

Die Ausgabedaten der RAMs führen an die Eingänge des 8-bit-Schieberegisters J25. Jeweils zu Beginn eines neuen Taktschrittes werden die acht Punktdaten hier übernommen, gesteuert durch J12. Während der BLANK-Zeit, gesteuert durch J10/25 BLK über J12, wird die feste "1" des Serial Inputs übernommen. Eine "1" entspricht einem dunklen Bildpunkt.

Die hinausgeschobenen Daten (J25/13) werden mit dem Punkttakt verknüpft über J12/12,13, mit dem Synchronisationssignal über SYNC(J10/34), J13 und den Widerständen R7, R9 und R10 vermischt.

Hinweis: Bei manchen Monitoren kann R10 (1K0hm) weggelassen werden - dies erhöht den Signalpegel.

T1 verstärkt das Signal und legt es Z-Richtig mit 75 Ohm an den Ausgang BAS.

Funktion des Bausteines 9366

Dieses Kapitel benötigen Sie nur dann, falls Sie den GDP direkt ansteuern wollen. Sie verwenden dazu den Befehl G a b

Sonst können Sie überblättern und direkt zum Kapitel "Quick Reference" oder "Anwendungsbeispiele" übergehen.

Die Steuerung des TERM1 übernimmt der "Graphik Display Prozessor EF 9366" von Thomson-CSF.

Tabelle 1 zeigt die Registeradressen und die Bedeutung:

TABLE 1 - REGISTER ADDRESS

ADDRESS REGISTER					REGISTER FUNCTIONS		Number of bits
Binary				Hexa	Read R/W = 1	Write R/W = 0	
A3	A2	A1	A0				
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL 1 (Write control and interrupt control)		7
0	0	1	0	2	CTRL 2 (Vector and symbol type control)		4
0	0	1	1	3	CSIZE (Character size)		8
0	1	0	0	4	Reserved		—
0	1	0	1	5	DELTAX		8
0	1	1	0	6	Reserved		—
0	1	1	1	7	DELTAY		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		8
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		8
1	1	0	0	C	XLP (Light-pen)	Reserved	7
1	1	0	1	D	YLP (Light-pen)	Reserved	8
1	1	1	0	E	Reserved		—
1	1	1	1	F	Reserved		—

Reserved : These addresses are reserved for future versions of the circuit. In read mode, output buffers D0-D7 force a high state on the data bus.

TABELLE 1 Registeradressen

Detaillierte Beschreibung der einzelnen Register:

0 Status / Kommando

Dieses Register ist das Schlüsselregister zum Baustein. Es kann vom Rechner gelesen werden - hier meldet der 9366 seinen Status - oder beschrieben werden - hiermit übergibt man ein Kommando zum Baustein.

STATUS REGISTER (LESEN)

Bits:

7	6	5	4	3	2	1	0		
I	I	I	I	I	I	I	IHIGH	Ende einer Lichtgriffelsequenz
I	I	I	I	I	I	I	IHIGH	Vertikal Blank
I	I	I	I	I	I	I	IHIGH	Bereit für neues Kommando
I	I	I	I	I				LOW	BUSY
I	I	I	I	I	I	I	IHIGH	Pen ausserhalb Anzeigebereich
I	I	I	I	I	I	I	IHIGH	Lichtgriffel verursachte IRQ
I	I	I							(Falls freigegeben)
I	I	I	I	I	I	I	IHIGH	Vertical Blank verursachte IRQ
I	I								(Falls freigegeben)
I	I	I	I	I	I	I	IHIGH	Bereit für neues Kommando ver-
I									ursachte IRQ (falls freigeg.)
I	I	I	I	I	I	I	IHIGH	Bits 5,6,7 verodert; HIGH
									falls überhaupt IRQ vorhanden

KOMMANDO - REGISTER (Schreiben)

Dieses Kommando-Register ist nicht ganz einfach zu verstehen - es hat fünf Funktionen:

Wert	Bedeutung
00H-0FH	Kommandos, wie Bildschirm , Register löschen usw.
10H-17H	Grundvektorbefehle Ein Vektor wird definiert durch den Betrag in den Registern DELTAX und DELTAY sowie durch das hier übergebene Vorzeichen
18H-1F	Vektoren mit Richtungsvorgabe Ein Richtungscode (8 Richtungen) wird übergeben, das kleinere der Register DELTAX oder DELTAY ignoriert und der Vektor mit der Länge des grösseren DELTA-Registers gezeichnet
20H-7FH	ASCII-Zeichen werden ausgegeben, Code-Zuordnung siehe Tabelle 2

80H-FFH Kurzvektoren

In einem Byte ist ein Vektor ollständig definiert

TABELLE 2 zeigt den Überblick über das Kommando-Register:

TABLE 2 - COMMAND REGISTER

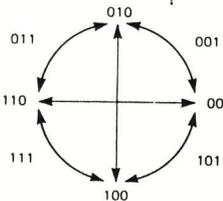
b7 b6 b5 b4	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1																										
	0	0	0	0	1	1	1	1	1	0	0	0	1	1	0	0	1	1																									
b3 b2 b1 b0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																											
0 0 0 0	0	Set bit 1 of CTRL1 : Pen selection								SMALL VECTOR DEFINITION : <table border="1" style="margin: 5px;"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>1</td><td> ΔX </td><td> ΔY </td><td colspan="5">Direction</td> </tr> </table> Dimension <table border="1" style="margin: 5px;"> <tr> <td>ΔX or ΔY</td><td>Vector length</td> </tr> <tr> <td>0 0</td><td>0 step</td> </tr> <tr> <td>0 1</td><td>1 step</td> </tr> <tr> <td>1 0</td><td>2 steps</td> </tr> <tr> <td>1 1</td><td>3 steps</td> </tr> </table> Direction 								b7	b6	b5	b4	b3	b2	b1	b0	1	ΔX	ΔY	Direction					ΔX or ΔY	Vector length	0 0	0 step	0 1	1 step	1 0	2 steps	1 1	3 steps
b7	b6	b5	b4	b3	b2	b1	b0																																				
1	ΔX	ΔY	Direction																																								
ΔX or ΔY	Vector length																																										
0 0	0 step																																										
0 1	1 step																																										
1 0	2 steps																																										
1 1	3 steps																																										
0 0 0 1	1	!	1	A	Q	a	q																																				
0 0 1 0	2	"	2	B	R	b	r																																				
0 0 1 1	3	#	3	C	S	c	s																																				
0 1 0 0	4	\$	4	D	T	d	t																																				
0 1 0 1	5	%	5	E	U	e	u																																				
0 1 1 0	6	&	6	F	V	f	v																																				
0 1 1 1	7	/	7	G	W	g	w																																				
1 0 0 0	8	(8	H	X	h	x																																				
1 0 0 1	9)	9	I	Y	i	y																																				
1 0 1 0	A	.	:	J	Z	j	z																																				
1 0 1 1	B	+	:	K	[k	{																																				
1 1 0 0	C	,	<	L	\	l																																					
1 1 0 1	D	-	=	M]	m	}																																				
1 1 1 0	E	.	>	N	^	n	_																																				
1 1 1 1	F	/	?	O	←	o	⊗																																				

TABELLE 2 Übersicht über das Kommando-Register

Bit Bedeutung

- 0 Im Register CTRL1 wird das BIT1 gesetzt; der "Schreibstift=Pen" wird angewählt
Dieses Bit bzw. Kommando ist vor jedem Schreiben zu geben
- 1 Auswahl zwischen PEN und "Radiergummi":
"1" PEN
"0" ERASER (Gummi)



- 2 Im Register CTRL1 wird Bit 0 gesetzt - der PEN oder Eraser wird abgesenkt
- 3 Im Register CTRL1 wird das BIT 1 gelöscht - Pen oder Eraser werden angehoben
- 4 Bidschirm löschen
- 5 Register X und Y = 0
- 6 X,Y=0 und Bildschirm löschen
- 7 Alle Register (ausser XLP,YLP) zu 0 setzen, Bildschirm löschen
- 8 Lichtgriffel initialisieren (WHITE aktivieren, Bildschirm blinkt 1 Zyklus weiss)
- 9 Lichtgriffel initialisieren
- A 5x8-Block-Zeichnen. Die Grösse des Blocks ist von Register CSIZE abhängig
- B 4x4-Block-Zeichnen
- C Bildschirm scannen - Pen oder Eraser wie CTRL1
- D X=0
- E Y=0
- F Direkter Bildzugriff im nächsten freien Zyklus

Diese Befehle gelten, falls das erste Halbbyte des Register "0" ist, also für die Ausgaben "00H" bis "0FH".

Das Zeichnen von Vektoren

Grundsätzlich ist ein Vektor durch seinen Startpunkt und seine Projektion auf die X- und Y-Achsen beschrieben.

Der Startpunkt wird durch den Inhalt der Register X und Y definiert

Die Projektionen auf die Achsen werden durch die Inhalte der Register DELAX und DELTAY übergeben. Das Vorzeichen wird erst im Kommando übergeben.

Während des Plottens werden die einzelnen Bildpunkte durch das X und Y-Register adressiert. Nach einem Plot zeigen diese Register auf den Endpunkt des Vektors.

Damit die Angelegenheit etwas komplizierter wird (und um die Flexibilität in der Programmierung zu erhöhen), gibt es 128 verschiedene Kommandos, Vektoren zu zeichnen. Wir wollen nun versuchen, Licht in das Dunkel zu bringen.

Die Vektorbefehle können in vier Gruppen aufgeteilt werden:

- Grundvektorbefehle
- Vektorbefehle parallel zur X oder Y-Achse
- Vektoren mit besonderer Richtungsvorgabe
- Kurzvektoren.

Grundvektorbefehle sind die allgemeinste Art der Vektordarstellung. Hier wird der Startpunkt des Vektors im Registerpaar X und Y angegeben; die Länge des Vektors (projiziert) in den Registern DELTAX und DELTAY. Das Vorzeichen dieser Register wird beim Befehl "Zeichne Grundvektor" übergeben.

Vektorbefehle parallel zur X oder Y-Achse sind ein Sonderfall der Grundvektorbefehle. Ein Register (DELTAX oder DELTAY) wird hierbei zu Null angenommen; im Befehl wird das Vorzeichen des anderen Registers übergeben.

Vektoren mit besonderer Richtungsvorgabe können nicht nur parallel zu den Achsen, sondern auch im 45-Grad - Winkel gezeichnet werden. Die Richtung wird hierbei im Befehl übergeben. Das kleinere der beiden Register DELTAX oder DELTAY wird ignoriert, da der Vektor ja durch die gegebene Richtung und eine (projizierte) Länge bereits vollständig beschrieben ist.

Grundvektorbefehle

```
0 0 0 1 0 X X 1
      I I
      I I..... Vorzeichen von DELTAX   0="+"
      I..... Vorzeichen von DELTAY   1="-"
```

Vektorbefehle parallel zu einer Achse

```
0 0 0 1 0 X X 0
      I I
      I I..... 00 DELTAY=0, DELTAX>0
                  01 DELTAX=0, DELTAY>0
                  10 DELTAX=0, DELTAY<0
                  11 DELTAY=0, DELTAX<0
```

Vektoren mit besonderer Richtungsvorgabe

- 2 Im Register CTRL1 wird Bit 0 gesetzt - der PEN oder Eraser wird abgesenkt
- 3 Im Register CTRL1 wird das BIT 1 gelöscht - Pen oder Eraser werden angehoben
- 4 Bildschirm löschen
- 5 Register X und Y = 0
- 6 X,Y=0 und Bildschirm löschen
- 7 Alle Register (ausser XLP,YLP) zu 0 setzen, Bildschirm löschen
- 8 Lichtgriffel initialisieren (WHITE aktivieren, Bildschirm blinkt 1 Zyklus weiss)
- 9 Lichtgriffel initialisieren
- A 5x8-Block-Zeichnen. Die Grösse des Blocks ist von Register CSIZE abhängig
- B 4x4-Block-Zeichnen
- C Bildschirm scannen - Pen oder Eraser wie CTRL1
- D X=0
- E Y=0
- F Direkter Bildzugriff im nächsten freien Zyklus

Diese Befehle gelten, falls das erste Halbbyte des Register "0" ist, also für die Ausgaben "00H" bis "0FH".

Das Zeichnen von Vektoren

Grundsätzlich ist ein Vektor durch seinen Startpunkt und seine Projektion auf die X- und Y-Achsen beschrieben.

Der Startpunkt wird durch den Inhalt der Register X und Y definiert

Die Projektionen auf die Achsen werden durch die Inhalte der Register DELAX und DELTAY übergeben. Das Vorzeichen wird erst im Kommando übergeben.

Während des Plottens werden die einzelnen Bildpunkte durch das X und Y-Register adressiert. Nach einem Plot zeigen diese Register auf den Endpunkt des Vektors.

Damit die Angelegenheit etwas komplizierter wird (und um die Flexibilität in der Programmierung zu erhöhen), gibt es 128 verschiedene Kommandos, Vektoren zu zeichnen. Wir wollen nun versuchen, Licht in das Dunkel zu bringen.

Die Vektorbefehle können in vier Gruppen aufgeteilt werden:

- Grundvektorbefehle
- Vektorbefehle parallel zur X oder Y-Achse
- Vektoren mit besonderer Richtungsvorgabe
- Kurzvektoren.

Grundvektorbefehle sind die allgemeinste Art der Vektordarstellung. Hier wird der Startpunkt des Vektors im Registerpaar X und Y angegeben; die Länge des Vektors (projiziert) in den Registern DELTAX und DELTAY. Das Vorzeichen dieser Register wird beim Befehl "Zeichne Grundvektor" übergeben.

Vektorbefehle parallel zur X oder Y-Achse sind ein Sonderfall der Grundvektorbefehle. Ein Register (DELTAX oder DELTAY) wird hierbei zu Null angenommen; im Befehl wird das Vorzeichen des anderen Registers übergeben.

Vektoren mit besonderer Richtungsvorgabe können nicht nur parallel zu den Achsen, sondern auch im 45-Grad - Winkel gezeichnet werden. Die Richtung wird hierbei im Befehl übergeben. Das kleinere der beiden Register DELTAX oder DELTAY wird ignoriert, da der Vektor ja durch die gegebene Richtung und eine (projizierte) Länge bereits vollständig beschrieben ist.

Grundvektorbefehle

```

0 0 0 1 0 X X 1
      I I
      I I..... Vorzeichen von DELTAX  0="+"
      I I..... Vorzeichen von DELTAY  1="-"

```

Vektorbefehle parallel zu einer Achse

```

0 0 0 1 0 X X 0
      I I
      I I..... 00  DELTAY=0,DELTAX>0
                   01  DELTAX=0,DELTAY>0
                   10  DELTAX=0,DELTAY<0
                   11  DELTAY=0,DELTAX<0

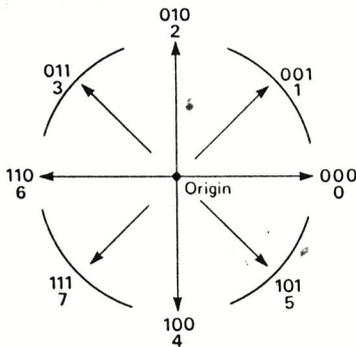
```

Vektoren mit besonderer Richtungsvorgabe

```

0 0 0 1 1 1 X X X
      I I I
      I I I...
    
```

Richtung, in folgendem Diagramm festgehalten



Das kleinere der Register DELTAX oder DELTAY wird ignoriert, der Vektor entsprechend der Richtung mit der Länge des grösseren Registers gezeichnet:

Kurzvektoren

```

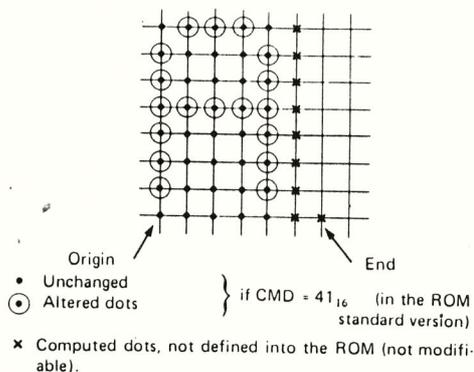
1 x x x x x x x
  I I I I I I I
  I I I I I-I-I..... Richtungscode
  I I I I
  I I I-I-I..... DELTAY von 0..3 ohne Vorz.
  I I
  I-I..... DELTAX von 0..3 ohne Vorz.
    
```

Da die gesamte Information über einen (Kurz-)Vektor in einem Byte übergeben werden kann, eignen sich die Kurzvektoren besonders gut, Umrisszeichnungen oder bewegte Bilder auszugeben. Die Vektoren können aufeinanderfolgend in einer Tabelle abgelegt werden und wie Texte ausgegeben werden.

Falls das Bit B7 (das höchstwertige) "0" ist und B6 bis B4 ungleich Null sind, so wird über das Kommandoregister ein ASCII-Zeichen übergeben. Dieses Zeichen wird an der Stelle X,Y mit der im CSIZE - Register angegebenen Grösse und der in CTRL2 definierten Richtung angezeigt.



Zeichen werden in einer 5 x 8-Matrix ausgegeben. Nach der Ausgabe eines Zeichens wird X um 6P erhöht. Dies verdeutlicht das folgende BILD:



Jeder der ausgegebenen Bildpunkte kann durch einen Block, der $P \times Q$ gross ist, ersetzt werden. P und Q können von 1 bis 15 reichen und werden in CSIZE übergeben.

Die Zeichen liegen von 20H bis 7FH und entsprechen den 96 Standard (USA) ASCII-Zeichen. Ein 97 stes Zeichen (OAH) erzeugt einen $5P \times 8Q$ -Block und kann dazu verwendet werden, andere Zeichen zu löschen

Das 98 ste Zeichen erzeugt ein $4P \times 4Q$ -Feld ohne Zwischenraum zum nächsten Zeichen. Mit diesem Zeichen können grobe graphische Zeichnungen (z.B. Balkendiagramme) erzeugt werden.

Ein Zeichen kann auf zweierlei Arten gelöscht werden: Entweder mit dem Zeichen OAH, oder indem man das gleiche Zeichen (mit dem gleichen Startpunkt X, Y) und eingeschaltetem Eraser überschreibt.

Hinweis: Das Blank (20H) löscht nicht, sondern positioniert nur den X-Wert ein Zeichen weiter.

1 CTRL1 Steuerregister 1

TABELLE 3 zeigt den Aufbau des Registers CTRL1

TABELLE 3 Aufbau des Registers CTRL1

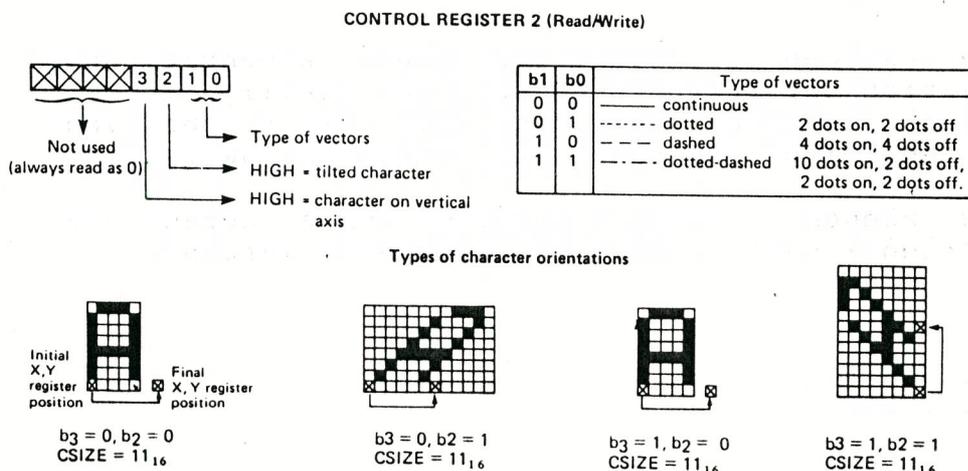
Einige Bits des Registers sind redundant, d.h. sie können auch über das Kontrollregister gesetzt bzw. gelöscht werden. Es sind dies die beiden (wichtigsten) niederwertigen Bits B0 und B1.

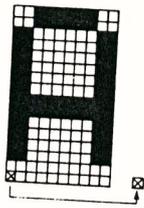
7	6	5	4	3	2	1	0		
I	I	I	I	I	I	I	I	HIGH=Pen unten LOW=Pen oben
I	I	I	I	I	I	I	I	HIGH=Pen LOW=Gummi
I	I	I	I	I	I	I	I	HIGH=Schnell schreiben ohne Ausgangssignal
I	I	I	I	I	I	I	I	HIGH Geschlossene Bildfläche, d.h. es wird auch geschrieben, wenn MSB X,Y ausserhalb der Bildfläche sind
I	I	I	I	I	I	I	I	IRQ-Freigabe für Lichtgriffel
I	I	I	I	I	I	I	I	IRQ-Freigabe bei VB
I	I	I	I	I	I	I	I	IRQ-Freigabe bei "Bereit"
I	I	I	I	I	I	I	I	Nicht verwendet (0)

2 CTRL2 Steuerregister 2

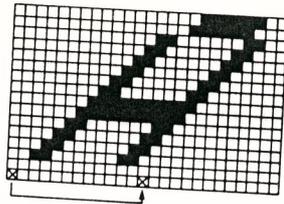
Steuerrregister 2 steuert die Art der gezeichneten Vektoren (Durchgezogen oder gepunktet oder strich-punktiiert) sowie die Art der Zeichendarstellung.

TABELLE 4 zeigt die Möglichkeiten des Registers 2





b3 = 0, b2 = 0
CSIZE = 22₆



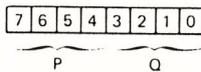
b3 = 0, b2 = 1
CSIZE = 22₆

3 Register CSIZE ZEICHENGRÖSSE

In diesem Register wird die Grösse der darzustellenden Zeichen übergeben. Die Grösse ist in X und Y-Richtung in 16 Schritten wählbar.

TABELLE 6 zeigt die Übergabe.

C-SIZE REGISTER (Read/Write)



P : Scaling factor on X axis
Q : Scaling factor on Y axis

P and Q may take any value between 1 and 16. This value is given by the leftmost or rightmost 4 bits for P and Q respectively. Binary value (0) means 16.

TABELLE 6 CSIZE-Register

5,7 Register DELTAX und DELTAY

Diese Register werden bei den Grundvektorbefehlen verwendet und bedeuten die Projektion der Vektorlänge auf die jeweilige Achse. Ihre Bedeutung erhalten diese Register erst bei der Ausgabe von Grundvektoren. Im Befehl wird dann auch das Vorzeichen übergeben.

Mit DELTAX=DELTAY=0 können einzelne diskrete Punkte, deren Lage durch die Register X und Y definiert sind, ausgegeben werden.

8,9,A,B Register X und Y

Diese Register beinhalten den Startpunkt für jede Operation (Vektor Zeichnen oder Schriftausgabe)

Die Übergabe erfolgt in 12 Bit; damit ergibt sich ein virtueller Raum von 4096 x 4096 Bit, der vom Baustein auch verwaltet wird. 512 (x) x 256 (y) Bit werden angezeigt.

C,D Lichtgriffelregister

Hier wird vom Baustein der aktuelle Stand des Lichtgriffels übergeben. Diese Funktion ist im TERM1 nicht vorgesehen.

Anwendungsbeispiele

Wir haben zehn kleine Programme zusammengestellt, damit Sie sich mit dem Gerät und der Art der Programmierung etwas vertraut machen können.

Arbeiten Sie diese Beispiele durch, tippen Sie sie in den Rechner ein und verwenden Sie die Befehlsübersicht dazu!

Dies Beispiele sind im Microsoft MBASIC geschrieben - Das Graphiksystem ist in diesen Beispielen an den Druckerausgang angeschlossen und deshalb mit LPRINT-Befehlen anzusprechen.

Sollte Ihr Graphiksystem als Monitor geschaltet sein, so ersetzen Sie einfach alle "LPRINT"-Befehle durch "PRINT"-Befehle.

Beispiellist schon etwas trivial: Es schreibt das Wort "TEST" auf den Bildschirm.

```

8>TYPE DEMO1.BAS
10 'DEMO1 GES V.1.00
20 'SCHREIBT LEDIGLICH "TEST" AUF DEN BILDSCHIRM
100 LPRINT "TEST"
110 END
  
```

Beispiell

Im Beispiel 2 wird schon mehrmals eine Zeile angesprochen - hier wird das Scrollen gezeigt.

```

8>TYPE DEMO2.BAS
10 'DEMO2 GES V.1.00
20 'SCHREIBT SCHON 30 MAL TEST 1 ... TEST30 AUF DEN BILDSCHIRM
30 'HIER WIRD DAS SCROLLEN GEZEIGT
40 '
100 FOR I=1 TO 30
110 LPRINT "TEST":I
120 NEXT I
130 END
  
```

Beispiel 2 zeigt das Scrollen

im Beispiel 3 ist die Auswahl des Zeichensatzes (Deutsch oder Amerikanisch) erläutert. Beim Arbeiten mit deutschen Umlauten wird die Verarbeitungsgeschwindigkeit erheblich geringer, da hier nicht mehr auf den eingebauten Zeichengenerator zurückgegriffen werden kann, sondern die Umlaute und das scharfe "ß" Punkt für Punkt geplottet werden.

```

8>TYPE DEMO3.BAS
10 'DEMO3 GES GMBH V 1.00
20 'LÖSCHEN EINER SEITE UND UMSCHALTEN ASCII - DEUTSCHE UMLAUTE
30 '
40 ESC%=CHR$(27) 'ESCAPE
100 LPRINT CHR%(&H1A) 'BILDSCHIRM NUN GELÖSCHT
110 '
120 GOSUB 1000 'ZUNÄCHST ASCII - GEHT SCHNELL
130 '
  
```

```

140 LPRINT ESC$:"z 1"
150 GOSUB 1000
151
160 END
1000 FOR I=1 TO 24
1010 LPRINT "VIELE SCHLAUE FÜCHSE KAUFEN JEDE WOCHTE WEITERE VIER BEQUEME PELZE; Ä Ö Ü ß"
1020 NEXT I
1030 RETURN

```

```

'UMSCHALTEN ZEICHENSATZ DEUTSCH
'GEHT LANGSAMER, DA NUN
'GEPLOTTET WIRD !

```

Beispiel 3 zeigt den Zeichensatz

Im Beispiel 4 zeigen wir einige Anwendungen der Steuerzeichen im Terminal-Betrieb. Zunächst wird der Bildschirm vollgeschrieben, dann kann der Cursor über das Bild gesteuert werden. Hier können auch Zeilen gelöscht oder eingefügt werden. Welche Tasten was bewirken, geht aus dem Programm hervor:

```

B>TYPE DEMO4.BAS
10 'DEMO4 GES GMBH V. 1.00
20 'CURSORSTEUERUNG
30 '
40 ESC$=CHR$(27)
100 LPRINT CHR$(&H1A)
110 '
120 GOSUB 1000
130 '
140 LPRINT "CURSORSTEUERUNG: 8 NACH OBEN, 6 NACH RECHTS, 2 NACH UNTEN UND 4 NACH LINKS"
145 LPRINT "7 ZEILE EINFÜGEN, 3 ZEILE LÖSCHEN, 0 ENDE"
150 GOSUB 2000
160 END
1000 FOR I=1 TO 20
1010 LPRINT "VIELE SCHLAUE FÜCHSE KAUFEN JEDE WOCHTE WEITERE VIER BEQUEME PELZE; Ä Ö Ü ß"
1020 NEXT I
1030 RETURN
2000 'EINHOLEN
2010 A$=INPUT$(1)
2020 IF A$="8" THEN LPRINT CHR$(&HB);
2030 IF A$="6" THEN LPRINT CHR$(&HC);
2040 IF A$="2" THEN LPRINT CHR$(&H16);
2050 IF A$="4" THEN LPRINT CHR$(&HB);
2052 IF A$="7" THEN LPRINT ESC$;"E";
2054 IF A$="3" THEN LPRINT ESC$;"R";
2060 IF A$="0" THEN RETURN
2070 GOTO 2010

```

```

ESCAPE
'BILDSCHIRM NUN GELÖSCHT
'BILDSCHIRM VOLLMACHEN
'EINHOLEN UND AUSFÜHREN
'UP
'RIGHT
'BACKSPACE
'INSERT LINE
'DELETE LINE
'END

```

Beispiel 4: Cursorsteuerung

Nun aber zur Graphik!. Die Graphik-Betriebsart erreicht man durch die Sequenz: Escape Escape G, wie in Zeile 50 gezeigt. Das einfache Beispiel zeichnet einige Linien.

```

B>TYPE DEMOS.BAS
10 'DEMOS  GS GMBH V 1.00
20 'ERSTE GRAPHIK
30
40 ESC#=CHR$(27)
50 LPRINT ESC#;ESC#;"G"          'UMSCHALTEN IN GRAPHIK
60 LPRINT "Z"                    'LÖSCHEN BILDSCHIRM
70 LPRINT "M0,0"                  'ZUM PUNKT 0,0 FAHREN
80 LPRINT "D0,100"                'NACH 0,100 LINIE ZIEHEN
90 LPRINT "D100,100"              'USW
100 LPRINT "D50,150"
110 LPRINT "D0,100"
120 LPRINT "M100,100"
130 END
140 LPRINT "D0,0"

```

Beispiel 5 : Einfache Graphik

Mit dem Befehl "G" kann man dem Graphik Display Prozessor direkte Steuerbefehle übermitteln. Dies ist im entsprechenden Kapitel erklärt. Wir wenden hier im Beispiel 6 und 7 folgende Steuermöglichkeiten an:

- Umschalten auf Lösch-Darstellung, um Linien zu löschen.
(Beispiel 6)
- Verändern der Buchstabengrösse und Richtung
(Beispiel 7)

```

B>TYPE DEMO6.BAS
5 'DEMO6
10 'ZEICHEN UND LÖSCHEN VON LINIEN
20 'ERSTE GRAPHIK
30
40 ESC#=CHR$(27)
50 LPRINT ESC#;ESC#;"G"          'UMSCHALTEN IN GRAPHIK
60 LPRINT "Z"                    'LÖSCHEN BILDSCHIRM
70 LPRINT "M 0,0"
80 LPRINT "D100,100"
90 LPRINT "D0,100"
100 LPRINT "D0,0"                'DREIECK ZEICHNEN
105 FOR I=1 TO 1000:NEXT I       'VERZÖGERUNG
110 LPRINT "G 1,1"              'GDP DIREKT KOMMANDO-
111                               'ALLE FOLGENDEN DRAW-
112                               'KOMMANDOS LÖSCHEN NUN
120 LPRINT "D100,100"
130 LPRINT "G 1,3"              'WIEDER NORMAL
140 LPRINT "D511,0"
150 END

```

Beispiel 6 Zeichnen und Löschen von Linien

```

10 'DEM07 GES GMBH GRAF
20 'BUCHSTABENDARSTELLUNG
30 '
35 LPRINT CHR$(27);CHR$(27);"G";LPRINT "Z"
36 LPRINT "MO,0"
40 LPRINT "BA"
50 LPRINT "G3 $33"
60 LPRINT "BA"
70 LPRINT "G2 $04"           'KURSIV
80 LPRINT "BA"
90 LPRINT "M300,0"
95 LPRINT "G3 $FF"         'MAX GRÖSSE
100 LPRINT "G2 $08"       'VERTIKAL
110 LPRINT "BA"

```

Beispiel 7 Buchstabengrößen und Richtung ändern

Unser Graphik-System verfügt über vier unabhängige Bildschirmseiten. Unabhängig deshalb, da man in eine Bildschirmseite eintragen kann und gleichzeitig eine andere Bildschirmseite anzeigen kann. Beispiel 8 zeigt diese Möglichkeit.

```

B>TYPE DEM08.BAS
5 'DEM08 GES GMBH GRAF
10 'SEITENUMSCHALTEN.
20 '
30 '
40 ESC$=CHR$(27)
50 LPRINT ESC$;ESC$;"G"           'UMSCHALTEN IN GRAPHIK
60 LPRINT "Z"
65 LPRINT "MO 0;G2 0; X 0"       'KOMMANDOS SHACHTELN MÖGLICH!
66 LPRINT "P 0"                 'SCHREIB=LESESEITE=0
70 LPRINT "R511 255"
75 LPRINT "M10 200"
80 LPRINT "G3 $22"
90 LPRINT "BDIES IST DIE SEITE 0: SIE IST IMMER NACH "
100 LPRINT "M 10 180"
110 LPRINT "BDEM EINSCHALTEN VOREINGESTELLT"
115 FOR I=1 TO 500:NEXT         'WARTEN
120 LPRINT "P 5"               'SCHREIBSEITE=1
121                             'LESESEITE =1 EINSTELLEN
130 LPRINT "M 10 160"
140 LPRINT "BUND DAS IST SEITE 1"
150 FOR I=1 TO 500:NEXT
160 LPRINT "P 10"              'SEITE 2:2*4+2=10
170 LPRINT "M10 140"
180 LPRINT "BSEITE 2"
185 FOR I=1 TO 500:NEXT

```

```

190 LPRINT "G3 $11"
195 LPRINT "P 15"
200 LPRINT "M10 100"
205 LPRINT "DWAHREND SIE HIER LESEN, WIRD IN SEITE 0 GEZEICHNET"
210 LPRINT "P 3"
215 LPRINT "M 5 5"
220 FOR IX=10 TO 100 STEP 5
230 FOR IY=10 TO 100 STEP 10
235 LPRINT "R ";IX;" ";IY
240 NEXT IY
250 NEXT IX
260 LPRINT "P0"
265 FOR I=1 TO 500: NEXT
270 LPRINT "X 50"
280 FOR I=1 TO 5000: NEXT I
290 LPRINT "X 1"
300 FOR I=1 TO 3000: NEXT
310 LPRINT "X 20"
999 END

```

'SCHRIFT KLEINER

'SEITE 3: 3*4+3

'SCHREIBSEITE=0. LESE=3

'JETZT ZEIGEN!

'UMSCHALTEN

'SCHNELL UMSCHALTEN

Beispiel 8 Vier Bildschirmseiten

DAS Graphik-System verfügt über eingebaute Funktionen zum Zeichnen von:

- Kreisen, Ellipsen und Kreissegmenten
- Dreiecken
- Polygonen
- Rechtecken

Beispiel 9 gibt hierüber Auskunft.

```

B>TYPE DEM09.BAS
10 'DEM09 GES GMBH GRAF 15.8.83
20 'KREISE UND KREISSEGMENTE
30 'SOWIE DREIECKE
40 '
50 ESC$=CHR$(27)
60 LPRINT ESC$;ESC$;"G"
70 LPRINT "Z"
80 LPRINT "M250 125"
90 LPRINT "D 30 20 0 0"
100
110
120 GOSUB 260
130 LPRINT "M 200 100"
140 LPRINT "D 30 20 0 0 1"
150 GOSUB 260
160 LPRINT "M 50 175"
170 LPRINT "D 100 90 110 140 1"
180
190 GOSUB 260
200 LPRINT "Z"
210 LPRINT "I 0 0 400 0 200 200"
220 GOSUB 260
230 LPRINT "Z"
240 LPRINT "L 0 0 100 0 130 100 50 100"
250 END
260 FOR I=1 TO 500:NEXT:RETURN

```

/ -35-

Beispiel 9 Ellipsen, Kreise, Polygoe

Im Beispiel 10 zeigen wir, was unseres Wissens kein System dieser Preisklasse kann: Das Definieren und Zeichnen von Figuren in Echtzeit.

Mit einem Befehl kann ein Fadenkreuz an jeder Bildschirmseite an jeder Position dargestellt werden. Mit einer einfachen ASCII-Zeichenkette kann die Figur des Fadenkreuzes zu einer beliebigen Figur umdefiniert werden. Nun kann diese vom Anwender definierte Figur wievorher das Fadenkreuz am Bildschirm positioniert werden. Aber nicht nur das - die Figur kann auch durch einen Befehl vergrößert oder verkleinert werden!

Beispiel 10 zeigt diese Möglichkeit - das Fadenkreuz wurde zu einem Auto umdefiniert.

```
B>TYPE DEMO10.BAS
10 'DEMO 10 GES GMBH GRAF 15.8.83
20 'FADENKREUZ UND AHNliche FIGUREN
30 '
40 '
50 ESC$=CHR$(27)
60 LPRINT ESC$;ESC$;"G"           'UMSCHALTEN IN GRAPHIK
70 LPRINT "Z"
75 LPRINT "WA"                    'FADENKREUZSYMBOL AUSWAHLEN
80 LPRINT "F 100 100 0"          'ZEICHNE FADENKRUZ
90 FOR IX=100 TO 300 STEP 20     'UND FAHRE RECHTS
100 LPRINT "F ";IX;" 100 0"
110 NEXT IX
120 '
150 FOR I=1 TO 500:NEXT
155 LPRINT "MO 92;R 511 2"
160 LPRINT "WA1XXOH1XXOH1XRHLAAATTTTTTTTTVVVVVVTTVPPPOYY1XXZTTTTVN"
165 LPRINT "WC 2"                'VERGRÖSSERN UM 2
170 LPRINT "F100 100 0"
180 FOR IX=100 TO 500
190 LPRINT "F ";IX;" 100 0"
200 NEXT IX
```

Beispiel 10 - Fadenkreuz und Auto

Wie entsteht der Befehl in Zeile 160?

Dazu dient das folgende Bild: Die Figur wird in sogenannte Kurzvektoren zerlegt; diese Kurzvektoren werden durch ein ASCII-Zeichen dargestellt. Eine "1" schaltet den "Zeichenstift" ein, eine "0" aus.

Beispiel: Schalte ein, gehe drei Schritte nach rechts "IX", dann einen Schritt nach oben "IXJ", dann einen Schritt nach links "IXJL", dann schalte aus "IXJLO" und gehe einen Schritt nach unten "IXJLON" und schalte wieder ein "IXJLON1" usw.

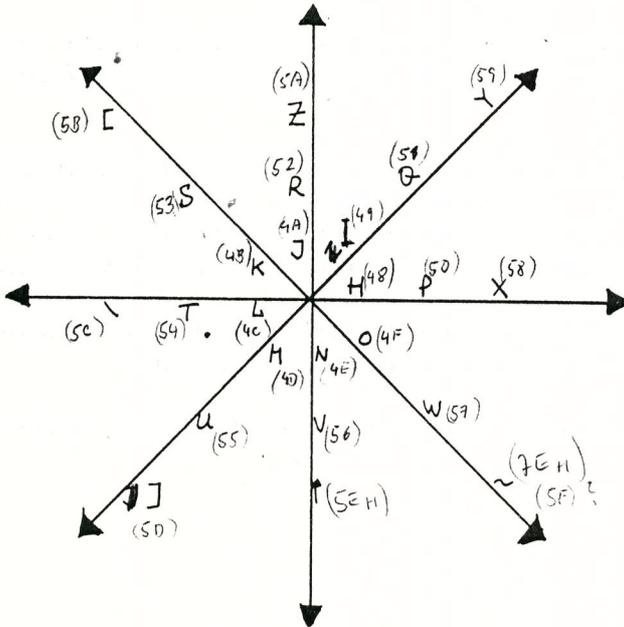


BILD: Kurzvektoren und Ihre Verschlüsselung"

Die im weiteren Anhang stehenden Beispielprogramme sollen nicht weiter erläutert werden. Hier ist für fortgeschrittene Programmierer die LOGO und Tectronix-Fähigkeit des Graphik-Systemes erläutert.

```

E>TYPE E:DEM01.BAS
10 CLEAR 2000
20 REM Beispiele fuer TERM 1
30 REM Rolf-Dieter Klein 830508
40 REM TERMINAL-TVI-950 Funktionen
50 PRINT CHR$(26); 'Clear Screen
60 FOR X=0 TO 79 'Alle Spalten
70 Y = SIN(X/79*2*PI)*11+11
80 Y = INT(Y)
90 PRINT CHR$(27);"=";CHR$(32+Y);CHR$(32+X);"*"
100 NEXT X
110 REM Wegschieben durch Scrollen
120 FOR I=1 TO 40:PRINT:NEXT I
130 REM
140 REM verschiedene Zeichensaetze
150 PRINT CHR$(27);"=";CHR$(32+5);CHR$(32)
160 PRINT CHR$(27);"z0" 'amerikan Zeichen
170 PRINT
180 FOR I=32 TO 127:PRINT CHR$(I);:NEXT I
190 PRINT
200 PRINT CHR$(27);"z1" 'deutsche Zeichen
210 FOR I=32 TO 127:PRINT CHR$(I);:NEXT I
220 PRINT
230 FOR I=1 TO 1000:NEXT I
240 REM Einfuegen, Loeschen etc.
250 FOR J=1 TO 2
260 PRINT CHR$(26);CHR$(27);"=$ "
270 PRINT "Testsatz fuer Einfuegen etc."
280 PRINT CHR$(11); ' cursor up
290 FOR I=1 TO 10:PRINT CHR$(27);"E";:NEXT I
300 FOR I=1 TO 10:PRINT CHR$(27);"R";:NEXT I
310 FOR I=1 TO 10:PRINT CHR$(27);"Q";:NEXT I
320 FOR I=1 TO 10:PRINT CHR$(27);"W";:NEXT I
330 PRINT 'cr ausgeben, wegen autocr von Basic
340 NEXT J

```

```

TYPE DEM02.BAS
10 REM Graphic Modus
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"GZPO" 'Graphic Modus
40 PRINT "MO 0;R511 255" 'Rechteck-Rahmen zeichnen
50 PRINT "M50 200;G3 $22;BGraphic-Befehle"
60 PRINT "M200 50;O100 50 0 0 1"
70 PRINT "1300 0 511 20 400 200"
80 REM Symbol definieren
90 PRINT "WA";"HHHHIJKKLLMMNO"
100 PRINT "Y1" 'zwei Seiten quasisimultan
110 PRINT "WC2 0" 'VERGROESSERUNG 2 DREHUNG 0
120 FOR X=0 TO 511
130 PRINT "F";X;INT(SIN(X/100*2*PI)*120+127);1
140 NEXT X
150 PRINT "A" 'in Alpha zurueck
999 PRINT "A"

```

```

TYPE DEM03.BAS
10 REM Statuszeile ueber Graphic Modus
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"G" 'ohne Loeschen
40 PRINT "PO;MO 0;R511 15" 'mit Rahmen
50 PRINT "P5;MO 0;R511 15" 'beide Seiten
60 PRINT "PO;G3 $22;M40 -1;BStatus Zeile doppelt hoch"
70 PRINT "P5;M40 -1;BStatus Zeile hoch"
80 PRINT "A" 'wieder zurueck in Textmodus
90 LIST

```

27_D ≙ 13_H ≙ ESC

22_D ≙ 14_H ≙ SYN(?)

```
TYPE DEMO4.BAS
10 REM Kleine LOGO-Demonstration
20 REM Rolf-Dieter Klein
30 PRINT CHR$(27);CHR$(27);"L" 'std mode
40 PRINT "M255 255 90" 'umgerechnet auf 512x256
50 FOR I=0 TO 360/10
60 GOSUB 130 'Muster 1
70 PRINT "P-10"
80 NEXT I
90 PRINT "TO" 'Turtle off
100 FOR I=1 TO 1000:NEXT I
110 PRINT "A"
120 END
130 REM Unterprogramm Figur
140 FOR N=1 TO 4
150 PRINT "F150;P";360/4
160 NEXT N
170 RETURN
```

```
E>TYPE DEMO5.BAS
10 REM Tektronix Demo
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"TO 0 1 2 1"
40 FOR X=0 TO 1023 STEP 4
50 Y=(SIN(X/1023*2*PI)*500+512)
55 Y=INT(Y)
60 GOSUB 100 'Tektronix Ausgabe
70 NEXT X
80 FOR I=1 TO 1000:NEXT I
90 PRINT CHR$(27);CHR$(27);"A":END
100 REM x,y trennen und Ausgeben
105 PRINT CHR$(28);CHR$(7); 'Pointplot mode
110 PRINT CHR$(INT(Y/32)+32);CHR$(Y MOD 32 + 96);
CHR$(INT(X/32)+32);CHR$(X MOD 32 + 64)
999 RETURN
```