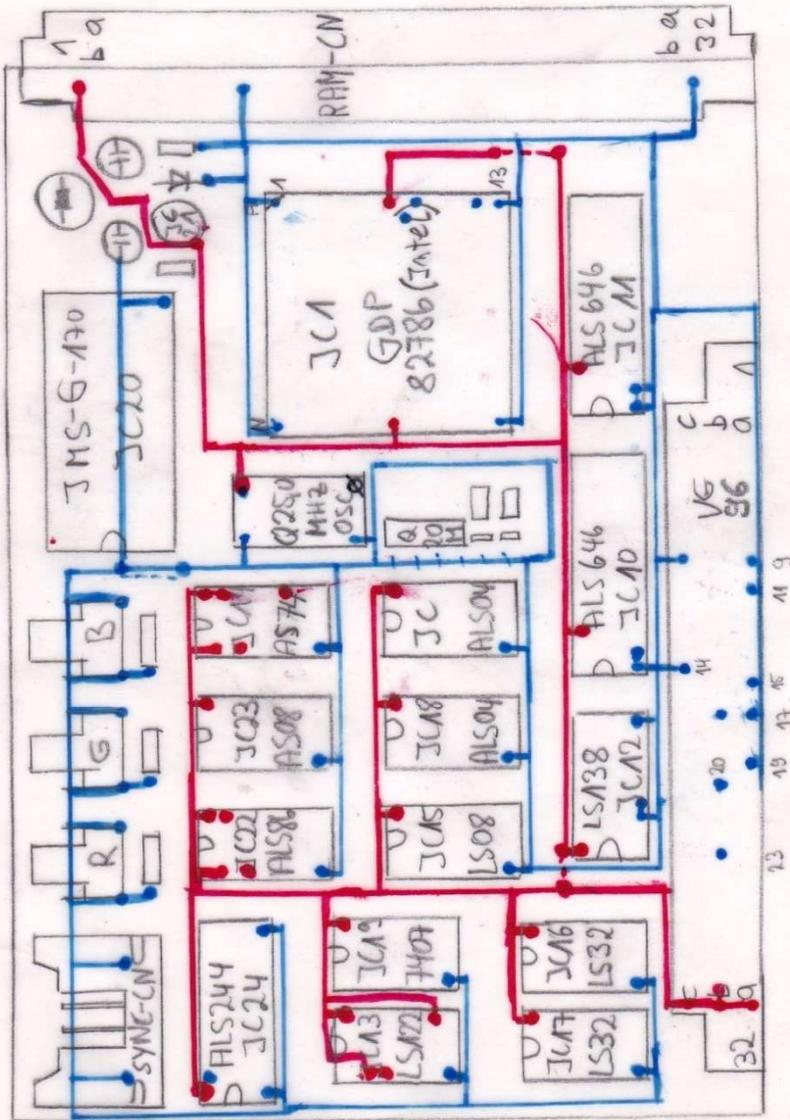


Top-View

GDP Card für 68k board
1.11.1990, Randolph Esser

RAM-Card
(max 40MB)

Bestückungs-Seite



Graphik-Steuerenteil

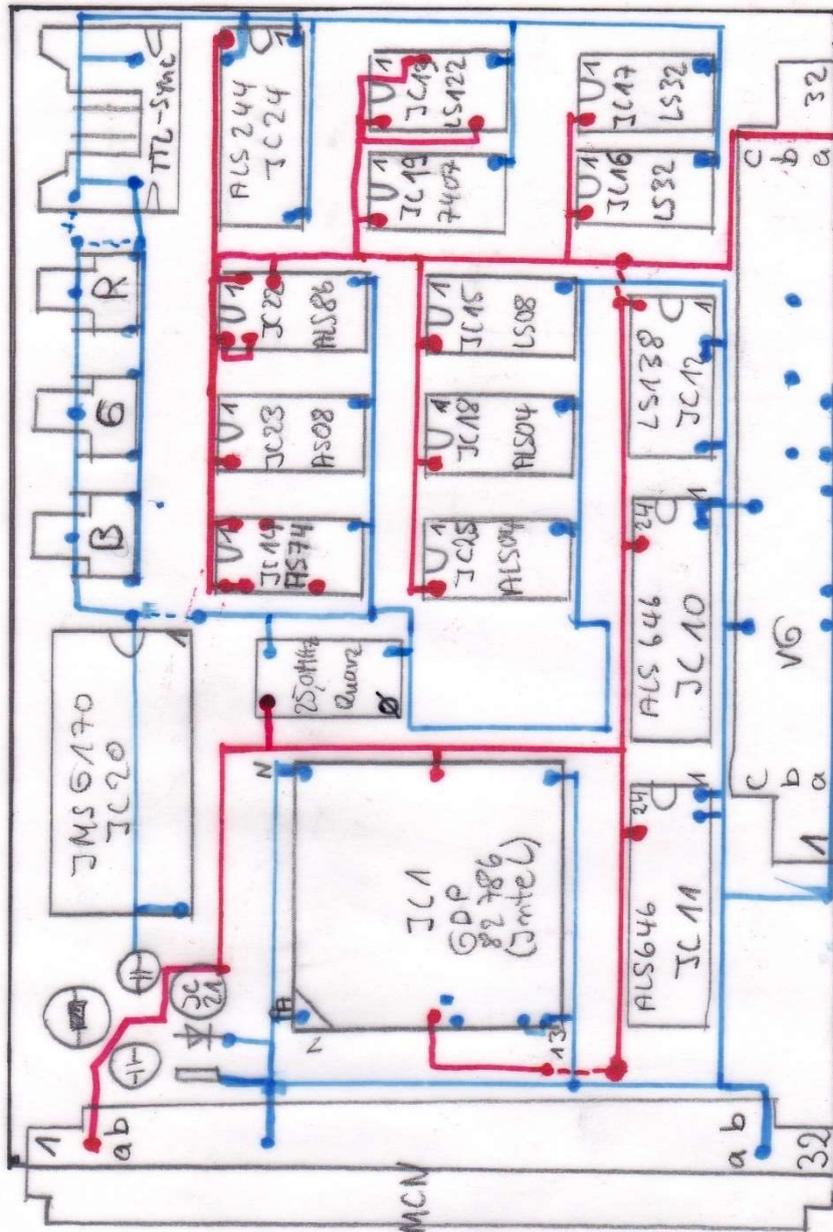
mc 68000 Board

Bottom - View

1. M. 1390 R. Esser

RAM-card. \leftarrow RAMCN
(max 4MB)

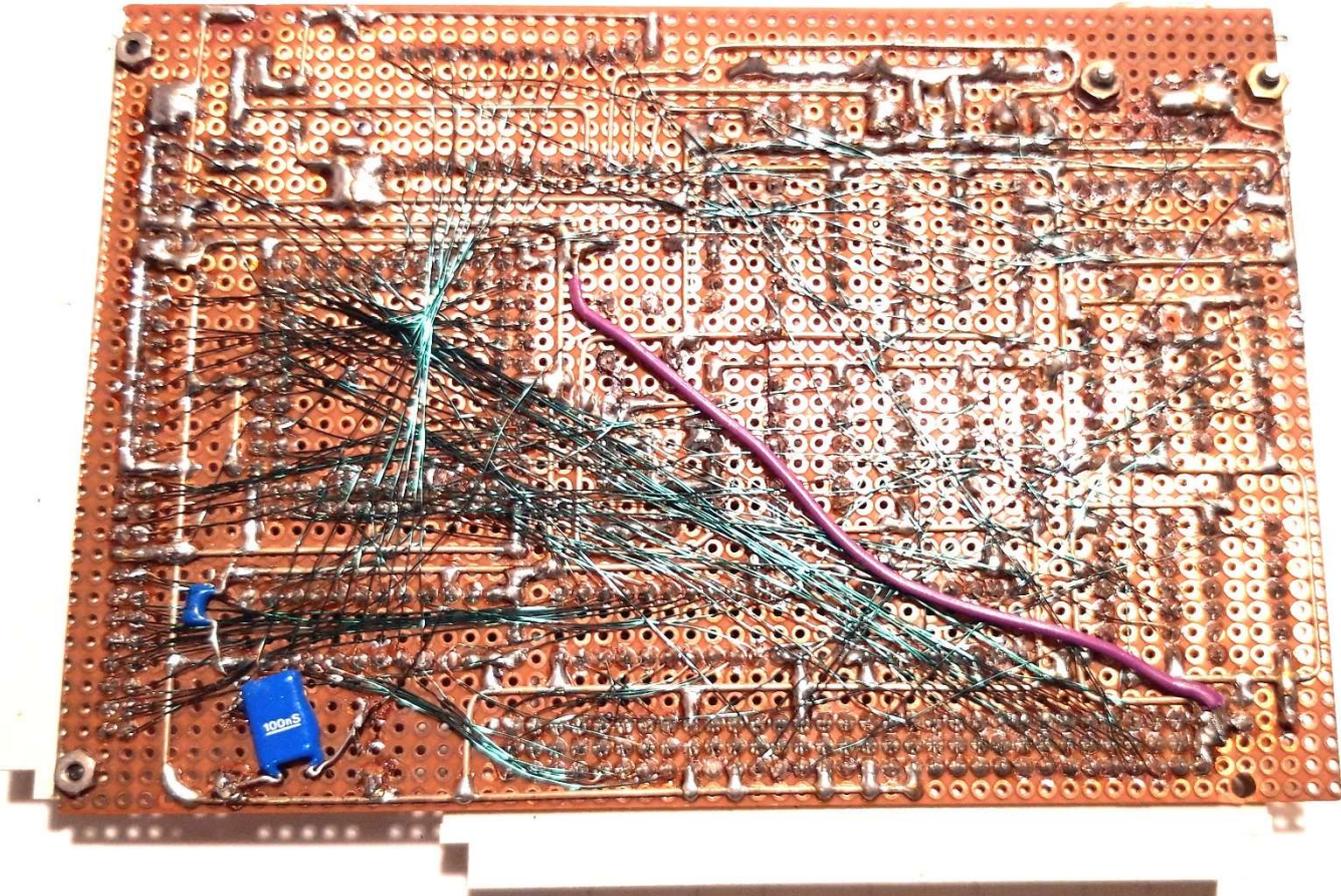
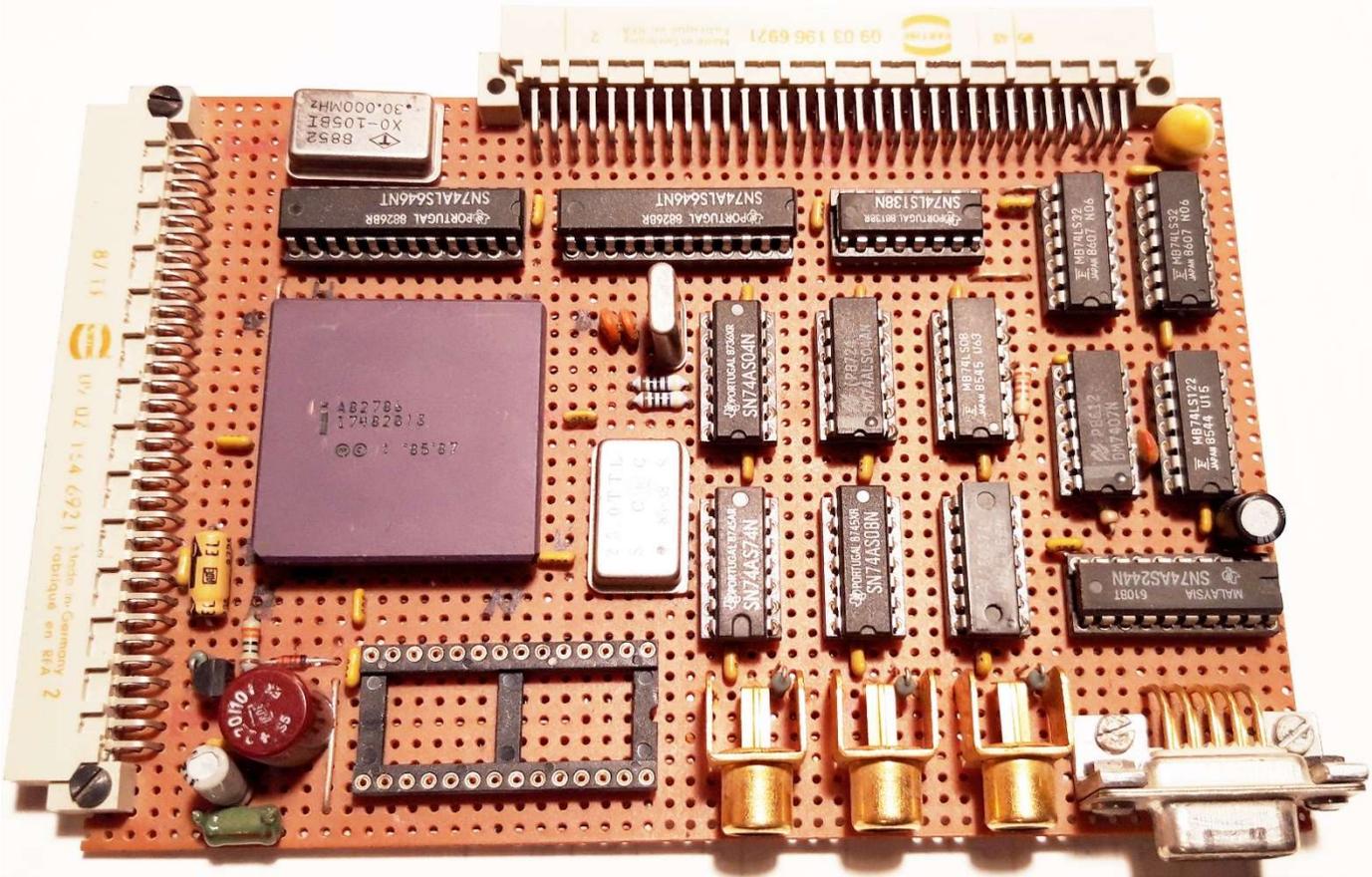
Löt-Seite



Graphik-Steuer teil

MC 68000

GDP68k – GPU Board - Top + Buttom View

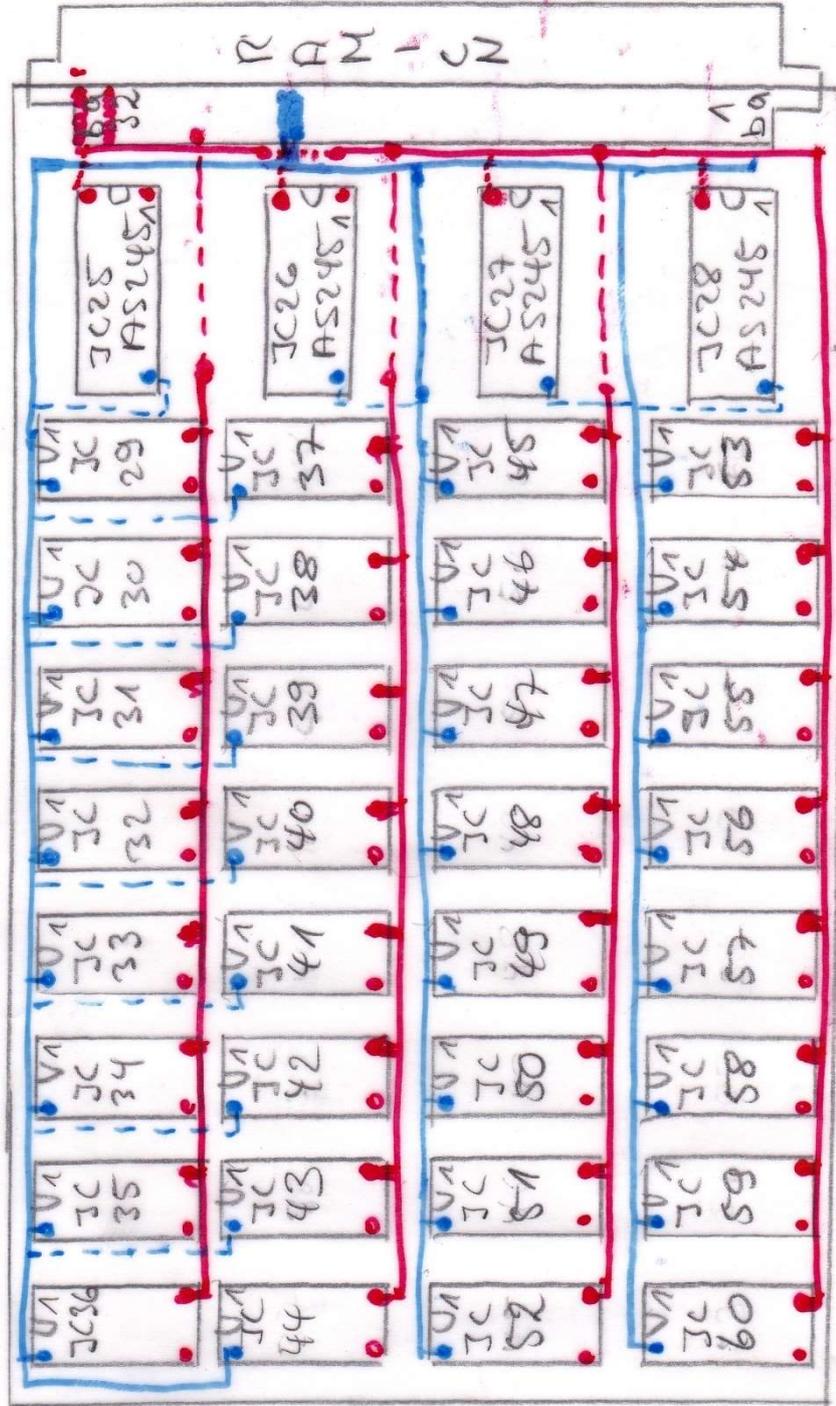


Bottom-View: GDP-Ram - CARD (64K---1M)

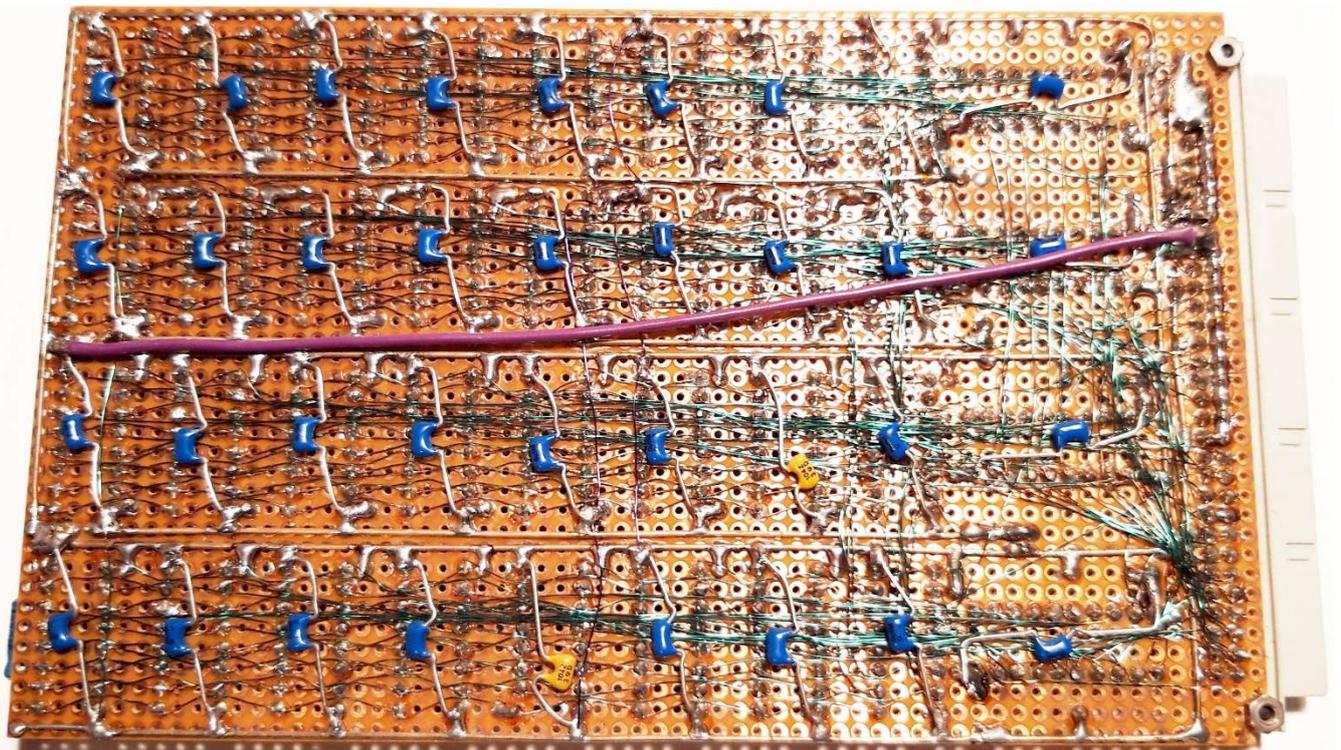
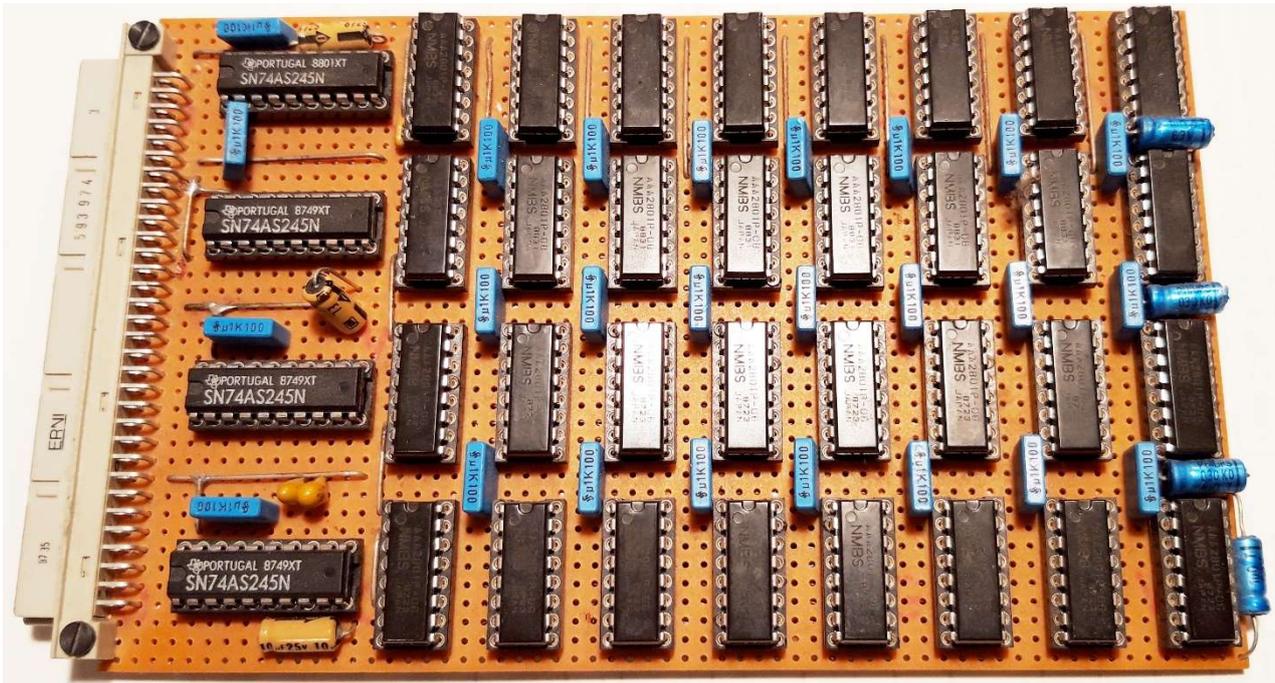
Löt-seite

1.11.1990

R. Esser



GDP68k DRAM (1MB) Board – Top & Bottom View

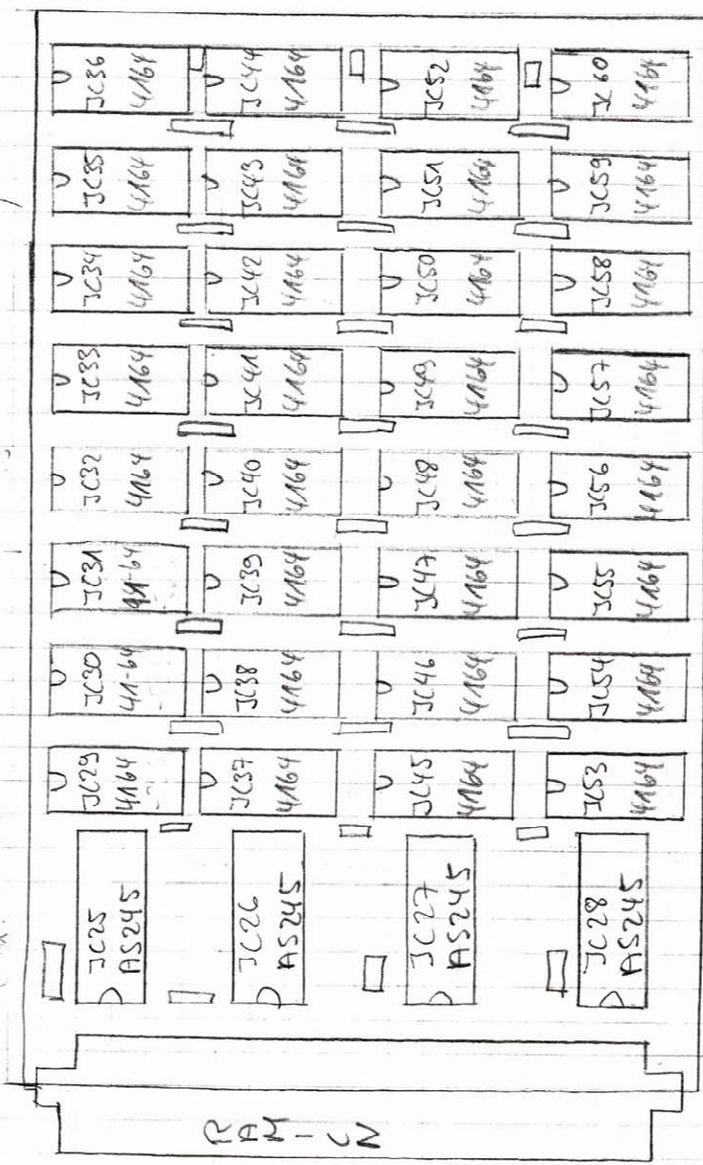


RAM-Connector:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
a	+	+	NC	NC	NC	NC	NC	+	-	-	NC	NC	-	WEH	CAS0	BEM0	-	D15	D14	D13	D12	D11	D10	D9	D8								
b	+	+	NC	NC	NC	NC	+	-	-	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	-	WEL	CAS1	BEM1	-	D7	D6	D5	D4	D3	D2	D1	D0
				RAS2	RAS2	RAS1	RAS1				DR18	DR17	DR16	DR15	DR14	DR13	DR12	DR11	DR10														

GDP- RAM-Card

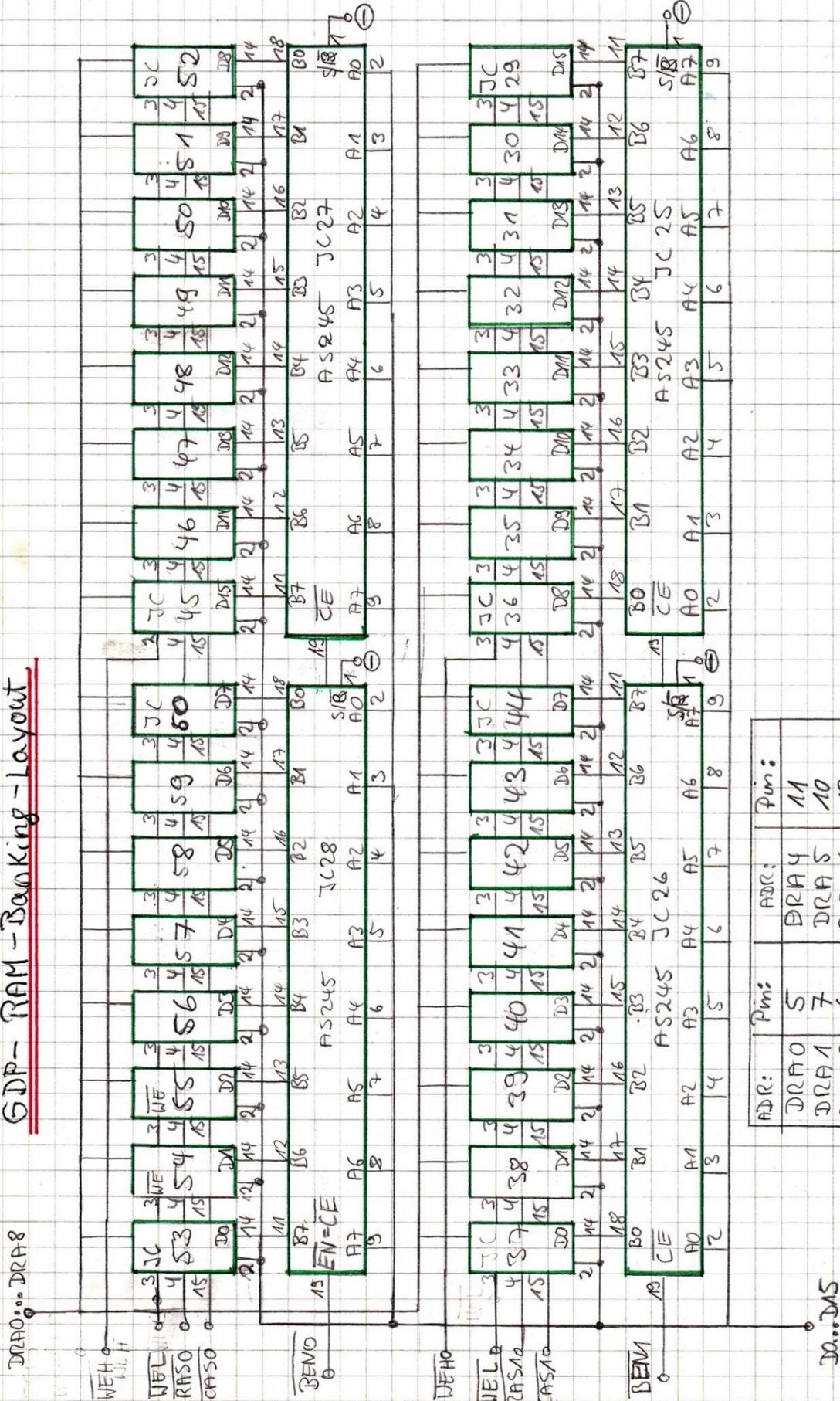
(4164: 64...256KByte; 41256: 256... 1MBByte)



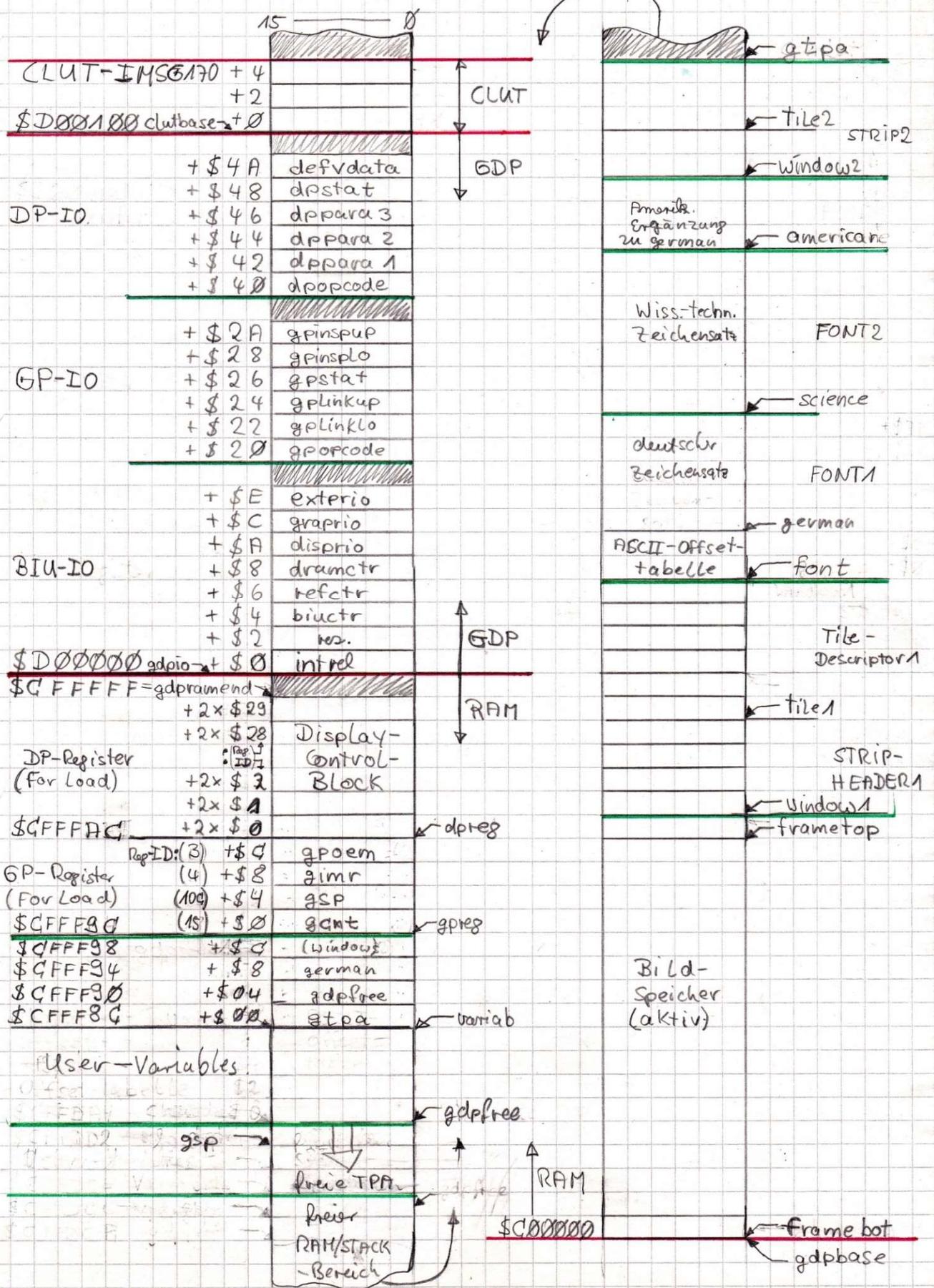
RAM-CN

GDP-Card (82786)

GDP - RAM - Banking - Layout



GDP-Speicher aufteilung



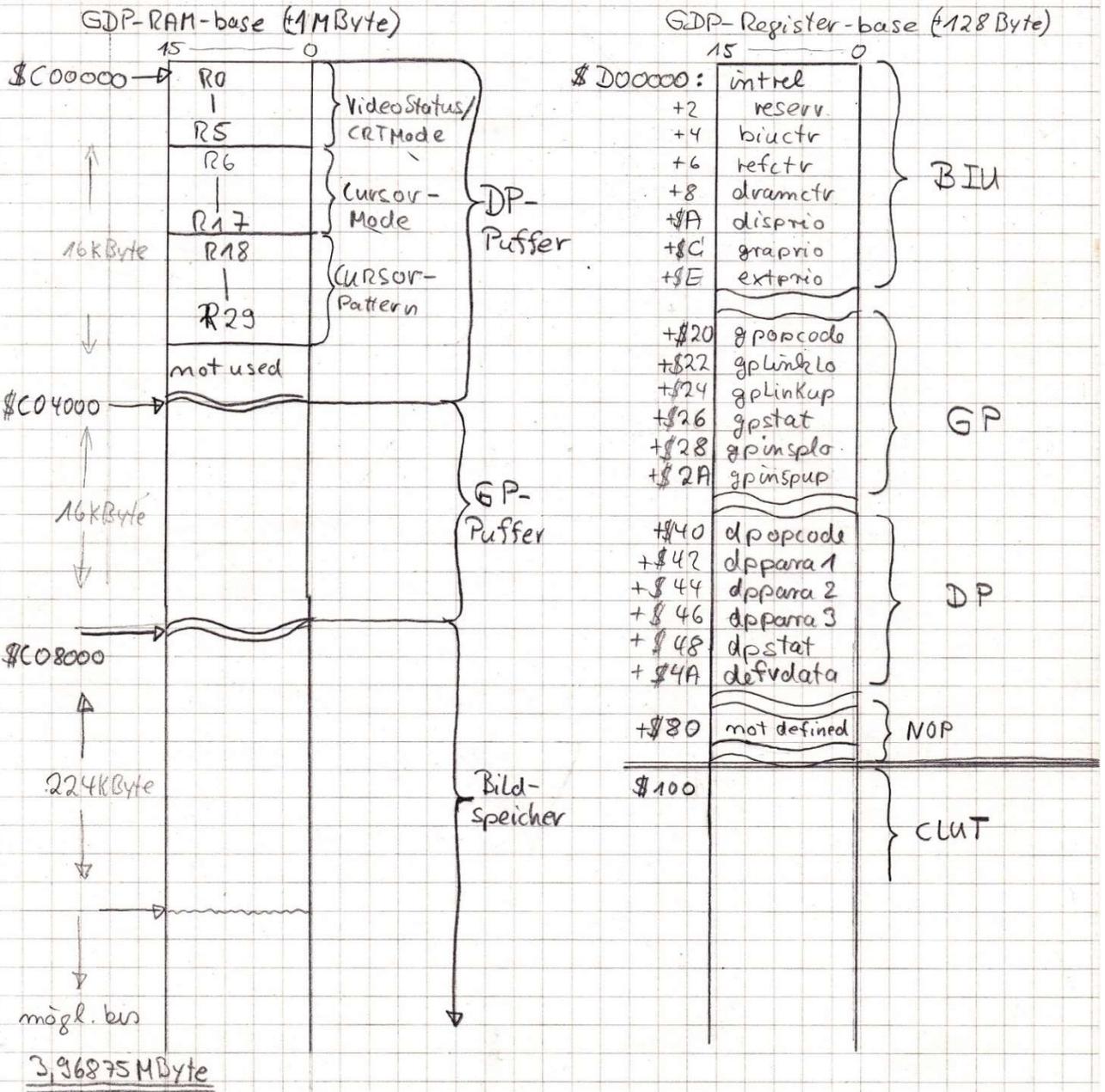
Int.-Level : # 3 Vector: \$80 → \$68CCC (JRG-routine)

Slottabelle:

\$FFF870: \$0702 8000 00C0 0000 0006 8EAC

entry:

	Kennung	GDP-base	GDP-UP-base-pointer
\$68EAC :	\$ 68CF0	ginit	; Sprungtabelle für
	\$ 68D3A	move_abs	
	\$ 68D84	move_rel	; GDP-standard-routinen!
	\$ 68DCE	line	
	\$ 68E18	circle	
	\$ 68ESC	rectan	



Offset: Variable

Init Symbol/Wert

Funktion

Offset	Variable	Init Symbol	Wert	Funktion
+4	RUNFLAG		\$0	Wait-Flag f. Rom-Wait-Routine
+6	IOBYTE	IOBYINIT	\$1	CRT=Bit0, LCD=Bit1, PRN=Bit2
0A	XSYZE	XSIZEIN	810	abt. Bit-size des Grafike -
+0A	YSYZE	YSIZEIN	600	Bitfeldes
0E	CURSMAX	curshm	80-1	Max. Anz. der Zeichen in
10	CURSYM	curshm	80-1	X und Y - Richtung
12	CURADR	textbot + \$0	\$0A70	Cursoradr. im Grafispeicher
16	CURSX	curshx	0	Cursor-home position im
18	CURSY	curshy	0	Zeichenfeld. (0,0)
1E	SPOTXHOME	spotxh	5	Spot-homeposition im
20	SPOTYHOME	spotyh	3	Bitfeld (Grafispeicher) (5,3)
26	SPOTX	spotxh	5	abt. Spot-Position im
28	SPOTY	spotyh	3	Grafik - Bitfeld
1A	CURSMHOME	curshx	0	Cursor-home-Position im
1C	CURSMHOME	curshy	0	Zeichenfeld.
2A	STEPX	stepix	10	Schrittweite des Cursors im
+2A	STEPY	stepiy	19	Bitfeld. / Zeichen (x, y)
2E	TABSTEP	itab	8	Tab-Schrittweite des Cursor im Z.-bild.
30	CHARTYP	CCHARTR	\$A7	Zeichenausgabemod f. GOMGDP
32	BUSY		0	Busyflag f. GDP-IRQ-Serv.
22	PIXCOL		\$BFBF	abt. Zeichenfarbe
24	BACKCOL		\$3030	Hintergrundfarbe auf d. d. Windows
08	IRQMEMB		0	IRQ-Verster-No. of last Interrupt.

Textspeicher darstellen:

CURSADR C76A74
 CURSX 4
 CURSY 0
 SPOTX 45
 SPOTY 3

~~CCHARTR~~ = 8 1740

C MOVABS
 C ~~home~~ x
 C ~~home~~ y
 call
 +Line

+Line: CCHARTR
 Line m low
 Line n up
 #80

text top
 text bot = zeilemax * spaltenmax - 1
 Textseite

VARTAB
 VARIEND

15FC 000010001

~~15FS~~

RAMTOP 1FBDC
 FDFC

CONG-DCX

~~Auf Eprom-Mode schalten.~~

~~IRQ-Tab kopieren -> \$0 - \$400 1100~~

~~RAM testen.~~

~~RAMTOP -> A 6
 RAMBOT -> A 6
 RAMTXT -> D 5~~

~~+PATOLTX
 -PATOPTX~~

CURS X MAX 79
 CURS Y MAX 29

CURS X } Cursor-Position / Zeichenfeld 0-79
 CURS Y } 0-29
 CURSADR } Offsetadr. d. Cursor im Zeichenfeld (Textbot + 0)
 SPOT X } Pixel-Cursor / Bitfeld. 0-809
 SPOT Y } 0-599

SPOT X MAX 809 = XSIZE - 1
 SPOT Y MAX 599 = YSIZE - 1

XSIZE 1 - XFRAME | XFrame 810
 YSIZE 1 - YFRAME | YFrame 600

CURS X HOME 0 } Cursorhome im Zeichenfeld
 CURS Y HOME 0

SPOT X HOME 5 } Pixel-Cursorhome im Bitfeld.
 SPOT Y HOME 3

~~PIXCOL~~ \$BFBF Zeichenfarbe
 BACKCOL \$0000 Hintergrundfarbe (aufhalb der Windows)

STEPX 10 Schrittweite des Cursors
 STEPY 7 bei Zeichenbewegungen