

auf Band ab ~ 1535, 1100089, od 1000 - 2704

28.12.82

## SC/MP Disassembler für ES 100

Benötigte Spannungen +5V u. -12V

Bedienungsanleitung: Das Programm ist frei verschiebbar geschrieben, muß jedoch innerhalb eines 4 K - Blocks stehen.

Starten mit run: XXXX

(z.B. run C000)

Es erscheint ad.... auf dem Display

Eingabe der Anfangsadresse XXXX

Eingabe der Endadresse YYY

Wenn jetzt irgendeine Taste (außer 'mo') gedrückt wird, startet das Programm.

Die Voreinstellung ist, daß die Maschine nach 39 Zeilen stoppt.

Eingabe: " mo "

Display: mo 2 39 ( 2 = Zeilenabstand 1<sup>1</sup>/2 ; 39 = Zeilenanzahl )

Nun sind folgende Eingaben möglich : 1) " ru " 2) " bl " 3) Zahlen

zu 1) " ru " bewirkt starten des Programms mit dieser Voreinstellung  
(zweispaltig)

zu 2) " bl " bewirkt starten des Programms mit dieser Voreinstellung  
(zweispaltig)

zu 3) Es muß zuerst der neue Zeilenabstand ( 1, 2, oder 3 ) eingeben werden.  
Dann erfolgt die Eingabe der neuen Zeilenanzahl

Danach sind folgende Eingaben möglich:

" bl " starten mit der neuen Einstellung - zweispaltig

" mo " zurück in den Modify Mode

" andere Taste " starten des Programms mit der neuen Einstellung  
- einspaltig.

---

### Aufrufen des "Eigentlichen Programms" als Subroutine:

Es müssen im Ram Bereich vorher bestimmte Voreinstellungen vorgenommen werden.

- 1) Start adresse ( z.B. 0C00 ) in Reg. OFEE (high) u. OFED (low)
- 2) Anfangsadresse in Reg. OFEC (high) und OFEB
- 3) Endadresse in Reg. OFE2 (high) und OFE1
- 4) Zeilenanzahl ( hexadezimal ) in Reg. OFF1 und Reg. OFF0
- 5) 00 in Reg OFF2

Programm zum Ausdrucken eines "Kerodumps"  
 (wird im Disassembly an entsprech. Stelle eingefügt)

```

3la8  c408  ldi 08
3laa  c915  st 1 15
3lac  c10c  ld 1 0c
3lae  36    xpah 2
3laf  c10b  ld 1 0b
3lbl  32    xpal 2
3lb2  c200  ld 2 00
3lb4  01    xae
3lb5  40    lde
3lb6  c900  st 1 00
3lb8  lc    sr
3lb9  lc    sr
3lba  lc    sr
3lbb  lc    sr
3lbc  dca0  ori a0
3lbe  01    xae
3lbf  c180  ld 1 80
3lc1  3f    xppc 3
3lc2  c100  ld 1 00
3lc4  d40f  ani 0f
3lc6  dca0  ori a0
3lc8  01    xae
3lc9  c180  ld 1 80
3lcb  3f    xppc 3
3lcc  c41b  ldi 1b
3lce  3f    xppc 3
3lcf  3f    xppc 3
3ld0  3f    xppc 3
3ld1  3f    xppc 3
3ld2  c10d  ld 1 0d
3ld4  02    ccl
3ld5  f4e2  adi e2
3ld7  c91d  st 1 1d
3ld9  c10e  ld 1 0e
3ldb  f406  adi 06
3ldd  c91c  st 1 1c
3ldf  3f    xppc 3
3le0  c10d  ld 1 0d
3le2  f44c  adi 4c
3le4  c91d  st 1 1d
3le6  c10e  ld 1 0e
3le8  f406  adi 06
3lea  c91c  st 1 1c
3lec  b915  dld 1 15
3lee  9cbb  jnz  bb
3lf0  c406  ldi 06
3lf2  3f    xppc 3
3lf3  c408  ldi 08
3lf5  c915  st 1 15
3lf7  90ae  jmp  ae
3lf9  a001  error 01
3l

```

SC/M P Disassembler für ES 100

### Verwendete Register:

\* Die roten Register dürfen durch ein anderes Programm nicht verändert werden;

die anderen werden zwar durch die betreffenden Routinen verändert, dürfen jedoch sonst verwendet werden

# "Eingabe"

c000	c201	ld 2 01	Hauptadresse in Reg. 0FEE (higher) und 0FED (lower) schreiben
c002	ca0d	st 2 0d	
c004	c202	ld 2 02	
c006	ca0e	st 2 0e	
c008	c427	ldi 27	Zeilenanzahl (Voreinstellung = H'27 ≈ D'39) in Reg. 0FF0 (Zeilenanzahl) und Reg. 0FF1 (Zeilenzahl) schreiben
c00a	ca10	st 2 10	
c00c	call	st 2 11	
c00e	c402	ldi 02	
c010	ca13	st 2 13	Zeilenabstand (Voreinstellung = H'02 ≈ 1½) in Reg 0FF3 eintragen
c012	c400	ldi 00	ptr. I auf Display (7-Segment)
c014	31	xpal 1	
c015	c407	ldi 07	
c017	35	xpah 1	
c018	c45f	ldi 5f	"ad...." an Display
c01a	c907	st 1 07	
c01c	c45e	ldi 5e	
c01e	c906	st 1 06	
c020	c400	ldi 00	"Gethex-1" in "Raut-adr." Register (0FFD-lower; 0FFC-higher)
c022	c900	st 1 00	
c024	c901	st 1 01	
c026	c480	ldi 80	
c028	c902	st 1 02	"Push-1" in Ptr. III
c02a	c903	st 1 03	
c02c	c904	st 1 04	
c02e	c905	st 1 05	
c030	c43e	ldi 3e	Anfangsadresse holen
c032	cald	st 2 1d	
c034	c402	ldi 02	
c036	calc	st 2 1c	
c038	c455	ldi 55	Anfangsadresse in Reg. 0FEC (higher) und Reg. 0FEB (lower) speichern
c03a	33	xpal 3	
c03b	c400	ldi 00	
c03d	37	xpah 3	
c03e	3f	xppc 3	Endadresse holen
c03f	c201	ld 2 01	
c041	ca0b	st 2 0b	
c043	c202	ld 2 02	
c045	ca0c	st 2 0c	
c047	3f	xppc 3	

c048	c40a	ldi 0a	} Adr. "Ld Kbd-1" in "Rout-Adr."	
c04a	cald	st 2 ld		
c04c	3f	xppc 3	auf Taste warten	
c04d	c208	1d 2 08		
c04f	e4e0	xri e0	"Modify"-Taste?	
c051	9c69	jnz 69	nein: auf Adr. C0BC (= Hilfsjump) → C130	
→ c053	c454	ldi 54	} "Mo---" an Display	
c055	c907	st 1 07		
c057	c45c	ldi 5c		
c059	c906	st 1 06		
c05b	c400	ldi 00		
c05d	c905	st 1 05		
c05f	c904	st 1 04		
c061	c903	st 1 03		
c063	c902	st 1 02		
c065	c400	ldi 00		} Zähler (Reg. 0FF0)
c067	ca00	st 2 00	auf 00	
c069	c210	1d 2 10	Zeilenanzahl laden	
→ c06b	01	xae	} Zeilenanzahl, die als Hex-Zahl in Reg. 0FF0 steht durch wiederholte Subtraktion von 10 im Decimalzahl umwandeln und auf Display 00 und 01 darstellen.	
c06c	02	ccl		
c06d	40	1de		
c06e	f4f6	adi f6		(D'-10)
c070	06	csa		
c071	9408	jp 08		
c073	aa00	i1d 2 00		
c075	40	1de		
c076	02	ccl		
c077	f4f6	adi f6		
c079	90f0	jmp f0	} Ptr III auf 7. Leg. Code Tabelle	
c07b	c401	ldi 01		
c07d	37	xpah 3		
c07e	c41f	ldi lf		
c080	33	xpal 3		
c081	c380	1d 3 80	Ld(3)+(Ex.t.)	
c083	c900	st 1 00	an Display 00	
c085	c200	1d 2 00		
c087	d40f	ani 0f		
c089	01	xae		
c08a	c380	1d 3 80		

c08c	c901	st 1 01	}	
c08e	c213	1d 2 13	Zeilenabstand (Reg. 0FF3) (1=1; 2=1½; 3=2) an Display 03	
c090	01	xae		
c091	c380	1d 3 80		
c093	c903	st 1 03		
c095	c400	1di 00	ptr. III auf "Push-1"	
c097	37	xpah 3		
c098	c455	1di 55		
c09a	33	xpal 3		
→c09b	3f	xppc 3	Taste von Keyboard	
c09c	c208	1d 2 08		
c09e	01	xae		
c09f	40	1de		
c0a0	e4f0	xri f0	"run" ?	
c0a2	9835	jz 35	ja: → C0D9	
c0a4	40	1de		
c0a5	e4b0	xri b0	"bl." ?	
c0a7	987a	jz 7a	ja: →	
c0a9	40	1de		
c0aa	d40f	ani 0f	nur 1, 2, oder 3 sind als Eingabe zulässig. (Veränderung: Zeilenabstand)	
c0ac	98ed	jz ed		
c0ae	40	1de		
c0af	d40c	ani 0c		
c0b1	9ce8	jnz e8		
c0b3	40	1de	Eingabe für neuen Zeilenabstand im Reg. 0FF3	
c0b4	d40f	ani 0f		
c0b6	ca13	st 2 13		
c0b8	9004	jmp 04		
→c0ba	9097	jmp 97	Hilfsjump: C11C → C0BA → C053	
→c0bc	9073	jmp 73	Hilfsjump: Ausprung - C051 → C0BC → C130	
→c0be	c207	1d 2 07	neuen Zeilenabstand an Display 03	
c0c0	c903	st 1 03		
→c0c2	3f	xppc 3	Taste holen	
c0c3	c208	1d 2 08	nur Zahlen von 0-9 für neue Zeilenwahl annehmen	
c0c5	d40f	ani 0f		
c0c7	01	xae		
c0c8	40	1de		
c0c9	02	ccl		
c0ca	f406	adi 06		

```

c0cc d4f0 ani f0
c0ce 9cf2 jnz f2
c0d0 40 lde
c0d1 ca00 st 2 00 } 1. Dezimalzahl in Reg. 0FED zwischenspeichern
c0d3 c207 1d 2 07 } 7 Reg. Code an Display 01
c0d5 c901 st 1 01
c0d7 9006 jmp 06
c0d9 9046 jmp 46 → C121
c0db 9000 jmp 00
c0dd 9000 jmp 00
c0df c400 ldi 00 } Display 00 löschen
c0e1 c900 st 1 00
c0e3 3f xppc 3 } 2. Tasten für neue Zeilenanzahl holen
c0e4 c208 1d 2 08
c0e6 d40f ani 0f
c0e8 01 xae } nur Zahlen 0-9 annehmen
c0e9 40 lde
c0ea 02 cc1
c0eb f406 adi 06
c0ed d4f0 ani f0
c0ef 9cf2 jnz f2
c0f1 c207 1d 2 07 } 7 Reg. Code d. 2. Zahl an Display 00
c0f3 c900 st 1 00
c0f5 c400 ldi 00
c0f7 01 xae } 1. Zahl war 0?
c0f8 c200 1d 2 00
c0fa 9809 jz 09 ja: → C105
c0fc 01 xae
c0fd 02 cc1
c0fe f40a adi 0a
c100 01 xae
c101 ba00 d1d 2 00
c103 90f3 jmp f3
c105 c208 1d 2 08
c107 d40f ani 0f
c109 02 cc1
c10a 70 ade
c10b ca10 st 2 10 } in Reg 0FFD = Zeilenanzahl
c10d call st 2 11 } und Reg 0FF1 = Zeilenzähler abspalten

```

dezimale Eingabe für neue Zeilenanzahl in hexadezimale Zahl umwandeln

sooft, wie es der 1. Eingabe entspricht H'0A (= D'10) aufzuhören

dann 2. Eingabe addieren

in Reg 0FFD = Zeilenanzahl und Reg 0FF1 = Zeilenzähler abspalten

c10f c400 ldi 00	} Display 06 und 07 löschen auf Taste warten
c111 c907 st 1 07	
c113 c906 st 1 06	
c115 3f xppc 3	
c116 c208 ld 2 08	
c118 01 xae	
c119 40 lde	"Modify"?
c11a e4e0 xri e0	ja: → C0BA → C053; zurück in Modify Routine
c11c 989c jz 9c	
c11e 40 lde	
c11f e4b0 xri b0	"bl"?
→ c121 9c0e jnz 0e	nein: → C131
c123 c47c ldi 7c	} "bl" an Display 07 und 06
c125 c907 st 1 07	
c127 c438 ldi 38	
c129 c906 st 1 06	
c12b c4ff ldi ff	} Reg. 0FF2 mit FF laden; FF bedeutet es soll im Block-Format (2-spaltig) gedruckt werden.
c12d cal2 st 2 12	
c12f 9004 jmp 04	} Reg. 0FF2 mit 00 laden; d.h. es soll nur 1-spaltig gedruckt werden.
→ c131 c400 ldi 00	
c133 cal2 st 2 12	
→ c135 c4ff ldi ff	} Reg. 0FF4 mit FF laden; FF = 1. Spalte 00 = 2. Spalte
c137 cal4 st 2 14	
c139 c20d ld 2 0d	} aus der Startadresse (steht in Reg. 0FEE und 0FED) (ist hier z.B. C000) und dem Abstand zum "eigentlichen Programm" deren effektive Adresse errechnen. Effektive Adresse → "Rout-Adr"-Register
c13b 02 cc1	
c13c f4 <b>4b</b> adi 4b	
c13e cald st 2 1d	
c140 c20e ld 2 0e	
c142 f4 <b>01</b> adi 01	
c144 calc st 2 1c	
c146 3f xppc 3	"Eigentliches Programm" als Subroutine auspringen
c147 c400 ldi 00	} nach Rückkehr aus "eigentlichem Programm" "Ebug" auspringen.
c149 33 xpal 3	
c14a 3f xppc 3	
c14b 08 nop	

---

"

~~c14c c4e0 ldi e0~~  
~~c14e 31 xpah 1~~  
~~c14f c40f ldi 0f~~  
~~c151 35 xpah 1~~

2116	c208	1d 2 08
2118	01	xae
2119	40	lde
211a	e4e0	xri e0
211c	989c	jz 9c
211e	40	lde
211f	e4b0	xri b0
2121	9c0f	jnz 0f
2123	c47c	ldi 7c
2125	c907	st 1 07
2127	c438	ldi 38
2129	c906	st 1 06
212b	c4ff	ldi ff
212d	cal2	st 2 12
212f	9004	jmp 04
2131	c400	ldi 00
2133	cal2	st 2 12
2135	c4ff	ldi ff
2137	cal4	st 2 14
2139	c20d	ld 2 0d
213b	02	<sup>22</sup> ne
213c	f44b	adi 4b
213e	cald	st 2 ld
2140	c20e	ld 2 0e
2142	f401	adi 01
2144	calc	st 2 lc
2146	3f	xppc 3
2147	c400	ldi 00
2149	33	xpal 3
214a	3f	xppc 3
214b	08	ne

→

214c	c4e0	1di e0	} Ptr I "Eigentliches Programm"
214e	31	xpal 1	
214f	c40f	1di 0f	} = 0FED X14C = Ausprungadresse als Subroutine"
2151	35	xpah 1	
2152	c455	1di 55	} "Push-1" → Ptr. III
2154	33	xpal 3	
2155	c400	1di 00	} aus Startadresse und Abstand effektive Adresse von "Druckentscheidung" berechnen "Druckentscheidung -1" → "Rout. Adr"
2157	37	xpah 3	
2158	c10d	1d 1 0d	
215a	02	ccl	
215b	f44c	adi 4c	
215d	c91d	st 1 ld	
215f	c10e	1d 1 0e	
2161	f406	adi 06	
2163	c91c	st 1 lc	
2165	c437	1di 37	
2167	3f	xppc 3	
2168	c42f	1di 2f	0
216a	c9a0	st 1 a0	1
216c	c43d	1di 3d	2
216e	c9a1	st 1 al	3
2170	c408	1di 08	4
2172	c9a2	st 1 a2	
2174	c40d	1di 0d	
2176	c9a3	st 1 a3	
2178	c412	1di 12	
217a			

217a	c9a4	st l a4	
217c	c417	ldi 17	5
217e	c9a5	st l a5	
2180	c41d	ldi 1d	6
2182	c9a6	st l a6	
2184	c419	ldi 19	7
2186	c9a7	st l a7	
2188	c425	ldi 25	8
218a	c9a8	st l a8	
218c	c421	ldi 21	9
218e	c9a9	st l a9	
2190	c43f	ldi 3f	A
2192	c9aa	st l aa	
2194	c41e	ldi 1e	B
2196	c9ab	st l ab	
2198	c414	ldi 14	C
219a	c9ac	st l ac	
219c	c411	ldi 11	D
219e	c9ad	st l ad	E
21a0	c40e	ldi 0e	
21a2	c9ae	st l ae	
21a4	c416	ldi 16	F
21a6	c9af	st l af	
21a8	c10c	ld l 0c	1. Adr. (high) laden
21aa	lc	kkne \$Y	
21ab	lc	kkne \$Y	4x SR
21ac	lc	kkne \$Y	
21ad	lc	kkne \$Y	
21ae	dca0	ori a0	Displacement um auf Tabelle zu kommen
21b0	01	xae	
21b1	c180	ld l 80	Ld (1) + (Ext.)
21b3	3f	xppc 3	
21b4	c10c	ld l 0c	
21b6	d40f	ani 0f	
21b8	dca0	ori a0	
21ba	01	xae	
21bb	c180	ld l 80	
21bd	3f	xppc 3	
21be	c10b	ld l 0b	
21c0	lc	kkne	
21c1	lc	kkne	
21c2	lc	kkne	
21c3	lc	kkne	
21c4	dca0	ori a0	
21c6	01	xae	
21c7	c180	ld l 80	
21c9	3f	xppc 3	
21ca	c10b	ld l 0b	
21cc	d40f	ani 0f	
21ce	dca0	ori a0	
21d0	01	xae	
21d1	c180	ld l 80	
21d3	3f	xppc 3	
21d4	c41b	ldi 1b	
21d6	3f	xppc 3	
21d7	3f	xppc 3	
21d8	c10c	ld l 0c	

Code für Zahlen in  
Tabelle schreiben  
(Adr. 0FAF - 0FAA)

"Loop"

1. Adr. (high) laden

4x SR

Displacement um auf Tabelle zu kommen

1. Ziffer d.  
Adr. ausgeben

2. Ziffer der Anfangsadresse (ff) ausgeben

3. Ziffer

4. Ziffer

2x Blank

21da	36	xpah 2	} Anfangsadresse (ff.) in Ptr. II Inhalt d. Anfangsadresse (ff) laden	
21db	c10b	1d 1 0b		
21dd	32	xpal 2		
21de	c200	1d 2 00		
21e0	01	xae		
21e1	40	lde		
21e2	940d	jp 0d		
21e4	c10d	1d 1 0d		
21e6	02	22ne		
21e7	f4f8	adi f8		
21e9	32	xpal 2	} Doppelbyte Befehl? Adresse von "Doppelbyte Befehl" berechnen und auspringen	
21ea	c10e	1d 1 0e		
21ec	f403	adi 03		
21ee	36	xpah 2		
21ef	9200	jmp 2 00		
21f1	40	lde		
21f2	c900	st 1 00		
21f4	1c	kkne		
21f5	1c	kkne		
21f6	1c	kkne		
21f7	1c	kkne	} "Ein-Byte Befehle" 1. Hex ziffer des Befehls ausgeben	
21f8	dca0	ori a0		
21fa	01	xae		
21fb	c180	1d 1 80		
21fd	3f	xppc 3		
21fe	c100	1d 1 00		
2200	d40f	ani 0f		
2202	dca0	ori a0		
2204	01	xae		
2205	c180	1d 1 80		
2207	3f	xppc 3	} 2. Hex ziffer des Befehls ausgeben 4x Blank	
2208	c41b	ldi 1b		
220a	3f	xppc 3		
220b	3f	xppc 3		
220c	3f	xppc 3		
220d	3f	xppc 3		
220e	c100	1d 1 00		
2210	01	xae		
2211	c10d	1d 1 0d		
2213	02	22ne		
2214	f4cc	adi cc	} Inhalt d. Adr. (Befehl) laden und in E suchen Adr. von "zeilenende (1-Byte Befehl)" berechnen und in Ptr. II	
2216	32	xpal 2		
2217	c10e	1d 1 0e		
2219	f403	adi 03		
221b	36	xpah 2		
221c	40	lde		
221d	08 ne			
221e	08 ne			
221f	e440	xri 40		
2221	9c0b	jnz 0b		
2223	c42d	ldi 2d	} LDE (2. Byte freigeschlagen)	
2225	3f	xppc 3		
2226	c411	ldi 11		
2228	3f	xppc 3		
2229	c40e	ldi 0e		
222b	3f	xppc 3		
222c	9200	jmp 2 00		
222e	40	lde		
222f	e401	xri 01		
2231				

2231	9c0b	jnz 0b	
2233	c40f	ldi 0f	XHE
2235	3f	xppc 3	
2236	c43f	ldi 3f	
2238	3f	xppc 3	
2239	c40e	ldi 0e	
223b	3f	xppc 3	
223c	9200	jmp 2 00	
223e	40	lde	
223f	e430	xri 30	XPA ausgeben
2241	d4f8	ani f8	
2243	9c18	jnz 18	
2245	c40f	ldi 0f	
2247	3f	xppc 3	
2248	c429	ldi 29	
224a	3f	xppc 3	
224b	c43f	ldi 3f	
224d	3f	xppc 3	Low oder high?
224e	40	lde	
224f	d404	ani 04	
2251	9c05	jnz 05	
2253	c42d	ldi 2d	
2255	3f	xppc 3	
2256	9003	jmp 03	
2258	c418	ldi 18	
225a	3f	xppc 3	2B
225b	9011	jmp 11	
225d	40	lde	
225e	e43c	xri 3c	
2260	d4fc	ani fc	
2262	9c34	jnz 34	
2264	c40f	ldi 0f	
2266	3f	xppc 3	
2267	c429	ldi 29	XPPC
2269	3f	xppc 3	
226a	3f	xppc 3	
226b	c414	ldi 14	
226d	3f	xppc 3	
226e	c41b	ldi 1b	
2270	3f	xppc 3	
2271	40	lde	
2272	d403	ani 03	Blank
2274	01	xae	
2275	40	lde	
2276	e401	xri 01	
2278	9c03	jnz 03	
227a	c43d	ldi 3d	
227c	3f	xppc 3	
227d	40	lde	
227e	e402	xri 02	I
2280	9c03	jnz 03	
2282	c408	ldi 08	
2284	3f	xppc 3	
2285	40	lde	
2286	e403	xri 03	
2288	9c03	jnz 03	
228a	c40d	ldi 0d	
228c	3f	xppc 3	II
228d	9200	jmp 2 00	

Ptr. I, II oder III?

II

III

228f	c100	1d 1 00	ANE
2291	01	xae	
2292	40	1de	
2293	e450	xri 50	
2295	9c0b	jnz 0b	
2297	c43f	ldi 3f	
2299	3f	xppc 3	
229a	c422	ldi 22	
229c	3f	xppc 3	
229d	c40e	ldi 0e	
229f	3f	xppc 3	ORE
22a0	9200	jmp 2 00	
22a2	40	1de	
22a3	e458	xri 58	
22a5	9c0b	jnz 0b	
22a7	c42c	ldi 2c	
22a9	3f	xppc 3	
22aa	c415	ldi 15	
22ac	3f	xppc 3	
22ad	c40e	ldi 0e	
22af	3f	xppc 3	XRE
22b0	9200	jmp 2 00	
22b2	40	1de	
22b3	e460	xri 60	
22b5	9c0b	jnz 0b	
22b7	c40f	ldi 0f	
22b9	3f	xppc 3	
22ba	c415	ldi 15	
22bc	3f	xppc 3	
22bd	c40e	ldi 0e	
22bf	3f	xppc 3	DAE
22c0	9200	jmp 2 00	
22c2	40	1de	
22c3	e468	xri 68	
22c5	9c0b	jnz 0b	
22c7	c411	ldi 11	
22c9	3f	xppc 3	
22ca	c43f	ldi 3f	
22cc	3f	xppc 3	
22cd	c40e	ldi 0e	
22cf	3f	xppc 3	ADE
22d0	9200	jmp 2 00	
22d2	40	1de	
22d3	e470	xri 70	
22d5	9c0b	jnz 0b	
22d7	c43f	ldi 3f	
22d9	3f	xppc 3	
22da	c411	ldi 11	
22dc	3f	xppc 3	
22dd	c40e	ldi 0e	
22df	3f	xppc 3	CRE
22e0	9200	jmp 2 00	
22e2	40	1de	
22e3	e478	xri 78	
22e5	9c0b	jnz 0b	
22e7	c414	ldi 14	
22e9	3f	xppc 3	
22ea	c43f	ldi 3f	
22ec	3f	xppc 3	
22ed	c40e	ldi 0e	
22ef	3f	xppc 3	
22f0	9200	jmp 2 00	
22f2	40	1de	
22f3	e4		

22f3	e419	xri 19	
22f5	9c0b	jnz 0b	S10
22f7	c40c	ldi 0c	
22f9	3f	xppc 3	
22fa	c427	ldi 27	
22fc	3f	xppc 3	
22fd	c42c	ldi 2c	
22ff	3f	xppc 3	
2300	9200	jmp 2 00	
2302	40	lde	
2303	e41d	xri 1d	
2305	9c0b	jnz 0b	
2307	c40c	ldi 0c	
2309	3f	xppc 3	
230a	c415	ldi 15	
230c	3f	xppc 3	
230d	c42d	ldi 2d	
230f	3f	xppc 3	
2310	9200	jmp 2 00	
2312	40	lde	
2313	e41f	xri 1f	RRL
2315	9c0b	jnz 0b	
2317	c415	ldi 15	
2319	3f	xppc 3	
231a	c415	ldi 15	
231c	3f	xppc 3	
231d	c42d	ldi 2d	
231f	3f	xppc 3	
2320	9200	jmp 2 00	
2322	40	lde	
2323	e402	xri 02	CCL
2325	9c0b	jnz 0b	
2327	c414	ldi 14	
2329	3f	xppc 3	
232a	c414	ldi 14	
232c	3f	xppc 3	
232d	c42d	ldi 2d	
232f	3f	xppc 3	
2330	9200	jmp 2 00	
2332	40	lde	
2333	e403	xri 03	SCL
2335	9c0b	jnz 0b	
2337	c40c	ldi 0c	
2339	3f	xppc 3	
233a	c414	ldi 14	
233c	3f	xppc 3	
233d	c42d	ldi 2d	
233f	3f	xppc 3	
2340	9200	jmp 2 00	
2342	40	lde	
2343	e405	xri 05	LEN
2345	9c0b	jnz 0b	
2347	c427	ldi 27	
2349	3f	xppc 3	
234a	c40e	ldi 0e	
234c	3f	xppc 3	
234d	c422	ldi 22	
234f	3f	xppc 3	
2350	9200	jmp 2 00	
2352	40	lde	
2353	e406	xri 06	
2355	9c0b	jnz 0b	
2357	c414	ldi 14	
2359	3f	xppc 3	
235a			

235a	c40c	ldi 0c	CSA
235c	3f	xppc 3	
235d	c43f	ldi 3f	
235f	3f	xppc 3	
2360	9200	jmp 2 00	
2362	40	lde	
2363	e407	xri 07	
2365	9c0b	jnz 0b	
2367	c414	ldi 14	CHS
2369	3f	xppc 3	
236a	c43f	ldi 3f	
236c	3f	xppc 3	
236d	c40c	ldi 0c	
236f	3f	xppc 3	
2370	9200	jmp 2 00	
2372	40	lde	
2373	e408	xri 08	NOP
2375	9c0b	jnz 0b	
2377	c422	ldi 22	
2379	3f	xppc 3	
237a	c42c	ldi 2c	
237c	3f	xppc 3	
237d	c429	ldi 29	
237f	3f	xppc 3	
2380	9200	jmp 2 00	SR
2382	40	lde	
2383	e41c	xri 1c	
2385	9c08	jnz 08	
2387	c40c	ldi 0c	
2389	3f	xppc 3	
238a	c415	ldi 15	
238c	3f	xppc 3	
238d	9200	jmp 2 00	RR
238f	40	lde	
2390	e41e	xri 1e	
2392	9c06	jnz 06	
2394	c415	ldi 15	
2396	3f	xppc 3	
2397	3f	xppc 3	
2398	9200	jmp 2 00	
239a	40	lde	HALT
239b	e400	xri 00	
239d	9c0e	jnz 0e	
239f	c418	ldi 18	
23a1	3f	xppc 3	
23a2	c43f	ldi 3f	
23a4	3f	xppc 3	
23a5	c42d	ldi 2d	
23a7	3f	xppc 3	DINT
23a8	c413	ldi 13	
23aa	3f	xppc 3	
23ab	9200	jmp 2 00	
23ad	40	lde	
23ae	e404	xri 04	
23b0	9c0e	jnz 0e	
23b2	c411	ldi 11	
23b4	3f	xppc 3	DINT
23b5	c427	ldi 27	
23b7	3f	xppc 3	
23b8	c422	ldi 22	
23ba	3f	xppc 3	
23bb	c413	ldi 13	
23bd	3f	xppc 3	
23be	9200	jmp 2 00	
23c0	c4		

23c0	c40e	ldi 0e	}	ERROR
23c2	3f	xppc 3		
23c3	c415	ldi 15		
23c5	3f	xppc 3		
23c6	3f	xppc 3		
23c7	c42c	ldi 2c		
23c9	3f	xppc 3		
23ca	c415	ldi 15		
23cc	3f	xppc 3		
23cd	c406	ldi 06		Line feed + CR
23cf	3f	xppc 3		effektive Adresse von "Adresse erhöhen" errechnen
23d0	c10d	ld 1 0d		"Adresse erhöhen - 1" → "Rout.-Adr." Register
23d2	02	22 ne		Adresse erhöhen
23d3	f4e2	adi e2		effektive Adresse von "Druckentscheidung" berechnen
23d5	c91d	st 1 ld		"Druckentscheidung - 1" → Rout. Adr.
23d7	c10e	ld 1 0e		effektive Adresse von "Loop" berechnen
23d9	f406	adi 06		"Loop - 1" → Ptr. II
23db	c91c	st 1 lc		Rücksprung nach Loop
23dd	3f	xppc 3		
23de	c10d	ld 1 0d		
23e0	02	22 ne		
23e1	f44c	adi 4c		
23e3	c91d	st 1 ld		
23e5	c10e	ld 1 0e		
23e7	f406	adi 06		
23e9	c91c	st 1 lc		
23eb	c10d	ld 1 0d		
23ed	02	22 ne		
23ee	f4a7	adi a7		
23f0	32	xpal 2		
23f1	c10e	ld 1 0e		
23f3	f401	adi 01		
23f5	36	xpah 2		
23f6	9200	jmp 2 00		
23f8	00	22 ne		
23f9	c402	ldi 02	}	Voreingestell. "Doppelbyte Befehl"
23fb	c90f	st 1 0f		"1. Byte"
23fd	40	1de		
23fe	c900	st 1 00		
→2400	40	1de		1. Hex Ziffer ausgeben (1. od. 2. Byte)
2401	1c	kkne		
2402	1c	kkne		
2403	1c	kkne		
2404	1c	kkne		
2405	dca0	ori a0		2. Hex Ziffer ausgeben (1. od. 2. Byte)
2407	01	xae		
2408	c180	ld 1 80		
240a	3f	xppc 3		
240b	c100	ld 1 00		
240d	d40f	ani 0f	}	2. Byte des Befehls ausgegeben ?
240f	dca0	ori a0		jz 11 → 242A
2411	01	xae		1. Byte des Befehls → Reg. 0FEA
2412	c180	ld 1 80		Anfangsadresse (ff) → Ptr. II
2414	3f	xppc 3		
2415	b90f	dld 1 0f		
2417	9811	jz 11		
2419	c100	ld 1 00		
241b	c90a	st 1 0a		
241d	c10c	ld 1 0c		
241f	36	xpah 2	}	2. Byte des Befehls laden
2420	c10b	ld 1 0b		
2422	32	xpal 2		
2423	c201	ld 2 01		
2425	e90			

2425 c900 st 1 00      2. Byte → Reg DFE0  
 2427 01 xae      2. Byte → E  
 2428 90d6 jmp d6      → 2400  
 } 2x Blank  
 242a c41b ldi 1b  
 242c 3f xppc 3  
 242d 3f xppc 3  
 242e c10a ldi 1 0a      1. Byte laden  
 2430 01 xae      XA E  
 2431 c10d ldi 1 0d  
 2433 02 22ne  
 } "2. Byte - 1" → Ptr. II  
 2434 f406 adi 06  
 2436 32 xpal 2  
 2437 c10e ldi 1 0e  
 2439 f406 adi 06  
 243b 36 xpah 2  
 243c 40 lde      1. Byte aus Ext. holen  
 243d e4c4 xri c4  
 243f 9c0b jnz 0b  
 2441 c42d ldi 2d  
 } LDI  
 2443 3f xppc 3  
 2444 c411 ldi 11  
 2446 3f xppc 3  
 2447 c427 ldi 27  
 2449 3f xppc 3  
 244a 9200 jmp 2 00  
 244c 40 lde  
 244d e4d4 xri d4  
 244f 9c0b jnz 0b  
 2451 c43f ldi 3f      AND  
 2453 3f xppc 3  
 2454 c422 ldi 22  
 2456 3f xppc 3  
 2457 c427 ldi 27  
 2459 3f xppc 3  
 245a 9200 jmp 2 00  
 245c 40 lde  
 245d e4dc xri dc  
 245f 9c0b jnz 0b  
 2461 c42c ldi 2c      ORI  
 2463 3f xppc 3  
 2464 c415 ldi 15  
 2466 3f xppc 3  
 2467 c427 ldi 27  
 2469 3f xppc 3  
 246a 9200 jmp 2 00  
 246c 40 lde  
 246d e4e4 xri e4  
 246f 9c0b jnz 0b  
 2471 c40f ldi 0f      XORI  
 2473 3f xppc 3  
 2474 c415 ldi 15  
 2476 3f xppc 3  
 2477 c427 ldi 27  
 2479 3f xppc 3  
 247a 9200 jmp 2 00  
 247c 40 lde  
 247d e4ec xri ec  
 247f 9c0b jnz 0b  
 2481 c411 ldi 11  
 2483 3f xppc 3  
 2484 c43f ldi 3f  
 2486 3f xppc 3  
 2487 c427 ldi 27  
 2489 3f xppc 3  
 248a 92

248a	9200	jmp 2 00	}	AD1
248c	40	lde		
248d	e4f4	xri f4		
248f	9c0b	jnz 0b		
2491	c43f	ldi 3f		
2493	3f	xppc 3		
2494	c411	ldi 11		
2496	3f	xppc 3		
2497	c427	ldi 27		
2499	3f	xppc 3		
249a	9200	jmp 2 00	}	CH1
249c	40	lde		
249d	e4fc	xri fc		
249f	9c0b	jnz 0b		
24a1	c414	ldi 14		
24a3	3f	xppc 3		
24a4	c43f	ldi 3f		
24a6	3f	xppc 3		
24a7	c427	ldi 27		
24a9	3f	xppc 3		
24aa	9200	jmp 2 00	}	DLY
24ac	40	lde		
24ad	e48f	xri 8f		
24af	9c0b	jnz 0b		
24bl	c411	ldi 11		
24b3	3f	xppc 3		
24b4	c42d	ldi 2d		
24b6	3f	xppc 3		
24b7	c409	ldi 09		
24b9	3f	xppc 3		
24ba	9200	jmp 2 00	}	Adresse für "Nr. des Pointers" berechnen "Nr. des Pointers - 1" → Ptr. II
24bc	c10d	ld 1 0d		
24be	02	<sup>22</sup> ne		
24bf	f4e1	adi e1		
24cl	32	xpal 2		
24c2	c10e	ld 1 0e		
24c4	f405	adi 05		
24c6	36	xpah 2		
24c7	40	lde		
24c8	e4a8	xri a8		
24ca	d4fc	ani fc	}	III
24cc	9c0b	jnz 0b		
24ce	c427	ldi 27		
24d0	3f	xppc 3		
24d1	c42d	ldi 2d		
24d3	3f	xppc 3		
24d4	c411	ldi 11		
24d6	3f	xppc 3		
24d7	9200	jmp 2 00		
24d9	40	lde		
24da	e4b8	xri b8	}	ILD
24dc	d4fc	ani fc		
24de	9c0b	jnz 0b		
24e0	c411	ldi 11		
24e2	3f	xppc 3		
24e3	c42d	ldi 2d		
24e5	3f	xppc 3		
24e6	c411	ldi 11		
24e8	3f	xppc 3		
24e9	9200	jmp 2 00		
24eb	40	lde	}	DLD
24ec	e490	xri 90		
24ee	d4fc	ani fc		
24f0	9c0b	jnz 0b		
24f2	c4			

24f2	c424	ldi 24	{}	JMP
24f4	3f	xppc 3		JNZ
24f5	c426	ldi 26		
24f7	3f	xppc 3		
24f8	c429	ldi 29		
24fa	3f	xppc 3		
24fb	9200	jmp 2 00		
24fd	40	lde		
24fe	e49c	xri 9c	{}	JP
2500	d4fc	ani fc		JZ
2502	9c0b	jnz 0b		
2504	c424	ldi 24		
2506	3f	xppc 3		
2507	c422	ldi 22		
2509	3f	xppc 3		
250a	c41f	ldi lf		
250c	3f	xppc 3	{}	Adresse von "@" berechnen
250d	9200	jmp 2 00		"@ - 1" → ptr.II
250f	40	lde		
2510	e494	xri 94		
2512	d4fc	ani fc		
2514	9c08	jnz 08		
2516	c424	ldi 24		
2518	3f	xppc 3		
2519	c429	ldi 29	{}	LD
251b	3f	xppc 3		
251c	9200	jmp 2 00		
251e	40	lde		
251f	e498	xri 98		
2521	d4fc	ani fc		
2523	9c08	jnz 08		
2525	c424	ldi 24		
2527	3f	xppc 3	{}	ST
2528	c41f	ldi lf		
252a	3f	xppc 3		
252b	9200	jmp 2 00		
252d	c10d	ld 1 0d		
252f	02	22ne		
2530	f4d6	adi d6		
2532	32	xpal 2		
2533	c10e	ld 1 0e	{}	
2535	f405	adi 05		
2537	36	xpah 2		
2538	40	lde		
2539	e4c0	xri c0		
253b	d4f8	ani f8		
253d	9c08	jnz 08		
253f	c42d	ldi 2d		
2541	3f	xppc 3	{}	
2542	c411	ldi 11		
2544	3f	xppc 3		
2545	9200	jmp 2 00		
2547	40	lde		
2548	e4c8	xri c8		
254a	d4f8	ani f8		
254c	9c08	jnz 08		
254e	c40c	ldi 0c	{}	
2550	3f	xppc 3		
2551	c413	ldi 13		
2553	3f	xppc 3		
2554	9200	jmp 2 00		
2556	40	lde		
2557	e4d0	xri d0		
2559	d4f8	ani f8		
255b	9c			

255b	9c0b	jnz 0b	AND
255d	c43f	ldi 3f	
255f	3f	xppc 3	
2560	c422	ldi 22	
2562	3f	xppc 3	
2563	c411	ldi 11	
2565	3f	xppc 3	
2566	9200	jmp 2 00	
2568	40	lde	
2569	e4d8	xri d8	
256b	d4f8	ani f8	OR
256d	9c08	jnz 08	
256f	c42c	ldi 2c	
2571	3f	xppc 3	
2572	c415	ldi 15	
2574	3f	xppc 3	
2575	9200	jmp 2 00	
2577	40	lde	
2578	e4e0	xri e0	
257a	d4f8	ani f8	
257c	9c0b	jnz 0b	XOR
257e	c40f	ldi 0f	
2580	3f	xppc 3	
2581	c42c	ldi 2c	
2583	3f	xppc 3	
2584	c415	ldi 15	
2586	3f	xppc 3	
2587	9200	jmp 2 00	
2589	40	lde	
258a	e4e8	xri e8	
258c	d4f8	ani f8	AND
258e	9c0b	jnz 0b	
2590	c411	ldi 11	
2592	3f	xppc 3	
2593	c43f	ldi 3f	
2595	3f	xppc 3	
2596	c411	ldi 11	
2598	3f	xppc 3	
2599	9200	jmp 2 00	
259b	40	lde	
259c	e4f0	xri f0	
259e	d4f8	ani f8	
25a0	9c09	jnz 09	
25a2	c43f	ldi 3f	
25a4	3f	xppc 3	
25a5	c411	ldi 11	
25a7	3f	xppc 3	
25a8	3f	xppc 3	
25a9	9200	jmp 2 00	
25ab	40	lde	AND
25ac	e4f8	xri f8	
25ae	d4f8	ani f8	
25b0	9c0b	jnz 0b	
25b2	c414	ldi 14	
25b4	3f	xppc 3	
25b5	c43f	ldi 3f	
25b7	3f	xppc 3	
25b8	c411	ldi 11	
25ba	3f	xppc 3	
25bb	9200	jmp 2 00	Adresse von "2. Byte" berechnen
25bd	c10d	ld 1 0d	
25bf	02	22 ne	
25c0	f406	adi 06	
25c2	32	xpal 2	
25c3			

25c3	c10e	1d 1 0e	} "2. Byte - 1" → Ptr. II
25c5	f406	adi 06	
25c7	36	xpah 2	
25c8	c40e	ldi 0e	
25ca	3f	xppc 3	
25cb	c415	ldi 15	
25cd	3f	xppc 3	
25ce	3f	xppc 3	
25cf	c42c	ldi 2c	
25d1	3f	xppc 3	
25d2	c415	ldi 15	
25d4	3f	xppc 3	

ERROR

25d5	9200	jmp 2 00	zu "2. Byte" springen
25d7	40	lde	
25d8	d404	ani 04	
25da	9806	jz 06	
25dc	c41b	ldi 1b	
25de	3f	xppc 3	
25df	c452	ldi 52	
25e1	3f	xppc 3	

"@ ?"      "@"

nein: → 25E2

Blank

@ (&) drucken

25e2	c41b	ldi 1b	} Blank      "Nr. des Pointers"
25e4	3f	xppc 3	
25e5	40	lde	
25e6	d403	ani 03	
25e8	e401	xri 01	
25ea	9c05	jnz 05	
25ec	c43d	ldi 3d	
25ee	3f	xppc 3	
25ef	9016	jmp 16	
25f1	40	lde	
25f2	d403	ani 03	
25f4	e402	xri 02	

Pointer I ?

25f6	9c05	jnz 05	} Pointer II ?
25f8	c408	ldi 08	
25fa	3f	xppc 3	
25fb	900a	jmp 0a	
25fd	40	lde	
25fe	d403	ani 03	
2600	e403	xri 03	
2602	9c03	jnz 03	
2604	c40d	ldi 0d	
2606	3f	xppc 3	

Pointer III ?

2607	c41b	ldi 1b	} Blank      "2. Byte"
2609	3f	xppc 3	
260a	c100	1d 1 00	
260c	lc	kkne	
260d	lc	kkne	
260e	lc	kkne	
260f	lc	kkne	
2610	dca0	ori a0	
2612	01	xae	
2613	c180	1d 1 80	
2615	3f	xppc 3	
2616	c100	1d 1 00	

1. Ziffer d. 2. Byte ausgeben

2618	d40f	ani 0f	} 2. Ziffer d. 2. Byte ausgeben
261a	dca0	ori a0	
261c	01	xae	
261d	c180	1d 1 80	
261f	3f	xppc 3	
2620	c10d	ld 1 0d	
2622	02	22 ne	
2623	f4e2	adi e2	
2625	c91d	st 1 ld	
2627	c10e	1d 1 0e	
2629	f4		

Adresse von "Adresse erhöhen" berechnen

"Adresse erhöhen - 1" → "Rout. Adv"

2629	f406	adi 06	}
262b	c91c	st 1 lc	
262d	3f	xppc 3	2x "Adresse erhöhen" anspringen
262e	3f	xppc 3	
262f	c10d	ld 1 0d	
2631	02	cc1	
2632	f44c	adi 4c	Adresse von "Druckentfernung" berechnen
2634	c91d	st 1 ld	
2636	c10e	ld 1 0e	"Druckentfernung - 1" → "Ront Adr"
2638	f406	adi 06	
263a	c91c	st 1 lc	
263c	c406	ldi 06	
263e	3f	xppc 3	Line feed + CR
263f	c10d	ld 1 0d	
2641	02	cc1	
2642	f4a7	adi a7	
2644	32	xpal 2	
2645	c10e	ld 1 0e	
2647	f401	adi 01	
2649	36	xpah 2	
264a	3e	xppc 2	
264b	00	halt	
264c	00	halt	

Rückprung "Loop"

# "Druckentscheidung"

264d	clee	1d 1 ee	Aus Startadresse und Abstand effektive Adresse von "Print" berechnen
264f	02	ccl	
2650	f41d	adi 1d	
2652	33	xpal 3	
2653	clef	1d 1 ef	
2655	f407	adi 07	
2657	37	xpah 3	"Print-1" → Ptr. III
2658	3f	xppc 3	zeichen drucken (zeichen wird in "Push" in Reg. DFF8 geschrieben)
2659	cle9	1d 1 e9	gedrucktes Zeichen aus Reg. DFE8 laden (siehe "Print")
265b	e406	xri 06	↖ ?
265d	9c6f	jnz 6f	nein: → 26CE
265f	b9f2	d1d 1 f2	Zeilenzähler decrementieren
2661	9c6b	jnz 6b	∅? nein: → 26CE
2663	clf3	1d 1 f3	zweispaltiger Druck gewünscht?
2665	9462	jp 62	nein: → 26C9
2667	clf5	1d 1 f5	1. od. 2. Spalte beendet?
2669	943e	jp 3e	2. Spalte fertig: → 26A9      1. Spalte fertig
266b	clf1	1d 1 f1	teilenanzahl → Zeilenzähler
266d	c9f2	st 1 f2	
→ 266f	clf4	1d 1 f4	Zeilenabstand + 1 → Reg. DFF4
2671	02	ccl	
2672	f401	adi 01	
2674	c9f5	st 1 f5	
→ 2676	c403	ldi 03	↑ → Druckregister DFF8
2678	c9f9	st 1 f9	drucken
267a	3f	xppc 3	
267b	b9f5	d1d 1 f5	
267d	9cf7	jnz f7	
267f	b9f2	d1d 1 f2	Reg. DFF4 decrementieren (Zeilenabstand)
2681	9cec	jnz ec	∅? nein: → 2676
2683	c403	ldi 03	zeilenzähler decrementieren
2685	c9f9	st 1 f9	∅? nein: → 266F
2687	3f	xppc 3	noch einmal "I" und "I" ausgeben
2688	c404	ldi 04	
268a	c9f9	st 1 f9	
268c	3f	xppc 3	
268d	c420	ldi 20	H'20 (D'32) mal Blank ausgeben
268f	c9f5	st 1 f5	
2691	c41b	ldi 1b	

2693	c9f9	st 1 f9	
2695	3f	xppc 3	
2696	b9f5	d1d 1 f5	
2698	9cf7	jnz f7	
269a	c400	ldi 00	
269c	c9f9	st 1 f9	{ ↪ ausgeben
269e	3f	xppc 3	
269f	c400	ldi 00	{ Reg. OFF4 auf "2. Spalte"
26a1	c9f5	st 1 f5	
26a3	clf1	ld 1 f1	{ zeilenanzahl → Zeilenzähler
26a5	c9f2	st 1 f2	
26a7	9025	jmp 25	→ 26CE

---

→ 26a9 c439 ldi 39 { ↪ ausgeben      2. Spalte fertig

26ab c9f9 st 1 f9 { ↪ ausgeben

26ad 3f xppc 3

26ae c437 ldi 37 { ↪ ausgeben

26b0 c9f9 st 1 f9 { ↪ ausgeben

26b2 3f xppc 3

26b3 c40b ldi 0b

26b5 c9f5 st 1 f5

26b7 c41b ldi 1b

26b9 c9f9 st 1 f9

26bb 3f xppc 3

26bc b9f5 d1d 1 f5

26be 9cf7 jnz f7

26c0 c400 ldi 00

26c2 c9f9 st 1 f9

26c4 3f xppc 3

26c5 c4ff ldi ff

26c7 c9f5 st 1 f5

→ 26c9 clf1 ld 1 f1

26cb c9f2 st 1 f2

26cd 00 halt

→ 26ce c408 ldi 08

26d0 33 xpal 3

26d1 c407 ldi 07

26d3 37 xpah 3

26d4 c300 ld 3 00

26d6 e490 xri 90

11× Blank ausgeben

↪ ausgeben

Reg. OFF4 auf "1. Spalte"

zeilenanzahl → Zeilenzähler

Stop

Abr. KBd im Btr. III

Keyboard laden

"down"-Taste ?

26d8 9c01 jnz 01 ja : halt  
26da 00 halt  
26db >c400 ldi 00 }  
26dd 37 xpah 3 } "Pull - 1" → Pv. III  
26de c414 ldi 14 }  
26e0 33 xpal 3 }  
26e1 3f xppc 3 zurück  
26e2 00 halt

## "Adresse erhöhen"

26e3	clec	1d 1 ec		(Ptr. I steht von "Push" auf 0FFF)
26e5	01	xae		Anfangsadresse (ff) lower = Endadresse - lower ?
26e6	cle2	1d 1 e2		
26e8	60	xre		nein: → 270E
26e9	9c23	jnz 23		
26eb	cled	1d 1 ed		Anfangsadresse (ff) higher = Endadresse higher ?
26ed	01	xae		
26ee	cle3	1d 1 e3		
26f0	60	xre		nein: → 270E
26f1	9clb	jnz 1b		
26f3	c100	1d 1 00		
26f5	32	xpal 2		STAKPT (in Adr. 0FFE u. 0FFF) → Ptr. II
26f6	clff	1d 1 ff		
26f8	36	xpah 2		
26f9	clee	1d 1 ee		
26fb	02	ccl		Adresse zum Einprung berechnen (hier: 2707)
26fc	f406	adi 06		und in den letzten Stack in das Register für Ptr III eintragen
26fe	ca03	st 2 03		
2700	clef	1d 1 ef		
2702	f407	adi 07		
2704	ca02	st 2 02		
2706	3f	xppc 3		zurück: d.h über "Pull" auf Adr. 2707
→ 2707	c400	ldi 00		
2709	37	xpah 3		"Pull-1" → Ptr. III
270a	c414	ldi 14		
270c	33	xpal 3		
270d	3f	xppc 3		zurück in das Programm, aus dem das "Eigentliche Programm" als Subroutine aufgerufen wurde
→ 270e	clec	1d 1 ec		
2710	02	ccl		Anfangsadresse (ff) incrementieren
2711	f401	adi 01		
2713	c9ec	st 1 ec		
2715	cled	1d 1 ed		
2717	f400	adi 00		
2719	c9ed	st 1 ed		
271b	3f	xppc 3		zurück ins "Eigentliche Programm"
271c	00	halt		
271d	00	halt		

# "Print"

271e 06 csa } Status in Reg. &FE5 zurücksetzen  
 271f c9e6 st 1 e6 }  
 2721 d4f7 ani f7 } DINT  
 2723 07 cas }  
 2724 37 xpah 3 }  
 2725 c9e5 st 1 e5 } Ptr. III in Reg. &FE4 (higher) u. &FE3 (lower)  
 2727 33 xpal 3 } zurücksetzen  
 2728 c9e4 st 1 e4 }  
 272a c411 ldi ll }  
 272c 36 xpah 2 } Ptr. II auf LED's  
 272d c400 ldi 00 }  
 272f 32 xpal 2 }  
 2730 c4cb ldi cb }  
 2732 ca00 st 2 00 } NData an Schreibmaschine ausgeben  
 2734 c4cf ldi cf }  
 2736 ca00 st 2 00 }  
 2738 c1f9 ld 1 f9 } zu druckendes Zeichen aus Reg &FF8 laden  
 273a c9e8 st 1 e8 } und in Reg. &FE7 zurücksetzen (neu)  
 273c 9002 jmp 02 }  
 273e 90de jmp de } Hilfsjump: 27A2 → 273E → 271E  
 01 xae }  
 2740 c408 ldi 08 } Reg. &FF8 als Zahl laden } Zeichen seriell  
 2743 c9f9 st 1 f9 }  
 → 2745 19 sio } ausgeben  
 2746 c4ce ldi ce }  
 2748 ca00 st 2 00 } Clock  $\overline{L}\overline{T}$   
 274a c4cf ldi cf }  
 274c ca00 st 2 00 }  
 274e b9f9 dld 1 f9 } 8x ?  
 2750 9cf3 jnz f3 }  
 2752 c4cd ldi cd }  
 2754 ca00 st 2 00 } NData ready an Schreibmaschine  
 2756 c406 ldi 06 }  
 2758 37 xpah 3 }  
 2759 c400 ldi 00 } Ptr. III auf Adresse des Multiplexers  
 275b 33 xpal 3 }  
 → 275c c404 ldi 04 } Reg. &FF8 als Zahl laden  
 275e c9f9 st 1 f9 }  
 → 2760 c400 ldi 00 }  
 2762 8f01 dly 01 } DLY ~ 1msec  
 2764 c300 ld 3 00 } liegt NData ready von Schreibmaschine an?  
 2766 9402 jp 02 } ja: → 2786  
 2768 901c jmp 1c }  
 276a b9f9 dld 1 f9 } nein: Zähler dekrementieren;  
 276c 9cf2 jnz f2 }  
 276e c4cf ldi cf } DDL ja: weiter  
 2770 ca00 st 2 00 } NData ready zurücknehmen  
 2772 8f60 dly 60 } DLY 60  
 2774 c300 ld 3 00 } liegt NData ready von Schreibmaschine jetzt an?  
 2776 9402 jp 02 } ja: → 2786  
 2778 900c jmp 0c }  
 277a c4cb ldi cb }  
 277c ca00 st 2 00 } NData Impuls ausgeben, NData ready inaktiv  
 277e c4cf ldi cf }  
 2780 ca00 st 2 00 }  
 2782 cle8 ld 1 e8 } zu drucken des Zeichen aus Reg &FE7  
 2784 90ba jmp ba } laden und Rückprung zu erneuter Ausgabe  
 → 2786 c300 ld 3 00 }  
~~2788 94d2~~

2788	94d2	jp d2	}	NData ready - Signal von Schreibmaschine liegt noch an? (= Entprellung) nein: → 275C
278a	c4cb	ldi cb		NACK - Impuls ausgeben, NData ready inaktiv
278c	ca00	st 2 00		
278e	c40f	ldi 0f		
2790	ca00	st 2 00		
2792	8f40	dly 40	}	Entprellzeit für Schreibmaschine gedrucktes Zeichen → Reg 0FE8
2794	cle8	ld 1 e8		
2796	c9e9	st 1 e9		
2798	cle5	ld 1 e5		
279a	37	xpah 3		
279b	cle4	ld 1 e4	}	Ptr. III wieder auf alten Stand
279d	33	xpal 3		
279e	cle6	ld 1 e6		
27a0	07	cas		
27a1	3f	xppc 3		
27a2	909a	jmp 9a	}	Rücksprung
27a4	00	halt		
27a5	31	xpal		

Status wieder auf alten Stand  
 Rücksprung  
 Rücksprung → 271E  
 (dadurch mehrfachen Anprung mit XPPC 3 möglich)