

Combo

**GRAPHIC
BOOSTER 128™**

(C) T. GIGER 1986

GRAPHIC BOOSTER 128™

Alle Rechte vorbehalten.

Urheberrecht und Copyright (C)1986 liegen bei T.Giger Solothurn Schweiz.
COMBO AG vertritt alle Rechte. Soft-und Hardware dürfen weder verviel-
fältigt noch veröffentlicht werden. Gerichtstand Solothurn Schweiz.

**COMBO AG
TUGGINERWEG 3
CH-4500 SOLOTHURN
TEL. 065 23 26 86**

Sehr geehrter GRAPHIC BOOSTER Besitzer,

Mit dem Erwerb des GRAPHIC BOOSTER steigen Sie in die Klasse der professionellen CAD Anwender ein. Die Möglichkeiten, die Ihnen diese Software bietet, ist einzigartig durch alle Reihen der Computergrafik. Das System ist so anwenderfreundlich und besitzt eine solche Vielzahl von professionellen Routinen, dass die Erstellung von 1.klassiger Software zum Genuss wird.

Vergleicht man andere Computersysteme mit gleichen Features, so können Sie feststellen, dass andere Grafikkarten mit Software (sofern überhaupt erhältlich) mindestens 10 mal teurer zu stehen kommen! Und dann bedingt das natürlich auch einer 10 mal teureren Hardwarekonfiguration!

Seit 1986 haben uns viele Programmierer Software zugeschickt, die sie mit dem GRAPHIC BOOSTER erstellt haben. Es hat bei uns einen richtigen Highresolution-Pool gegeben mit Programmen aus allen möglichen Gebieten. Denn jedes Programm, das mit einer Superauflösung des GRAPHIC BOOSTER läuft ist Spitze! Deshalb haben wir uns entschlossen, für Sie eine Kontaktstelle für GRAPHIC BOOSTER Anwenderprogramme zu gründen. Und das funktioniert so:

Jeder, der in den GRAPHIC BOOSTER Pool ein Programm einsendet, bekommt eine Diskette voll(ca.20 Programme) neue Superhighresolution Programme. Sie können Programme jeder Größe und aus beliebigem Gebiet einsenden. Hier einige Anregungen: Mathematische Kurven, 1 oder 2 Monitorprogramme, CAD, Spiele, Malprogramme, Symbole, digitalisierte Bilder, Tools also Unter-routinen, Windowtechnik im Textmodus wie beschrieben GRAPHIC BOOSTER und der Textmodus und und und.....

GRAPHIC BOOSTER (C) 1986/87/88/89 Combo AG

Der GRAPHIC BOOSTER ist ein Developers package, dessen Rechte alle bei Combo AG, Solothurn, Schweiz liegen.

Bitte beachten Sie die Schutzrechte des GRAPHIC BOOSTER 128 sowie der vorliegenden Beschreibung:
Der GRAPHIC BOOSTER ist ein Produkt der Combo AG und wird ausschliesslich durch Combo AG vertrieben.
Alle Rechte sind vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung ist nicht zulässig.
Der GRAPHIC BOOSTER darf nicht zu Lehrzwecken verwendet und nicht öffentlich vorgeführt werden.
Der GRAPHIC BOOSTER darf weder ganz noch teilweise veröffentlicht oder besprochen werden.
In jedem Fall von Verbreitung, Besprechung und Veröffentlichung bedarf es der schriftlichen Genehmigung der Firma Combo AG.

Im Besonderen sei darauf hingewiesen, dass der Bildschirmtreiber des GRAPHIC BOOSTER für Auflösungen grösser als 640 x 200 und speziell der Interlace Modus für die hohen Auflösungen unter das weltweite Copyright 1986/87/88/89 Combo AG fallen!

Adresse: Combo AG
GRAPHIC BOOSTER Pool
Tugginerweg 3
CH-4500 Solothurn

Befehlsklärung

Programmiert wird in Basic 7.0. Orientieren Sie sich anhand des C-128 Handbuchs.

neue Befehle: GRAPHIC1,1 schaltet die Grafik ein und löscht den Bildschirm. Interlaced Modus 640x720 b.z.w. 720x700 Punkte.
GRAPHIC1,0 schaltet die Grafik ein ohne den Bildschirm zu löschen.
GRAPHIC2,1 schaltet die Grafik ein und löscht den Bildschirm. Normalmodus 640x360 b.z.w. 720x360 Punkte.
GRAPHIC2,0 schaltet die Grafik ein ohne den Bildschirm zu löschen.
GRAPHIC0 kehrt von der Grafik in den Textmodus zurück. (Nur beim Arbeiten mit einem Monitor und bei gedrückter 80 Zeichentaste).

GRAPHIC3,1 720x400 oder 640x400, je nach Auswahl im Startprogramm.
GRAPHIC3,0 Einschalten ohne Bildschirm löschen.

GRAPHIC4 wie GRAPHIC2, aber 2. Bildschirm
GRAPHIC4,2 schaltet GRAPHIC4 ein und kopiert GRAPHIC 2 nach GRAPHIC 4
GRAPHIC4,3 schaltet GRAPHIC4 ein, zeichnet aber in GRAPHIC2
GRAPHIC4,4 schaltet GRAPHIC4 ein und kopiert GRAPHIC4 nach GRAPHIC2
GRAPHIC4,5 schaltet GRAPHIC4 ein und löscht GRAPHIC2

Mit diesen Befehlen kann im Versteckten gezeichnet werden und auf Befehl das ganze Bild hervorgeholt werden. Ebenfalls kann von der 1750 Exp. versteckt geladen werden. Bei GRAPHIC2 hat die 2. Kommastrichstelle dieselben Auswirkungen, wie bei GRAPHIC4.

Bei Multicolorbetrieb ist nur GRAPHIC1 aktiv.

TEXT(0-1),x,y,(Zeichenkette),(invers) entspricht dem Charbefehl des Basic 7.0 schreibt aber nur auf den Textschirm

OLD installiert ein durch NEW gelöschtes Basicprogramm

OFF schaltet den GRAPHIC BOOSTER aus (Einschaltzustand)

Die Grafikbefehle für den 80Zeichenschirm:

GRAPHIC

DRAW	Syntax wie Basic 7.0 von 40Zeichenschirm
BOX	" " "
CIRCLE	" " "
SSHAPE	" " "
GSHAPE	" " "

Die Grafikbefehle für den 40Zeichenschirm:

VGRAPHIC	Syntax wie Basic 7.0
VDRAW	" "
VDRAW	" "
VBOX	" "
VCIRCLE	" "
VPAINT	" "
VCHAR	" "
VSSHAPE	" "
VGSHAPE	" "
VCOLOR	" "
VWIDTH	" "
VSCNCLR	" "
VSPRITEDEF	" "

Die Befehle können beliebig gemischt werden die gleichen Shapes können auf beiden Bildschirmen verwendet werden. Sprites können während des Programmablaufes definiert (VSPRITEDEF) und in die Shapes des 80Zeichenschirmes übertragen werden!
Sprites zusammen mit den BYTE und FBYTE Befehlen erlauben einen kontinuierlichen Übergang vom 40Zeichen- zum 80Zeichenmonitor. Will man in x-Richtung ebenfalls ein pixelweises Verschieben geht man folgendermassen vor:
Mit VSSHAPE oder mit SPRSAV die Matrix in eine Stringvariable übernehmen, mit GSHAPE 7mal um ein Pixel versetzt am unteren Rand 80Z abspeichern und mit FBYTE swapen. Solche Tools finden immer wieder Verwendung und sollten bei jeder Programmiersitzung erreichbar sein.
Wie oben schon erwähnt bauen Sie solche Tools folgendermassen auf: die ersten paar REM Zeilen geben Auskunft über die Eingabe- und die Ausgabe-parameter dann kommt die Funktion und mit einem RETURN wird abgeschlossen. Bei einem späteren MERGE müssen die Zeilennummern nicht vor dem Laden neu nummeriert werden wenn sie von Anfang an solche Tools mit hohen Zeilennummern versehen und für das gleiche Tool immer die gleiche Startzeilennummer gebrauchen.

GRAPHIC BOOSTER und der Textmodus

Laden Sie eine Auflösung zweifarbig und 640 breit Höhe spielt keine Rolle. Die Grafikbefehle eignen sich sehr gut für den Textmodus. BYTE,FBYTE und auch SSHAPE und GSHAPE erlauben eine Bearbeitung wie sie es sich bisher nur erträumten. Echte Windows und Bildschirmbereiche füllen, Charakterpokes Attributpokes setzen und löschen einzelner Bits(Unterstreichen, Blinken,Revers,Farbe und Intensität setzen,Manipulation des Zeichensatzes wie ganz austauschen, einzelne Charakter swapen. Denn GRAPHIC BOOSTER verwaltet im Textmodus volle 64K!
In medias res: S/GSHAPE nimmt 3 Textzeilen 80 Zeichen breit auf oder 251/Breite=Höhe oder 15 breit 15 hoch. Die X-Koordinate errechnet sich folgendermassen: x mal 8 da das Koordinatensystem von SHAPE in X-Richtung 640 breit ist.
z.B. POKE13480,PEEK(2606):SSHAPE\$,20x8,4,35x8,14:PRINT"CLR HOME":GSHAPE\$,15x8,10

POKE13480,PEEK(2606) entspricht dem Start des Text Bildschirms wollen Sie das Attribut des entsprechenden Zeichens verändern oder kopieren schreiben Sie:
POKE13480,PEEK(2607) LOCATE0,0 entspricht dann dem Attribut links oben in der Ecke.
Setzen Sie folgende Parameter vor dem Start:POKE2606,0:POKE2607,16: BANK15:SYS52684,16,20:GRAPHIC4:GRAPHICO
z.B.Text auf den Bildschirm
POKE13480,0:PRINT"CLR HOME":FORI=0TO4:PRINT"GRAPHIC BOOSTER ";:NEXT FBYTE1,0,1,80:DRAW,0,0TO0,20
z.B. unterstreichen von Text und wieder aus
POKE13480,PEEK(2607):WIDTH20,10:BYTE2,32:DRAW,10,10:SLEEP2:DRAW,10,10

Systemverbesserungen:Die 1750 Ramexpansion kann jetzt viel einfacher betrieben werden.Die Befehle STASH, FETCH und SWAP funktionieren auch im Fastmodus.Bank1 kann auch von der Ramexpansion erreicht werden. Man schreibt einfach Bank1
Negative Koordinaten führen nicht zu einem illegal Quantity. Der Charbefehl wird jetzt korrekt ausgeführt. Vorher war es nicht möglich den CHAR Befehl auf dem 80Zeichenschirm zu verwenden.

Übernahme von Programmen der GRAPHIC BOOSTER Version Nummer 1:
Listing auf Bildschirm listen; ca.15 zeilenweise und mit RETURN übernehmen.
Der COLOR Befehl muss korrigiert werden.
Dann abspeichern und Testlauf.

Drucken: Mit Epson oder IBM kompatiblen Druckern
 Syntax: Epson(mode),(Geräteadresse),(Anzahl Zeilen mal 8)
 Mode 0-13 Geräteadresse 4-7 Anzahl Zeilen 1-255
 Mode 0-6 die Ausgabe erfolgt über den seriellen Bus
 Mode 7-13 die Ausgabe erfolgt über Centronics am Userport (entsprechendes Spezialkabel ist im Fachhandel erhältlich) Mode 0=7 1=8 usw
 mode0 480 Punkte pro Zeile (esc"K")
 mode1 960 Punkte pro Zeile (esc"L")
 mode2 960 Punkte pro Zeile (esc"Y") Fast
 mode3 1920 Punkte pro Zeile (esc"Z")
 mode4 640 Punkte pro Zeile
 mode5 576 Punkte pro Zeile
 mode6 720 Punkte pro Zeile

Beispiel: EPSON4,4,65

IBM(mode),(Geräteadresse),(Anzahl Zeilen mal 8)
 Mode 0-8
 Mode 0-3 seriell
 Mode 4-8 parallel Cenrtonics 4=0 5=1 usw
 Mode 0 480 Punkte pro Zeile
 Mode 1 960 Punkte pro Zeile
 Mode 2 960 Punkte pro Zeile Fast
 Mode 3 1920 Punkte pro Zeile

Die Sekundäradresse ist in 16248. Voreingestellt ist 4
 Sekundäradresse 1 einstellen SCREENB:POKE 16248,1
 640 breit drucken mit MPS 1000 SCREENB:POKE14814,32:POKE14314,7
 IBM3,4,65
 Rücksetzen SCREENB:POKE14814,76:POKE14314,2
 Drucken GRAPHIC2 voreingestellt ist GRAPHIC2 SCREENB:POKE14782,127
 GRAPHIC4 SCREENB:POKE14782,255
 Linefeed aus: SCREENB:POKE12250,13 ein: SCREENB:POKE12250,10

SCREENB schaltet auf VDC Grafik 80 Zeichenschirm

SCREENV schaltet auf VIC Grafik 40 Zeichen

MOUSE(0-3) aktiviert Commodore Maus 1351 am Controlport 2
 MOUSE0 ein
 MOUSE1,20,30 setzt Mausparameter auf x=20 y=31
 MOUSE2,a,b liest Mausparameter und übergibt sie in die Variablen a und b
 MOUSE3 aus
 Maustasten links t=JOY(2) t=128
 rechts t=JOY(2) t= 1
 beide t=JOY(2) t=129
 keine t=JOY(2) t= 0

TRANS(0-3) Verschiebt 40 Zeichen Bitmapgrafik zum Zeichensatzram und umgekehrt. Es können Zeichensätze auf dem 40Zeichenschirm VIC neu gestellt werden oder ganze Bilder als neue Buchstaben verwendet werden. Das Zeichensatzram befindet sich von \$E000-\$EFFF. Siehe CHAR
 TRANS0 verschiebt \$2000-\$2FFF nach \$E000-\$EFFF 1.Hälfte VIC
 TRANS1 " \$E000-\$EFFF " \$2000-\$2FFF "
 TRANS2 " \$E000-\$EFFF " \$3000-\$3FFF 2.Hälfte VIC
 TRANS3 " \$3000-\$3FFF " \$E000-\$EFFF "

z.B. BLOAD"ZEICHENSATZ",U8,ONBØ,PDEC("E000")
 TRANS1:VGRAPHIC1:VCIRCLE3,3,3:TRANS0:CHAR,30,30,1,1,1,1,1,1
 mit allen 40 Zeichenbefehlen kann der Zeichensatz verändert werden (siehe VGRAPHIC Befehle)

CHAR schreibt Text in die GRAPHIC BOOSTER GRAFIK
 CHAR(set),X,Y,(Breite),(Höhe),(Mode),(String oder Zahl)
 set 0 oder 1 setzen/löschen
 Breite Höhe = Vergrößerungsfaktor 1-255
 Mode 0 Grossschrift/Grafik aus CharROM
 1 Kleinschrift/Grossschrift aus CharROM
 2 Grossschrift/Grafik aus RAM
 3 Kleinschrift/Grossschrift aus RAM
 4 Pokecodes aus CharROM
 5 Pokecodes aus RAM

z.B. CHAR,40,40,1,2,0,"GRAPHIC BOOSTER 128"
 CHAR,40,60,1,2,1,"Graphic Booster 128"
 mode4 und 5 ermöglichen einen bequemen Transfer des CharakterROMs und RAMs| CHAR,40, 80,1,2,4, 0,1,2,3,4,5,6,7,8,9 usw
 oder FORI=0T020:CHAR,40,120+i 8,2,2,4,122,153,1,250,255:NEXT
 oder FORI=0T020:CHAR,40+i 8,100,1,2,4,I:NEXT

Anstelle des Strings können beliebig viele Werte durch Komma getrennt folgen, welche einen Pokecode des CharROMs oder RAMs repräsentieren
 Werte zwischen 0 und 511 sind erlaubt
 Die Vergrößerung arbeitet auch mit dem eigenen Zeichensatz. 40Zeichen Grafiken können sehr einfach auf dem GRAPHIC BOOSTER Bildschirm vergrößert dargestellt werden.
 Reverse Zeichen Mode0-3 werden durch CONTROL RVSON in der Zeichenkette dargestellt "(RVS)TEXT"

PAINTX,Y,(Mode) füllen von von Zeichenflächen mit und ohne Muster
 X,Y Position im Koordinatensystem

Mode0-7
 0 füllen Punkt setzen
 1 füllen Punkt setzen, Muster fein
 2 füllen neu, test auf Punkt nicht gesetzt
 3 löschen neu, Punkt gesetzt
 4,(0-3),(0-3),(0-3) testet auf nicht gesetzt, Muster fein
 5,(0-3),(0-3),(0-3) testet auf nicht gesetzt, Muster grob
 6,(0-3),(0-3),(0-3) testet auf gesetzt, Muster fein
 7,(0-3),(0-3),(0-3) testet auf gesetzt, Muster grob

Mode 0 und 1 verhalten sich wie Paint auf dem 40Zeichenschirm
 Mode 2 bis 7 füllen oder löschen mit und ohne Muster. Es werden die vier Quadranten um einen Punkt abgetastet. Ebenfalls wird nicht bis ins Uferlose gefüllt, was bei Zeichenprogrammen von Vorteil ist.
 Gefüllt wird mehr als doppelt so schnell gegenüber Mode 0 und 1.
 PAINT verändert auch PATTERN, so muss nach PAINT mit PATTERN0 auf volle Strichdicke geschaltet werden wenn erwünscht.

z.B. CIRCLE,50,50,50:PAINT30,30,4,2,3,2:PATTERN0

PATTERN(Mode0-2)
 PATTERN0 kein Muster
 PATTERN1,(0-3),(0-3),(0-3) Muster fein (0-3) Werte für Muster
 PATTERN2,(0-3),(0-3),(0-3) Muster grob

z.B. PATTERN1,0,0,0 feine Quadrate
 " 1,1,1,1 "
 " 1,2,2,2 " gröbere "
 " 1,3,3,3 " grobe "

PATTERN findet bei allen Zeichnungsbefehlen wie DRAW usw Verwendung.
 Grobe Muster sieht man erst bei Veränderung der Strichdicke mit WIDTH oder die Linie wird gestrichelt dargestellt.

RLOAD(Bank),(0-2) ladet ein ein Bild von der Ramexpansion
 Bank.0-7 1750 Bank 0-1 1700
 (0-2) 0 Grafik1 interlace Bild
 1 Grafik2
 2 Grafik2

RSAVE(Bank),(0-2) speichert ein Bild auf die Ramexpansion
 Parameter wie RLOAD

GLOADa\$,Geräteadresse ladet ein Bild von Disk Device 8-15
 z.B.GLOAD"BILD",8 od. A\$="BILD":GLOADa\$,8

GSAVEa\$,8 speichert ein Bild auf Disk 8
 Es werden GRAPHIC1 und GRAPHIC2 Bilder gespeichert
 Ein Überlanges Grafikbild 2 speichern mit GRAPHIC1:GSAVEa\$,8
 oder auch GRAPHIC4

MERGE(A\$),U8 hängt ein Basicprogramm an das im Rechner befindende Programm
 MERGE"TOOL" od. A\$="TOOL":MERGE(A\$),U8
 Man kann seine GRAPHIC BOOSTER Unterroutinen, welche mit
 GOSUB aufgerufen werden, aneinanderhängen und anschliessend
 mit RENUMBER lauffähig machen.

BWINDOW(0-12) ruft zwei verschiedene Windows auf eines am rechten Bildrand
 (für Malprogramme) und eines in der Bildmitte,
 Das Window am rechten Rand ist 80 Punkte breit und so hoch wie
 der Bildschirm. Das in der Bildmitte ist 320 Punkte breit und
 174 Punkte (GRAPHIC1) 87 Punkte (GRAPHIC4) hoch.
 Der Startpunkt vom linken Rand ist 160 bzw. 200 Punkte je
 nach Auflösung, vom oberen Rand 143 (GRAPHIC1) 72 (GRAPHIC4)
 entfernt.BOX,160,143,479,316
 Die Parameter 0-3 beziehen sich auf das Window am rechtem Rand

BWINDOW0
 Swap wechselt Daten mit Puffer
 1 Fetch holt Daten aus Puffer
 2 Stash schreibt Daten in Puffer
 3 löscht Window (bei Multicolor wird die Farbe separat gesetzt)

Die Parameter 4-7 beziehen sich auf das Window Bildmitte bei
 GRAPHIC4
 4 Swap
 5 Fetch
 6 Stash
 7 löschen

Die Parameter 8-12 beziehen sich auf das Window Bildmitte bei
 GRAPHIC1
 8 Swap
 9 Fetch
 10 Stash
 11 löschen
 12 setzt Farbe welche in ECOL definiert ist
 Speicherbereich von Windowpuffer siehe BUFFER

BUFFER(0-4),(0-2) Überträgt Speicherbereiche vom 80Zeichenschirm in einen
 Puffer
 BUFFER0 bezieht sich auf das Attributram bei Multicolor Betrieb
 1 bezieht sich auf die ersten 16Kbyte im VDC
 2 " " " " zweiten " "
 3 " " " " dritten " "
 4 " " " " vierten " "

BUFFER0,0 Swap tauscht Attributram mit Puffer
 x,1 Fetch holt Daten aus Puffer und schreibt ins VDC ram
 x,2 Stash schreibt Daten vom VDC in den Puffer

BUFFER 1-4 können dazu verwendet werden ein Bild nicht mit
 GLOAD GSAVE zu laden/speichern sondern über den Puffer
 ebenfalls im Textmodus ist dies von grossem Nutzen, da
 GRAPHIC BOOSTER den Textmodus mit vollen 64K unterstützt
 Der Puffer wird von BWINDOW ebenfalls benützt so werden
 Daten gegebenenfalls gelöscht.
 schreiben Sie SCREENV und der Puffer befindet sich von
 \$2000-\$5FFF Dez 8192 - 24575 in Bank1
 dieser Speicherbereich kann jetzt auf Disk oder in die
 Ramerweiterung 1750/1700 gespeichert/geladen werden
 Siehe Beispiele
 Sie haben sich sicher gefragt wie man denn von Bank1
 in die Ramerweiterung speichert.Ganz einfach GRAPHIC BOOSTER
 nimmt die Daten aus derjenigen Bank, die mit dem BANK-Befehl
 definiert wurde. Ebenfalls können Sie im FASTmodus mit den
 Befehlen STASH FETCH und SWAP auf die Ramexpansion zugreifen
 was vorher zu einem Absturz des Rechners führte!(siehe System-
 verbesserungen).
 Den Puffer vom Window am rechten Rand erreicht man durch
 BANK1:SCREENV er liegt an der Adresse 8192 -16383
 den Puffer vom Window in der Bildmitte ebenfalls durch
 BANK1:SCREENV er liegt an der Adresse 16384-25856

Beispiel: Bildschirm über Puffer speichern

Die speziellen Kopier- und Fill-Befehle
 Die folgenden Befehle verändern den aktuellen Modus MODE. Mit
 TMODEa den aktuellen Modus in die Variable a retten und bei
 Rückkehr wieder MODEa.

PIXEL(Mode),w,z Kopiert Pixelweise
 w X-Offset zum Zeichenpunkt (LOCATE DRAW)
 z Y-Offset zum Zeichenpunkt

Mode0-6
 PIXEL0,w,z Swap austauschen
 1,w,z kopieren
 2, invers kopieren
 3 EOR und invers kopieren
 4 nur EOR kopieren
 5 OR kopieren
 6 AND kopieren

Beispiel: den Bereich 150,100 200,250 nach 0,0 kopieren
 PIXEL1,-150,-100
 WIDTH50,1
 DRAW,150,100to150,250
 oder WIDTH1,1
 PIXEL1,-150,-100
 FORI=150TO200:DRAW,I,100TO1,250:NEXT
 oder WIDTH1,1
 PIXEL1,-150,-100
 BOX,150,100,200,250,

Mit dem pixelweisen Kopieren kann vergrössert und verkleinert werden,
 indem man PIXEL in die Schleife miteinbezieht und die Werte mit einem
 Faktor von DRAW verändert. Überlappungen von Ziel und Quelle führen
 zu speziellen Effekten.

Befindet sich das Ziel in der Quelle muss folgendermassen vorgegangen werden:

Der Startpunkt des Ziels ist kleiner als der Startpunkt der Quelle
DRAW kleine Zahl zu grosser Zahl für X und Y
Der Startpunkt des Ziels ist grösser als der Startpunkt der Quelle
DRAW grosse Zahl zu kleiner Zahl für X und Y

Bei Swap muss das erste Koordinatenpaar von Draw durch LOCATE_{x,y} definiert werden, da DRAW den ersten Punkt zweimal setzt und somit zweimal austauscht. Wirkung: der erste Punkt wird nicht ausgetauscht
LOCATE10,10:DRAWTO10,50 statt DRAW,10,10to10,50
Dies muss bei allen folgenden Swapbefehlen erfolgen.

BYTE(0-9) Kopier- und Swapbefehle Byteweise bei 640 Pixelbreite sind dies 80 Bytes

BYTE0,Wert	setzt ein Byte Wert 0-255
1,Wert	" " OR verknüpft
2,Wert	" " EOR
3,Wert	" " AND
4,Wert1,Wert2	setzt nur Byte Wert1 wenn gleich wie Wert2
5,Wert1,WERT2	setzt Byte nur wenn Wert1 nicht gleich Wert2
6,x,y	wie Pixel kopiert EOR verknüpft
7,x,y	kopiert OR verknüpft
8,x,y	kopiert AND verknüpft
9,x,y	Swap tauscht aus

Was für fantastische Möglichkeiten dieser Befehl im Textmodus hat können Sie hier nur erahnen! (siehe GRAPHIC BOOSTER und der Textmodus)

z.B. GRAPHIC1,1
WIDTH3,8
BYTE0,170
CIRCLE,40,300,20,160
WIDTH1,1
BYTE1,112
BOX,0,0,40,200
BYTE2,20
BOX,0,0,20,100
SLEEP1
BOX,0,0,20,100
WIDTH15,0
BYTE6,30,150
LOCATE0,0|DRAWTO0,100
SLEEP2
LOCATE0,100:DRAWTO0,0
WIDTH10,0
BYTE9,50,300
LOCATE0,0:DRAWTO0,100
SLEEP1
LOCATE0,100:DRAWto0,0

Wollen Sie ein Bild revers darstellen oder drucken:

WIDTH80,1
BYTE2,255
DRAW,0,0to0,500

FBYTE(Mode),x,y,Breite	schnelles Kopieren und Swapen Mode 0-7
FBYTE0,x,y,Breite	Kopiert mit Offset x y mit der Breite Byte
1,x,y,b	Swap
2,x,y,b	kopiert Attribut Multicolor nach Pixelbitmap
3,x,y,b	kopiert Pixelbitmap nach Attribut
4,x,y,b	Swap Attribut mit Pixel
5,x,y,b	Swap Pixel mit Attribut
6,x,y,b	kopiert Attribut mit Attribut
7,x,y,b	Swap Attribut mit Attribut

Bei Zweifarbooster nur Mode 0 und 1
Warum Mode2-5 ? Wird nur mit Blockgrafik gearbeitet kann der Bitmapspeicher als versteckter Zusatzspeicher genutzt werden. ECOL2,2,2,2 Vorder- und Hintergrundfarbe geichsetzen.

Beispiele:

FILLO,Breite,Wert	füllt Bereich mit Byte Wert 0-255
1,Breite	füllt Bereich mit Farbe aus ECOL
TBYTEa	liest Byte an aktueller Cursorposition LOCATE20,200:TBYTEA:PRINTA
USER	für eigene Maschinenroutinen, springt über Vektor \$1BEB \$1BEC (7149 7150) POKE7149,0 POKE7150,12 USER startet Programm ab \$0C00

Allgemeines

Die wichtigsten Befehle können miteinander kombiniert werden. Z.B. beim Kopieren kann mit einem Kreisring oder einer Linie beliebiger Dicke eine Fläche kopiert werden.

PIXEL, BYTE, FBYTE,FILL arbeiten im Zusammenhang mit den Zeichenbefehlen wie DRAW, CIRCLE u.s.w.. D.h. anstatt dass ein Punkt gesetzt wird, wird einer der Spezialbefehle ausgeführt.

Der GRAPHIC BOOSTER ist so programmiert, dass mit einfachsten Befehlen bereits Profissoftware erstellt werden kann.

Für Einsteiger: Üben Sie mit den Zeichenbefehlen im MODE 0 1 und 2 im 2 Farben-GRAPHIC BOOSTER.

Wichtig: Am Anfang des Programmes immer MODE, PATTERN, WIDTH und ECOL definieren.

Die Multicolorbefehle können nicht im 2-Farbenmodus verwendet werden.

CTEST:ECOL:RECOL:CHANGE:SCLR

Am Ende des Directory der Disk Seite 1 sind verschiedene Tools, die Sie gebrauchen können.

Im Anwenderprogramm "Grafik Tablet" können Sie das File Shuttle anstatt Körper laden. Das sind die Daten des Space Shuttle.Mit diesen können Sie jetzt das Grafik Tablet erweitern: Machen Sie mal ein Programm zum Vergrössern,

Verschieben, Drehen und lassen Sie die Daten durchlaufen.

Auch dieses Programm können Sie einschicken oder auch die anderen Programme auf der Disk erweitern.

Die Daten des Shuttle dürfen nur in Zusammenhang mit GRAPHIC BOOSTER verwendet werden; copyright Combo AG.

WIDTHX,Y stellt die Strichdicke ein
 Parameter von 1-255 0 entspricht 1
 stellen Sie die Strichdicke auf WIDTH1,1 wenn ein neuer Modus(MODE)
 ausprobiert wird. Will eine Zeichenfunktion nicht enden ist die
 Strichdicke sicher zu gross gewählt. Die Funktion wird aber korrekt
 ausgeführt.
 z.B. MODE2:WIDTH16,16:CIRCLE,150,150,100
 MODE0:WIDTH16,16:CIRCLE,350,150,100

MODE(0-17) stellt verschiedene Zeichenmodi ein

0	setzt Punkt	
1	" " EOR (Gummiband)	
2	" " EOR (Gummiband) für	CIRCLE
3	Vorder- und Hintergrundfarbe und	Punkt
4	" " ohne Punkt	
5	nur Vordergrundfarbe und Punkt	
6	" Hintergrundfarbe und Punkt	
7	" Vordergrundfarbe ohne Punkt	
8	" Hintergrundfarbe ohne Punkt	
9	Vorder- und Hintergrundfarbe	Colorblock
10	nur Vordergrundfarbe	"
11	" Hintergrundfarbe	"
12	CHANGE nur Vordergrundfarbe	
13	" " Hintergrundfarbe	
14	" " Vordergrundfarbe	Colorblock
15	" " Hintergrundfarbe	"
16	" Vorder- und Hintergrundfarbe	
17	" " " "	Colorblock

Für den zweifarbigen GRAPHIC BOOSTER sind nur die Modi 0,1 und 2
 aktiv.
 Colorblock bedeutet, dass nur die Farbattribute gesetzt werden; in einem
 Koordinatennetz von 80 mal 96 Punkten.

TESTX,Y,(var) testet ob Punkt gesetzt ist oder nicht. Das Resultat wird in
 einer Variablen übergeben. 0 = gesetzt 1 = nicht gesetzt
 z.B. TEST10,59,a:PRINTa

CTESTa,b,c,d liest die Farbattribute an der aktuellen Cursorposition und
 übergibt sie den Variablen a b c und d

ECOLa,b,c,d setzt die Farbparameter für die Zeichenfarbe Werte von 0-15
 ECOL2,2,8,8
 ECOLa,b ist auch erlaubt

RECOLa,b,c,d liest die aktuelle Zeichenfarbe von ECOL

die ersten zwei Parameter von CTEST ECOL RECOL bedeuten die Vordergrundfarbe
 die zweiten zwei die Hintergrundfarbe. Mit dem entsprechenden PATTERN1,0,0,0
 lassen sich so über 65000 Farben darstellen. Vertauscht man a und b ergeben
 sich wiederum andere Farben. Mit ungeraden und geraden Y-Koordinaten lassen
 sich auch andere Effekte einstellen. Rechnet man das Fading vom Colorbefehl
 dazu 0-15 ergibt sich eine Totalanzahl von rund 980 000 Farbmöglichkeiten.
 Die Colorblöcke 8 mal 6 Punkte können unabhängig gesetzt werden.
 256 Vorder- und 256 Hintergrundfarben. Wird diese Matrix mit PATTERN1,0,0,0
 aufgefüllt lassen sich in dieser Matrix 65000 Farben darstellen. Bei einer
 Totalanzahl von 7200 solchen Blöcken 80 mal 94 sind die Möglichkeiten
 unberentzt; ebenfalls ist man nicht auf das 8 mal 6 Raster begrenzt, da
 diese Matrizen mit Mustern auch nur angeschnitten werden können und z.B.
 mit MODE0 ohne Beeinflussung der Farben gezeichnet werden kann.
 Freilich eignen sich nicht alle Farben zum direkten Arbeiten auf dem
 Bildschirm. Für Bildschirmfotos ist aber jede Kombination möglich.
 Abhilfe schafft ein ein Bildschirmnetz oder ein nachleuchtender Monitor z.B
 Commodore 2020.
 Der CHANGE Befehl kann hier optimal eingesetzt werden. Man zeichnet zuerst
 mit Falschfarben und wechselt sie dann mit CHANGE.

CHANGEa,b,c,d wechselt die vorgegebenen Farben in die mit ECOL definierten
 dazu verwendet man die Modi 12-17
 Man verwendet wie gewohnt die Zeichenbefehle DRAW usw
 Es werden also nur die Farben gewechselt, die mit CHANGE
 übereinstimmen:
 z.B. MODE0:ECOL2,2,8,8:GRAPHIC1,1:SCLR:CIRCLE,50,50,50
 PAINT50,50,2:LOCATE50,50:CTESTa,b,c,d:CHANGEa,b,c,d:
 ECOL5,5:MODE14:BOX,0,0,14,20,,1
 Dies ist ein sehr einfaches Beispiel. Kommen in dem Pixel-
 bereich 100 mal 100 noch andere Farben vor werden diese
 nicht verändert.

COLOR(Hintergrundfarbe),(Vordergrundfarbe),(Fading)
 0-15 0-15 0-15

COLOR2,8,5

SCLR setzt die in ECOL definierten Farben auf dem Ganzen Bildschirm
 im Multicolorbetrieb müssen nach GRAPHIC1,1 immer mit SCLR
 die Farbattribute gesetzt werden.

SWITCH schaltet von Multicolor auf Zweifarbbetrieb und umgekehrt
 das laden von der 1750 erfolgt mehr als doppelt so schnell

TMODEa liefert den aktuellen Zeichenmodus
 siehe Beispiel auf disk Maus (TOOLS)
 TMODEa:?a

UPa scrollt den Bildschirm pixelweise nach oben nur Zweifarbbetrieb
 Werte 1-255 UP20

DOWNa scrollt den Bildschirm pixelweise nach unten nur Zweifarbbetrieb
 DOWN 20

UP ohne Parameter nur für Multicolorbetrieb

Down ohne Parameter nur für Multicolorbetrieb

BLEFT(0-9) schiebt den Bildschirm nach links. Für Monitoreinstellung und
 Effekte. BLEFT0 = GRAPHIC1

BRIGHT(0-9) schiebt den Bildschirm nach rechts. BRIGHT4

EINBAU DER STECKKARTE

Netz kabel und Peripherieverbindungen ausziehen!

Commodore 128

Öffnen Sie den Rechner mittels 6 Schrauben auf der Gehäuseunterseite. Beim Abheben des Tastaturdeckels drücken Sie mit einem Schraubenzieher oberhalb des Joystickport 1 die untere Gehäusehälfte ein wenig nach innen.

Ziehen Sie den Tastatur- und den PowerLED-Stecker heraus. Entfernen Sie das Abschirmblech, das die ganze Platine überspannt. Weiters entfernen Sie den kleinen Blechdeckel durch anheben. Jetzt hebeln Sie den Chip 8563 (links im Blechkasten) mit 2 Schraubenziehern heraus. Beachten Sie die Richtung der Kerbe, die beim Wiedereinbau in die selbe Richtung zeigen muss. Stecken Sie jetzt die Platine an den Platz des 8563 (Biegen Sie den Transistor und die Kondensatoren ev. zur Seite b.z.w. flach auf die Platine).

Den Sockel, der mit der Steckkarte verbunden ist, stecken Sie in den freien Steckplatz U36. Der Sockel muss so liegen, dass der befestigte Draht unten links, also neben U4 zu liegen kommt. (siehe Zeichnung).

Stecken Sie den 8563 in den Sockel auf der Steckkarte. Die Kerbe muss Richtung Userport, Kassettenport zeigen. Achten Sie darauf, dass keine Beinchen verbogen und abgebrochen werden.

Als Test schliessen Sie die Tastatur wieder an und drücken die 80-Zeichentaste. Schliessen Sie die Stromversorgung an, schalten den Rechner ein und das gewohnte Einschaltbild muss erscheinen.

Wenn nicht, schalten Sie den Rechner nach 4 Sekunden aus und überprüfen Ihren Einbau.

Nach erfolgreichem Test montieren Sie wieder den Blechdeckel. Er lässt sich nun auf der linken Seite nicht mehr ganz nach unten drücken. Dadurch entstehen links kleine Schlitze, durch welchen Sie den Draht zu U36 führen.

Abschirmblech montieren, Rechner zusammenschrauben - fertig.

Commodore 128D

1. Gehäuseunterseite: 4 Schrauben lösen mit Kreuzschraubenzieher.

2. Traggriff zur Seite klappen

3. Deckel abheben

4. Floppyplatine lösen: 3 Schrauben

5a. Wichtig!!!

Markieren Sie alle Steckverbindungen mit einem Filzstift d.h. ziehen Sie einen Strich über Stecker und Buchse, um beim Zusammenbau ein falsches Zusammenstecken zu vermeiden!

5b. sechs Stecker auf Floppyplatine herausziehen

-Stecker vom Netzteil her

-3-Polstecker senkrecht auf Platine

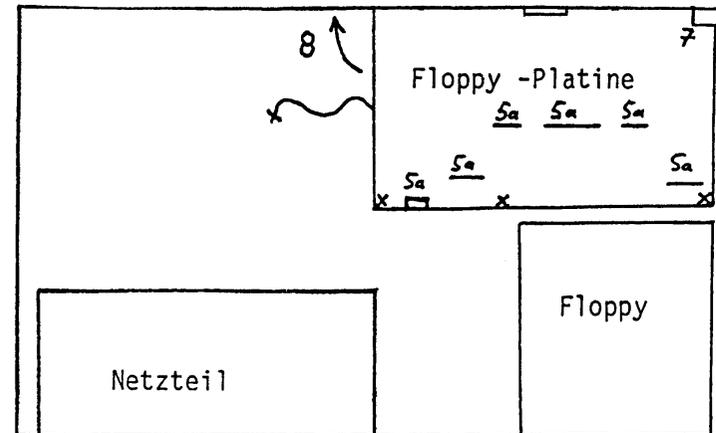
-10-Polstecker senkrecht auf Platine

-6-Polstecker senkrecht auf Platine

-4-Polstecker senkrecht auf Platine

-3-Polstecker vom Netzteil

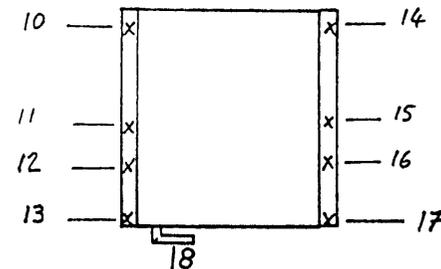
6. Floppyplatine nach hinten klappen



7. Stecker von Netzteil auf der Platine herausziehen.

8. Schraube von Abdeckblech der Floppy lösen

9. "



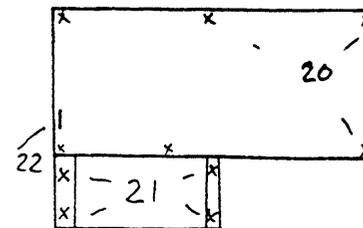
10.-17. Schrauben von Floppy lösen.

18. Floppyverschlusshebel herausziehen

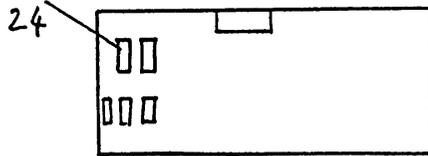
19. Floppy anheben und herausnehmen

20. Abschirmblech, welches die ganze Rechnerplatine überspannt lösen (7 Schrauben)

21. Netzteil lösen (4 Schrauben).



Jetzt kann das Abschirmblech über der Rechnerplatine weggenommen werden. Am Abschirmblech drücken Blechzungen auf die verschiedenen Bausteine. Gegenüber vom Userport befinden sich die ersten zwei Blechzungen. Die erste von links muss ca 1cm nach oben gebogen werden, um wie wir später sehen keinen Kontakt mit dem Zusatzsockel auf U36 zu verursachen!



25. Der Einbau der Steckkarte erfolgt jetzt wie beim 128

26. Beim Zusammenbau beachten Sie die Markierungen, (Stecker), die Sie sicher mit einem wasserfesten Filzstift zeichnen.

Der Umbau geschieht auf Ihr eigenes Risiko. Da der Rechner geöffnet werden muss, erlöschen eventuelle Garantieansprüche.

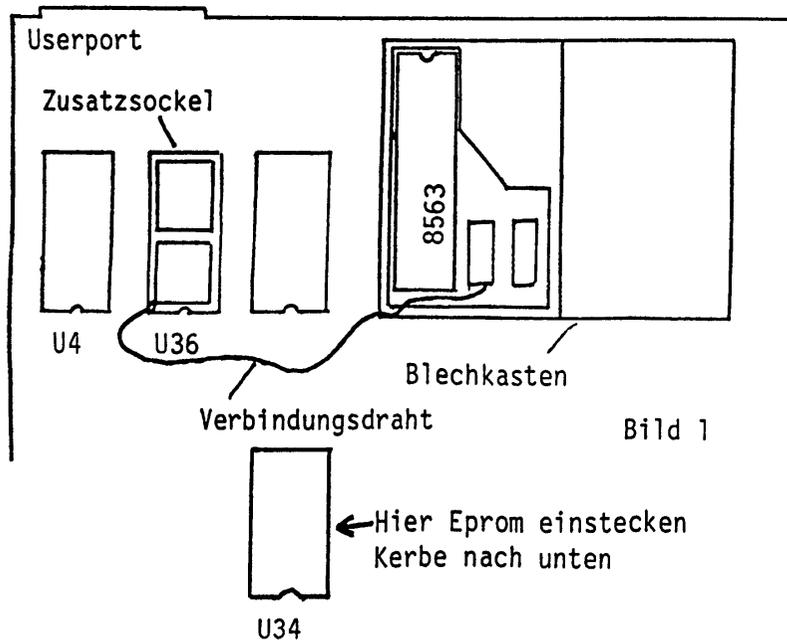
Bei vorsichtigem Einbau sind jedoch keine Schäden zu befürchten.

Ansonsten lassen Sie den Umbau von Ihrem Fachgeschäft durchführen.

C - 128 DCR (128D im Blechgehäuse)

Einbau: 5 Schrauben Gehäuse Rückwand und Seite lösen. Deckel nach hinten schieben und wegnehmen. EProm auf Platz U34 mit Schraubenzieher heraushebeln und durch neues EPROM ersetzen. Kerbe muss Richtung Frontplatte zeigen also so wie altes EProm. Gehäusedeckel wieder montieren. Schluss.

Technische Änderungen vorbehalten.



Anwenderprogramme zum GRAPHIC BOOSTER 128

- *****
1. Boxes 1 screen:
Selbsterklärend
 2. Boxes 2 screen:
2 Monitorbetrieb, selbsterklärend
 3. Circle 1 screen:
Selbsterklärend
 4. Circle 2 screen:
2 Monitorbetrieb, selbsterklärend
 5. Grafiktafel:
Durch Voranstellen eines "-" Zeichens bei der X-Koordinate kann ein neuer Startpunkt gewählt werden. Z.B. x=-200 y=300, d.h. die Linie beginnt bei 200/300. Ohne "-" wird von Punkt zu Punkt gezeichnet. Programm beenden: "END"
 6. Artillerie:
Selbsterklärend
 7. 3-D Konstruktionen:
Nach Setzen des ersten Punktes mit dem Feuerknopf kurz warten. Pfeiltaste links berechnet 3-D Daten. "W" speichert auf RAM-Expansion 1750. "R" ladet von RAM-Expansion 1750. "E" gleich neuer Start.