

# CCD-Modul

## Kurzanleitung

Die im folgenden beschriebenen Funktionen stehen Ihnen zur Verfügung, sobald Sie das CCD-Modul in einen der Modulschächte Ihres Rechners eingesetzt haben. Außer den im Katalog 2 erscheinenden Funktionen erweitert das CCD-Modul einige Funktionen des Betriebssystems. Diese sind:

### Erweiterungen des Betriebssystems

#### Die Kataloge:

Alle neuen Kataloge lassen sich mit **R/S** anhalten, sodann mit **SST** und **BST** durchgehen. Das SHIFT-Flag bleibt gesetzt und **SHIFT R/S** läßt die Kataloge rückwärts laufen. Der laufende Katalog kann durch das Drücken einer beliebigen Taste (außer **R/S** und **ON**) beschleunigt werden.

**CAT'0** Zeigt die ID oder die AID von allen Geräten in der angeschlossenen Interface Loop an. Bei angehaltenem Katalog kann mit der **ENTER**-Taste das angezeigte Gerät selektiert werden, mit der **C**-Taste kann es zurückgesetzt werden (Device clear)

**CAT'1** Wie bisher.

**CAT'2** Zeigt zunächst nur die „Überschriften“ der eingesteckten Module an. Mit **ENTER** gelangt man in den Funktionsblock oder wieder zur „Überschrift“ zurück. Mit der **XEQ**-Taste kann eine angezeigte Funktion direkt aus dem Katalog heraus ausgeführt oder program-

miert werden, die **A**-Taste ermöglicht die Zuweisung der angezeigten Funktion ebenfalls direkt aus dem Katalog heraus.

- CAT'3** Wie bisher.
- CAT'4** Zeigt die Namen und die Länge sowie die Typen von Files im erweiterten Speicherbereich (X-Memory) an. Die zusätzlichen File-Typen des CCD-Moduls (B = Buffer, M = Matrix, K = Key) werden richtig angezeigt.
- CAT'5** Ruft die Funktion **ALMCAT** des TIME-Moduls auf.
- CAT'6** Zeigt alle Tastenbelegungen in der Reihenfolge der Tastencodes an. Auch Zuweisungen von synthetischen Funktionen werden richtig angezeigt (z.B. „RCL M“ oder „TEXT 7“). Die Taste **C** löscht die angezeigte Tastenzuweisung.
- CAT'7** Ruft die Funktion **DIR** des IL-Moduls auf.
- CAT'8** bis  
**CAT'F** Zeigt den Katalog 2 beginnend mit dem Modul auf dem Adressblock 8 bis F an.

## Die Funktionen **ASN** und **XEQ**:

Mit diesen Funktionen können jetzt jeder beliebige Zwei-Byte-Befehl und jede XROM-Nummer direkt eingegeben werden.

- ASN** Nach Drücken der Taste **ASN** kann direkt dezimal eingegeben werden, z.B. 144 117 für „RCL M“; mit **ALPHA** gelangt man in die normale **ASN**-Funktion.

## **XEQ**

Mit der Taste **ENTER** gelangt man zur dezimalen Zwei-Byte-Eingabe, durch nochmaliges Drücken von **ENTER** (oder mit der Taste **ALPHA**) gelangt man wieder in die normale **XEQ**-Funktion.

Die Zwei-Byte-Funktion kann dezimal oder hexadezimal eingegeben werden: die **H**-Taste ermöglicht die hexadezimale Eingabe, mit der **.**-Taste gelangt man zurück zur dezimalen Eingabe. Das nochmalige Drücken der **XEQ**-Taste erlaubt die Ausführung (Programmierung) bzw. Zuordnung von XROM-Nummern.

## **Direkte und indirekte Speicherzugriffsfunktionen:**

Alle Speicherzugriffsfunktionen (**RCL**, **STO**, **X<>** usw.) können jetzt auch direkt von der Tastatur aus (ohne Tastenzuweisungen) jedes Statusregister ansprechen. Der Zugriff auf die Register M, N, O, P, Q, r, a, b, c, d und e ist somit genauso einfach geworden wie bei den Registern X, Y, Z, T und L. Die Tastenfolge hierfür ist z.B. **RCL . d**.

## **Die Eingabe von beliebigen ALPHA-Zeichen:**

Dies ist der sogenannte Kleinschreibmodus. Er ist bei ausgeschaltetem USER-Modus aktiv ( **ALPHA an!**). Nun ist es möglich, Texte, die Kleinbuchstaben, andere Sonderzeichen oder Druckersteuerzeichen enthalten, in das ALPHA-Register oder in ein Programm zu schreiben. Beim Drücken einer Taste erscheint zunächst das normale Zeichen, nach Loslassen der Taste wird es in einen Kleinbuchstaben bzw. in ein Sonderzeichen verwandelt. Die meisten dieser Sonderzeichen werden als „Black Box“ angezeigt. Nach der Tastenfolge **SHIFT ENTER** kann man das gewünschte Zeichen auch mit seinem Dezimalwert eingeben, nach anschließendem Drücken der **H**-Taste auch hexadezimal.

## Im Katalog 2 erscheinende Funktionen:

### -W&W CCD A

- B?** (Buffer?) Existiert ein Ein-/Ausgabebuffer mit der im X-Register angegebenen Identitätszahl *aa*?
- CAS** (Clear auto start) Löscht den Autostart-Modus, siehe **SAS**.
- CLB** (Clear buffer) Löscht den Ein-/Ausgabebuffer mit der im X-Register angegebenen Identitätszahl *aa*.
- RNDM** (Random) Erzeugt eine Zufallszahl, Ausgabe im X-Register.
- SAS** (Set autostart) Setzt den Autostart-Modus, d.h., nach jedem Einschalten beginnt der HP 41 sofort mit dem Programmablauf (wie bei gesetztem Flag 11).
- SEED** Speichert einen Startwert für **RNDM** im Ein-/Ausgabebuffer des CCD-Moduls, erwartet eine Zahl kleiner 1 im X-Register.
- SORT** Sortiert die Registerinhalte von  $R_{iii}$  bis  $R_{jjj}$  (auch Alpha-Daten, größten Wert nach  $R_{jjj}$ ); *iii, jjj* wird im X-Register spezifiziert.

### -ARR FNS

Dieser Funktionsblock dient zur Bearbeitung von Datenfeldern. Die benötigten Datenfelder werden mit ihren Namen im ALPHA-Register in der Form „Operand 1, Operand 2, Ergebnis (Result)“ angegeben. Im folgenden wird die verkürzte Schreibweise *OP1, OP2, RES* verwendet. Wird als Name der Buchstabe *X* angegeben, so wird der X-Wert als Operand verwendet. Datenfel-

der werden entweder im Benutzerspeicher oder im erweiterten Speicherbereich angelegt (siehe **MDIM**).

- >C+** (*X to column and increment i*) Schreibt ein Element eines Datenfeldes in die aktuelle Zeigerposition und erhöht anschließend den Reihenzeiger um eins (spaltenweises Eingeben von Datenfeldelementen).
- >R+** (*X to row and increment j*) Schreibt ein Element eines Datenfeldes in die aktuelle Zeigerposition und erhöht anschließend den Spaltenzeiger um eins (reihenweises Eingeben von Datenfeldelementen).
- ?IJ** (*?ij*) Liest die aktuelle Zeigerposition *iii,jjj*.
- ?IJA** (*?ij by ALPHA*) Liest die Zeigerposition *iii,jjj* des angegebenen Datenfeldes.
- C<>C** (*Exchange columns*) Vertauscht Spalte *kkk* mit Spalte *lll* des angegebenen Datenfeldes; *kkk,lll* wird im X-Register erwartet.
- C>+** (*From column to X and increment i*) Liest das Element auf der aktuellen Zeigerposition und erhöht anschließend den Reihenzeiger um eins (spaltenweises Lesen eines Datenfeldes).
- C>-** (*From column to X and decrement i*) Liest das Element auf der aktuellen Zeigerposition und vermindert anschließend den Reihenzeiger um eins (spaltenweises Lesen eines Datenfeldes).
- CMAXAB** (*Column maximum absolute*) Setzt den Zeiger auf das größte absolute Element der im X-Register spezifizierten Spalte des angegebenen Feldes und liest es.

**CNRM** (Column norm) Berechnet die Spaltennorm des angegebenen Feldes.

$$\| A \|_C = \max_{1 < j < n} \sum_{i=1}^m | a_{ij} |$$

**CSUM** (Column sum) Berechnet alle Spaltensummen eines Datenfeldes; erwartet im ALPHA-Register *OPI,RES*.

**DIM** (Dimension) Liest die Dimension des angegebenen Datenfeldes.

**FNRM** (Froebius norm) Berechnet die Froebius-Norm des angegebenen Datenfeldes.

$$\| A \|_F = \left( \sum a_{ij}^2 \right)^{1/2}$$

**IJ=** (ij=) Setzt den Zeiger auf das Element *ij* des aktiven Feldes.

**IJ=A** (ij= ALPHA) Setzt den Zeiger auf das Element *ij* des angegebenen Feldes.

**M+** (Matrix +) Addiert alle Elemente von zwei Datenfeldern (Formel s.u.); erwartet *OPI,OP2,RES* im ALPHA-Register.

$$c_{ij} = a_{ij} + b_{ij}$$

**M-** (Matrix -) Subtrahiert alle Elemente von zwei Datenfeldern (Formel s.u.); erwartet *OPI,OP2,RES* im ALPHA-Register.

$$c_{ij} = a_{ij} - b_{ij}$$

**M\*** (Matrix \*) Multipliziert alle Elemente von zwei Datenfeldern (Formel s.u.); erwartet *OPI,OP2,RES* im ALPHA-Register.

$$c_{ij} = a_{ij} \cdot b_{ij}$$

**M\*M** (*Matrix \* matrix*) Multipliziert zwei Matrizen (Formel s.u.); erwartet *OP1, OP2, RES* im ALPHA-Register.

$$c_{ij} = \sum_{j=1}^n a_{ij} \cdot b_{jk}$$

**M/** (*Matrix /*) Dividiert alle Elemente von zwei Datenfeldern (Formel s.u.); erwartet *OP1, OP2, RES* im ALPHA-Register.

$$c_{ij} = a_{ij} / b_{ij}$$

**MAX** (*Maximum element*) Stellt den Zeiger auf das größte Element des angegebenen Datenfeldes und liest es.

**MAXAB** (*Maximum element absolute*) Stellt den Zeiger auf das größte absolute Element des angegebenen Datenfeldes und liest es.

**MDIM** (*Matrix dimensioning*) Legt ein Datenfeld (Typ „M“ = Matrix) der Dimension *mmm \* nnn* und dem angegebenen Namen im erweiterten Speicher an. Hat der Name die Form *Rlll* (wobei *lll* ein Ziffernstring ist), so wird die Matrix im Datenspeicher, beginnend mit dem Register *Rlll*, angelegt. Dieses Register ist das Statusregister des Feldes. *mmm, nnn* wird im X-Register erwartet.

**MIN** (*Minimum element*) Stellt den Zeiger auf das kleinste Element des angegebenen Datenfeldes und liest es.

**MOVE** (*Move*) Verschiebt einen Teil eines Datenfeldes in ein zweites. Dieser Teil ist durch zwei gegenüberliegende ECKELEMEN-TE  $a_{ij}$  und  $a_{kl}$  definiert. Der Teilblock im Zielfeld wird durch das Element  $b_{mn}$

spezifiziert.

ALPHA : *OPI,RES*  
X-Reg. : *iii,jjj*  
Y-Reg. : *kkk,lll*  
Z-Reg. : *mmm,nnn*

**PIV** (*Pivot*) Stellt den Zeiger auf das größte absolute Element der im X-Register spezifizierten Spalte der unteren Dreiecksmatrix des angegebenen Datenfeldes und liest es.

**R-PR** (*Row = P \* row*)

mit  $Q = a_{kl} / a_{ll}$  gilt:  
für  $l < j \leq n$  :  $a_{kj} = a_{kj} - Q \cdot a_{lj}$   
für  $j = l$  :  $a_{kl} = Q$   
für  $1 \leq j \leq l$  :  $a_{kj} = a_{kj}$

Wird verwendet für die LR-Zerlegung einer Matrix.

**R-QR** (*Row = Q \* row*)

mit  $Q = a_{kl} / a_{ll}$  gilt:  
für  $1 \leq j \leq n$  :  $a_{kj} = a_{kj} - Q \cdot a_{lj}$

Wird verwendet für den Gauß-Algorithmus.

**R<>R** (*Exchange Rows*) Vertauscht Reihe *kkk* mit Reihe *lll* des angegebenen Feldes. *kkk, lll* wird im X-Register erwartet.

**R>+** (*From row to X and increment j*) Liest das Element auf der aktuellen Zeigerposition und erhöht anschließend den Spaltenzeiger um eins (reihenweises Lesen eines Datenfeldes).

**R>-** (*From row to X and decrement j*) Liest das Element auf der aktuellen Zeigerposition und vermindert anschließend den

Spaltenzeiger um eins (reihenweises Lesen eines Datenfeldes).

## R>R?

(Row  $\geq$  row?) Vergleicht spaltenweise die Elemente der Reihe *kkk* mit denen der Reihe *lll* des angegebenen Datenfeldes und überspringt den nächsten Programmschritt, wenn ein Element in Reihe *lll* größer als das Element der Reihe *kkk* (der gleichen Spalte) ist. Sobald ein Element der Reihe *lll* kleiner als das Element der Reihe *kkk* ist, wird der nächste Programmschritt ausgeführt. *kkk, lll* wird im X-Register erwartet.

## RMAXAB

(Row maximum absolute) Setzt den Zeiger auf das größte absolute Element der im X-Register spezifizierten Reihe des angegebenen Feldes und liest es.

## RNRM

(Row norm) Berechnet die Reihennorm des angegebenen Feldes.

$$\|A\|_R = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$$

## RSUM

(Row sum) Berechnet alle Reihensummen eines Datenfeldes; erwartet im ALPHA-Register *OPI, RES*.

## SUM

(Sum) Addiert alle Elemente des angegebenen Datenfeldes.

## SUMAB

(Sum absolute) Addiert die Absolutwerte aller Elemente des angegebenen Datenfeldes.

## SWAP

(Swap) Vertauscht zwei Teilblöcke zweier Datenfelder, wobei ein Teilblock durch zwei gegenüberliegende Eckelemente  $a_{ij}$  und  $a_{kl}$  definiert ist. Der Teilblock im Zielfeld wird durch das Ele-

ment  $b_{mn}$  spezifiziert.

ALPHA : *OPI,RES*  
X-Reg. : *iii,jjj*  
Y-Reg. : *kkk,lll*  
Z-Reg. : *mmm,nnn*

**YC+C** (*Y \* column ± column*) Addiert Spalte *kkk* multipliziert mit Y zu Spalte *lll*; *kkk,lll* des angegebenen Datenfeldes wird im X-Register erwartet.

## -HEX FNS

Dieser Funktionsblock beinhaltet alle notwendigen Funktionen zur Bitmanipulation. Sie sind zusammen mit ihren Ein-/Ausgabefunktionen **ARCLH**, **PMTH**, **VIEWH** und **XTOAH** abhängig von der eingestellten Wortbreite und dem gewählten Modus (Einer-, Zweierkomplement oder Unsigned).

**1CMP** (*1's complement*) Setzt den Einerkomplement-Modus.

**2CMP** (*2's complement*) Setzt den Zweierkomplement-Modus.

**AND** (*And*) Logische „und“-Verknüpfung von X und Y.

**bc?** (*bit clear?*) Ist das Bit *bb* des Y-Registers gelöscht? *bb* wird im X-Register spezifiziert.

**bs?** (*bit set?*) Ist das Bit *bb* des Y-Registers gesetzt? *bb* wird im X-Register spezifiziert.

**Cb** (*Clear bit*) Löscht das Bit *bb* des Y-Registers. *bb* wird im X-Register spezifiziert.

**NOT** (*Not*) Invertiert die Werte aller Bits der Zahl im X-Register.

<b>OR</b>	( <i>Or</i> ) Logische „oder“-Verknüpfung von X und Y.
<b>R&lt;</b>	( <i>Rotate left</i> ) Rotiert den Inhalt von X um 1 Bit nach links. Flag 0 ist das Carry-Bit (Übertrags-Bit).
<b>R&gt;</b>	( <i>Rotate right</i> ) Rotiert den Inhalt von X um 1 Bit nach rechts. Flag 0 ist das Carry-Bit (Übertrags-Bit).
<b>S&lt;</b>	( <i>Shift left</i> ) Verschiebt den Inhalt von X um 1 Bit nach links. Flag 0 ist das Carry-Bit (Übertrags-Bit).
<b>S&gt;</b>	( <i>Shift right</i> ) Verschiebt den Inhalt von X um 1 Bit nach rechts. Flag 0 ist das Carry-Bit (Übertrags-Bit).
<b>Sb</b>	( <i>Set bit</i> ) Setzt das Bit <i>bb</i> des Y-Registers. <i>bb</i> wird im X-Register spezifiziert.
<b>UNS</b>	( <i>Unsigned</i> ) Setzt den Unsigned-Modus, in dem kein Vorzeichen benutzt wird.
<b>WSIZE</b>	( <i>Wordsize</i> ) Stellt die Wortbreite für alle Hexadezimalfunktionen ein. Erwartet einen Wert zwischen 0 und 32 im X-Register.
<b>XOR</b>	( <i>Exclusive or</i> ) Logische „exklusiv-oder“-Verknüpfung von X und Y.
<b>-I/O FNS</b>	
<b>ABSP</b>	( <i>ALPHA backspace</i> ) Löscht von rechts ein Zeichen aus dem Alpha-Register.
<b>ACAXY</b>	( <i>Accumulate ALPHA and X by Y</i> ) Überträgt den Inhalt des ALPHA-Registers linksbündig und den X-Wert rechtsbündig auf der durch Y eingestellten Druck-

breite in den Druckbuffer. Im F/E-Format (s.u.) wird der X-Wert im ENG-Format mit Buchstaben statt Exponent übertragen. Bei gesetztem Flag 20 werden Text und Zahl durch Punkte verbunden. Der X-Wert wird in das LAST-X-Register geschrieben und der Stack nach unten verschoben.

- ACLX** (*Accumulate line by X*) Überträgt *aa* Zeichen mit dem Zeichencode *bbb* in den Druckbuffer; *aa,bbb* wird im X-Register erwartet.
- ARCLE** (*ALPHA recall engineer*) Wie **ARCL X**, ersetzt den Exponenten aber durch Buchstaben (z.B. k für 10 hoch 3).
- ARCLH** (*ALPHA recall hex*) Hängt die hexadezimale Darstellung der Zahl im X-Register an den Inhalt des ALPHA-Registers an (abhängig von der eingestellten Wortbreite und dem gewählten Modus, siehe -HEX FNS).
- ARCLI** (*ALPHA recall integer*) Hängt den ganzzahligen Anteil der Zahl im X-Register an den Inhalt des ALPHA-Registers an.
- CLA-** (*Clear Alpha -*) Löscht das ALPHA-Register von rechts bis zum ersten Leerschritt (Space).
- F/E** (*Fix/eng-mode*) Setzt den Fix/Eng-Modus (Flag 40 und 41 werden gesetzt).
- INPT** (*Input*) Universelle Dateneingabefunktion im laufenden Programm. Das ALPHA-Register wird zusammen mit  $R_{iii}$  angezeigt und die Programmausführung unterbrochen. Nach einer Sekunde kann der angezeigte Wert übernommen oder geändert werden. Durch Drücken

von **R/S** wird der eingegebene Wert mit den Eingabegrenzen verglichen. Liegt er innerhalb dieser Grenzen, so wird er in  $R_{iii}$  abgespeichert. Sodann wird  $iii$  um  $cc$  erhöht und die nächste Programmzeile übersprungen, wenn gilt:  $iii > fff$ .

Kontrollregister:

$R_{000}$ :  $iii, fffc$

$R_{001}$ : minimaler Eingabewert

$R_{002}$ : maximaler Eingabewert

## **PMTA**

(Prompt ALPHA) Zeigt den Inhalt des ALPHA-Registers an, unterbricht die Programmausführung und fordert zur Texteingabe auf.

## **PMTH**

(Prompt hex) Eingabefunktion, die eine hexadezimale Eingabe ermöglicht (abhängig von der eingestellten Wortbreite und dem gewählten Modus, siehe -HEX FNS). Zusätzlich zu den Zifferntasten sind auch die Tasten A, B, C, D, E und F aktiv. Im Programmablauf wird der Inhalt des ALPHA-Registers angezeigt und die Programmausführung unterbrochen.

## **PMTK**

(Prompt key) Menüfunktion: zeigt den Inhalt des ALPHA-Registers an, unterbricht die Programmausführung und erwartet einen Tastendruck. Die zulässigen Tasten werden im ALPHA-Register durch ihren zugehörigen Buchstaben „L“ spezifiziert: z.B. „TEXT LLLL“ (max. 24 Zeichen; Text und L's müssen durch mindestens ein Leerzeichen getrennt sein). Der Stellenwert der gedrückten Taste wird im X-Register ausgegeben. Bei leerem ALPHA-Register wird „KEY?“ angezeigt und der Tastencode der als nächstes gedrückten Taste ausgegeben.

**PRAXY** (*Print ALPHA and X by Y*) Drückt den Inhalt des ALPHA-Registers linksbündig und das X-Register rechtsbündig in eine Zeile. Die Breite der Druckspalte wird durch den Y-Wert bestimmt. Im F/E-Format (s.o.) wird der X-Wert im ENG-Format mit Buchstaben statt Exponent gedruckt. Bei gesetztem Flag 20 werden Text und Zahl durch Punkte verbunden. Der X-Wert wird in das LAST-X-Register geschrieben und der Stack nach unten verschoben.

**PRL** (*Print line*) Drückt 24 (bzw. 12, wenn Flag 12 gesetzt ist) „-“-Zeichen.

**VIEWH** (*View hex*) Zeigt den Inhalt des X-Registers hexadezimal an (abhängig von der eingestellten Wortbreite und dem gewählten Modus, siehe -HEX FNS).

**XTOAH** (*X to ALPHA hex*) Hängt ein oder mehrere Zeichen entsprechend dem hexadezimalen Äquivalent des X-Registers an den Inhalt des ALPHA-Registers an (abhängig von der eingestellten Wortbreite und dem gewählten Modus, siehe -HEX FNS).

## **-ADV FNS**

**A+** (*Increment address*) Erhöht die Adresse im X-Register um ein Byte.

**A+B** (*Address + bytes*) Addiert zu einer Adresse im Y-Register die Anzahl Bytes aus dem X-Register.

**A-** (*Decrement address*) Vermindert die Adresse im X-Register um ein Byte.

**A-A** (*Address - address*) Subtrahiert die Adresse im X-Register von der Adresse im Y-Register.

- DCD** (*Decode*) Hängt das hexadezimal decodierte Abbild des X-Registers an den Inhalt des ALPHA-Registers an.
- PC<>RTN** (*Exchange Programcounter with 1. returnaddress*) Vertauscht den Programmzeiger mit der ersten Rücksprungadresse.
- PC>X** (*Programcounter to X*) Liest den Programmzeiger.
- PEEKb** (*Peek byte*) Liest den Dezimalwert des Bytes  $b$  aus dem absoluten Register  $R_{nnn}$ ;  $nnn, b$  wird im X-Register erwartet.
- PEEKr** (*Peek register*) Liest den Registerinhalt des absoluten Registers  $R_{nnn}$ ;  $nnn$  wird im X-Register spezifiziert.
- PHD** (*Program head*) Schreibt die Adresse des ersten Bytes des Programms, dessen Name im ALPHA-Register spezifiziert wird, in das X-Register.
- PLNG** (*Program length*) Fordert zur Eingabe des Programmnamens auf und zeigt anschließend die Länge dieses Programms in Bytes an.
- POKEb** (*Poke byte*) Schreibt das Byte  $bbb$ , dessen Dezimalwert im X-Register angegeben ist, in die Byteposition  $c$  des absoluten Registers  $R_{aaa}$ ;  $aaa, c$  wird im Y-Register erwartet.
- POKER** (*Poke register*) Schreibt den Inhalt des X-Registers in das absolute Register  $R_{aaa}$ ;  $aaa$  wird im Y-Register spezifiziert.
- PPLNG** (*Programmable program length*) Gibt die Länge eines Programms in Bytes im X-Register an. Der Programmname wird im ALPHA-Register erwartet.

**X>PC** (*X to programcounter*) Setzt den Programmzeiger auf die im X-Register angegebene Adresse.

**X>RTN** (*X to returnstack*) Schiebt die im X-Register angegebene Adresse auf den Rückprungstapel.

**XR>RTN** (*XROM to returnstack*) Schiebt die Adresse einer Programmzeile eines ROM-Programms auf den Rückprungstapel. Im X-Register wird die Nummer *nnnn* der Programmzeile und im Y-Register die XROM-Nummer *kk,ll* erwartet.

## **-XF/M FNS**

Diese Funktionen arbeiten nur in Verbindung mit dem X/F-Modul oder einem HP41CX.

**GETB** (*Get buffer*) Kopiert den im ALPHA-Register spezifizierten Bufferfile in den Ein-/Ausgabebuffer zurück.

**GETK** (*Get keys*) Löscht die bestehenden Tastenbelegungen und reaktiviert die Tastenbelegungen aus dem angegebenen Keyfile.

**MRGK** (*Merge keys*) Fügt die Tastenbelegungen aus dem angegebenen Keyfile zu den bestehenden Tastenbelegungen hinzu, ohne diese zu verändern.

**SAVEB** (*Save buffer*) Speichert den Ein-/Ausgabebuffer mit der Identitätszahl *aa* in dem angegebenen Bufferfile ab; *aa* wird im X-Register erwartet.

**SAVEK** (*Save key assignments*) Speichert alle Tastenzuweisungen im angegebenen Keyfile ab.

**SORTFL** (*Sort file*) Sortiert alle Elemente des angegebenen Datenfiles.

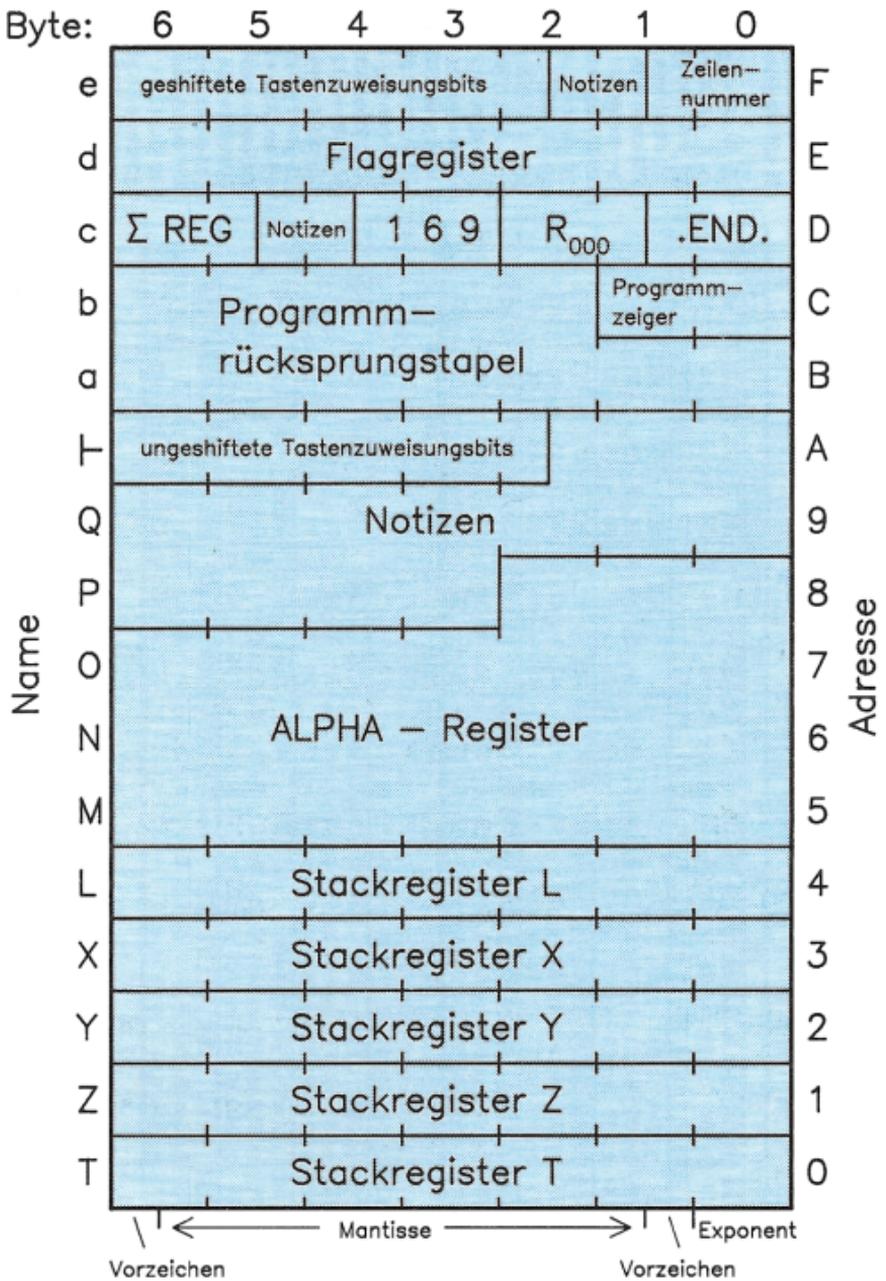
## Fehlermeldungen

Im folgenden finden Sie eine Kurzbeschreibung der Fehlermeldungen des CCD-Moduls. Eine Funktion, die eine Fehlermeldung verursacht, wird nicht ausgeführt (Ausnahme: bei Datenfeldoperationen bricht die betreffende Funktion an der fehlerverursachenden Stelle ab; d.h. die Funktion ist dann bereits teilweise ausgeführt worden).

<b>Fehler</b>	<b>Bedeutung</b>
<b>ALPHA DATA</b>	Nicht numerische Daten.
<b>CAT EMPTY</b>	Keine Alarme vorhanden.
<b>DATA ERROR</b>	Unzulässiger Operand.
<b>DIM ERROR</b>	Angegebenes Datenfeld hat falsche Dimension.
<b>DIR EMPTY</b>	Keine Files im erweiterten Speicher.
<b>END OF FL</b>	Zeiger ist am Ende des Datenfeldes oder Files.
<b>FL NOT FOUND</b>	Angegebenes Datenfeld oder angegebener File existiert nicht.
<b>FL TYPE ERR</b>	Unzulässiger Filetyp.
<b>MEMORY LOST</b>	Permanentspeicher wurde gelöscht.

<b>NAME ERR</b>	Unzulässiger Filetyp.
<b>NONEXISTENT</b>	Angegebene Register oder Funktion nicht vorhanden.
<b>NO ROOM</b>	Nicht genügend Speicherplatz vorhanden.
<b>NO DEVICE</b>	Keine Geräte in der Interface Loop vorhanden.
<b>NO DRIVE</b>	Kein Massenspeichermedium in der Interface Loop vorhanden.
<b>NO FAT</b>	(No function address table) Keine Funktionen im angegebenen Modul vorhanden.
<b>NO HPIL</b>	Keine Interface Loop vorhanden.
<b>NO KEYS</b>	Keine Tastenzuweisungen vorhanden.
<b>NO PRINTER</b>	Kein Drucker vorhanden.
<b>NO ROM</b>	Kein Modul auf der angegebenen Adresse vorhanden.
<b>NO TIMER</b>	Kein TIME-Modul vorhanden.
<b>NO XF/M</b>	Kein erweiterter Speicherbereich vorhanden.
<b>ROM</b>	Zugriff oder Veränderung von Programmen im Festwertspeicher (ROM) unzulässig.
<b>TRANSMIT ERROR</b>	Interface Loop offen oder Übertragungsfehler.

# Statusregister



HP-41C QUICK REFERENCE CARD FOR SYNTHETIC PROGRAMMING

© 1982, SYNTHETIX

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
CAT	@c (GTO.)	DEL	COPY	CLIP	R/S	SIZE	BST	SST	ON	PACK	←(PRGM)	USR/P/A	2	SHIFT	ASN
NULL	LBL 00	LBL 01	LBL 02	LBL 03	LBL 04	LBL 05	LBL 06	LBL 07	LBL 08	LBL 09	LBL 10	LBL 11	LBL 12	LBL 13	LBL 14
0	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	EEX	NEG	GTO	XEQ	W	T
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
RCL 00	RCL 01	RCL 02	RCL 03	RCL 04	RCL 05	RCL 06	RCL 07	RCL 08	RCL 09	RCL 10	RCL 11	RCL 12	RCL 13	RCL 14	RCL 15
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
STO 00	STO 01	STO 02	STO 03	STO 04	STO 05	STO 06	STO 07	STO 08	STO 09	STO 10	STO 11	STO 12	STO 13	STO 14	STO 15
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
+	*	/	^	X<Y?	X>Y?	XSY?	Z+	Z-	HMS+	HMS-	MOD	%	%CH	P→R	R→P
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
LN	X↑2	SQRT	Y↑X	CHS	ETX	LOG	10↑X	SIN	COS	TAN	ASIN	ACOS	ATAN	→DEC	←DEC
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
1/X	ABS	FACT	X≠0?	X>0?	LN1+X	X<0?	X=0?	INT	FRC	D→R	R→D	→HMS	←HR	RND	→OCT
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110
CLS	X<Y	PI	CLST	R↑	RDN	LASTX	CLX	X=Y?	X≠Y?	SIGN	X&0?	MEAN	SDEV	AVIEW	CLD
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

FLAGS (Register d)

- 00-10 general purpose
- 11 auto execute
- 12 doublewide
- 13 lower case
- 14 overwrite
- 15-16 IL printer
- 0 0 MAN
- 0 1 NORM
- 0 0 TRACE
- 1 1 TR/STACK record
- incomplete
- 18 general use
- 19 cleared at
- 20 J turn-on
- 21 prtr enable
- 22 num. entry
- 23 alpha entry
- 24 range ignore
- 25 error ignore
- 26 audio enable
- 27 USER mode
- 51 SST
- 28 dec./comma
- 52 PGRM digit grouping
- 30 CAT
- 31 timer
- 32 manual IL I/O

- 33 IL absolute manual
- 34 ADROFF
- 35 disable autostart
- 36-39 number of digits display
- 40-41 SCI
- 0 0 ENG
- 0 1 FIX
- 0 1 FIX/ENG
- 42-43 trig mode
- 0 0 DEG
- 0 1 RAD
- 1 0 GRAD
- 1 1 RAD
- 44 cont. ON
- 45 system
- 46 partial key data entry
- 47 SHIFT sequence
- 48 ALPHA
- 49 low BAT
- 50 message
- 51 SST
- 52 PGRM
- 53 I/O
- 54 PSE
- 55 printer existence

bit numbers in a 7-byte register

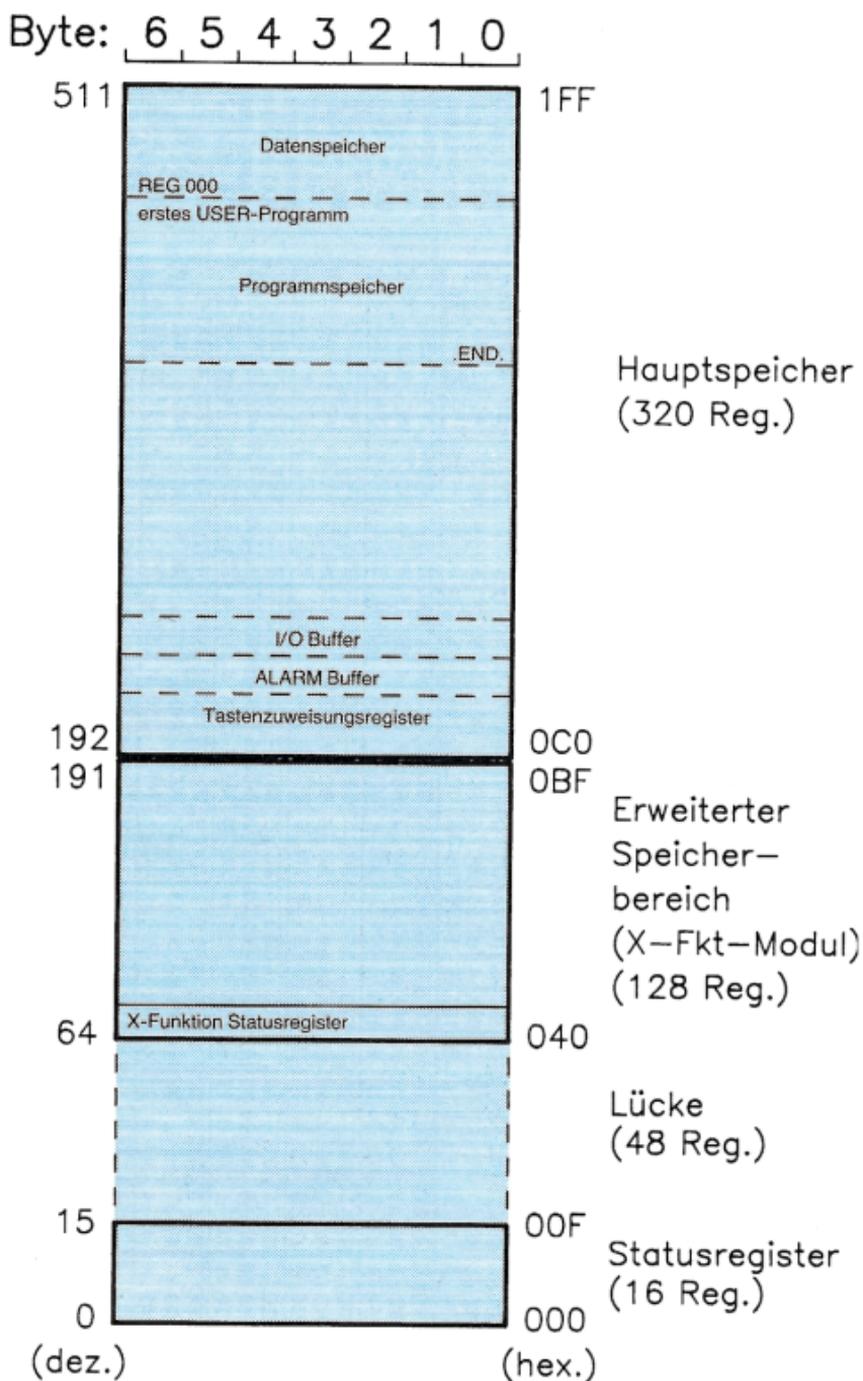
HP-41C QUICK REFERENCE CARD FOR SYNTHETIC PROGRAMMING © 1982, SYNTHETIX

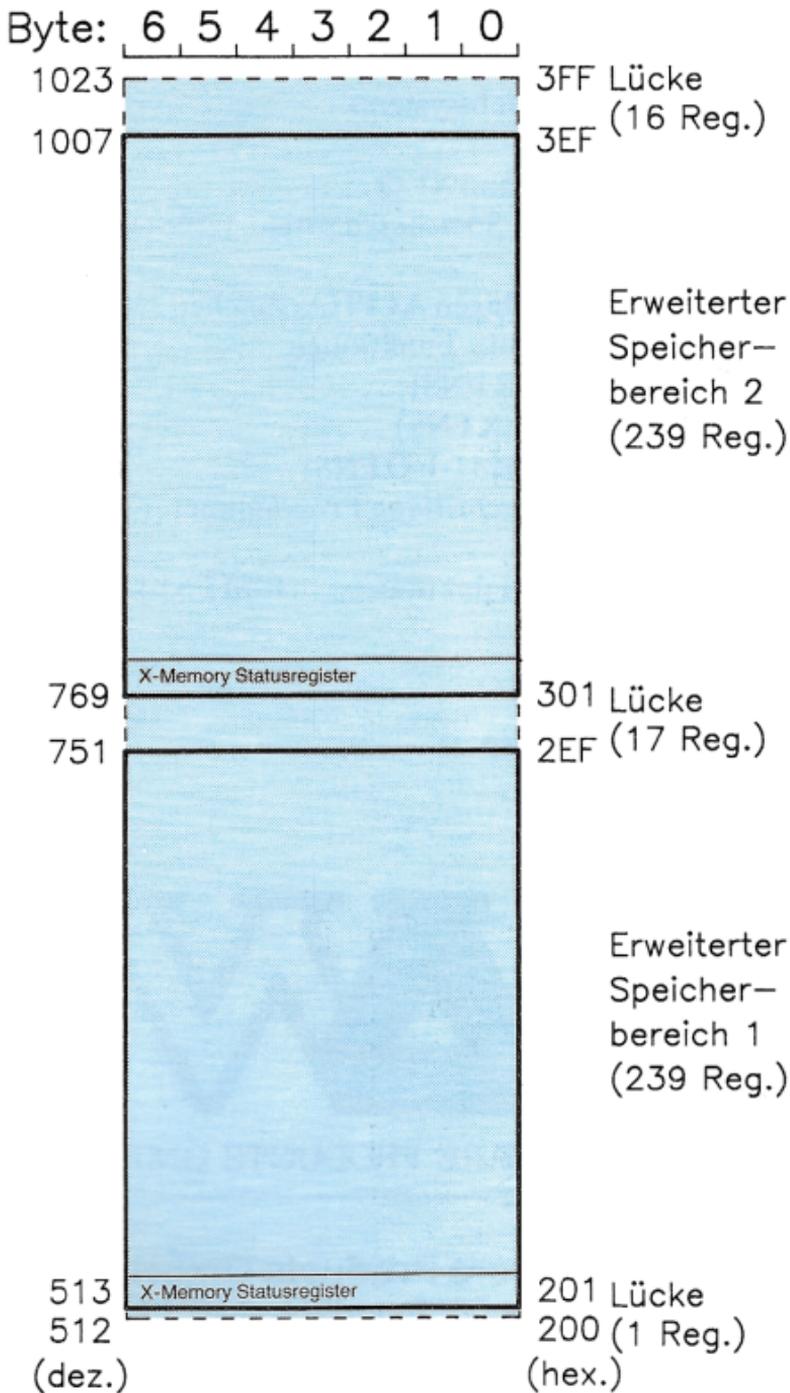
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DEG	RAD	GRAD	STOP	ENTER↑	RTN	BEEP	CLA	ASHF	PSE	CLRG	AOFF	AON	OFF	PROMPT	ADV	
8	IND 00	IND 01	IND 02	IND 03	IND 04	IND 05	IND 06	IND 07	IND 08	IND 09	IND 10	IND 11	IND 12	IND 13	IND 14	IND 15
	128 ↕	129 ×	130 ÷	131 ←	132 α	133 β	134 Γ	135 ↓	136 Δ	137 σ	138 ⬆	139 ↗	140 μ	141 ⋄	142 ↖	143 ✎
RCL	STO	ST+	ST*	ST-	ST/	ISG	DSE	VIEW	Σ REG	ASTO	ARCL	FIX	SCI	ENG	STONE	
9	IND 16	IND 17	IND 18	IND 19	IND 20	IND 21	IND 22	IND 23	IND 24	IND 25	IND 26	IND 27	IND 28	IND 29	IND 30	IND 31
	144 ♂	145 Ω	146 ♂	147 ♀	148 α	149 ρ	150 α	151 ♂	152 ♂	153 ♂	154 ∩	155 ♂	156 ♂	157 ♂	158 ♂	159 ♂
XR 0-3	XR 4-7	XR8-11	X12-15	X16-19	X20-23	X24-27	X28-31	SF	CF	F5?C	FC?C	F5?	FC?	GTO IND	SPARE	
A	IND 32	IND 33	IND 34	IND 35	IND 36	IND 37	IND 38	IND 39	IND 40	IND 41	IND 42	IND 43	IND 44	IND 45	IND 46	IND 47
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	IND 48	IND 49	IND 50	IND 51	IND 52	IND 53	IND 54	IND 55	IND 56	IND 57	IND 58	IND 59	IND 60	IND 61	IND 62	IND 63
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
SPARE	GTO 00	GTO 01	GTO 02	GTO 03	GTO 04	GTO 05	GTO 06	GTO 07	GTO 08	GTO 09	GTO 10	GTO 11	GTO 12	GTO 13	GTO 14	
B	IND 64	IND 65	IND 66	IND 67	IND 68	IND 69	IND 70	IND 71	IND 72	IND 73	IND 74	IND 75	IND 76	IND 77	IND 78	IND 79
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
C	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	GLOBAL	X<->	LBL
	IND 80	IND 81	IND 82	IND 83	IND 84	IND 85	IND 86	IND 87	IND 88	IND 89	IND 90	IND 91	IND 92	IND 93	IND 94	IND 95
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
D	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO
	IND 96	IND 97	IND 98	IND 99	IND 100	IND 101	IND 102	IND 103	IND 104	IND 105	IND 106	IND 107	IND 108	IND 109	IND 110	IND 111
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
E	TEXT 0	TEXT 1	TEXT 2	TEXT 3	TEXT 4	TEXT 5	TEXT 6	TEXT 7	TEXT 8	TEXT 9	TEXT 10	TEXT 11	TEXT 12	TEXT 13	TEXT 14	TEXT 15
	IND 2	IND 3	IND 4	IND 5	IND 6	IND 7	IND 8	IND 9	IND 10	IND 11	IND 12	IND 13	IND 14	IND 15	IND 16	IND 17
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
F	IND 20	IND 21	IND 22	IND 23	IND 24	IND 25	IND 26	IND 27	IND 28	IND 29	IND 30	IND 31	IND 32	IND 33	IND 34	IND 35
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Structure of multi-byte instructions  
 Two-byte instructions  
 STO 16 = 145, 16 DSE IND 55 = 151, 183  
 LBL e = 207, 127 F5?C IND Y = 170, 242  
 RCL b = 144, 124 TONE 89 = 159, 89  
 X<->M = 206, 117 V T W H N = 146, 246  
 LBL Q = 207, 121 ST+ IND(109) = 152, 109  
 Two-byte special cases  
 GTO IND = 174, reg. XEQ IND = 174, 128 + r  
 GTO IND 09 = 174, 9 XEQ IND X = 174, 243  
 XROM i, j = 160 + i/4, 64(i mod 4) + j  
 WSTS = XROM 30, 10 = 167, 138  
 short form GTO = 177 + label, 0  
 GTO 12 = 189, 0  
 Three-byte instructions  
 long form GTO = 208, 0, label  
 GTO 32 = 208, 0, 32  
 XEQ = 224, 0, label  
 XEQ D = 224, 0, 105  
 END = 192, 0, 9 + sum of status indicators  
 32(, END.), 4(rePACK), 2(decompile)  
 Variable length instructions  
 TEXT = 240 + n, n character bytes  
 Append symbol counts as first char.  
 †& = 241, 38 †r? = 243, 127, 41, 63  
 GTO † = 29, 240 + n, n character bytes  
 GTO † XYZ = 29, 243, 88, 89, 90  
 XEQ † = 30, 240 + n, n character bytes  
 XEQ † A = 30, 241, 65 (synthetic)  
 LBL † = 192, 0, 241 + n, (key), n chars.  
 LBL † = 192, 0, 242, 0, 58 (synthetic)

For price information and a list of dealers in your area, send a self-addressed stamped envelope to: SYNTHETIX, 1540 Matthews Ave., Manhattan Beach, CA 90266, USA

# Speicheraufteilung des HP-41





# Inhalt

Erweiterungen des Betriebssystems	
Die Kataloge .....	1
Die Funktionen ASN und XEQ .....	2
Direkte und indirekte Speicherzugriffs- funktionen .....	3
Die Eingabe von beliebigen ALPHA-Zeichen .....	3
Im Katalog 2 erscheinende Funktionen	
Feldfunktionen (-ARR FNS) .....	4
Binärfunktionen (-HEX FNS) .....	10
Ein- Ausgabefunktionen (-I/O FNS) .....	11
Funktionen für fortgeschrittene Programmierer (-ADV FNS) .....	14
X-Funktion-Memory Funktionen (-XF/M FNS) .....	16
Fehlermeldungen .....	17
Statusregister .....	19
Byte-Tafel .....	20
Speicheraufteilung des HP-41 .....	22



**W&W Software Products GmbH**  
Im Aehlemaar 20, Postfach 800 133  
5060 Bergisch Gladbach 2, Tel.: 02202/85068



COMPUTER  
PRODUKTE

PARTNER-  
LÖSUNGEN

© W&W Software Products GmbH 1985