

\*\*\*\*\*  
 \* SEDEZIMAL-CODE \*  
 \* TRICKS mit \*  
 \* SONDERZEICHEN, \*  
 \* SONDERBEFEHLEN \*  
 \* und PASSWOERTERN \*  
 \* (C) 1982 Welker \*  
 \*\*\*\*\*

**1. PROBLEMSTELLUNG**

Die Befehle und Zeichen sind intern sedezimal (= hexadezimal) codiert, d.h. durch Ziffern der Basis 16 dargestellt:

0123456789. ' ° -

Die sechs letzten Ziffern, die in der Literatur sonst mit ABCDEF

bezeichnet werden, haben dezimal folgende Werte:

- ' = 10
- ° = 11
- = 12
- ° = 13
- = 14
- = 15 (Blank)

Jeden Befehl ist eine 2-stellige Sedezimalzahl zugeordnet, z.B. 2° 35 '- Es gibt daher 16x16=256 Befehle, die zwischen 2 ALPHA-Befehlen als Zeichen bzw. Zeichenbefehle interpretiert werden.

Unser Ziel ist es, den internen SEDEZIMAL-Code kennenzulernen und anzuwenden. Dazu muss der Rechner zunächst ueberlistet werden, damit er mit Sedezimalzahlen rechnet.

**2. SEDEZIMALZAHLEN**

**a) GRUNDZIFFER**

Da die Sedezimalziffern nicht ueber die Tastatur eingegeben werden koennen, muss zuerst eine GRUNDZIFFER erzeugt werden. Dazu ein kleines Programm:

```

MODE2
P0 .0 HLT ""
÷ 10 = MinF HLT

MODE1
START: P0
ANZEIGE: 0.0
EINGABE: invALPHA
          inv#
          FST FST EXE
ANZEIGE: '
          = Grundziffer
          (= Dezimal-Pkt.)
    
```

b) RECHENREGELN  
 Ausgehend von der Grundziffer ' koennen die uebrigen Sedezimalziffern und -zahlen durch arithmetische Operationen erzeugt werden, z.B.:

- ' - 11 + 10 = °
- ' - 11 + 12 = °
- ' - 11 + 13 = -
- ' - 11 + 14 = -
- ' - 11 + 20 = 1°
- ' - 11 + 21 = 1'
- ' - 11 + 22 = 1°
- ' - 11 + 23 = 1-
- ' - 11 + 24 = 1-
- ' - 11 + 30 = 2°
- ' - 11 + 31 = 2'
- ' - 11 + 32 = 2°
- .....
- ' - 11 + 93 = 8-
- ' - 11 + 94 = 8-
- .....
- ' - 11.1 + 10 = ..9
- ..9 INT = .
- . ++ = 1.
- = 2.
- = 3.
- .....

```

3.55555550E15 ÷ ' =
= 1. .... E 14
Min00
MR00- 3.11111110E14
= -1. E14
+/- = 1. E14
Min01
MR01- 3.81403581E 14
= -1.2° - ° 2° E 14
    
```

Um Ziffern wie - u. beim Weiterrechnen nicht zu verlieren, wenden Sie zweckmaessig Speicherarithmetik an, z.B.:

```

' Min00
11 M-00 13 M+00
    
```

**3. INDIREKTE SPEICHERADRESSIERUNG**

Die Sedezimalzahlen ermöglichen die indirekte Adressierung von Speichern, die ueber die Dezimaladressen z.T. nicht erreichbar sind. Die auf diese Weise ansprechbaren Speicher ergeben sich aus der folgenden Tabelle:

TABELLE 1:  
 SEDEZIMAL-ADRESSEN

#	°	'	°	'	°	'	°	'	°
0	F	L1	L2	L3	L4	L5			
1	1F	L6	L7	L8	L9	L10			
2	2F	20	20	20	20	20			
3	3F	25	26	27	28	29			
4	4F	35	36	37	38	39			
5	5F	45	46	47	48	49			
6	6F	55	56	57	58	59			
7	7F	65	66	67	68	69			
8		75	76	77	78	79			

a) KLAMMERREGISTER  
 Die Register L1-L10 haben eine Bedeutung fuer das Rechnen mit Klammern (Klammer Ebenen 1-10). Es sind vollwertige Datenspeicher, die jetzt durch entsprechende indirekte Adressierung in den Rechenablauf einbezogen werden koennen, wenn man auf Klammerrechnungen in den betr. Ebenen verzichtet. Sie werden durch MAC nicht seloescht!

b) F - REGISTER  
 Ueber die Sedezimal-Adressen: . bis 7. koennen die sonst nur direkt adressierbaren Datenspeicher: F bis 7F auch indirekt adressiert werden!  
 BEISPIEL:  
 M00 = 1.  
 MIF = 25  
 IND MR00 → 25

**c) HOEHERWERTIGE SPEICHERADRESSEN**

Ein weiterer Teil der Sedezimal-Adressen stimmt mit den Dezimal-Adressen: 20, 25-29, ... 75-79 ueberein. Da die zugehoerigen Datenspeicher auch ueber die Dezimal-Adressen indirekt adressierbar sind, wird hierdurch scheinbar nichts hinzugewonnen. Bis auf eine Ausnahme:

d) PROGRAMMSPEICHER  
 Gerade diese Ausnahme hat es aber in sich. Sie verschafft uns voellig neue Einblicke in die interne Befehlsstruktur und eroeffnet interessante Anwendungsmoeglichkeiten. Dazu zunaechst einise Versuche:

Geben Sie ein:  
 MODE3 MAC  
 MODE1 MODE.77  
 Damit sind die Datenspeicher M00-76 verfuegbar.  
 Speichern Sie ab:  
 M76 = 76  
 M00 = 8'  
 M01 = 8°  
 M02 = 8-  
 Versuchen Sie nun:  
 MR76 → 76  
 IND MR00 → 76  
 MR77 geht nicht  
 IND MR01 → - . E65  
 IND MR02 → M ERROR  
 Ueberraschend ist das Ergebnis bei IND MR01.  
 Eigentlich haette man hier M ERROR erwartet, wie bei IND MR02.  
 Nach obiger Tabelle gehoert naemlich die Sedezimaladresse 8° zum Datenspeicher 77 der sarnicht verfuegbar sondern dem Programmspeicher zugezuehrt ist.

Stammt die mysterioese Zahl - . E65 also aus dem Programmspeicher? Um dies nachzupruuefen, geben Sie ein:  
 MODE2  
 P0: 4444 ..... 4444  
 ganz auffuellen bis zur Pos. 55  
 Versuchen Sie nun:  
 MODE1  
 IND MR01 +  
 ...1.1.1.1 E-80 !

Der Inhalt von Sp.8° hat sich tatsaechlich geaendert!  
 Speichern Sie nun ab 2.132435465 E67  
 IND Min01  
 und sehen Sie sich anschliessend P0 an:  
 MODE2  
 P0: 4444 ..... 4444  
 Min07 X-M01  
 GSBP6 G0T05  
 LBL4 RND3 RND2 4

Damit haben wirs: Die nach Sp.8° abgespeicherte Zahl gelangte zum ENDE des PROGRAMMSPEICHERS und wird dort als eine folse von 7 Befehlen interpretiert (die letzte 4 blieb erhalten).

ERGEBNIS:  
 Das ENDE (7 steps) des PROGRAMMSPEICHERS ist ueber diejenige SEDEZIMAL - ADRESSE adressierbar, die lt. obiger Tabelle der Zahl XX in MODE.XX entspricht.  
 Jedem step entsprechen 2 Stellen in der zugehoerigen Sedezimalzahl. Die Zuordnung zu Befehlen und Zeichen erfolgt nach dem internen SEDEZIMAL-CODE:

\*\*\*\*\*  
 \* SEDEZIMAL-CODE \*  
 \*\*\*\*\*

4. SEDEZIMAL-CODE

Wenn wir die letzten 8 Befehlsadressen im Programmspeicher in Gedanken rueckwaerts mit 8-1 durchnummern:  
 ... 8 7 6 5 4 3 2 1  
 so sind diesen Adressen die folgenden Doppelstellen in der zusehoerigen Sedezimalzahl zugeordnet:  
 2.3344556677 e88

Jeden Befehl entspricht eine 2-stellige sedezimale Codezahl. Zwischen zwei ALPHA-Befehlen (Codezahl 2") werden die Codezahlen als Zeichen oder Zeichenbefehle interpretiert. Die betreffenden Zuordnungen sind in der beiliegenden CODE-TABELLE zusammengestellt:

Im oberen Teil der Kaestchen ist der jeweilige BEFEHL und im unteren Teil das ALPHA-ZEICHEN angegeben. Sofern mit dem CASIO-Drucker FP-10 ein vom angezeigten Zeichen abweichendes Zeichen ausgedruckt wird, ist unten rechts jeweils noch das DRUCKZEICHEN angegeben. Die ZEICHEN-BEFEHLE sind in Klammern geschrieben. Entsprechendes gilt fuer einise SONDERBEFEHLE.

Bei Durchsicht der CODE-TABELLE werden Sie feststellen, dass eine Reihe von Befehlen und Zeichen mit der Tastatur nicht erreichbar ist. Mit einem kleinen Programm und einigen Tricks lassen sich diese SONDERBEFEHLE und SONDERZEICHEN aber einfach in der gewuenschten Reihenfolge synthetisieren und in ein Programm einfuegen.

5. SYNTHESE-PROGRAMM

Programm und Daten eingeben:

MODE.75

PROGRAM LIST  
 M00-74, F-6F 072steps  
 FILE: SONDER

\*\*\* P0  
 M02 = MRF = Min00 -  
 M03 = +/- Min00 - 3

HLT EXP 14 = Min00  
 IND Min01  
 HLT  
 ...023steps

\*\*\* P1  
 44444444444444444444  
 44444444444444444444  
 44444444  
 ...049steps

MEMORY LIST

M01 = 8"  
 M02 = 3.555555550e 15  
 M03 = 3.111111110e 14  
 M0F = '

(Zur Beziehung zwischen XX in MODE.XX und Inhalt von M01 s. Ziff.3d-ERGEBNIS)

Das Programm erzeugt aus vier 2-stelligen dezimalen Eingabedaten: D3, D4, D5, D6 Sedezimal-Codezahlen S3, S4, S5, S6 und speichert diese auf die Positionen 3,4,5,6 (s. Ziff.4) am Ende des Programmspeichers (die auf Pos. 2,7,8 gelangenden Codezahlen lassen sich nicht eindeutig festlegen).

Die Ziffernzuordnung ist dabei wie folgt:

D| 0123456789  
 S| . 0123

Bei Loeschen des Befehls +/- in P0 erhaelt man folgende D/S-Ziffernzuordnung

D| 0123456789  
 S| 6789.

START: P0  
 ANZEIGE: 3. HLT  
 EINGABE: D3 D4 D5 D6  
 (je 2-stellig)  
 EXE  
 ANZEIGE: S2.S3...eS8  
 abgespeichert → P1

BEISPIEL:

D3D4D5D6 = 81403581  
 S = -1.2... 14  
 S2 S3S4S5S6 S8  
 Programmspeicher P1:  
 Pos.: . 6 5 4 3 .  
 Inhalt: " " " "  
 Druck: \$ &

6. SYNTHESE von SONDERZEICHEN & SONDERBEFEHLEN

a) SYNTHESE-PROGRAMM  
 - MODE3 invMAC  
 - Prgm. P0 (Ziff.5) einseben

b) PROGRAMM-AUSWAHL  
 - Ausgewaehletes Programm Px, in das die SONDERZEICHEN bzw. SONDERBEFEHLE gelangen sollen, mit MODE3 AC loeschen  
 - Dann Px im WRT-Betrieb mit Fuellsteps ganz fuellen z.B. mit:  
 4444...4  
 (Bef. & Zeichen)  
 oder mit:  
 "4444...4"  
 (Zeichen)

c) DIREKTE SYNTHESE  
 - Anhand der CODE-TABELLE und der D/S-Tabelle vier Codezahlen Di auswaehlen

- START: P0  
 ANZEIGE: 3.  
 EINGABE: D3D4D5D6  
 EXE  
 (s.Beisp. Ziff.5)  
 - nicht benoetigte Befehle u. Zeichen in Px im WRT-Betrieb mit C-Taste loeschen.

d) INDIREKTE SYNTHESE  
 Fuer SONDERZEICHEN, die ueber das Programm P0 nicht erreichbar sind, z.B. die CODE-Zahlen:

-9(0) oder -4(1)  
 - Synthese eines singularen ALPHA-Befehls 2" (D=81) mit P0, z.B.:  
 D3D4D5D6=81707070  
 - MODE2 Px BST  
 - Loeschen der nicht benoetigten Befehle u. Zeichen sowie einiger vor " stehender Fuellsteps (4 → k!) mit der C-Taste  
 - Einfuegen des dem gewuenschten SONDERZEICHEN laut CODE-TABELLE code-naessig entsprechenden Befehls hinter " z.B.:  
 M+19 fuer 0  
 M+17 fuer "  
 M+14 fuer ↑  
 - dahinter mit Fuellsteps bis zum Speicherende auffuellen (mindestens 8 steps)  
 - Synthese eines zweiten singularen ALPHA-Befehls 2" (D=81) mit P0  
 - nicht benoetigte Befehle u. Zeichen in Px mit C-Taste loeschen.

e) WIEDERHOLUNG

- Vor einer wiederholten Synthese ist dafuer zu sorgen, dass die letzten 8 steps im Programmspeicher zum Uberschreiben frei sind. Dazu sind vorne einige Fuellsteps zu loeschen und hinten anzufuegen.  
 - Soll statt Px ein anderes Programm Py gewaehlt werden alle Fuellsteps in Px loeschen und fuer Py wie unter b) verfahren.

f) PROGRAMMEINGABE

Wenn die gewuenschten SONDERZEICHEN und -BEFEHLE alle in der richtigen Reihenfolge im Programmspeicher stehen, kann das Syntheseprogramm P0 geloescht werden. Die uebrigen Befehle und Zeichen eines Programms koennen dann im WRT-Betrieb direkt ueber die Tastatur eingeseben werden.

7. SONDERZEICHEN

Der Zeichenvorrat ergibt sich aus der CODE-TABELLE. Nachstehend sind alle SONDERZEICHEN zusammengestellt, die im ALPHA-Modus mit dem Drucker FP-10 ausdrueckbar sind:

a) DRUCK = ANZEIGE  
 CODE.....DRUCK  
 7" ..... "  
 7' ..... '  
 7° ..... °  
 - ..... -  
 00 ..... 0  
 0- ..... 8

b) DRUCK+ANZEIGE(A)  
 CODE... A .....DRUCK  
 0 ..... 0  
 0 ..... \$  
 -4 ..... 4  
 -5 ..... 5  
 -7 ..... 7  
 -9 ..... 9  
 - ..... -  
 -- ..... 2  
 82 ..... 82steps

c) ZENTRIERUNGSPUNKTE

CODE A DRUCK  
 81 : ...  
 8- : .....

Die 5 Zentrierungspunkte: ..... koennen im ALPHA-Modus direkt mit der Tastatur eingeseben werden:  
 " MODE invEXE BST "

8. SONDERBEFEHLE

a) FIXn  
Die FIXn-Befehle beanspruchen bekanntlich 2 steps. Sie sind zusammengesetzt aus:  
F1 RNDn  
mit den Codezahlen  
2' In

F1 sowie F2 bis F4 koennen mit dem SYNTHESE-Programm auch isoliert erzeugt werden (D=82-85). Sie werden im WRT-Modus zwar nicht als solche angezeigt, sind aber an einem Sprung in der Programmadresse erkennbar.

b) HOEHERWERTIGE SPEICHERADRESSEN  
Auch die Speicherbefehle mit Adressen >19 und >1F nehmen 2 steps ein.  
Es ist dort eine Erhoehung der Speicheradresse erforderlich die durch ein vorangestelltes F1 bis F4 erfolgt:

F1 Code..... Mn  
F1 2' ..... +20  
F2 2° ..... +40  
F3 2- ..... +60  
F4 2w ..... +80  
z.B.:  
2' 64=F1 Min04=Min24  
2- 8=F3 MR18 =MR78  
2° 8=F2 M+1F =A+5F

c) 6 weitere LBLi

Die Befehle HLT \* F1 F2 F3 F4 liesen in der LBLi-Reihe (Code 2i) der CODE-TABELLE. Sie haben zusatzlich die Bedeutung von LBLi mit sedezimaler Kennziffer:

HLT = LBL.  
" = LBL"  
F1 = LBL'  
F2 = LBL°  
F3 = LBL-  
F4 = LBLw

Sie koennen indirekt mit IND GOTOi angesprungen werden, wenn in M0i die sedezimale Kennziffer steht. Der betreffende Befehl wird beim Ansprungen nicht ausgefuehrt:  
- bei HLT wird nicht angehalten  
- bei " wird nicht auf ALPHA geschaltet - die Zeichen werden als codegleiche Befehle interpretiert  
- z.B. MR35= F1 MR15 wird als MR15 interpretiert.

d) 6 weitere Pi

Die Zahlenweisungen 0 1 2 3 . EXP liesen in der Reihe der Programmstart-Labels Pi (Code 0i) der CODE-TABELLE. Sie haben die zusatzliche Bedeutung von Programmstart-Labels mit sedezimaler Kennziffer:

0 = P.  
1 = P°  
2 = P'  
3 = P-  
EXP = Pw  
Sie koennen indirekt mit

IND GSB(0) angesprungen werden, wenn in M00 die sedezimale Kennziffer steht. Es wird dabei die Stelle im Programmspeicher angesprungen, an der die betr. Zahlenweisung (zB 2) zum erstenmal steht.

Das Programm wird von dort aus - ohne Ausfuehrung der Zahlenweisung - wie ein uebliches Unterprogramm ausgefuehrt.

e)(SAC) (SAC:) (GRA)  
Diese SONDERBEFEHLE haben die Wirkung anderer, bekannter Befehle:  
BEFEHL WIRKUNG  
(SAC) x0  
(SAC:) x0EL  
(GRA) ISZ

f) NOP (10<sup>x</sup>) (e<sup>x</sup>)  
- NOP  
No Operation  
- (10<sup>x</sup>)

Bei Erreichen dieses Befehls wird der Operationsspeicher geloescht wie mit C-Taste (= programmierte C-Taste) Befehl kann nur im PCL-Modus geloescht werden, nicht mit C-Taste im WRT-Mode - (e<sup>x</sup>)  
Programm haelt bei Erreichen dieses Befehls an mit einer Anzeige wie bei BST (= programmierte BST-Taste). Anzeige sperrt automatische Rechnerabschaltung.  
(s. auch Ziff.9c)

g) ERROR (Code -. und -°)  
WRT-Modus:  
-Error-Anzeige, die nur durch Aus- und Einschalten des Rechners geloescht werden kann!  
-ANWENDUNG: Bildung eines Passwort-Sonderzeichens (s.10c)  
-Loeschen der Befehle nur im PCL-Modus.  
RUN-Modus:  
-arithmetische Rechenoperation (unbekannt?)

h) SA - FLAG (Code -)  
Bei Erreichen dieses Schritts fuehrt das Programm noch den nachfolgenden Befehl aus und haelt dann an.

ANWENDUNG: Bedingter Programmstopp mit einer gewuenschten Anzeige ohne LBL und GOTO (auch im ALPHA-Modus!).  
BEISPIEL 1:  
.. x=0 SA MR01 ..  
(Stop + Anzeige M01, wenn x=0)  
BEISPIEL 2:  
.. x=F  
" AR02 km SA " ..  
(Stop + Anzeige M02 km, wenn x=F).

i) BLANK (Code -)  
Ein BLANK (Leerspeicher) wird im Programmspeicher als Programmende interpretiert. Etwas Schritte hinter dem BLANK werden im RUN-Modus nicht und im WRT-Modus erst nach Auffuellen des bzw. der BLANKS mit Fullsteps erreicht.

j) PROGRAMMSTART-LABELS Pi (Code 0i)  
Mit dem SYNTHESE-Programm koennen gleiche Pi auch mehrfach erzeugt werden. Dies kann auch unbeabsichtigt ueber die nicht eindeutig festlegbaren Codezahlen S2 u. S7 (vgl. Ziff.5) erfolgen.  
In diesem Fall wird im RUN- und im WRT-Modus nur das vorderste Pi im Programmspeicher erreicht.  
Beim Loeschen mit MODE3 Pi AC wird nur das jeweils vorderste Pi geloescht.

9) ZEICHENBEFEHLE

Zeichenbefehle sind Anweisungen, die im ALPHA-Modus bestimmte Operationen ausfuehren, wie:  
ARi [ ] [s]  
Sie sind im ALPHA-Teil der CODETABELLE durch Klammern gekennzeichnet.

a) 12 weitere ARi  
Die Zeichenbefehle: GRA DSZ x=0  
x=F RAN# x  
x° x°° P+R  
R+P % invEXE  
liesen in den AR0i- und AR1i-Reihen (Code 6i und "i) der CODE-TABELLE. Sie haben die Bedeutung von AR-Anweisungen (direkte Inhaltsanzeige) fuer Speicher mit sedezimaler Adresse (vgl. Ziff.3).  
Beim Programmausdruck im PCL-Modus werden sie als ARi mit besonderem i gekennzeichnet:

TABELLE 2:  
SONDER-AR-BEFEHLE  
Anzeige im WRT-Modus  
Druck im PCL-Modus  
(AR= sedez. Adresse)

CODE	WRT	PCL	AD
6.	GRA	ARF	.
6°	DSZ	ARX	'
6'	x=0	ARD	'
6°	x=F	ARL	°
6-	RAN#	AR-	-
6w	x	ARE	w
"	x°	AR1F	1.
"	x°°	AR1X	1°
"	R+P	AR10	1°
"	P+R	AR1L	1°
"	%	AR1-	1-
"	invEXE	AR1E	1w

Durch ein vorangestelltes F1 bis F4 (Ziff.8b) koennen auch AR-Befehle

\*\*\*\*\*  
 \* SEDEZIMAL-CODE \*  
 \*\*\*\*\*

fehlt fuer die hoehwertigen sede-  
 zimalen Speicheradres-  
 sen erzeugt werden:

TABELLE 3:  
 Sedezimale Adressen  
 d. SONDER-AR-BEFEHLE

AR-BEF	F1	F2	F3	F4
GRA	2	4	6	8
DSZ	2	4	6	8
x=0	2	4	6	8
x=F	2	4	6	8
RAM#	2	4	6	8
x	2	4	6	8
x <sup>y</sup>	1	3	5	7
x <sup>y</sup> y	1	3	5	7
R+P	1	3	5	7
P+R	1	3	5	7
%	1	3	5	7
INVEXE	3	5	7	9

Setzt man in Tab.3  
 die Speicheradressen  
 aus Tab.1 (Ziff.3)  
 ein, so ergibt sich  
 folgende Zuordnung:

TABELLE 4:  
 Interne Adressen der  
 SONDER-AR-BEFEHLE

AR-BEF	F1	F2	F3	F4
GRA	F	2F	4F	6F
DSZ	L1	20	35	55
x=0	L2	20	36	56
x=F	L3	20	37	57
RAM#	L4	20	38	58
x	L5	20	39	59
x <sup>y</sup>	1F	3F	5F	7F
x <sup>y</sup> y	L6	25	45	65
R+P	L7	26	46	66
P+R	L8	27	47	67
%	L9	28	48	68
EXE	L10	29	49	69

Neu ist also, dass  
 der Inhalt der Klam-  
 merregister Li di-  
 rekt angezeigt wer-  
 den kann.  
 Mit dem zur Adresse  
 XX in MODE.XX gehoe-  
 renden Zeichenbefehl  
 kann auch der Inhalt  
 des Programmspeicher-  
 endes direkt ange-  
 zeigt werden (vgl.  
 Ziff.3c-Ergebnis).

b) Mx F & Mx I F  
 (Code 0-9)  
 Die Zeichenbefehle  
 " Mx F " & " Mx I F "  
 werden alle wie  
 " AR F " & " AR I F "  
 interpretiert.

c) AC NOP 10<sup>x</sup> e<sup>x</sup>  
 - AC (Code 0-9)  
 Bei Erreichen von  
 " .. AC .. "  
 werden der Zeichen-  
 Pufferspeicher und  
 die Operationspei-  
 cher gelöscht.  
 - NOP (Code 0-9)  
 No Operation  
 - 10<sup>x</sup> (Code 0-9)  
 Bei Erreichen von  
 " .. 10<sup>x</sup> .. "  
 wird der Zeichen-  
 Pufferspeicher  
 gelöscht.  
 Befehl kann nur im  
 PCL-Modus gelöscht  
 werden.

- e<sup>x</sup> (Code 0-9)  
 Bei Erreichen von  
 " .. e<sup>x</sup> .. "  
 blinkt Anzeige kurz  
 auf. Ist eine be-  
 liebige Taste ge-  
 drueckt, so haelt  
 Programm an (Anzei-  
 ge wie bei BST).

d) SA  
 (CODE 0-9)  
 Bei Erreichen von  
 " .. SA X .. "  
 haelt Programm nach  
 dem Schritt X im  
 ALPHA-Modus an.  
 ANWENDUNG:  
 Direkt eingebbarer  
 Kennungsdruck.  
 BEISPIEL:  
 - "Name SA : "  
 SAVE INVEXE ...  
 - ANZEIGE: Name:  
 - EINGABE: Maier  
 EXE  
 - DRUCK: Name:Maier  
 (s. auch Beispiel 2  
 unter Ziff.8h).

10. PASSWORT

a) Pi pass  
 (Code -)  
 Der Befehl  
 Pi pass-----  
 besitzt vier freie  
 Stellen, die mit je  
 einem ALPHA-Zeichen  
 belegt werden koen-  
 nen, die zusammen  
 das Passwort (Lo-  
 sungswort) bilden.  
 Die Angabe Pi stimmt  
 dabei mit dem davor-  
 stehend am naechsten  
 kommenden Program-  
 start-Label Pi  
 ueberein. Auch die  
 Zahlenweisungen  
 0 1 2 3 . EXP  
 werden hier als (se-  
 dezimale) Pi sedeu-  
 tet (vgl. Ziff.8d).  
 Die Belegung der  
 Passwortstellen er-  
 folgt durch die vier  
 BEFEHLE, die unmit-  
 telbar hinter dem  
 benachbarten Pi ste-  
 hen. Sie werden auf  
 den Passwortstellen  
 als codesgleiche  
 ALPHA-ZEICHEN inter-  
 pretiert.

Eine Programmverrie-  
 gelung geht nur bei  
 den dezimalen P0-P9.  
 Und nur, wenn Pipass  
 auf dem 5.Step hin-  
 ter Pi folgt.

BEISPIEL:  
 Erzeugen Sie mit dem  
 SYNTHESE-Programm  
 zunaechst den Befehl  
 Pi pass (0=42)  
 und sodann im  
 WRT-Modus folgende  
 Schrittfolgen:

- P1 453
- P<sup>0</sup>pass-----
- P1 453 GOTO3 GOTO2  
 GOTO1 GOTO0  
 P<sup>0</sup>passABCD
- P1 45 GOTO3 GOTO2  
 GOTO1 GOTO0  
 P<sup>1</sup>passCD1k
- P1 GOTO3 GOTO2  
 GOTO1 GOTO0  
 P<sup>1</sup>passABCD
- P1 druecken:  
 P<sup>1</sup>pass-----?
- ABCD einseben:  
 P<sup>1</sup>passABCD

b) LOAD-507-TRICK  
 PASSWORT-KNACKEN 1

Bei einem verriegel-  
 ten Programm steht  
 der Pi pass-Befehl  
 immer an 5.Stelle  
 hinter dem Pi am An-  
 fang eines Programms.  
 Zum Entschlüsseln  
 eines unbekanntem  
 Passworts muss der  
 Pi pass-Befehl unter  
 Erhaltung der davor-  
 stehenden codeslei-  
 chen Befehle elimi-  
 niert werden.  
 Hierzu gehen Sie  
 zweckmaessig folgen-  
 dermassen vor:

- MODE.20
- Loeschen aller  
 nicht-verriegelter  
 Programme mit  
 MODE3 Pi AC
- Abspeichern auf  
 Band mit  
 MODE3 SAVE EXE
- Loeschen mit  
 MODE3 MAC
- Einlesen des zuvor  
 abgespeicherten  
 Programms vom Band  
 MODE3 LOAD EXE
- Sobald Ladeanzeige  
 PF 000  
 erscheint, sofort  
 unterbrechen mit  
 AC-Taste
- Die Schrittzahlan-  
 zeige muss jetzt  
 auf 507 stehen!
- Ist dies nicht der  
 Fall, muss nach  
 MODE3 MAC  
 erneut geladen und  
 abgestoppt werden
- Die vier Befehle,  
 die jetzt im WRT-  
 Modus eingesehen  
 werden koennen,  
 sind anhand der  
 CODE-TABELLE in  
 die codesgleichen  
 Zeichen umzuwan-  
 deln. Diese bilden  
 von hinten nach  
 vorn gelesen das  
 PASSWORT
- Sind mehrere Pro-  
 gramme verriegelt,  
 kann das jeweils  
 entschlueselte  
 Programm nach Ein-  
 gabe des Passworts  
 mit MODE3 Pi AC  
 geloescht werden.

c) PASSWORT aus  
 SONDERZEICHEN

Ein verriegeltes Pro-  
 gramm wird ueblicher-  
 weise dadurch entrie-  
 gelt, dass ueber die  
 TASTATUR das rich-  
 tige Passwort einge-  
 geben wird.

Wenn zur Bildung des  
 Passworts SONDERZEI-  
 CHEN verwendet  
 werden, kann es zwar  
 entschlueselt wer-  
 den. Eine Entriege-  
 lung ist aber trotz-  
 dem nicht moeglich,  
 weil die Sonderzei-  
 chen auf der Tasta-  
 tur nicht vorhanden  
 sind.

Dazu gehen Sie zweck-  
 maessig wie folgt  
 vor:

- Erzeugung eines  
 Pi pass -Befehls  
 mit dem SYNTHESE-  
 Programm
- Vor Pi pass vier  
 Befehle setzen,  
 deren codesgleiche  
 Zeichen nicht auf  
 der Tastatur sind
- Loeschen der vor  
 Pi pass stehenden  
 Fuellschritte, so  
 dass zwischen Pi  
 und Pi pass noch  
 5 Schritte stehen
- Das zu verriegeln-  
 de Programm hinter  
 Pi pass schreiben
- Den letzten Fuell-  
 schritt loeschen
- Pi druecken  
 Damit kann das Pro-  
 gramm ueber die Ta-  
 statur nicht mehr  
 entriegelt werden.

d) SINGLE-ALPHA-TRICK  
 PASSWORT-KNACKEN 2  
 - In beliebig. Px ein  
 singulaeres ALPHA  
 (Code 2<sup>0</sup>) erzeugen  
 (s.Ziff.6d)  
 - Danach! verschlues-  
 seltes Py von Band  
 einlesen  
 - Dann: MODE1  
 GSBPx FST FST ...  
 - ANZEIGE: Passwort  
 von Py -rueckwaerts

\*\*\*\*\*  
 \* SEDEZIMAL-CODE \*  
 \*\*\*\*\*

CASIO

Seite 5

E. Welker  
 Steinpilzweg 18  
 7 STUTTGART 70

HEXCODE - TABELLE

\*\*\*\*\*  
 \* CASIO FX-601P \*  
 \* CASIO FX-602P \*  
 \* CASIO FP-10 \*  
 \*\*\*\*\*

B E F E H L  
 ' A L P H A '  
 Anzeige Druck

S <sub>0</sub>	0 <sub>6</sub>	1 <sub>7</sub>	2 <sub>8</sub>	3 <sub>9</sub>	4	5	6	7	8	9	A <sub>0</sub>	B <sub>11</sub>	C <sub>12</sub>	D <sub>0</sub>	E <sub>-</sub>	F <sub>9</sub>	S <sub>0</sub>	
0 <sub>6</sub>	P0 P1 P2 P3 P4 P5 P6 P7 P8 P9	P1 P2 P3 P4 P5 P6 P7 P8 P9	P2 P3 P4 P5 P6 P7 P8 P9	P3 P4 P5 P6 P7 P8 P9	P4 P5 P6 P7 P8 P9	P5 P6 P7 P8 P9	P6 P7 P8 P9	P7 P8 P9	P8 P9	P9	0 1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	2 3 4 5 6 7 8 9	3 4 5 6 7 8 9	4 5 6 7 8 9	EXP L	0	0
1 <sub>7</sub>	RND0 a	RND1 b	RND2 c	RND3 d	RND4 e	RND5 f	RND6 g	RND7 h	RND8 i	RND9 j	4 k	5 l	6 m	7 n	8 o	9 p	1	1
2 <sub>8</sub>	LBL0 q	LBL1 r	LBL2 s	LBL3 t	LBL4 u	LBL5 v	LBL6 w	LBL7 x	LBL8 y	LBL9 z	HLT *	"	F1 (F1x) (+20)	F2 (+40)	F3 (+60)	F4 (+80)	2	2
3 <sub>9</sub>	GOTO0 A	GOTO1 B	GOTO2 C	GOTO3 D	GOTO4 E	GOTO5 F	GOTO6 G	GOTO7 H	GOTO8 I	GOTO9 J	XD K	(SAC) L	ENG M	o11 N	log O	Ln P	3	3
4	GSBP0 Q	GSBP1 R	GSBP2 S	GSBP3 T	GSBP4 U	GSBP5 V	GSBP6 W	GSBP7 X	GSBP8 Y	GSBP9 Z	+/- X	( ) ÷	sin +	cos -	tan =	tan L	4	4
5	X+00 :	X+01 :	X+02 ?	X+03 	X+04 N	X+05 <	X+06 >	X+07 Σ	X+08 ( )	X+09 ,	π	E	□	□	□	□	5	5
6	Min00 (AR00)	Min01 (AR01)	Min02 (AR02)	Min03 (AR03)	Min04 (AR04)	Min05 (AR05)	Min06 (AR06)	Min07 (AR07)	Min08 (AR08)	Min09 (AR09)	(GRA) (GRA)	DSZ (DSZ)	x=0 (x=0)	x=F (x=F)	RAN# (RAN#)	π (π)	6	6
7	MRO0 0	MRO1 1	MRO2 2	MRO3 3	MRO4 4	MRO5 5	MRO6 6	MRO7 7	MRO8 8	MRO9 9	ISZ .	x≥0 "	x≥F 1	σ <sub>n</sub> 0	σ <sub>n-1</sub> -	σ <sub>n-1</sub> L	7	7
8	M-00 :	M-01 :	M-02 :	M-03 :	M-04 :	M-05 :	M-06 :	M-07 :	M-08 :	M-09 :	PAUSE p	IND 10	SAVE L	LOAD in	MAC .	SAC L	8	8
9	M+00 a	M+01 b	M+02 c	M+03 d	M+04 e	M+05 f	M+06 g	M+07 h	M+08 i	M+09 j	XDEL k	(SAC) L	ENG m	o11 n	10 <sup>x</sup> o	e <sup>x</sup> p	9	9
A <sub>0</sub>	X-M10 q	X-M11 r	X-M12 s	X-M13 t	X-M14 u	X-M15 v	X-M16 w	X-M17 x	X-M18 y	X-M19 z	ABS *	INT /	FRAC →	sin <sup>-1</sup> ←	cos <sup>-1</sup> %	tan <sup>-1</sup> ←	A <sub>0</sub>	4
B <sub>11</sub>	Min10 (AR10)	Min11 (AR11)	Min12 (AR12)	Min13 (AR13)	Min14 (AR14)	Min15 (AR15)	Min16 (AR16)	Min17 (AR17)	Min18 (AR18)	Min19 (AR19)	x <sup>y</sup> (x <sup>y</sup> )	x <sup>1/y</sup> (x <sup>1/y</sup> )	R→P (R→P)	P→R (P→R)	% (%)	invEXE (invEXE)	B <sub>11</sub>	5
C <sub>12</sub>	MRI0 Q	MRI1 R	MRI2 S	MRI3 T	MRI4 U	MRI5 V	MRI6 W	MRI7 X	MRI8 Y	MRI9 Z	GSB(0) X	X↔Y ÷	√ +	x <sup>2</sup> -	1/x =	x! L	C <sub>12</sub>	6
D <sub>0</sub>	M-10 :	M-11 :	M-12 ?	M-13 !	M-14 N	M-15 <	M-16 >	M-17 Σ	M-18 ( )	M-19 ,	DEG π	RAD π	GRA π	sinh ;	cosh #	tanh L	D <sub>0</sub>	7
E <sub>-</sub>	M+10 A	M+11 R	M+12 √*	M+13 %/	M+14 ≥	M+15 σ	M+16 x!	M+17 2 <sup>u</sup>	M+18 -1)	M+19 -1)	Error y &	Error d	Pipass !	sinh <sup>-1</sup> x ?	cosh <sup>-1</sup> 1/2	tanh <sup>-1</sup> L	E <sub>-</sub>	8
F <sub>9</sub>	X-MF (X-MF)	MinF (ARF)	MRF (MRF)	M-F (M-F)	M+F (M+F)	X-MIF (X-MIF)	MinIF (ARIF)	MRF (MRF)	M-IF (M-IF)	M+IF (M+IF)	AC (AC)	NOP (NOP)	(10 <sup>x</sup> ) (10 <sup>x</sup> )	(e <sup>x</sup> ) (e <sup>x</sup> )	SA (SA)	(BLANK) :	F <sub>9</sub>	9
/	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	S	