
* SEDEZIMAL-CODE *
* 2. TEIL *
* (C) 1982 Welker *

Seit der Veröffentlichung meiner 1. Ausarbeitung ueber den Hex-Code des FX-602P haben sich weitere Erkenntnisse ergeben, die den Einblick in die interne Speicherstruktur vertiefen und neue Tricks bei der Handhabung des Rechners ermöglichen.

Viele Anregungen erhielt ich bei einem resen Erfahrungsaustausch mit 602-Fans. Insbesondere moechte ich mich hierfür bei den Herren KOPPENHÜFER und STESSUN bedanken.

Die neuen Erkenntnisse sind im vorliegenden 2. Teil zusammengestellt. Der Uebersichtlichkeit halber wurde die Kapitelnummerierung des ersten Teils uebernommen und ersaenzt.

1. AUSGANGSPUNKT

Das SYNTHESPROGRAMM aus Kap. 5 ermöglicht das synthetische Programmieren, d.h. die Erzeugung von Befehlen und Zeichen unter Anwendung des internen Hexcodes. Damit lassen sich auch solche SONDERBEFEHLE und -ZEICHEN in Programme einfügen, die sonst mit der Tastatur nicht eingesegeben werden koennen.

TABELLE:
Interne Darstellungen der HEX-Ziffern

0123456789ABCDEF
0123456789.°°°

2. SEDEZIMALZAHLEN

a) GRUNDZIFFER
Das einfachste Programm zur Erzeugung einer sedezimalen Grundziffer enthaelt nur 2 ALPHA-Befehle:

Pi °°

Durchfuehrung:
MODE1
VORGABE: a
(=beliebige Zahl)
EINGABE: GSBPi
invALPHA
inv# inv#
FST FST FST

ANZEIGE: 'a (= Ca)
INHALT MF: 'a!
(fuer bestimmte a gibt es Ausnahmen!)

BEISPIELE:

a = 0 → '0 (C0)
a = '0 → '0 (C0)
a = 1 → '1 (C1)
a = . → . (CA)
aber:
a = °6 → °6 (886)

b) RECHENREGELN

'0 ÷ 10 = ' (C)
'1 ÷ '1 = . (A)
'0 ÷ 10 = °6 (86)

Der Dezimalpunkt wurde jeweils weglassen.

3. INDIREKTE SPEICHERADRESSIERUNG

a) KLAMMERREGISTER
Die Klammerregister werden beim Laden von Datenbloeken vom Band mit LOAD EXE nicht wie die uebrigen Speicher ueberschrieben.

ANWENDUNG:
(fuer Interface mit Fernsteuerung: FA-2)
Bei programmgesteuerter Datenblock-Eingabe vom Band koennen Variable in Klammerregister zwischenspeichert werden.

Dazu muss der Datenblock lediglich die sedezimale Klammerregister-Adresse oder eine Grundziffer fuer deren Berechnung enthalten. Damit kann man z.B. auch eine automatische Dateisuche programmieren:

Vergleich von Dateinummern (Pointern) in einem Speicher (z.B. MF) der Datenfiles und einem Klammerregister.

b) F-REGISTER

MUSIC-Betrieb
Die Sedezimalziffern ergeben in MF sechs weitere Tempustufen, schneller als die Stufe 9. Weitere Tonhoehen koennen aber nicht erzeugt werden.

d) PROGRAMMSPEICHER

Im 1. Teil heisst es dass die auf Pos. 2, 7, 8 des Programmspeichers gelangenden Codezahlen nicht eindeutig festlesbar seien (vgl. Kap. 5). Dem wollen wir weiter nachsehen.

Geben Sie ein:
MODE3 MAC MODE.79
MODE2
PI: 4444 4444
ganz auffuellen bis zur Pos. 39
MODE1
MO1 = 8..

Speichern Sie ab:
2.132435465 E67
IND Min01
Sehen Sie P1 im WRT-Modus von hinten heran:

Adr	Pos	Befehl	Code
039	1	4	1.
038	2	RND2	12
037	3	RND3	13
036	4	LBL4	24
035	5	GOTO5	25
034	6	GSBP6	46
033	7	X-M01	50
032	8	Min07	67
031	9	4	1.

In den unterstrichenen Codestellen erkennen wir die abgespeicherte Zahl. Die Vorzeichen-Kennziffer ist einsekreist.

-Pos.1:
1. = Fuellstep 4 bleibt erhalten.
-Pos.2 Zehnerstelle:
1 =Zehnerstelle von Fuellstep (1.) bleibt erhalten.
NICHTIG:

Bei Fuellsteps aus der 0-Reihe der Hexcode-Tabelle, also bei 0 1 2 3. EXP. wird in Pos.2 ein Pi erzeugt. Deshalb Fuellsteps aus anderen Hexcode-Reihen wahlen, zB 4!

-Pos.2 Einerstelle:
Vorkommastelle Mantissee.
-Pos.3-6:
8 Nachkommastellen Mantissee.
-Pos.7 Zehnerstelle:
9. Nachkommastelle Mantissee.
-Pos.7 Einerstelle:
Vorzeichenkennziffer (s. unten).
-Pos.8:
Exponent (s.u.).
-Pos.9 ..
Fuellsteps bleiben erhalten.

VORZEICHENKENNZIFFER
Je nach Vorzeichen von Mantissee (M) und Exponent (E) der abgespeicherten Zahl aendert sich in unserem Beispiel Pos.7 wie folgt:

M	E	Pos	Befehl	Code
+	+	7	X-M01	51
-	+	7	X-M06	56
+	-	7	X-M08	58
-	-	7	X-M05	55

TABELLE:
Vorzeichenkennziffer

	MANTISSE	
	+	-
E +	1	6
X		
P -	0	5

EXPONENT
Je nach Vorzeichen des Exponenten aendert sich Pos.8 wie folgt:

E	Pos	Befehl	Code
+	8	Min07	67
-	8	GOTO3	33

Bei negativen EXP-Vorzeichen ergibt sich als Codezahl des Exponenten also das 100er-Komplement:

$$100 - 67 = 33$$

All dies gilt, wenn Daten abgespeichert werden. Wir haben damit die Struktur der Datenspeicherung kennengelernt.

DATENABRUF aus PROGRAMMSPEICHER
Durch Einschreiben von Befehlen im WRT-Modus koennen auf die Einerstelle der Pos. 7 des Programmspeichers ausser den Kennziffern 0 1 5 6 auch alle anderen Ziffern gelangen. Diese werden beim Abruf mit IND MR.. vorzeichenmaessig wie folgt interpretiert:

TABELLE: Abruf-Vorzeichenkennziffer

Ziffer	Mant.	Exp
0	+	-
1	+	+
2	Error	
3	0	
4	0	
5	-	-
6	-	+
7	Error	
8	0	
9	0	
.(A)	+	-
.(B)	+	+
.(C)	Error	
.(D)	0	
-(E)	0	
-(F)	-	-

 * SEDEZIMAL-CODE *
 * 2. TEIL *

EXPONENT bei DATEN-ABRUF aus P-SPEICHER

Im Exponenten einer Zahl koennen nur 2-stellige DEZIMALZAHLEN mit positivem o. negativem Vorzeichen stehen.

Steht auf Pos.8 des P-Speichers ein Befehl mit beliebiger HEXCODEZAHLE, so wird diese beim Abruf mit IND MR.. im Exponent automatisch in die zuehoerige DEZIMALZAHLE umgewandelt. Bei NEGATIVEN Exponenten (Vorzeichenkennziffer Pos.7 !) wird zusaetzlich 100 subtrahiert. Exponenten ueber +100 ergeben beim Abruf Error-Anzeige.

BEISPIELE

Pos.8 Code	E
4 1. →	10+10= 20
E negativ	-100=-80
log 3-	30+14= 44
E negativ	-100=-56
SAVE 8'	80+12= 92
E negativ	-100=-88
ENG 9'	90+12=102
E positiv	Error
E negativ	-100= 82
COSH 0-	130+14=144
E positiv	Error
E negativ	-100= 44
Blank ~	150+15=165
E positiv	Error
E negativ	-100= 65

Abruf aus leeren Programmspeicher ergibt also die mysterioese Zahl - . e65 !

Abruf bei Fuellsteps 4444..444 ergibt: ...1.1.1.1.1 E-80 (In einer Teilaufgabe des 1.Teils ist auf S.1, rechts oben diese Zahl versehentlich mit e21 angedruekt).

e)PROGRAMMSPEICHER 2

In Kap.3d wurde gezeigt, wie man durch geeignete Wahl einer Sedezimaladresse das ENDE des P-Speichers indirekt adressieren kann.

Auf aehnliche Weise kann man auch noch zu einer anderen Stelle im P-Speicher gelangen. Dies geht zB mit der Sedezimaladresse '. (CA).

Setzen Sie: MF = '. (vgl. Kap.2a+b)
 Geben Sie ein: MODE3 MAC MODE.20
 MODE2
 P1 bis ADR.50 (Fuellsteps aus 0-Reihe der HEXCODE-Tabelle !)
 Speichern Sie ab: 2.132435465 E67
 IND MinF

Sehen Sie P1 und P0 im WRT-Modus an:

*** P1
 0n RND6
 X-M00 RAN#
 X-M04 los
 LBL4
 los
 LBL1

 ...047steps
 *** P0
 ...
 ...004steps

Die Abspeicherung erfolgt hier also mit Luecken auf folgende P-Speicher-Adressen:

ADR	Pos	Befehl	Code
047	1	P0	00
046	2	.	0-
039	3	LBL1	21
038	4	los	3-
031	5	LBL4	24
030	6	los	3-
023	7	X-M04	54
022	8	RAN#	6-
015	9	X-M00	50
014	10	.	0-
007	11	RND6	06
006	12	0n	1-

In den angestrichelten Codestellen erkennen wir wieder die abgespeicherte Zahl und die Vorzeichenkennziffer.

Pos.1:
 Auf Adresse 47 gelangt immer ein P0 (Code 00).
 ANWENDUNG: Programm-Entriegelung: siehe unten Kap.11.

Pos.2 Zehnerstelle: wird der abgespeicherten Zahl als VORKOMMNSTELLE zugeschlagen und wird dann selbst auf Codenziffer 0 gesetzt. (Dieser Effekt ist nur erkennbar, wenn Fuellstep auf ADR.46 nicht aus der 0-Reihe ist - s. unten!)

Pos.2,4,6,8,10,12 Einerstelle (-): = Einerstelle der Fuellsteps (0-), bleibt erhalten.

Pos.3 Zehnerstelle: Vorkommastelle der Mantisse.

Pos.3 Einerstelle + Pos.5,7,9 + Pos.4,6,8 Zehnerst.: Nachkommastellen d. Mantisse.

Pos.10 Zehnerstelle: 11. Nachkommastelle der Mantisse, wird beim Abspeichern immer 0 gesetzt.

Pos.11 Zehnerstelle: Vorzeichenkennziffer
 Pos.11 Einerstelle + Pos.12 Zehnerstelle: Exponent.

VORKOMMNSTELLE
 Waehlt man den Fuellstep auf Pos.2 (ADR. 46) nicht aus der 0-Reihe (0 1 2 3 . EXP) der HEXCODE-Tabelle, sondern aus einer der Reihen 1 - 9 (zB sin - Code 4°), so wird beim Abspeichern die Zehnerstelle von Pos.2 intern auf die Zehnerstelle

von Pos.3 quasi als neue Vorkommastelle vor die abgespeicherte Zahl gesetzt. Zusiech wird die Mantisse um 1 Stelle nach rechts verschoben und der Exponent um 1 erhoehrt. Die so gebildete neue Zahl wird nach dem indirekten Abspeichern auf '. (CA) auch im Display angezeigt.

BEISPIEL:
 FUELLSTEPS: wie vor, aber ADR.46: sin (Code 4°)

EINGABE:
 2.132435465 E67
 IND MinF
 ANZEIGE:
 4.213243547 E68
 ABGESPEICHERT:

ADR	Pos	Befehl	Code
047	1	P0	00
046	2	.	0°
039	3	GSBP2	42 !
038	4	8	1-
031	5	GOTO2	32
030	6	cos	4-
023	7	GOTO5	35
022	8	cos	4-
015	9	NR00	70
014	10	.	0-
007	11	RND6	06
006	12	MAC	8-

Waehlt man den Fuellstep auf Pos.2 (ADR. 46) aus den Code-Reihen . bis ~ (zB sinh - Code 0°), so wird die sedezimale Zehnerstelle beim Abspeichern in die zuehoerige 2-stellige Dezimalzahl umgewandelt (zB 0°=13) und vor die Mantisse gesetzt.

BEISPIEL:
 FUELLSTEPS: wie vor, aber ADR.46: sinh (Code 0°)

EINGABE:
 2.132435465 E67
 IND MinF
 ANZEIGE:
 1.3213243547 E69
 ABGESPEICHERT:

ADR	Pos	Befehl	Code
047	1	P0	00
046	2	.	0°
039	3	RND3	13 !
038	4	F3	2-
031	5	RND3	13
030	6	F3	2-
023	7	GSBP3	43
022	8	=	5-
015	9	GSBP7	47
014	10	.	0-
007	11	RND6	16
006	12	10°	9-

Die Befehle F3 RND3 in der vorstehenden Tabelle erscheinen im WRT-Modus als P-pass... (unechter Passbefehl, Kap.8a) und werden im PCL-Modus als FIX3 ausgedruekt.

DATENABRUF aus PROGRAMMSPEICHER 2
 Der indirekte Datenabruf ueber Adresse '. (CA) funktioniert nur, wenn die Zehnerstelle der Codenzahl in Pos.2 (ADR 46) = 0 ist. Sonst erscheint Betriebsanzeige, die nur durch Aus- und Einschalten geloescht werden kann.

Beim Datenabruf werden auch die Codenzahlen der Einerstelle von Pos.9 und der Zehnerstelle von Pos.10 als 11. und 12. Mantissenstelle ins Anzeigenregister geholt (aber nicht angezeigt). Bekanntlich rechnet der Rechner mit 12 Mantissenstellen, aber nur 10 Stellen werden angezeigt.

ZUSAMMENFASSUNG
 Die vorstehenden Untersuchungen waren noetig, um die komplexen Vorgaenge bei der Datenspeicherung zu verstehen. Jetzt ist es moeglich nach sinnvollen Anwendungen zu suchen. Einige davon finden Sie in Kap.11.

 * SEDEZIMAL-CODE *
 * 2. TEIL *

**4. HEXCODE-TABELLE
 (HEX-SCHLUESSEL)**

Die beiliegende HEX-
 CODE-Tabelle wurde
 ersaenzt:

-Die Randreihen und
 -spalten enthalten
 neben den Sedezimal-
 ziffern in der uebli-
 chen und in der
 rechnerinternen Dar-
 stellung auch die
 zugeordneten Dezi-
 mal-Einsabziffern
 D bzw. D' fuer das
 Synthesprogramm.
 -Die ueber die Tas-
 tatur eingebbaren
 Zeichen sind durch
 senkrechte Balken
 markiert.

Zur HEXCODE-Tabelle
 fuer Passwort-Zei-
 chen (PASS-CODE):
 siehe Kap.10.

5. SYNTHESE-PROGRAMM

Das Synthesprogramm
 lasst sich wie folgt
 vereinfachen:

*** P0
 MR02 ÷ MRF = Min00 -
 MR03 = +/- - 3.
 HLT EXP 14 = IND
 Min01
 HLT
 ...021steps

In einer Teilaufgabe
 des 1. Teils wurden
 unter Kap.5 als Fuell-
 steps in P1 bzw. P2
 die Zahlenweisungen
 0 (0-Reihe der HEX-
 CODE-Tabelle) vorge-
 schlagen. Dies fuhrt
 zur Erzeugung eines
 unerwunschten Pi
 auf Pos.2 des Pro-
 grammspeicherendes
 (vgl 1. Teil 8j).
 Um dies zu vermeiden
 sollten die Fuell-
 steps aus einer an-

deren HEXCODE-Reihe
 gewaehlt werden (vgl.
 2. Teil 3d), z.B.:

*** P1
 44444444444444444444
 44444444444444444444
 4444444444
 ...051steps

Es ist auch moeslich,
 die Abspeicherung an
 das Programmspeicher-
 ende vor dem Auffuel-
 len mit Fuellsteps
 vorzunehmen. Wenn an-
 schliessend ein be-
 liebiges (freies) Px
 mit Fuellsteps voll-
 getastet wird, ste-
 hen dahinter die syn-
 thetisierten Steps.
 Die leeren Program-
 mspeicherstellen ent-
 halten BLANKS (Code_{hex}
 = FF - Kap.8i).

**6. SYNTHETISCHES
 PROGRAMMIEREN**

Mit den 2 Varianten
 des Synthesprogramms
 (mit und ohne +/-)
 lassen sich zwar die
 Ziffern 4 und 5 so-
 wie die Zweierkombi-
 nationen aus 0123
 einerseits und 6789
 andererseits in den
 abzuspeichernden Hex-
 code-Zahlen nicht
 erzeugen.
 Wie die HEXCODE-Ta-
 belle zeigt, koennen
 jedoch alle Befehle
 aus diesem Code-Ber-
 eich ueber die Tas-
 tatur eingegeben
 werden. Mit der im
 1. Teil unter Kap.6d
 beschriebenen Metho-
 de lassen sich daher
 auch die zuehoerigen
 Sonderzeichen gene-
 rieren.

7. SONDERZEICHEN

d) Sonder-SPACE
 Code 8_{hex} (8F)
 Kann direkt mit Tas-
 tatur eingegeben
 werden:
 "MODE invEXE FST"
 Wirkung:
 entspricht dem nor-
 malen SPACE (invEXE
 = Code 7_{hex}), fuehrt
 aber nicht zu dem
 kurzzeitigen Anhal-
 ten der Rollanzeige
 Anwendung:
 Beschleunigung des
 Ausdrucks von Funk-
 tionsgraphen und
 Tabellen mit vielen
 SPACES.

e) SONDER-SPACE 2
 Code 5_{hex} (5F)
 Anzeige: Space
 DRUCK: nichts
 (kein Space)

8. SONDERBEFEHLE

a) PREFIX F1, F2, F3, F4
 Code 2' 2° 2- 2_{hex}
 = 2C 2D 2E 2F
 Die Bezeichnungen F1
 bis F4 sind willkuer-
 lich. Sie wurden von
 FX-502P uebernommen.
 Es handelt sich um
 sog. PREFIXE von 2-
 (oder Mehr-) -Byte-
 befehlen.
 Als POSTFIXE kommen
 in Betracht:

-RNDn + FIXn
 (1. Teil 8a)
 Dies gilt auch fuer
 die F2 - F4.
 Letztere werden im
 WRT-Modus als
 Pi-Pass___ ange-
 zeigt und im PCL-
 Modus als FIXn aus-
 gedruckt.
 Diese uechten Pass-
 Befehle koennen aber
 nicht verriegelt
 werden. Sie sind im
 WRT-Modus mit der
 C-Taste loeschbar.

-Speicherbefehle
 + hoeherwertige
 Speicheradressen
 (1. Teil 8b + 9a)
 -alle uebrigen Befeh-
 le + ohne Wirkung.
 Da die F1 - F4 auch
 die Bedeutung von
 LBLi mit sedezima-
 ler Kennziffer i
 haben, koennen sie
 als Prefix vor je-
 der Anweisung indi-
 rekt mit IND GOTOj
 angesprungen werden.
 (s. 1. Teil 8c)

b) MEHRBYTE-BEFEHLE
 Vor einer Anweisung
 koennen auch mehrere
 Fi-Prefixe stehen.
 In Verbindung mit
 RNDn und den Spei-
 cherbefehlen wirkt
 nur das unmittelbar
 davorstehende Fi.
 Beim Loeschen mit der
 C-Taste wird nur das
 Postfix mit dem un-
 mittelbar davorste-
 henden Fi geloescht.
 Die weiteren Fi wer-
 den dem folgenden
 Befehl als Prefix
 zugeschlagen.
 Alle Fi eines Mehr-
 bytebefehls koennen
 getrennt mit IND
 GOTOj angesprungen
 werden.

f) 10^x Code w' (FC)
 Synthetischer PASS-
 Befehl mit Verrie-
 gelungsmoeglichkeit
 siehe Kap.12 !

f') e^x Code w° (FD)
 Das Programm haelt
 bei Erreichen die-
 ses Befehls nur an,
 wenn er von einem
 unmittelbar davor-
 stehenden Sprungsbe-
 fehl (ISZ, DSZ, x=0,
 x=0, x=F, x=F) ange-
 sprungen wird.
 Sonst blinkt Anzei-
 ge kurz auf und Pro-
 gramm laeuft weiter.
 Ist eine Taste ge-
 drueckt, so haelt
 Programm an.

g) Error Code - =EA
 Code - =EB
 Die im WRT-Modus
 neben der Error-An-
 zeige stehende drei-
 stellige Zahl Z er-
 rechnet sich wie
 folgt aus der abso-
 luten Adresse A des
 Error-Steps im Pro-
 grammspeicher (Pi
 mitzesaehlt, bei 0
 beginnend):
 Z = A + 792

h) SA Code w- (FE)
 Dieser Sonderbefehl
 entspricht in sei-
 ner Wirkung der FST
 -Taste (= program-
 mierter FST).

k) Pipass
 Code -' (EC)
 Pipass hat als un-
 abhaengiger Befehl
 (ohne Verriegelung)
 im Run-Modus die Wir-
 kung:

$$\log(x+\sqrt{x^2+1})$$

Diese Funktion erin-
 nert an die Formel:

$$\sinh^{-1}x = \ln(x+\sqrt{x^2+1})$$

$$= \ln10 \times \log(x+\sqrt{x^2+1})$$

10. PASSWORT

e) PASSCODE-Tabelle
 Auf den Passwortstel-
 len des Pipass wer-
 den nur echte Zei-
 chen angezeigt, kei-
 ne Zeichenbefehle:
 s. PASSCODE-Tab.
 In der 0-Reihe ste-
 hen die 16 Pi. Die
 uebrigen Zeichen sind
 fast alle doppelt.
 Interessant ist, dass
 beim Entriegeln (mit
 Tastatur) kein Unter-
 schied zwischen den
 doppelt vorkommenden
 Zeichen gemacht wird:
 So kann zB mit 6 so-
 wohl das Codezeichen
 36 (GOTO6) als auch
 B6 (Min16) entrie-
 gelt werden.

TABELLE: PASS-CODE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	A	B	C	D	E	F	A	B	C	D	P	P	P	P	P	P	0
1	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	1
2	r	s	t	u	v	w	x	y	z	/							2
3	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	3
4	Q	R	S	T	U	V	W	X	Y	Z	x	+	-	=	u	4	
5	:	?	!	~	<	>	Σ	()	x	E	:	u	5		5	
6	A	R	F	Y	Z	Σ	2	-	1	y	x	y	x	y	u	6	
7	0	1	2	3	4	5	6	7	8	9	.	°	°	°	°	u	7
8	F	:	?	!	~	<	>	Σ	()	x	E	:	u	8	8	
9	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	9
A	r	s	t	u	v	w	x	y	z	/	+	-	=	u	A	A	
B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	B
C	Q	R	S	T	U	V	W	X	Y	Z	x	+	-	=	u	C	
D	:	?	!	~	<	>	Σ	()	x	E	:	u	D	D	D	
E	A	R	F	Y	Z	Σ	2	-	1	y	x	y	x	y	u	E	
F	0	1	2	3	4	5	6	7	8	9	.	°	°	°	°	u	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

f) LBL-TRICK
 PASSWORT-KNACKEN 3

- Programm in beliebiges Px eingeben: MODE2 Px " LBL1
- MODE3 Verriegeltes Programm (* Px) laden (vor oder nach Px)
- START: MODE1 GSBPx FST AC GOTO1
- ANZEIGE: gesamter Inhalt des Programmspeichers im ALPHA-Modus (codesleich): umlaufende Rollanzeige in richtiger Reihenfolge
- DAUER eines Umlaufs ca. 40 sec mit Schnellrechner: ca. 15 sec
- Rollanzeige kann mit HLT angehalten und mit FST durchgetastet werden.
- PASSWOERTER erscheinen in umgekehrter Reihenfolge.
- Zeichen werden als codesleiche Befehle ausgeführt.
- GO Error entsteht, wenn ein Programm die Buchstaben A-J (+ GOTOi) enthaelt. weiter: AC EXE
- Leerspeicher erscheinen als (+ BLANK).

11) PASS-ENTRIEGELUNG

Ein verschluesselttes Programm wird normalerweise durch Eingabe des richtigen Passworts entriegelt. Dies geht aber nicht bei syntnetischen Passwoertern, die mit Tastatur nicht erreichbar sind (s. 1. Teil 10c). Wie man auch solche Programme entriegeln kann, erfahren Sie in folgenden:

a) CHECK-TRICK

- In beliebiges Programm Pi 5 Fuellsteps eingeben und auf Cassette abspeichern:
 - MODE3 MAC
 - MODE2 Pi 44444
 - MODE3 SAVE EXE
- Verschluesselttes Programm Pj auf Anfang des leeren Programmspeichers laden. j kann i sein.
- Danach Pi wie folgt laden:
 - FA-1/2: CHECK
 - FX-602: MODE3 LOAD EXE
 - FA-1/2: umschalten: SAVE/LOAD bzw NT
- erst jetzt: Recorder starten!

Dadurch wird LOAD-Sperre (OP-Error) ueberlistet!

Pi und die 5 Fuellsteps werden dabei ueber Pj + Passwort + Pipass geschrieben. Die nachfolgenden Schritte bleiben erhalten und werden entriegelt. -Wenn der 5. Fuellstep (statt 4) ein Pk ist, steht das entriegelte Programm in Pk.

b) CA-47-TRICK

Bei der indirekten Abspeicherung einer Zahl auf die Hex-Adresse '. (CA) wird auf Adresse 47 immer ein P0 erzeugt (vgl. 2. Teil 3e). Zum Entriegeln muss man also nur dafuer sorgen, dass Pipass auf ADR.47 steht:

- MODE3 MAC
- Schreiben Sie in ein Programm Pj *P0 beliebige Fuellsteps bis Pos.41
- Laden Sie nun das verschluesselte Programm Pi vom Band: MODE3 LOAD Pi EXE
- MF = '. (CA)
- Speichern Sie ab: J.3333333333e 33 (Zahl ohne 0 !)
- IND MinF
- Was passiert?

Adr	vorher	nachher
0	Pj	Pj
1	↓	↓
41		
42	Pi	Pi
43	x	x
44	x	x
45	x	x
46	x	Pk
47	Pipass	P0
48	Progrm	Progrm
↓	↓	↓

Das entriegelte Programm steht in P0 !! ADR.46 wird zu einem Pk (0-Reihe, vgl. 3e Pos.2), z.B. zu P0. Dieses P0 kann mit MODE3 P0 AC geloescht werden. Entsprechendes gilt, wenn Pi = P0 ist.

c) CA-46-TRICK
 PASSWORT-KNACKEN 4
 Wenn in Pj nur 40 (statt 41) Fuellsteps sind, gelangt Pipass beim Laden auf Ad.46. Beim indirekten Abspeichern auf '. (CA) wird Pipass dort zur Zahlenweisung 2 (vgl. Kap.3e Pos2).

Adr	vorher	nachher
0	Pj	Pj
↓	↓	↓
40		
41	Pi	Pi
42	x	x
43	x	x
44	x	x
45	x	x
46	Pipass	2
47	Progrm	P0
↓	↓	↓

Das Passwort kann in Pi eingesehen werden (codesleiche Befehle von hinten).

d) GEHEIMPROGRAMME

Durch Anwendung der CHECK- und CA-TRICKS ist es moeglich, mehrere Programme mit dem gleichen Pi zu versehen. Davon kann nur das vorderste direkt angesprochen werden (1. Teil 8j). Die versteckten Programme koennen indirekt mit IND GSB(0) erreicht werden, wenn dort am Anfang eines der SONDER-Pi (0 1 2 3 . EXP) steht (vgl. 1. Teil 8d).

12. FC- VERRIEGELUNG

Der Befehl 10* mit Code ' (FC) ist ein Passbefehl wie Pipass. In einem verriegelten Programm steht 10* an 5. Stelle hinter dem zugehoerigen Pi. Die 4 Befehle zwischen Pi und 10* bilden das codesleiche Passwort. Verriegelung: wie in 1. Teil, Kap.10a+c: 10* ← 0=52, D'=96. Im WRT- und PCL- Modus wird beim Anwaehlen eines verriegelten Programms auf ALPHA umgeschaltet.

Eine Entriegelung ueber die Tastatur ist moeglich. Allerdings nicht fuer alle Tastaturzeichen und mit recht merkwuerdigen Effekten:

-Bei 4 gleichen Passwortezeichen sind entriegelbar die HEXCODE-Zeichen: 30-3A, 40-4C, 50-5C (zB 8888).

-Hiervon sind nicht alle Kombinationen entriegelbar: So sind die Zeichen aus den 0-9-Spalten miteinander kombiniert entriegelbar (z.B. ABST), nicht aber mit allen Zeichen aus den A-F-Spalten. Dagegen funktioniert es mit manchen Zeichenkombinationen aus anderen Zeilen (z.B. x**x). Eine Regel hierfür ist nicht erkennbar.

-Zum Entriegeln muss zuerst eine beliebige Taste aus den HEXCODE-Spalten A-F gedrueckt werden (zB P oder +), auch wenn dieses Zeichen nicht im Passwort vorkommt. In der Anzeige erscheinen meist zunaechst 1-5 Hexziffern, die zur Code-Spalte der gedruckten Taste gehoeren (bei + zB ')

-Fuer das Passwort sind die 4 rechten Anzeigestellen vorgesehen. Beim 4. Zeichen wird zurueckgeschaltet oder entriegelt. Die Entriegelung ist an der Anzeige 10* mit Passwortbefehlen erkennbar.

Beruhigend ist, dass mit den gleichen Tricks wie für Pipass auch diese Passwoerter geknackt und Programme entriegelt werden koennen.