / \text{INT}(254/(\text{Schlüssellänge}+1))$

D Anzahl Plattenblöcke für den Masterbereich:
 $\text{Anzahl Coarse-Blöcke} / \text{INT}(254/(\text{Schlüssellänge}+1))$

(Satz- und Schlüssellänge in Worten, 1 Block 256 Worte)

Die Gesamtanzahl an Plattenblöcken für die Datei (ohne Header) ergibt sich nun aus der Summe von A und für jedes Verzeichnis B, C und D, wobei die Einzelergebnisse jeweils aufzurunden sind.

Ist allerdings:

$$\text{Gesamtanzahl Blöcke} * 256 / \text{Satzlänge} > 65533,$$

muß die Berechnung mit

$$\text{Anzahl Sätze} * 2 / \text{INT}(254/(\text{Schlüssellänge}+2)),$$

eingesetzt in B, wiederholt werden, da dann die Satzzeit zwei Worte einnehmen.

Minimale Satzlänge: 1 Wort
Maximale Satzlänge: 64 K Worte
Minimale Schlüssellänge: 1 Wort
Maximale Schlüssellänge: 15 Worte

Dateibearbeitung	Dateibearbeitung

11.4.1 Erstellen von Indexdateien

Beim Erstellen einer Indexdatei per BASIC-Programm müssen nacheinander

- der Dateibereich angelegt,
- jedes Schlüsselverzeichnis einzeln definiert,
- alle Schlüsselverzeichnisse gemeinsam initialisiert werden und
- die (freien) Datensätze verkettet werden.

Der Dateibereich wird angelegt mit der Anweisung

BUILD #

wie bei einer relativen Datei durch Voranstellen der Anzahl Sätze und Satzlänge in Worten. Diese beiden Angaben müssen in einer Klammer stehen und durch ":" oder "?" voneinander getrennt sein. Die Anzahl Sätze setzt sich zusammen aus der Anzahl Datensätze und der Anzahl Sätze (gleicher Länge), die den Indexbereich belegen könnten.

Die Definition und Initialisierung der Schlüsselverzeichnisse sowie die Verkettung der Datensätze erfolgt mit der Anweisung **SEARCH #**:

SEARCH #<NAusdr1>,<NAusdr2>,<NAusdr3>;<SVar>,<NVar1>,<NVar2>

<NAusdr1>: Kanalnummer.

<NAusdr2>: Modus: 0 (Definieren und Initialisieren),
6 (Verkettung der Datensätze)

<NAusdr3>: Verzeichnisnummer (ohne Bedeutg. bei Modus 6):

 #0: Definieren des Verzeichnisses

 =0: Initialisieren aller definierten Verzeichnisse

<SVar> : Ohne Bedeutung.

<NVar1> : Beim Definieren eines Verzeichnisses ist hier die Schlüssellänge, beim Initialisieren der Verzeichnisse die Anzahl Datensätze zu übergeben (ohne Bedeutung bei Modus 6).

<NVar2> : Statusvariable.

Das Definieren der Verzeichnisse muß bei 1 beginnend lückenlos aufsteigend erfolgen.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.“

Dateibearbeitung

Bei der Festlegung der Länge der Datensätze sind unbedingt die Berechnungsregeln, wie sie unter "Erstellung relativer Dateien" aufgeführt sind, einzuhalten, damit die Datensatzlänge in jedem Fall ausreicht. Wird bei Indexdateien versucht, mit einer WRITE # -, MAT WRITE # - oder PRINT # - Anweisung über das Datensatzende hinaus zu schreiben, wird BASIC-Fehler 82 gemeldet.

• Beispiel zum Erstellen einer Indexdatei:

Erstellen der Indexdatei AUF mit 2000 Datensätzen von je 30 Worten Länge und zwei Schlüsselverzeichnissen mit einer Schlüssellänge von je 6 Worten.

Anzahl Blöcke für einen Fine-Bereich:
 $2000 * 2 / \text{INT}(254/(6+1)) = 112$ (aufgerundet)

Anzahl Blöcke für einen Coarse-Bereich:
 $112 / \text{INT}(254/(6+1)) = 4$ (aufgerundet)

Anzahl Blöcke für einen Master-Bereich:
 $4 / \text{INT}(254/(6+1)) = 1$ (aufgerundet)

Zusammen benötigt der Indexbereich also

$2 * 112 = 224$ Blöcke, das entspricht $224 * 256$ Worten und damit 1997 Sätzen zu 30 Worten

Die Datei muß also mit 3997 Sätzen von je 30 Worten Länge erstellt werden:

```

10 DIM V$(12)
20 IF ERR 0 GOSUB ... /* FEHLERBEHANDLUNG
30 BUILD #3, "(3997:30)AUF" /* DATEIBEREICH
40 LET N=6
50 FOR V=1 TO 2
60 SEARCH #3,0,V;V$,N,S /* VERZEICHNIS DEFINIEREN
70 IF S GOTO ... /* STATUS ≠ 0
80 NEXT V
90 LET N=2000
100 SEARCH #3,0,V;V$,N,S /* VERZEICHNISSE INIT.
110 IF S GOTO ... /* STATUS ≠ 0
120 SEARCH #3,6,0;V$,N,S /* DATENSÄTZE VERKETTEN
130 IF S GOTO ... /* STATUS ≠ 0
:

```

Dateibearbeitung

11.4.2 Zugriff auf Indexdateien

Bevor auf Datensätze in einer Datei zugegriffen werden kann, muß die Datei mit der Anweisung

OPEN #

auf einem beliebigen freien Kanal eröffnet werden.

Indexdateien belegen wie relative Dateien einen zusammenhängenden Bereich auf der Magnetplatte. Der reservierte Bereich unterteilt sich in:

- 1 bis 15 Schlüsselverzeichnisse zur Aufnahme der Ordnungsbegriffe (Indexbereich) und
- den Bereich zur Aufnahme der Datensätze (Datenbereich).

Indexdateien ermöglichen sowohl sequentiellen als auch direkten Zugriff auf die Datensätze. Dabei bestehen die folgenden Möglichkeiten:

- Sequentielles Lesen des Datenbereiches.
- Sequentielles Lesen des Indexbereiches und direkter Zugriff auf die Datensätze.
- Direkter Zugriff im Indexbereich durch Vorgabe eines Ordnungsbegriffs (= Schlüssel) und direkter Zugriff auf den zugehörigen Datensatz.

Zugriffe auf Indexdateien erfolgen in der Regel mit mehreren BASIC-Anweisungen.

Der Zugriff auf Datensätze in Indexdateien erfolgt mit den Anweisungen:

- READ #
- WRITE #
- MAT READ #
- MAT WRITE #
- PRINT #

Für den Zugriff auf Datensätze gelten die gleichen Regeln wie bei relativen Dateien. Allerdings ist darauf zu achten, daß der erste Datensatz im Datenbereich nicht Satz 0 ist. Die Nummer des ersten Datensatzes relativ zum Dateianfang kann mit der Anweisung SEARCH # -Modus 1- ermittelt werden.

Die Verwendung des Semikolons in einer PRINT #-Anweisung hinter auszugebenden Zeichen führt zu Fehlern. In Indexdateien sollte stattdessen mit WRITE # geschrieben werden.

Dateibearbeitung	Dateibearbeitung
------------------	------------------

Der Zugriff auf Datensätze erfolgt in der Regel durch die Vorgabe eines Ordnungsbegriffs über den Indexbereich. Ein Datensatz kann in mehreren Ordnungsbegriffverzeichnissen eingetragen sein. Zur Bearbeitung der Indexbereiche steht die Anweisung:

SEARCH #

zur Verfügung.

11.4.3 Informationen über die Indexdatei

Die Funktion CHF sowie der SEARCH-Modus 1 ermöglichen es, verschiedene Informationen über die Indexdatei einzuholen.

Mit der Funktion CHF kann die größte Datensatznummer und die aktuelle Zugriffsposition im Datenbereich ermittelt werden.

SEARCH-Modus 1

Mit SEARCH-Modus 1 können die Längen der Ordnungsbegriffe in den Verzeichnissen, die Satznummer des ersten Datensatzes und die Anzahl freier Datensätze abgefragt werden.

Anweisung	Funktion
SEARCH #C,1,V;V\$,N,S	Die Schlüssellänge von Verzeichnis V wird in N übergeben (V\$ und S beliebig).
SEARCH #C,1,0;V\$,N,S mit S=0	Die Nummer des ersten Datensatzes wird in N übergeben (V\$ beliebig).
SEARCH #C,1,0;V\$,N,S mit S=1	Die Anzahl freier Datensätze wird in N übergeben (V\$ beliebig).

Dateibearbeitung	Dateibearbeitung

11.4.4 Datenbereich sequentiell lesen

Um den Datenbereich sequentiell zu lesen, muß zuerst die Nummer des ersten Datensatzes ermittelt und mit ihr der erste Satz gelesen werden:

```
LET S=0
SEARCH #C,1,0;V$,N,S
IF S GOSUB ...
READ #C,N;...
```

(C = Kanalnummer, V\$ beliebig)
Alle weiteren Datensätze können dann mit

```
READ #C,-1;...
```

gelesen werden.

11.4.5 Datenbereich in Verzeichnisordnung lesen

Diese Reihenfolge wird durch den SEARCH-Modus 3 ermöglicht. Es wird ein Ordnungsbegriff vorgegeben, der kleiner ist als jeder tatsächliche Ordnungsbegriff:

```
LET V$=""
SEARCH #C,3,V;V$,N,S
IF S GOSUB ...
READ #C,N;...
```

(C = Kanalnummer, V = Verzeichnisnummer, N und S bel.)

Nach Verarbeitung des Satzes kann ohne weitere Aufbereitungen auf die SEARCH-Anweisung zurückgesprungen werden, da in V\$ jeweils der nächstgrößere Ordnungsbegriff übergeben wird.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und
 Verbreitung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zu
 gestattet. Nachdruck, Verbreitung, Vervielfältigung, Verbreitung, auch
 auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG.
 Zur Patentierung oder Gebrauchsmustereintragung vorbehalten.

	Dateibearbeitung	
--	------------------	--

11.4.6 Wahlfreier Zugriff über einen Ordnungsbegriff

Das Lesen eines Satzes nach Vorgabe eines Ordnungsbegriffs ermöglicht SEARCH-Modus 2.
In V\$ muß der Ordnungsbegriff vorgegeben werden.

```
LET V$="Ordnungsbegriff" oder INPUT V$
SEARCH #C,2,V;V$,N,S
IF S GOSUB ...
READ #C,N;...
```

(V = Verzeichnisnummer, N und S beliebig)

11.4.7 Datensatz mit Ordnungsbegriffen einfügen

Zunächst muß der erste freie Datensatz gesucht werden:

```
LET S=2
SEARCH #C,1,0;V$,N,S
IF S GOSUB ...
```

(C = Kanalnummer, V\$ und N beliebig)

Die Satznummer wird in N übergeben und aus der Liste der freien Datensätze gelöscht.

Nun kann der Datensatz geschrieben werden:

```
WRITE #C,N;...
```

Schließlich sind noch alle auf den Datensatz verweisenden Ordnungsbegriffe mit SEARCH-Modus 4 in die Verzeichnisse einzutragen:

```
SEARCH #C,4,V;V$,N,S
IF S GOSUB ...
```

(V = Verzeichnisnummer, V\$ = Ordnungsbegriff, S beliebig)

	Dateibearbeitung	
--	------------------	--

11.4.8 Datensatz mit Ordnungsbegriffen löschen

Zunächst sind mit SEARCH-Modus 5 sämtliche auf den Datensatz verweisenden Ordnungsbegriffe aus den Verzeichnissen zu löschen:

```
SEARCH #C,5,V;V$,N,S
IF S GOSUB ...
```

(C = Kanalnummer, V = Verzeichnisnummer, V\$ = Ordnungsbegriff, N = Nummer des zu löschenden Satzes, S beliebig)

Anschließend ist der Datensatz selbst freizugeben:

```
LET S=3
SEARCH #C,1,0;V$,N,S
IF S GOSUB ...
```

(V\$ beliebig, N = Nummer des zu löschenden Satzes)

11.4.9 Indexdatei löschen

Zum Löschen einer Indexdatei werden zwei Möglichkeiten geboten:

- Physikalisches Löschen der Datei durch KILL:

```
CLOSE #C (falls Datei eröffnet)
KILL <DatName> (<DatName> = Name der Indexdatei)
```

- Löschen der Verzeichnisse und Freigeben aller Datensätze unter Beibehaltung der Dateistruktur mit SEARCH-Modus 6:

```
SEARCH #C,6,0;V$,N,S
IF S GOSUB ...
```

(C = Kanalnummer, V\$, N und S beliebig)

© Weitergabe sowie Vervielfältigung dieser Unterlage, Vervielfältigung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Dateibearbeitung

11.4.10 Statusmeldungen

Nach Durchführung einer jeden SEARCH #-Anweisung ist die Variable S unbedingt abzuprüfen, da in ihr Hinweise über den Verlauf der SEARCH-Operation übergeben werden. Folgende Statusmeldungen können nach Durchführung der verschiedenen Funktionen von SEARCH # in S abgestellt sein:

S	Bedeutung
0	Funktion konnte fehlerfrei durchgeführt werden.
1	Funktion konnte nicht erfolgreich durchgeführt werden.
2	Ende des Verzeichnisses erreicht.
3	Kein freier Datensatz vorhanden.
4	Keine Indexdatei.
5	Satznummer negativ, Modus > 7, S > 3, Verzeichnisnummer zu groß, Schlüsselfeldlänge zu klein oder anderer schwerer Fehler. Die Datei ist möglicherweise zerstört.
6	Unzulässige Reihenfolge bei Angabe der Verzeichnisse (nur Modus 0).
7	Datei ist keine relative Datei (nur Modus 0).
8	Verzeichnisse initialisieren wurde bereits ausgeführt.
9	Kein Verzeichnis angegeben.
10	Zu viele Verzeichnisse (max. 15 pro Datei).
12	Datei ist zu klein, um alle Verzeichnisse aufzunehmen.
15	Datei von anderem Teilnehmer eröffnet.

11.4.11 Das Schließen der Datei

Nach der Bearbeitung der Datei ist der Kanal, auf dem diese Datei eröffnet wurde, mit der Anweisung

CLOSE #

wieder zu schließen.

Dateibearbeitung

11.4.12 Probleme bei der Bearbeitung von Indexdateien

Die gleichzeitige Bearbeitung einer Indexdatei kann zu Problemen führen, weil

- Zugriffe auf Indexdateien in der Regel mit mehreren Anweisungen erfolgen,
- für einen Zugriff mehrere Zeitscheiben benötigt werden.

Um Fehler, die bis zur Zerstörung der Datei führen können, zu vermeiden, wird die Einhaltung folgender Regeln empfohlen:

- Soll ein Datensatz mit Ordnungsbegriffen in mehreren Verzeichnissen eingefügt werden, ist zuerst der Datensatz zu schreiben. Danach sind alle auf den Datensatz verweisenden Ordnungsbegriffe in die Verzeichnisse einzutragen.
- Ist ein Datensatz mit Ordnungsbegriffen in mehreren Verzeichnissen verkettet, sind zuerst alle auf den Datensatz verweisenden Ordnungsbegriffe aus den Verzeichnissen zu löschen. Anschließend kann der Datensatz selbst freigegeben werden.
- Nach Durchführung einer SEARCH #-Anweisung grundsätzlich den Status (<NVar2>) abfragen.
- Der Ordnungsbegriff sollte auch im Datensatz vorhanden sein. Er kann gegen den vorgegebenen Ordnungsbegriff verglichen werden und bei Verlust des Indexbereichs zur Rekonstruktion herangezogen werden.

In den folgenden Beispielen wird erläutert, worin die Probleme bestehen und wie Fehler vermieden werden können.

Dateibearbeitung

• Beispiele:

- Programm 1 liest den Satz mit dem Ordnungsbegriff "AA" und verändert den Datensatz. Programm 2 löscht den Ordnungsbegriff "AA" und trägt den dazugehörigen Datensatz in die Liste der freien Sätze ein.

Zeitscheibe	Programm 1	Programm 2
1	LET V\$="AA" SEARCH #C,2,1;V\$,N,S	
2		LET V\$="AA" SEARCH #C,5,1;V\$,N,S
3	IF S GOTO ..(Fehler) READ #C,N;R\$	
4		IF S GOTO ..(Fehler) LET S=3 SEARCH #C,1,1;V\$,N,S
5	Satz verändern WRITE #C,N;R\$	

Bei dieser Anweisungsfolge wäre nach dem Zurückschreiben des Datensatzes durch Programm 1 die Datei zerstört. Durch Programm 2 wurde in Zeitscheibe 4 der Satz in die Liste der freien Datensätze eingetragen. Die Verkettung der freien Datensätze erfolgt in den ersten beiden oder ersten vier Bytes der freien Sätze. Da Programm 1 den Datensatz in Zeitscheibe 5 ohne Displacement, also ab Byte 0, zurückgeschrieben hat, ist die Verkettung der freien Sätze zerstört.

Dieser Fehler kann ausgeschlossen werden, wenn beim Verändern eines Datensatzes in einer Indexdatei bereits bei der SEARCH-Anweisung (hier in Zeitscheibe 1) eine Satzsperrung gesetzt wird:

```
SEARCH #C,2,1;V$,N,S:
```

Dateibearbeitung

- Programm 1 liest den Satz mit dem Ordnungsbegriff "AA" und verändert den Datensatz. Programm 2 löscht den Ordnungsbegriff "AA", verändert den Datensatz und fügt für ihn den Ordnungsbegriff "BB" ein.

Zeitscheibe	Programm 1	Programm 2
1	LET V\$="AA" SEARCH #C,2,1;V\$,N,S	
2		LET V\$="AA" SEARCH #C,5,1;V\$,N,S IF S GOTO ..(Fehler) WRITE #C,N;R\$
3	IF S GOTO ..(Fehler) READ #C,N;R\$ Satz verändern	
4		LET V\$="BB" SEARCH #C,4,1;V\$,N,S IF S GOTO ..(Fehler)
5	WRITE #C,N;R\$	

Programm 1 wollte den zu dem Ordnungsbegriff "AA" gehörenden Datensatz verändern. Da allerdings Programm 2 vor dem Lesen des Datensatzes durch Programm 1 mit dem zum Ordnungsbegriff "BB" gehörenden Datensatz überschrieben hat, erhält Programm 1 nicht die gewünschten Daten und verändert außerdem den von Programm 2 für den Ordnungsbegriff "BB" geschriebenen Datensatz.

Dieser Fehler kann vermieden werden, indem der Ordnungsbegriff auch im zugehörigen Datensatz selbst vermerkt und nach dem Lesen mit dem im Schlüsselverzeichnis gesuchten verglichen wird. Außerdem sollte wie schon in Beispiel a) bei der ersten SEARCH-Anweisung eine Satzsperrung gesetzt werden.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlung unter Strafe. Nachdruck, Verbreitung, Verbreitung, Verbreitung oder Gebrauchsmusteranmeldung vorbehalten.“

	Dateibearbeitung	Dateibearbeitung
--	------------------	------------------

11.5 Textdateien

Eine Textdatei besteht aus dem Datei-Kennsatz (Header) und der entsprechenden Anzahl Datenblöcken, die erforderlich sind, um die aktuelle Datenmenge aufzunehmen.

Belegen die Daten mehr als 128 Plattenblöcke, sind zusätzlich zum Datei-Kennsatzblock 1 bis 128 Erweiterungsblöcke vorhanden. Ein Kennsatz-Erweiterungsblock ermöglicht die Verwaltung von 256 Datenblöcken.

Eine Textdatei belegt keinen zusammenhängenden Bereich auf der Magnetplatte. Erweiterungen werden über eine Liste der freien Plattenblöcke vorgenommen.

Der Inhalt einer Textdatei besteht aus einer einzigen Zeichenkette, deren maximale Länge

16777216 Byte

beträgt. In diesem Fall werden 128 Kennsatz-Erweiterungsblöcke angelegt.

Diese Zeichenkette wird in Seiten und Zeilen aufgeteilt. Zeilen werden durch CR-Codes und Seiten durch FF-Codes voneinander getrennt.

Das Ende der Zeichenkette, zugleich auch das Ende der Textdatei, wird durch ein Grenzzeichen angezeigt.

Textdateien sind für sequentielle Verarbeitung konzipiert, erlauben aber auch in folgender Weise direkten Zugriff:

Ein direkter Zugriff auf bestimmte Zeilen oder Seiten ist nicht möglich. Zugegriffen werden kann nur auf Plattenblöcke - durch die Vorgabe der Blocknummer relativ zum Dateianfang als Satznummer in der READ/WRITE-Anweisung - und gezielt auf Daten in diesem Block - durch die Vorgabe einer Byte-Adresse relativ zum Anfang des adressierten Blocks.

Dateibearbeitung	Dateibearbeitung

11.5.1 Erstellen von Textdateien

Beim Anlegen von Textdateien mit BUILD # muß vor der Dateibezeichnung ein "+" angegeben werden. Textdateien bestehen bei der Anlage zunächst nur aus dem Datei-Kennsatzblock.

Die Angaben:

- Anzahl Sätze
- Satzlänge

sind beim Anlegen von Textdateien nicht zugelassen.

● Beispiel:

Ersetzen und gleichzeitiges Eröffnen einer Textdatei mit dem Namen TEST auf Kanal 2 und der LU-Nummer 2.

50 BUILD #2,+ "2/TEST!"

Als Besonderheiten sind zu beachten:

- Bei der Anweisung READ & wird die Übertragung durch Auftreten eines 'CR'-Codes, eines Brückensatzes oder des dimensionsierten Endes im Zielfeld beendet.
- Wird die Übertragung durch Erreichen des dimensionsierten Endes des Zielfelds beendet, so überträgt der Dateileger das auf den übertragene Text folgende Leerzeichen der Textdatei, auch wenn dieses kein Größere ist. Falls es sich bei dem übertragene Zeichen nicht um kein 'CR'-Zeichen handelt, muß durch eine geeignete READ & Anweisung der Dateileger zurückschrittweise werden (mittels CHF (200+(Kanalnr.)).).
- Die Angabe von (Kanalnr) = -2 bewirkt nicht den Zugriff auf den zuletzt bearbeiteten Satz, sondern die Fortsetzung der Übertragung bei Auftreten eines 'CR'-Codes.
- In der WRITE & Anweisung dürfen keine numerischen Variablen und keine numerischen Ausdrücke angegeben sein.
- Die Funktion der Trennzeichen " " und ";" sind bei der Anweisung PRINT & unbedingt zu beachten.
- PRINT & stellt grundsätzlich hinter dem zuletzt übertragenen Zeichen und WRITE & unter bestimmten Umständen (siehe "Beschreibung der Anweisungen") ein Größere ein, das bei separaten Schritten abdrucken wieder überschrieben wird.

© Weitergabe sowie Vervielfältigung dieser Unterlagen, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.

Dateibearbeitung

11.5.2 Zugriff auf Textdateien

Bevor auf Datensätze in einer Datei zugegriffen werden kann, muß die Datei mit der Anweisung

OPEN #

auf einem beliebigen freien Kanal eröffnet werden.

Der Zugriff auf Daten in Textdateien ist mit den Anweisungen

- READ #
- WRITE #
- PRINT #

möglich.

In Textdateien dürfen nur String-Literale und/oder der Inhalt von String-Variablen geschrieben werden (Ausnahme PRINT #). Beim Lesen aus Textdateien sind als Zielvariable nur String-Variable zugelassen.

Als Besonderheiten sind zu beachten:

- Bei der Anweisung READ # wird die Übertragung durch Auftreten eines 'CR'-Codes, eines Grenzzeichens oder des dimensionierten Endes im Zielstring beendet.
- Wird die Übertragung durch Erreichen des dimensionierten Endes des Zielstrings beendet, so überspringt der Dateizeiger das auf den übertragenen Text folgende Zeichen der Textdatei, auch wenn dieses kein Grenzzeichen ist. Falls es sich bei dem übersprungenen Zeichen ggfs. um kein 'CR'-Zeichen handelt, muß durch eine geeignete READ #-Anweisung der Dateizeiger zurückpositioniert werden (mittels CHF (200+<Kanalnr.>)).
- Die Angabe von <NAusdr2> = -2 bewirkt nicht den Zugriff auf den zuletzt bearbeiteten Satz, sondern die Fortsetzung der Übertragung bei Auftreten eines 'CR'-Codes.
- In der WRITE #-Anweisung dürfen keine numerischen Variablen und keine numerischen Ausdrücke angegeben sein.
- Die Funktion der Trennzeichen "," und ";" sind bei der Anweisung PRINT # unbedingt zu beachten.
- PRINT # stellt grundsätzlich hinter dem zuletzt übertragenen Zeichen und WRITE # unter bestimmten Umständen (siehe "Beschreibung der Anweisungen") ein Grenzzeichen ab, das bei sequentiellem Schreiben allerdings wieder überschrieben wird.

Dateibearbeitung

Der Record-Pointer für Textdateien besteht aus der aktuellen Blocknummer und der Adresse des dem letzten übertragenen Zeichen folgenden Bytes relativ zum Blockanfang.

- Sequentieller Zugriff

Sequentieller Zugriff erfolgt, wenn in der Anweisung die relative Satznummer nicht angegeben oder als relative Satznummer -1 oder -2 vorgegeben ist.

Anmerkung:

Wird auf eine Textdatei sequentiell zugegriffen, muß, falls codiert, die Byte-Angabe = 0 sein.

• Beispiel:

Sequentielles Lesen und Ausgeben der gelesenen Zeilen am Bildschirm. Die Anzahl auszugebender Zeilen ist in der Variablen I vorgegeben.

```

:
150 FOR I1=0 TO I-1
160 READ #C;R$; /*TEXTDATEI LESEN
170 PRINT R$ /*BILDSCHIRMAUSGABE
180 NEXT I1

```

- Zugriff auf den zuletzt bearbeiteten Satz

Diese Möglichkeit wird vom IOCS für Textdateien nicht unterstützt. Die Realisierung ist nur durch direkten Zugriff, verwaltet durch das Anwenderprogramm, möglich.

Dateibearbeitung

- Direkter Zugriff

Da Zeilen in einer Textdatei verschieden lang sind, ist ein gezielter Zugriff auf Zeilen nicht möglich. Um trotzdem auf ein bestimmtes Byte in einer Textdatei zuzugreifen, wird für den direkten Zugriff bei Vorgabe von <NAusdr2> > 0 eine Satzlänge von 512 Byte angenommen. Die relative Satznummer ist bei Textdateien also die Nummer des Blocks (Sektors) relativ zum Dateianfang. Es besteht jederzeit die Möglichkeit, die Zugriffsart zu wechseln.

• Beispiel:

Direktes Lesen ab Byte 0 im Block 3 und anschließende sequentielle Verarbeitung.

```

:
150 READ #C,3;R$ /* DIREKT LESEN
160 GOSUB 2000
170 READ #C;R$; /* SEQUENTIELL LESEN
    
```

Anmerkung:

Aufgrund der Tatsache, daß jede Übertragung in eine Textdatei mit einem Grenzzeichen abgeschlossen wird (außer: Schreiben von Teilstrings mit WRITE #) und ein Grenzzeichen auch das Ende der Textdatei anzeigt, ist direktes Schreiben in Textdateien wenig sinnvoll. Falls in einer Textdatei ein Grenzzeichen steht, muß, um Daten hinter dem Grenzzeichen zu adressieren, direkt auf dem Byte hinter dem Grenzzeichen aufgesetzt werden.

Dateibearbeitung

● Beispiel:

```

:
9000 IF ERR 0 GOTO 9100 /*FEHLERBEHANDLUNG
9010 READ #3;R$ /*TEXTDATEI LESEN
9020 IF R$="" GOSUB 9200 /*LEERSTRING ??
9030 ....
9040 .... /*VERARBEITUNG
9050 ....
:
9100 IF SPC 8=52 PRINT "DATEIENDE"
9110 CHAIN ""
:
9200 READ #3,CHF103,CHF203+1;R$; /*AUFSETZEN HINTER
DEM GRENZ-
ZEICHEN
9210 IF R$="" GOTO 9200 /*LEERSTRING
9220 RETURN
    
```

- Feldverarbeitung

Bei Textdateien existieren keine Felder. Der Zugriff auf bestimmte Bytes innerhalb einer Textdatei ist durch die Angabe der relativen Nummer eines Bytes zum Anfang des adressierten Blocks möglich. Der maximale Wert, der vorgegeben werden darf, ist 511. Diese Angabe ist nur bei direktem Zugriff erlaubt!

● Beispiel:

Lesen aus einer Textdatei ab einer in den Variablen X und Y vorgegebenen Adresse in die Variable R\$.

```

:
150 READ #C,X,Y;R$;
    
```

Der Wert von Y darf nicht größer als 511 sein.

Dateibearbeitung

- Dateiendebehandlung

Das Dateiende in Textdateien kann festgestellt werden durch:

- Lesen eines Grenzzeichens (der Ziel-String ist "" oder kürzer als die dimensionierte Länge bei <NAusdr2> = -2).
- Auftreten des BASIC-Fehlers 52. Dies ist allerdings nur dann der Fall, wenn ein Block direkt adressiert wird, in den noch kein Zeichen geschrieben wurde.

Die Funktion CHF ermöglicht dem Programmierer, die aktuelle Größe der Datei oder die augenblickliche Zugriffsposition innerhalb der Datei zu ermitteln.

● Beispiel:

Eine Textdatei, die als Log-Datei benutzt wird, soll nicht größer als 50 Blöcke werden. Beim Erreichen von 40 Blöcken wird eine Warnung ausgegeben. Ist der 51. Block angelegt, wird ein Merker gesetzt und das aktuelle Programm bei der nächsten Schnittstelle abgebrochen, um die Log-Datei zu drucken.

```
:
9000 PRINT #3,CHF103,CHF203;M$; /*DIREKT SCHREIBEN
9010 IF CHF(103)>50 GOTO 9100 /*ABBRUCH?
9020 IF CHF(103)<40 RETURN /*WARNUNG?
9030 PRINT TAB(0,24);'LD';"LOG-DATEI DRUCKEN!
9040 SIGNAL 3,20
9050 RETURN
9100 LET M=1 /*MERKER FÜR ABBRUCH
9110 PRINT TAB(0,24);'LD';"LOG-DATEI WIRD GEDRUCKT";
9120 RETURN
```

- Das Schließen der Datei

Nach der Bearbeitung der Datei ist der Kanal, auf dem diese Datei eröffnet wurde, mit der Anweisung

CLOSE #

wieder zu schließen.

Dateibearbeitung

11.6 Sequentielle Dateien

Sequentielle Dateien beschränken sich im System 8870 auf die Datenträger

- Magnetband
- Magnetbandcassette
- Lochkarten
- Lochstreifen
- Drucker

Der Aufbau und die Programmierung der sequentiellen Dateien auf diesen Datenträgern ist jedoch teilweise voneinander abweichend, sodaß die Beschreibung der Programmierung bei der Beschreibung der jeweiligen Peripheriegeräte erfolgt.

Sequentielle Dateien erlauben, wie der Name schon sagt, lediglich sequentiellen Zugriff, d.h. eine Datei muß grundsätzlich von ihrem Beginn aus in ihrer physikalischen Reihenfolge bearbeitet werden.

	Die Drucker	Die Drucker
--	-------------	-------------

12 Die Drucker

Es können fünf Druckertypen angeschlossen werden:

- Nadeldrucker 150/210 Zeichen/Sekunde
- Nadeldrucker 100 Zeichen/Sekunde
- Kompakt-Drucker 100 Zeichen/Sekunde
- Typenrad-Drucker 45 Zeichen/Sekunde
- Zeilendrucker 18000 oder 36000 Zeilen/Stunde

Für den Programmierer bestehen je nach Druckeranschluß bestimmte Regeln und Einschränkungen.

Die grundsätzlichen Unterschiede zwischen den Druckertypen sind:

- Verschiedene Anzahl von Schreibstellen
- Der Nadeldrucker kann mit zwei Papiervorschüben und einem Formulareinzug ausgerüstet sein.
- Beim Nadel-, Kompakt-, Schnell- und Typenradrucker können erste und letzte Druckposition per Programm festgelegt werden. Nur darf die letzte Druckposition die physikalisch letzte Druckposition des Druckers nicht überschreiten.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Die Drucker

12.1 Zugriff auf Druckerdateien

Für die Drucker bildet ein logisches IOCS (Input-Output-Control-System) die Schnittstelle zur Sprache Business-BASIC. Das IOCS übernimmt die Verwaltung der Datenbestände. Alle auf dem Drucker ausgegebenen Daten sind als Dateien organisiert. Die einzig mögliche Organisationsart ist sequentiell. Die Bearbeitung einer Druckerdatei erfolgt über einen Datenkanal.

Für die Ausgabe von Daten auf einem Drucker gelten die gleichen Regeln wie für Ausgaben in Textdateien.

Ein Zeilendrucker kann nicht gleichzeitig von mehreren Teilnehmern oder von einem Teilnehmer auf mehreren Kanälen eröffnet sein. (Das gilt generell für alle Drucker mit nur einem Vorschubaggregat!)

Bei Nadeldruckern mit doppeltem Vorschub können die beiden Vorschübe und der Einzelformulareinzug von verschiedenen Teilnehmern gleichzeitig eröffnet werden, da jede Formularsteuerung als logisches unabhängiges Gerät betrieben wird. Der linke Leporello und der Formulareinzug können aus Sicherheitsgründen nur von einem Benutzer zu einer Zeit eröffnet werden. Aus mechanischen Gründen muß der Einzelformulareinzug auf der linken Druckerseite betrieben werden.

Es können z.Z. max. 2 Vorschubaggregate parallel betrieben werden; erlaubte Kombinationen sind: linker Leporello + rechter Leporello oder Formulareinzug + rechter Leporello.

Die Übergabe von Druckdaten und der Druck selbst erfolgen asynchron. Die vom Anwender übergebenen Daten werden solange aufgesammelt, bis entweder ein Auslösecode (Zeilen-vorschub) kommt oder das logische Zeilenende erreicht wird. In diesem Fall werden die Daten zum Drucker geschickt.

Die Synchronisation zwischen Zentraleinheit und dem Arbeitsplatz ist bezüglich der Druckerdaten auftragsbezogen. Werden nacheinander mehrere Felder auf die gleiche Zeile ausgegeben, so führt das zu extremer Druckverlangsamung. Deshalb ist unbedingt zeilenweise auszugeben.

Die Drucker

12.1.1 Eröffnen von Druckerdateien

Vor der Datenausgabe auf einem Drucker muß dieser auf einem beliebigen Kanal mit der Anweisung

```
OPEN # <DatName>
```

eröffnet werden.

Als <Datname> ist der Name des jeweiligen Druckerkanalprogramms (Driver) anzugeben:

- \$LPT = Nadeldrucker, linker Vorschub
- \$LPTR = Nadeldrucker, rechter Vorschub
- \$LPT1 = Zeilendrucker (Schnelldrucker)
- \$LPT1S = 2. Zeilendrucker
- \$LPT2 = Drucker (nur für USA)
- \$LPT2S = 2. Drucker (nur für USA)
- \$LPTS = 2. Nadeldrucker, linker Vorschub
- \$LPTRS = 2. Nadeldrucker, rechter Vorschub
- \$RPLX = Arbeitsplatzdrucker x, linker Vorschub
- \$RPRX = Arbeitsplatzdrucker x, rechter Vorschub
- \$RPFx = Arbeitsplatzdrucker x, Formulareinzug
- \$LCP = Kompaktdrucker
- \$LCPS = 2. Kompaktdrucker
- \$ORDP = Banddrucker
- \$ORDPS = 2. Banddrucker
- \$HDD = Systemkompaktdrucker
- \$HDDS = 2. Systemkompaktdrucker
- \$HDDSAS = Systemkompaktdrucker über SAS
- \$HDDSASS = 2. Systemkompaktdrucker über SAS
- \$ORDPSR = Banddrucker über PSR
- \$ORDPSRS = 2. Banddrucker über PSR

Alle Drucker können über den logischen Druckernamen \$<Name> betrieben werden. Über die Druckerzuordnungstabelle wird \$<Name> in Abhängigkeit vom Arbeitsplatz ein physikalischer Drucker zugeordnet, wenn ein Eintrag dafür existiert.

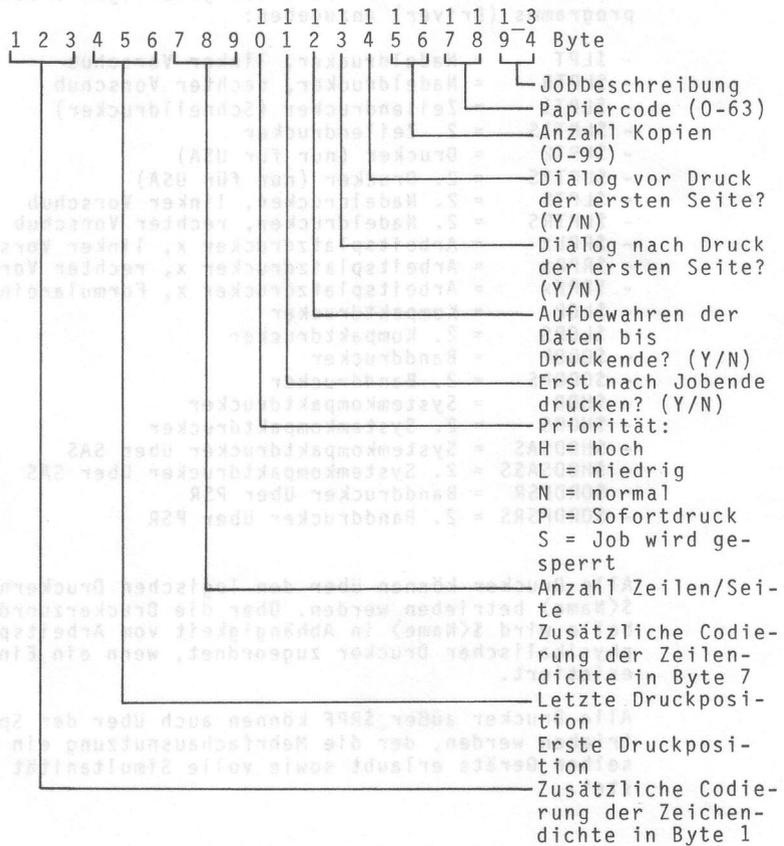
Alle Drucker außer \$RPF können auch über den Spooler betrieben werden, der die Mehrfachausnutzung ein und desselben Geräts erlaubt sowie volle Simultanität gewährleistet.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patentierung oder Gebrauchsmusterantragung vorbehalten.“

	Die Drucker	
--	-------------	--

Zusätzlich können in der OPEN #-Anweisung noch Parameter zur Formular- und Spoolersteuerung vorgegeben werden. Diese Parameter werden in einem Stringliteral oder einer Stringvariablen, die für mindestens 34 Byte dimensioniert ist, an das Druckerkanalprogramm übergeben. Es ist darauf zu achten, daß dieser String mit einem Grenzzeichen endet.

Der String hat folgenden Aufbau:



Die Angabe von "Kleinbuchstaben" für die Priorität bewirkt einen Schutz gegen Löschen.

Die Drucker

Bei Nadeldruckern mit zwei Vorschubaggregaten dürfen sich die Druckbereiche nicht überlappen.

Bei der Anzahl Zeilen/Seite sind folgende Punkte zu berücksichtigen:

- Bei Einzelformularen sind 12,6 mm Formulkopf und 26,2 mm Formularfuß von der tatsächlichen Formularhöhe abziehen.
- Beim Einzelformulareinzug beträgt die Zeilendichte 5 oder 6 Zeilen/Zoll. Beim Zeilendrucker ist die Zeilendichte nur manuell umschaltbar (8 oder 6 Zeilen/Zoll).
- Bei Nadeldruckern können je nach Typ unterschiedliche Zeilendichten realisiert werden.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patentierung oder Gebrauchsmustereintragung vorbehalten.“

Die Drucker	Die Drucker
-------------	-------------

Werden keine Angaben gemacht, gelten folgende Standardwerte:

Kanalprogramm (Driver)	Erste Druckposition	Letzte Druckposition	Formular- höhe
\$LPT	000	176	048
\$LPTR	177	177	048
\$LPT1	000	135	048
\$LPT1S	000	135	048
\$LPTS	000	176	048
\$LPTRS	177	177	048
\$RPLx	000	131	048
\$RPRx	145	177	048
\$RPFx	000	131	062
\$LCP	000	131	048
\$LCPS	000	131	048
\$ORDP	000	135/*131	048
\$ORDPS	000	135/*131	048
\$HDD	000	131/*157	048
\$HDDS	000	131/*157	048
\$HDDSAS	000	131	048
\$HDDSASS	000	131	048
\$ORDPSR	000	131/*157	048
\$ORDPSRS	000	131/*157	048

* einstellbar durch Codetabelle im Driver

Die Drucker

Übrige Parameter:

Priorität = N
 Erst nach Jobende drucken = N
 Aufbewahren der Daten bis Druckende = N
 Dialog nach Druck der ersten Seite = N
 Dialog vor Druck der ersten Seite = N
 Anzahl Kopien = 00
 Papiercode = 00

Jobbeschreibung:

Wird keine Jobbeschreibung vorgegeben, wird bei einem OPEN aus einem BASIC-Programm der Selektorname der 3. Ebene und bei einem OPEN von einem Prozessor (z.B. LIBR) der Prozessorname übernommen. Diese Angabe wird aus dem Common-Bereich gelesen; sie kann durch ein Anwenderprogramm überschrieben worden sein.

Bei logischem Druckernamen werden die Parameter nach folgenden Prioritäten eingesetzt:

Erste/letzte Druckposition	Zeilen pro Seite
1. Druckerzuordnungstabelle	1. OPEN #-Anweisung
2. OPEN #-Anweisung	2. Druckerzuordnungstabelle
3. Standardwerte	3. Standardwerte

Zeichendichte:

Zum Inhalt von Byte 1 zu addierender Wert	Bedeutung
0	Standardwert = 10 Zeichen pro Zoll (16,5 für 80-stelligen Mini-Kompakt-Drucker)
2	10 Zeichen pro Zoll
4	12 Zeichen pro Zoll
6	15 Zeichen pro Zoll
8	16,5 Zeichen pro Zoll

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Nachdruck, Vervielfältigung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der Patenberichterstattung vorbehalten.“

Die Drucker

Die Drucker

Zeilendichte:

Zum Inhalt von Byte 7 zu addierender Wert	Bedeutung
0	Standardwert = 6 Zeilen pro Zoll
2	3 Zeilen pro Zoll
4	4 Zeilen pro Zoll
6	6 Zeilen pro Zoll
8	8 Zeilen pro Zoll

12.1.2 Ausgabe auf dem Drucker

Die Ausgabe am Drucker erfolgt mit den Anweisungen:

- PRINT #
- WRITE #

Mit WRITE # dürfen nur Stringliterals oder der Inhalt von String-Variablen ausgegeben werden.

Als Besonderheiten sind zu beachten:

- Falls <NAusdr2> und <NAusdr3> codiert sind, werden diese Angaben ignoriert.
- In einer WRITE #-Anweisung dürfen keine numerischen Ausdrücke angegeben sein.
- Die unterschiedlichen Funktionen der Trennzeichen ";" und "," sind bei PRINT # unbedingt zu beachten.
- In der PRINT #-Anweisung ist als Ausgabeoperand die TAB(<NAusdr>)-Funktion zugelassen.
- Wird in einer PRINT #-Anweisung nach dem letzten Ausgabeoperanden kein ; oder , codiert, wird nach Ausführung der Anweisung ein Zeilenvorschub durchgeführt.
- Gerätefunktionen (<DevFkt>) müssen bei der WRITE #-Anweisung oktal codiert sein, in der PRINT #-Anweisung dagegen werden auch Mnemonics akzeptiert.

Die Drucker

- Pseudo Wait

Ein WRITE-Auftrag, der nur aus einem Grenzzeichen besteht, veranlaßt den Driver, solange zu warten, bis alle Druckdaten abgearbeitet sind. Dieser Auftrag sollte vor einem CLOSE abgesetzt werden, damit evtl. auftretende Fehler noch vorher gemeldet werden können.

12.1.2.1 Formularsteuerung

Die Formularsteuerung erfolgt mit Hilfe von Formularsteuerzeichen, die als Oktal-Code oder in symbolischer Form im auszugebenden String vorgegeben werden müssen:

Oktal-Code	Symbol	Funktion
210	'EOL'	Zeilenende ohne Vorschub
211	'LFB'	Zeilenschaltung rückwärts
212	'OPN'	Klappenöffnung für Formulareinzug
213	'IN'	Formular auf 1. Druckzeile einziehen
214	'FF'	Formular-Auswurf oder Vorschub auf neues Blatt
215	'CR'	Zeilenvorschub

Die Funktionen 'LFB' und 'IN' gelten für den Drucker \$RPFx (ND 02) und die anderen für alle Drucker.

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Vervielfältigung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

Die Drucker

12.1.2.2 Einzelformularverarbeitung

Das am Korrespondenzdrucker (TD01) anschließbare Einzelblattzuführgerät (1305) ermöglicht das programmgesteuerte Verarbeiten von Einzelformularen. Dabei ist zu beachten :

- Die Verarbeitung von Endlos- und Einzelformularen kann nicht gleichzeitig erfolgen.
- Bei Einzelformularverarbeitung muß das Endlosformular aus dem Leporello entfernt werden.
- Das Nachfüllen der Papiermagazine während des Druckbetriebs ist nicht zulässig.
- Die Einzelformularverarbeitung wird vom Druck-Spooling nicht unterstützt.

Im folgenden werden Druckerfunktionen beschrieben, die mit dem DAP 4 und den Drucker-Kanalprogrammen \$RPL, \$RPR und \$RPF genutzt werden können.

Code	Funktion
'SIS'	Kippentfernung für Formularanzug
'IN'	Formular auf 1. Druckzeile einrasten
'FF'	Formular-Auswurf über Vorschub auf neues Blatt
'OR'	Zeilenvorschub

Die Funktionen 'FF' und 'IN' gelten für den Drucker \$RPF (NB 05) und die anderen für alle Drucker.

	Die Drucker	
--	-------------	--

12.1.2.3 Papiervorschub

Syntax:

$$\left\{ \text{"} \leftarrow 376 \leftarrow 236 \leftarrow 227 \leftarrow \left\{ \begin{array}{c} 200 \\ : \\ 260 \end{array} \right\} \leftarrow \text{"} \right\}$$

<NAusdr> : Anzahl der Papiervorschübe zu je 1/24 Zoll
(1 bis 48)

Die Ausführung eines Einzugsbefehls bewirkt die vorgegebene Anzahl Papiervorschübe (Mikrotakte).
Mikrotaktvorschübe können (falls der Drucker es zuläßt) auch innerhalb von Endlosformularen ausgeführt werden.
Die Angabe 00 bzw. -200 - bewirkt die Synchronisation auf die nächste volle Zeile.

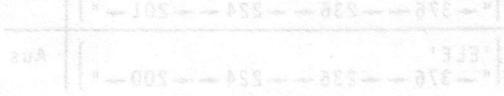
● Beispiel:

```
PRINT #X;MCL(24);
```

oder:

```
PRINT #X;" ← 376 ← 236 ← 227 ← 230 ← ";
```

Wirkung: Papiervorschub um 1 Zoll



© Weitergabe sowie Vervielfältigung dieser Unterlage, Vervielfältigung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Die Drucker

12.1.2.4 Papiercassettenauswahl

Syntax:

$\left\{ \begin{array}{l} 'SC \left\{ \begin{array}{l} 1 \\ 2 \end{array} \right\} ' \\ " \leftarrow 376 \leftarrow 236 \leftarrow 226 \leftarrow \left\{ \begin{array}{l} 201 \\ 202 \end{array} \right\} \leftarrow " \end{array} \right\}$

Eine Cassettenauswahl bleibt bis zur nächsten Umschaltung bzw. bis zum nächsten CLOSE bestehen. Standardmäßig werden die Formulare aus Cassette 1 zugeführt.

• Beispiel:

```
100 PRINT #X;'SC2';
```

Beim nächsten Einzugsbefehl wird das Formular aus Cassette 2 zugeführt.

12.1.2.5 Verbreiterung des Schriftbilds

Syntax:

$\left\{ \begin{array}{l} 'ELO' \\ " \leftarrow 376 \leftarrow 236 \leftarrow 224 \leftarrow 201 \leftarrow " \end{array} \right\}$	Ein
$\left\{ \begin{array}{l} 'ELE' \\ " \leftarrow 376 \leftarrow 236 \leftarrow 224 \leftarrow 200 \leftarrow " \end{array} \right\}$	Aus

Die Funktion "Elongated Druck" bewirkt ein verbreitertes Schriftbild beim Nadeldrucker und gesperrte Ausgabe (d.h. jeweils ein Blank eingefügt) bei Typenrad- oder Zeilendruckern.

Mit 'ELO' wird der "Elongated Druck" ein- und mit 'ELE' wieder ausgeschaltet.

• Beispiel:

```
100 PRINT #X;'ELO';B$;
:
200 PRINT #X;'ELE';
```

Die Drucker

12.1.2.6 Zeichendichte

Syntax:

$$\left\{ \begin{array}{l} \text{'LC [1]} \\ \text{[2]} \\ \text{" } \leftarrow 376 \leftarrow 236 \leftarrow 222 \leftarrow \left\{ \begin{array}{l} 200 \\ 201 \\ 202 \end{array} \right\} \leftarrow \text{"} \end{array} \right\}$$

Diese Funktion bewirkt die Umschaltung der Zeichendichte

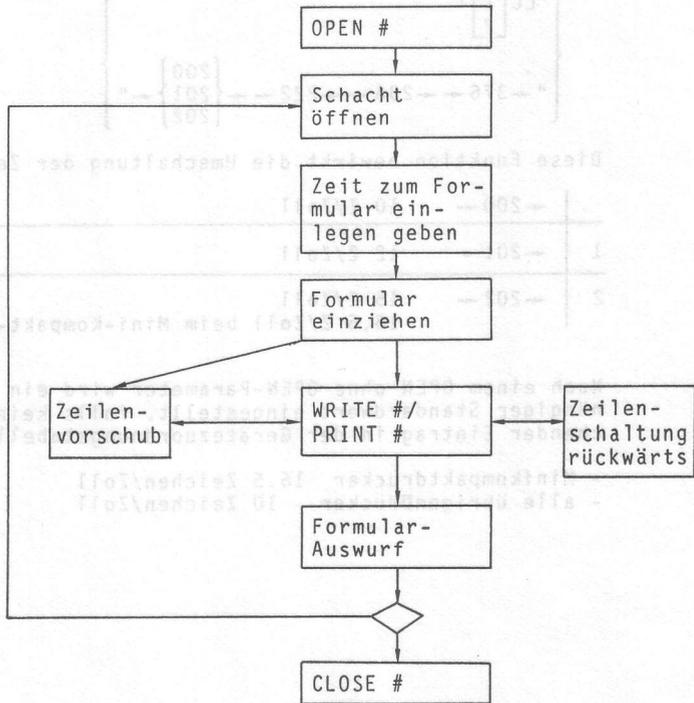
	$\leftarrow 200 \leftarrow$	10 Z/Zoll
1	$\leftarrow 201 \leftarrow$	12 Z/Zoll
2	$\leftarrow 202 \leftarrow$	15 Z/Zoll 16,5 Z/Zoll beim Mini-Kompakt-Drucker

Nach einem OPEN ohne OPEN-Parameter wird ein druckerabhängiger Standardwert eingestellt, falls kein entsprechender Eintrag in der Gerätezuordnungstabelle vorliegt:

- Minikomputerdrucker 16.5 Zeichen/Zoll
- alle übrigen Drucker 10 Zeichen/Zoll

Die Drucker

Beim Bedrucken von Einzelformularen per Formulareinzug ist eine bestimmte Verarbeitungsreihenfolge einzuhalten, die im folgenden Diagramm dargestellt ist:



Auch im Fehlerfall sollte CLOSE # nur nach einem Auswurf (←214←) gegeben werden.

Zeilenschaltungen rückwärts können aus technischen Gründen nach der Verarbeitung einer der letzten Zeilen nicht durchgeführt werden.

Die Drucker

Ein Zeilenvorschub wird automatisch dann durchgeführt, wenn:

- PRINT # ohne abschließendes Semikolon oder Komma zur Ausführung kommt
- mehr Zeichen ohne Steuercode ausgegeben werden, als eine Druckzeile Zeichen aufnehmen kann

Die Behandlung von Textdateien ist weitgehend identisch mit der Behandlung von Druckerdateien, d.h. es kann, falls Druckausgaben gemacht werden sollen und der Drucker belegt ist, der auszugebende Datenbestand in einer Textdatei zwischengespeichert werden.

• Beispiele:

- Zwischenspeichern einer Druckerdatei:

```

1000 IF ERR 0 GOTO 9000 /*FEHLERBEHANDLUNG
1010 OPEN #3;P$,"$LPT" /*DRUCKERERÖFFNUNG
: /*VERARBEITUNG
1100 PRINT #3;R$ /*DRUCKAUSGABE
1110 GOTO 1020
:
9000 IF ERR 0 GOTO 9100 /*FEHLER BEI BUILD
9010 BUILD #3,+D$ /*DATEINAME IN D$
9020 IF ERR 0 GOSUB ... /*STANDARD FEHLERBEH.
9030 GOTO 1020 /*AUFSETZEN NACH OPEN
:
9100 LET D$(LEN D$+1)="!"/ *DATEI ERSETZEN
9110 GOTO 9010

```

- Zeilenweises Drucken mit WRITE #:

```

1000 DIM N$(15),2%,L,Z$(130)
1010 OPEN #3;"005124072","$LPT"
1020 ...
:
1050 GOSUB 9000
1060 GOTO 1020
:
9000 LET Z$=" ",Z$
9010 IF LEN N$=0 GOTO 9030
9020 LET Z$(21,20+LEN N$)=N$
9030 LET Z$(61,69)=L USING "##,###.##"
9040 LET Z$(72)="DM -215 -"
9050 WRITE #3;Z$;
9060 IF CHF 3>=61 WRITE #3;" -214 -";
9070 RETURN

```

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.“

Die Drucker

12.1.3 Die Funktion CHF

Für Druckerdateien wird vom IOCS ein Satzzeiger (Zeilenzähler) geführt, sobald eine Druckerdatei eröffnet ist. Mit der Funktion CHF besteht die Möglichkeit, die aktuelle Druckzeile festzustellen. Als <NAusdr> ist die Nummer des Kanals vorzugeben, auf dem der Drucker eröffnet ist.

Außerdem kann mit CHF (C+200) (C = Kanalnummer) festgestellt werden, ob der Drucker über den Spooler betrieben wird:

CHF (C+200)	Bedeutung
0	Drucker läuft nicht über Spooler
<>0	Drucker läuft über Spooler

● Beispiel:

Vorschub auf die nächste Seite, wenn die Zeilennummer gleich der Formularhöhe -4 ist. Die Formularhöhe ist in der Variablen F abgestellt.

```
150 PRINT #C;A$; /*DRUCKAUSGABE
160 IF CHF C=F-4 PRINT #C;'FF';
```

Anmerkung:

Allerdings ist zu beachten, daß die Auswirkung der Funktion CHF bei Textdateien nicht identisch ist mit der bei Druckerdateien.

12.1.4 Das Schließen des Druckers

Nach dem Drucken ist der Kanal, auf dem der Drucker eröffnet wurde, mit der Anweisung:

```
CLOSE #
```

wieder zu schließen. Dabei wird ein Zeilenvorschub auf Formularanfang durchgeführt.

	Der Lochkartenleser
--	---------------------

13 Der Lochkartenleser

An das System kann ein Lochkartenleser mit einer Leistung von 90 Karten/min angeschlossen werden.

13.1 Zugriff auf Lochkartendateien

Für den Lochkartenleser bildet ein logisches IOCS (Input-Output-Control-System) die Schnittstelle zur Sprache Business-BASIC. Das IOCS übernimmt die Verwaltung der Datenbestände. Alle über den Lochkartenleser eingelesenen Daten sind als Dateien organisiert. Die Bearbeitung einer Lochkartendatei erfolgt über einen Datenkanal.

Eine Lochkartendatei besteht aus einer beliebigen Anzahl von Datensätzen mit einheitlicher Länge. Ein Datensatz entspricht in der Regel der Anzahl Spalten der verwendeten Lochkarten:

- 80 Zeichen.

Die einzig mögliche Verarbeitungsart ist sequentiell.

Der Lochkartenleser kann nicht gleichzeitig von mehreren Teilnehmern oder von einem Teilnehmer auf mehreren Kanälen eröffnet werden.

Die Umwandlung des Standard-Lochkartencodes in ASCII-Code wird intern vom IOCS vorgenommen.

13.1.1 Eröffnen von Lochkartenleserdateien

Bevor Daten über den Lochkartenleser gelesen werden, muß dieser auf einem Datenkanal mit der Anweisung

OPEN #

eröffnet werden. Bei der Eröffnung kann die Anzahl Spalten (1-80) der zu verarbeitenden Lochkarte als Parameter angegeben werden.

Als Dateiname ist der Name des Lochkartenleserkanalprogramms

\$CRD1

anzugeben.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.

	Der Lochkartenleser	
--	---------------------	--

13.1.2 Eingaben über Lochkartenleser

Die Eingabe von Daten über den Lochkartenleser erfolgt mit der Anweisung

READ #

Die angegebenen Zielvariablen müssen Strings sein. Sind mehrere Zielvariablen angegeben, werden soviele Karten gelesen wie Variablen codiert sind. Die Übertragung wird beendet, wenn das Ende der Zielvariablen erreicht ist.

Die Anwendung der Funktion CHF auf Lochkartenleserdateien bleibt ohne Auswertung.

13.1.3 Schließen des Lochkartenlesers

Das Schließen des Kanals, auf dem der Kartenleser eröffnet ist, erfolgt mit der Anweisung

CLOSE #

13.2 Beispiel

Übernahme einer Lochkartendatei in eine relative Datei auf einer Platte. Als Endekriterium für die Lochkartendatei ist eine /*-Karte vorgelegt.

```
:
9000 OPEN #3,"$CRD1" /*LOCHKARTENDATEI
9010 OPEN #4,$ /*RELATIVE DATEI
9020 READ #3,$ /*LOCHKARTE LESEN
9030 IF R$="/*" GOTO 9100 /*DATEIENDE ?
9040 WRITE #4;R$ /*DATENSATZ SCHREIBEN
9050 GOTO 9020 /*NÄCHSTE KARTE
9100 CLOSE #3 /*LOCHKARTENDATEI
9110 CLOSE #4 /*RELATIVE DATEI
9120 CHAIN "" /*CHAIN SCOPE
```