Vier Pseudo-VICs mit 32 Sprites

Sie wollen mit 32 Sprites und vier Bildschirmbereichen gleichzeitig arbeiten? Nichts leichter als das. Mit Provic 64 können simultan Grafik, Text oder veränderte Zeichensätze dargestellt werden.

Die Autoren (Jürgen, 21, Physikstudent, und Stefan, 18, Schüler) haben sich Mitte 1983 einen Commodore 64 angeschafft. Schon nach kurzer Zeit stellte sich der bei den C 64-Fans übliche Frust über die schlechte Dokumentation und die schwierige Informationsbeschaffung ein, besonders wenn es um die speziellen Grafikfähigkeiten dieses Computers geht. So sitzen wir oft stundenlang vor dem Bildschirm, der nur undefinierbare Zeichen zeigt, weil wir bei dem Versuch, die Geheimnisse des C 64 zu enträtseln, in irgendeinen unbekannten Darstellungsmodus geraten sind.

Dabei entdeckten wir, daß der C 64 nicht nur acht, sondern auch 16, 24, 32 oder noch mehr Sprites gleichzeitig zeigen kann. Zusätzlich ergibt sich die Möglichkeit, mehrere Bildschirmmodi zu mischen.

Nun haben wir uns entschlossen, den Kunstgriff, der dies ermöglicht, anderen C 64-Fans nicht vorzuenthalten. Also entwickelten wir ein Programm in Maschinensprache und dazu ein kleines Demonstrationsprogramm in Basic.

Zum Programm

Durch Aufruf der Initialisierungsroutine wird der Interruptmechanismus des C 64 verändert. Nicht mehr der Timer der CIA 1, sondern der VIC 6567 löst jetzt den Interrupt aus, und zwar synchron zum Takt des Bildschirmsignals. Außerdem werden vier sogenannte Pseudo-VICs eingerichtet. Alle PO-KEs, von Spritebewegung über Bildschirmfarbe bis zur Grafikkonfiguration, müssen ab jetzt in diese Pseudo-VICs erfolgen. Jeder dieser Pseudo-VICs ist für einen der vier Bildschirmbereiche zuständig:

Der Bildschirm wird in vier waagerechte Bereiche aufgeteilt, deren Grenzlinien fast beliebig nach oben oder unten verschoben werden können. Jeder einzelne Bereich kann acht Sprites darstellen und eine eigene Farb- und Grafikkonfiguration aufweisen. So können zum Beispiel Normalschrift, HiRes-Grafik, Multicolor-Grafik und eventuell ein selbstdefinierter Zeichensatz gleichzeitig auf dem Bildschirm gezeigt werden.

Provic 64 kann selbstverständlich wieder abgeschaltet werden (bei Kassetten- oder Diskettenoperationen nötig).

Für Maschinensprachefreaks nun eine kurze Funktionsbeschreibung der Interruptroutine:

Bei Aufruf der Einschaltroutine (SYS 52544) wird der IRQ-Vektor auf die Hauptroutine des Provic 64 gestellt und der bisherige Interrupt durch den Timer A der CIA 1 verboten, während der Raster-IRQ des VIC 6567 erlaubt wird.

Sobald der Bildschirmstrahl die eingestellte Rasterzeile erreicht, wird ein Interrupt ausgelöst und der Prozessor bearbeitet die Hauptroutine des Provic 64. In dieser wird zunächst an-

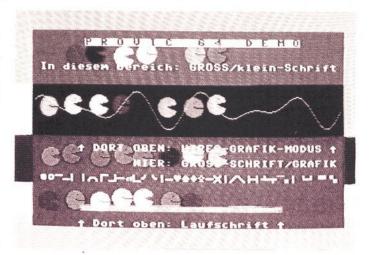
hand eines Zählers (\$ CFFF) festgestellt, welcher Bildschirmbereich an der Reihe ist. Dann wird die Rasterzeile, die das Ende dieses Bildschirms kennzeichnet, eingestellt.

Anschließend werden, falls ein entsprechendes Flag gesetzt ist, die Sprite- und andere Bildschirmparameter in den VIC 6567 übertragen. Nach dem Weiterzählen des IR-Zählers (\$ CFFF) wird entweder der Interrupt beendet, oder zur IRQ-Routine des Betriebssystems gesprungen (nach jedem vierten Interrupt). So werden in der Sekunde 200 Interrupts (vier pro Fersehbild) ausgelöst und 50 mal in der Sekunde (einmal pro Bild) die normale IRQ-Routine abgearbeitet. Dadurch zählt die interne Uhr TI in 50stel Sekunden und TI\$ wird unbrauchbar.

Beim Aufruf der Ausschaltroutine wird der Raster-IRQ des VIC 6567 unterbunden, der Interrupt des Timers A in CIA 1 erlaubt und der IRQ-Vektor auf die IRQ-Routine des Betriebssystems eingestellt.

Handhabung der Pseudo-VICs

Im Grunde ist jeder der vier Pseudo-VICs wie der echte VIC zu behandeln. Ausnahmen sind hier nur die Register 30 (Sprite-Sprite-Kollision) und 31 (Sprite-Hintergrund-Kollision), die sich auf den jeweils vorausgegangenen Bildschirmbereich beziehen. Die Register 19 und 20 (Lightpenkoordinaten), sowie 25 und 26 (IRQ-Flags und -maske) werden nicht behandelt, da diese Funktionen nur direkt über den VIC 6567 sinnvoll gehandhabt werden können. Außerdem hat jeder Pseudo-VIC noch zusätzliche Register für zwei Flags (REG 47 und REG 57), acht Sprite-Pointer (REG 48 bis REG 55), Videomatrix-Anfangsadresse Highbyte (REG 56) und die CIA 2, REG 0, Bits 0 und 1 (REG 58) (Adreßbits 14 und 15 des VIC 6567).



So werden die Fähigkeiten von »Provic 64« demonstriert

 Die vier Basisadressen der PVICs sind:

 PVIC 1
 52992 (\$ CF00)
 = REG 0

 PVIC 2
 53056 (\$ CF40)
 = REG 0

 PVIC 3
 53120 (\$ CF80)
 = REG 0

 PVIC 4
 53184 (\$ CFC0)
 = REG 0

Da die Pseudo-VICs praktisch gleichberechtigt sind, hier die Registerbeschreibung eines Pseudo-VICs:

REG 0:

X-Koordinate des Sprite 0

REG 1:

Y-Koordinate des Sprite 0 Beachte: Die Y-Koordinaten sollten im Bereich

des zugehörigen Bildschirmbereichs liegen, sonst ist der Sprite nicht zu sehen. Näheres sie-

he unten.

REG 2 bis 15: REG 16: Wie REG 0 und 1, aber für Sprites 1 - 7 MSB (höchstes Bit) der X-Koordinaten

REG 17:

Bits 0 bis 2: Y-Abstand der Zeichen vom oberen

Bildrand in Rasterzeilen (Softscrolling!)

REG 17:

Bit 3: Umschaltung 24/25-Zeilendarstellung Bit 4: Bild an/aus: es hat keinen Sinn, das Bild teilweise ausschalten zu wollen, da der VIC dieses Bit nur einmal pro Fernsehbild prüft (also ent-

weder das ganze Bild an oder aus) Bit 5: HiRes-Grafik-Modus an Bit 6: Hintergrundmehrfarb-Modus an

Bit 7: Nummer der Interrupt-Rasterzeile Bit 8; es hat wenig Sinn, dieses Bit zu setzen, da so nur Rasterzeilen angesprochen werden, die unterhalb des Bildschirms liegen. Ist in irgendeinem Pseudo-VIC die 9-Bit-Zahl für die Rasterzeile größer als 311, so wird überhaupt kein IRQ mehr

ausgelöst.

REG 18:

Nummer der Rasterzeile Bits 0 — 7; hier ist anzugeben, wann der nächste Interrupt ausgelöst werden soll, das heißt wo der Bildschirmbereich dieses PVICs zu Ende sein soll. Dabei sollte folgende Reihenfolge eingehalten werden: REG 18: PVIC 1 < PVIC 2 < PVIC 3 < PVIC 4 (Zykli-

sche Vertauschungen möglich!)

REG 19 und 20:

nicht verwendet (siehe oben) Sprite enable (einschalten)

REG 21: Sprite enable (einschalt REG 22: Bits 0 bis 2: Soltscrollin

Bits 0 bis 2: Soltscrolling in X-Richtung

Bit 3: Umschaltungen 38/40-Spaltendarstellung

Bit 4: Multicolor-Modus ein Bit 5 bis 7: unbenutzt

REG 23:

Sprite vergrößern in Y-Richtung

REG 24:

Bits 1 bis 3: Adresse Zeichengenerator (Bits 11

bis 13)

Bits 4 bis 7: Adresse Video-RAM (Bits 10 bis 13)

REG 25 und 26: nicht verwendet (siehe oben)
REG 27: Sprite-Priorität vor Hintergrund
REG 28: Flags für Multicolor-Sprites
REG 29: Sprite vergrößern in X-Richtung
REG 30: Sprite-Sprite-Kollision
REG 31: Sprite-Hintergrund-Kollision

Achtung: Geben die Kollisionen des vorangegangenen Bildschirmbereichs an: Findet im Bereich von PVIC 3 eine Kollision statt, wird dies im PVIC 4 registriert. Kollisionen im Bereich von PVIC 4 werden im PVIC 1 registriert. Dieses Register muß gelöscht werden, um neue Kollisionen anzei-

gen zu können!

REG 32: Rahmenfarbe

REG 33 bis 36: Hintergrundfarben 0 bis 3
REG 37 und 38: Multicolor-Sprite-Farben 0 und 1
REG 39 bis 46: Farben für Sprites 0 bis 7

REG 47:

RFG 58:

Flag für Spritebehandlung; nur wenn der Inhalt dieses Registers nicht Null ist, werden die Register, die etwas mit Sprites zu tun haben, vom PVIC in den VIC 6567 übertragen. Das sind REG 0 bis REG 16, REG 21, REG 23, REG 27 bis REG 31, REG 37 bis 46 sowie REG 48 bis 55. Ist der Inhalt Null, gelten für die Sprites die Werte des vorherigen PVICs, während die Kollisionen erst im nächsten PVIC, in dem REG 48 ungleich

Null ist, angezeigt werden.

REG 48 bis 55: Sprite-Pointer für Sprites 0 bis 7; Die Pointer auf

die Bitmuster der Sprites werden nicht mehr in die Speicherzellen 2040 bis 2047 geschrieben,

sondern in diese Register des PVICs.

REG 56: In diesem Register muß das Highbyte der Video-

RAM-Anfangsadresse plus 3 stehen; normalerweise also 4+3=7 (da der Bildschirm nach dem Einschalten des Computers bei 1024 beginnt, 1024 = \$ 0400). Bei Verlegung des Video-RAMs ist also der Inhalt dieses Registers

zu korrigieren.

REG 57: Flag für Bildschirmparameter-Behandlung; nur

wenn der Inhalt dieses Registers nicht Null ist, werden die REG 17, 22, 24, sowie 32 bis 36 und REG 58 in den VIC 6567 übertragen. Bits 0 und 1: Adressbits 14 und 15 des VIC

6557; werden nach CIA 2 REG 0 Bits 0 und 1 übertragen. Mit diesen Bits kann Video-RAM, Charaktergenerator, Grafik-Bitmap in 16-KByte-

Schritten verschoben werden. Da die Bits lowaktiv sind, sind sie beim Einschalten gesetzt (also REG 58 = 3). Bits 2 bis 7: unbenutzt, immer 0.

Übergang eines Sprites zwischen zwei Bildschirmbereichen:

Soll ein Sprite zwischen zwei Bildschirmbereichen wechseln, muß in beiden Bereichen derselbe Sprite (also zum Beispiel beidesmal Sprite 4) die gleiche Position besitzen, und zwar so lange, wie der Sprite die Trennlinie zwischen den Bereichen überdeckt. Wird dies nicht beachtet, werden die entsprechenden Sprites zerschnitten und verschoben.

Aktivieren von Provic 64: Von Basic aus mit SYS 52544 und von Maschinensprache aus mit JSR \$CD40.

Ausschalten von Provic 64: Von Basic aus mit SYS 52970 und von Maschinensprache aus mit JSR \$CEEA.

Der Basic-Lader:

Der Lader erzeugt Provic 64 aus den DATA-Zeilen und falls kein Prüfsummenfehler vorliegt, wird Provic 64 sofort als Maschinenprogramm auf Floppy oder Datasette (Zeile 400 entsprechend ändern!) abgespeichert. Dieses Maschinenprogramm enthält auch gleich die Standardwerte der Pseudo-VICs.

Laden von Provic 64: Im Programm am besten mit der Zeile IF PEEK (52544) > < 120 THEN LOAD "PROVIC 64", Gerätenummer, 1 die am Anfang des Basic-Programms stehen sollte

Das Demonstrationsprogramm zeigt einige der Vorzüge von Provic 64. Es ist nur als Anregung gedacht, deshalb verzichten wir hier auf eine nähere Beschreibung.

Provic 64 ist nicht nur für Basic-Programmierer, sondern vor allem auch für Maschinensprache-Freaks gedacht, da erst durch schnelle Maschinenprogramme die Möglichkeiten von Provic 64 voll ausgeschöpft werden können.

(Jürgen und Stefan Haas/rg)



Tabellarische Übersicht zu Provic		
Belegter Adre	eßraum:	
\$CD40	Einschaltroutine	
\$CD58	Interruptroutine	
\$CEEA	Ausschaltroutine	
\$CF00	Pseudo-VIC 1	
\$CF40	Pseudo-VIC 2	
\$CF80	Pseudo-VIC 3	
\$CFC0	Pseudo-VIC 4	
\$CFFF	Interruptzähler	
Provic 64 eins	schalten:	
in Basic: SYS	52544	
	sprache: JSR \$CD40	
Provic 64 aus		
in Basic: SYS		
	sprache JSR \$CEEA	
Benutzte RAM		
in der Zero-Page: 187 (\$BB)		
188 (\$BC)		
STATE OF THE PARTY	zuwachs bei aktiviertem Provic 64:	
	ags (REG 47) und Bildschirmparameterfla	
(REG 57) ge		+ 2,5%
für jedes ge	setzte Spriteflag (REG 47):	zirka
		+ 2,4%
für jedes ge	setzte Bildschirmflag (REG 57):	zirka
		+ 0,5%
alle Sprite- L	und Bildschirmflags gesetzt:	zirka
Falls day Days		+ 15,0%
	nner abstürzt rettet Run-Stop/Restore!	
	e TI zählt bei aktiviertem Provic 64 in 50	istel Sekunden
(statt 60stel); TI\$ wird somit	Luphroughbor	
		¢0FFF
Zeiger für Interruptaussprung von PVIC 3 bis PVIC 4: \$CEE5		
Zeiger für Interruptaussprung von PVIC 1: \$CEE8		

Basic-	ader	»Provic	64"
DUSIO	Lauci	"I LOAIC	0411

10 REM INTERRUPT-ROUTINE ZUR ERZEUGUNG 11 REM 12 REM EINES MAXIMAL VIERTEILIGEN 13 REM 14 REM BILDSCHIRMES UND 32 SPRITES	<118>
11 REM	<154>
12 REM EINES MAXIMAL VIERTEILIGEN	<159>
13 REM	<156>
14 REM BILDSCHIRMES UND 32 SPRITES	<140>
16 REM AUF EINEM COMMODORE 64 COMPUTER	<103>
17 REM	<160>
18 REM 1984 BY GEBR. HAAS	<125>
19 REM	<162>
17 REM 18 REM 1984 BY GEBR. HAAS 19 REM 100 REM PRUEFSUMMEN-KONTROLLE 101 REM 110 RESTORE:PS=0 120 FOR A=0 TO 511 130 READ WERT 140 PS=PS+WERT 150 NEXT A 160 IF PS<>60913 THEN PRINT"DATA-PRUEESUM	<049>
101 REM	<244>
110 RESTORE:PS=0	<185>
120 FOR A=0 TO 511	<087>
130 READ WERT	<075>
140 PS=PS+WERT	<112>
150 NEXT A	<089>
160 IF PS<>60913 THEN PRINT"DATA-PRUEFSUM	ME
LSPACE JFALSCH [SPACE]!": END: *	<123>
199 REM	<089>
200 REM MASCHINENCODE UEBERTRAGEN	<060>
201 REM	<088>
210 RESTORE	<094>
220 FUR A=0 TO 447	<195>
230 READ WERT	<175>
240 PUKE 52544+A, WERT	<228>
250 NEXT A	<189>
260 FUR A=0 TO 63	<182>
265 READ WERT	<211>
ISPACE JFALSCHISPACE]!":END:* 199 REM 200 REM MASCHINENCODE UEBERTRAGEN 201 REM 210 RESTORE 220 FOR A=0 TO 447 230 READ WERT 240 POKE 52544+A, WERT 250 NEXT A 260 FOR A=0 TO 63 265 READ WERT 270 FOR B=0 TO 3 280 POKE 52992+B*64+A, WERT	<139>
280 PURE 52992+8*64+A, WERT	<022>

		<113>
295 NEX		<089>
299 REN		<187>
		<029>
301 REN		<189>
		<220>
	R A=1 TO LEN(N\$)	<016>
340 NEX		<107>
	e in the second	<024>
		<054>
		<005>
	KE 193,64: POKE 194,205	(121)
390 POK		<074>
400 POK	(E 186,8: REM FUER FLOPPY	
: 8	FUER DATASETTE	: 1
100000000000000000000000000000000000000	<174>	
		<100>
420 END		<037>
999 REM		<121>
		<104>
1001 RE		<123>
140	ATA 120,169,88,162,205,141,20,3,142,2	
1003 00	,1,141,13,220,141,26,208 ATA 141,255,207,88,96,169,1,141,25,20	<170>
172		<105>
1004 De	ATA 240,3,162,192,44,162,128,44,162,6	4 44
162.		<253>
	ATA 189,47,207,208,3,76,141,206,173,3	
208,	29,30,207,157,30,207,173,31	<048>
1006 DA	ATA 208,29,31,207,157,31,207,189,21,2	07,
		<200>
	ATA 208,189,29,207,141,29,208,189,0,2	07,
141,	0,208,189,1,207,141,1,208	<0000>
1008 DA	TA 189,2,207,141,2,208,189,3,207,141	
		<004>
	ATA 207,141,5,208,189,6,207,141,6,208	
1010 00	77,141,7,208,189,8,207,141 ATA 8,208,189,9,207,141,9,208,189,10,	<114>
141		<001>
1011 DA	ATA 208,189,12,207,141,12,208,189,13,	
		<254>
1012 DA	ATA 208,189,15,207,141,15,208,189,16,	207,
141,	16,208,189,27,207,141,27	<019>
	TA 208,189,28,207,141,28,208,189,37,	
		<038>
	ATA 208,189,39,207,141,39,208,189,40,40,208,189,41,207,141,41	CONTROL OF THE PARTY OF THE PAR
	TA 208,189,42,207,141,41	<019>
141-		(020)
1016 D	OATA 208,189,45,207,141,45,208,189,46	.207.
		<094>
1017 DA	ATA 169,248,133,187,160,0,189,48,207,	145,
187,	200,189,49,207,145,187,200	<142>
1018 DA	TA 189,50,207,145,187,200,189,51,207	
187,	200,189,52,207,145,187	<245>
1019 DA	TA 200,189,53,207,145,187,200,189,54	
	187,200,189,55,207,145 TA 187,189,57,207,240,67,189,58,207,	(241)
		<037>
	TA 29,58,207,141,0,221,189,17,207,14	
208,	189,22,207,141,22,208,189	(119)
1022 DA	TA 24,207,141,24,208,189,32,207,141,	32,
208,	189,33,207,141,33,208,189	(Ø19>
	TA 34,207,141,34,208,189,35,207,141,	
1024 00	189,36,207,141,36,208,224	(024)
2017	TA 192,208,5,169,255,141,255,207,238 138,240,3,76,129,234,76	
	TA 49,234,120,169,49,162,234,141,20,	<Ø43>
	D4 7 4/D 100 100 100	<Ø58>
1026 DA	TA 0,76,80,205,234,0,0,0,0,0,0,0,0,0	.0.0.
Ø,Ø,	0,0,0,0,27,95,0,0,0,200	(139)
1027 DA	TA 0,21,121,240,0,0,0,0,0,14,6,1,2,3	
1,2,	3,4,5,6,7,8,0,0,0,0,0,0	(140)
1028 DA	TA 0,0,0,7,1,3,0,0,0,0,0	(094>

78 **EFE**



Demo-Listing »Provic 64«

6 REM DIESES KURZE DEMO-PROGRAMM SOLL		
	<212>	
8 REM EIN PAAR DARSTELLUNGSFORMEN	(246)	
D KEN EIN FHAK DAKSTELLUNGSFURNEN		
10 REM ZEIGEN, WIE SIE MIT PROVIC-64	<161>	
12 REM RELATIV EINFACH ERREICHBAR SIND.	<211>	
14 REM 1984 BY GEBR. HAAS	(121)	
19 REM	<162>	
20 REM LADEN DER PROVIC-64 ROUTINE	<114>	
21 REM	<164>	
TO 1 (1) (1) (1)		
30 IF PEEK(52544)=120 THEN 100	<061>	
40 PRINT"[DOWN2, SPACE]AUF[SPACE]WELCHEM[SP	ACE]	
DATENTRAEGER (SPACE) IST (SPACE) PROVIC-64"	(216)	
50 PRINT"[DOWN, SPACE] VERFUEGBAR[SPACE]([SP		
FLOPPY (SPACE)=8 (SPACE) / (SPACE) DATASETTE		
[SPACE]=1[SPACE])"	(229)	
60 INPUT"[SPACE]"; A\$: A=VAL(A\$)	<077>	
70 IF A<>1 AND A<>8 THEN 40	<062>	
80 LOAD"PROVICESPACE 364", A, 1	< 046>	
99 REM	<242>	
100 REM PSEUDO-VIC'S INITIALISIEREN	<101>	
101 REM	<244>	
	1277/	
110 P1=52992:P2=53056:P3=53120:P4=53184		
:REM BASISADRESSEN DER PSEUDO-VIC'S	<192>	
120 POKE P1+21,255:POKE P1+24,22:POKE P1+4	7.1	
:POKE P1+27,255:POKE P2+21,255	<112>	
130 POKE P2+17,59:POKE P2+24,24:POKE P2+32	,7	
:POKE P2+47,1:POKE P2+27,255	<244>	
440 DOVE DT. 04 DEC. DOVE DT. 70 0. DOVE DT. 47		
140 POKE P3+21,255:POKE P3+32,9:POKE P3+47		
:POKE P3+27,255:POKE P4+32,5	<253>	
150 POKE P4+21,255:POKE P4+24,22:POKE P4+4	7.1	
:POKE P4+18.230:POKE P4+27.255		
:FURE P4+18,230:FURE P4+27,255	<155>	
199 REM	< 086>	
2000 REM SPRITE-DATEN HERERTRAGEN	<251>	
200 REH SHIZTE DATEN SEDERHAGEN		
201 REM	< 088>	
210 RESTORE	<094>	
220 FOR A=0 TO 126	<189>	
220 1011 11 0 10 120		
230 READ WERT	<175>	
230 READ WERT 240 POKE 832+A,WERT		
230 READ WERT 240 POKE 832+A,WERT 250 NEXT A	<175> <125>	
:POKE P4+18,230:POKE P4+27,255 199 REM 200 REM SPRITE-DATEN UEBERTRAGEN 201 REM 210 RESTORE 220 FOR A=0 TO 126 230 READ WERT 240 POKE 832+A,WERT 250 NEXT A	<175> <125> <189>	
299 REM	<175> <125> <189> <187>	
230 READ WERT 240 POKE 832+A,WERT 250 NEXT A 299 REM 300 REM BILSCHIRM AUFBAUEN	<175> <125> <189>	
299 REM 300 REM BILSCHIRM AUFBAUEN	<175> <125> <189> <187>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM	<175> <125> <189> <187> <160>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <187> <160> <189>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMO"	<175> <125> <189> <187> <160> <189> 89	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMO"	<175> <125> <189> <187> <160> <189> 89	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PRO LL C 6 4 DE MO" 320 PRINT"[DOWN2, SPACE][NISPACE]DIESEMISPA	<175> <125> <189> <187> <160> <189> <092> CEJ	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMO" 320 PRINT"[DOWN2, SPACE] INISPACE IDIESEMISPA BEREICH: [SPACE] GROSS/KLEIN-SCHRIFT[DOWN	<175> <125> <189> <187> <160> <189> <092> CEJ	
279 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PRO LE C 6 4 DE DOWN 320 PRINT"[DOWN2, SPACE] INTSPACE ID LESEM (SPACE) BEREICH: (SPACE) GROSS/KLEIN-SCHRIFT (DOWN 4117)	<175> <125> <189> <187> <160> <189> <092> CEJ	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMO" 320 PRINT"[DOWN2, SPACE] INISPACE IDIESEMISPA BEREICH: [SPACE] GROSS/KLEIN-SCHRIFT[DOWN	<175> <125> <189> <187> <160> <189> <092> CEJ	
279 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PRO LLC 6 4 DEMO" 320 PRINT"[DOWN2, SPACE] JN SPACE ID IESEM [SPABEREICH: [SPACE] GROSS/KLEIN-SCHRIFT [DOWN (117)] 330 FOR A=0 TO 7: PRINT"[BLUE]	<175> <125> <189> <187> <187> <160> <160> <189> <092> CEJ ":	
279 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMO" 320 PRINT"[DOWN2, SPACE] IN LSPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN-SCHRIFT[DOWN <117> 330 FOR A=0 TO 7:PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <189> <187> <160> <187> <160> <189> <189>	
279 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <160> <187> <160> <189> <r <013=""></r>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <160> <187> <160> <189> <189>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <160> <187> <160> <189> <r <013=""></r>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROPERTY OF A PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROPERTY OF A PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROPERTY OF A PRINT"[CLEAR, WHITE] 335 FOR A PRINT"[CLEAR, WHITE] PROPERTY OF A PRINT"[CLEAR, WHITE] 340 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROPERTY OF A PRINT "[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROPERTY OF A PRINT "[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROPERTY OF A PRINT "[CLEAR, WHITE, DOWN, RVSON, RVSON,	<175> <125> <189> <187> <160> <160> <160> <189> <092> CEJ J": R <013> <029>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <160> <187> <160> <189> <r <013=""></r>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <187> <160> <160> <160> <160> <187> <160> <189> <092> CEI]": R <013> <029> <150>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <160> <160> <160> <189> <092> CEJ J": R <013> <029>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMO! 320 PRINT"[DOWN2, SPACE] IN [SPACE] DIESEM [SPABEREICH: [SPACE] GROSS/KLEIN-SCHRIFT[DOWN (117)] 330 FOR A=0 TO 7: PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <189> <187> <160> <160> <160> <189> <092> CE J J": R <013> <029> <150> <096>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMON 320 PRINT"[DOWN2, SPACE] INISPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN—SCHRIFT[DOWN 4117> 330 FOR A=0 TO 7:PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <189> <187> <160> <160> <160> <187> <1092> CEJ]": R <013> <029> <150> <xcvbh< td=""></xcvbh<>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <160> <160> <160> <187> <6092> CEJ J": R <013> <029> <150> <096> <xcvah <187=""></xcvah>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMON 320 PRINT"[DOWN2, SPACE] INISPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN—SCHRIFT[DOWN 4117> 330 FOR A=0 TO 7:PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <189> <187> <160> <160> <160> <187> <6092> CEJ J": R <013> <029> <150> <096> <xcvah <187=""></xcvah>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] 2 8 0 V 1 C 6 4 D E M 0" 320 PRINT"[DOWN2, SPACE] INISPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN-SCHRIFT[DOWN	<175> <125> <189> <187> <160> <160> <160> <187> <6092> CEJ J": R <013> <029> <150> <096> <xcvah <187=""></xcvah>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROBLEM 20 PRINT"[DOWN2, SPACE] INISPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN-SCHRIFT[DOWN 117> 330 FOR A=0 TO 7:PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <189> <1879 <160> <160> <1679 <1092> CEJ]": R <013> <029> <150> <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <094 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <0945 <	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"ICLEAR, WHITE, DOWN, RVSON, RIGHT7] PROVICE 64 DEMON 320 PRINT"IDOWN2, SPACE JUISPACE DIESEMISPA BEREICH: ISPACE JGROSS/KLEIN-SCHRIFT IDOWN (117) 330 FOR A=0 TO 7: PRINT"IBLUE J RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <189> <187> <167> <1687> <1697> <1697> <1092> CE J J": R <013> <029> <150> XCVBN <187> <100BEN <246>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] PROBLEM 20 PRINT"[DOWN2, SPACE] INISPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN-SCHRIFT[DOWN 117> 330 FOR A=0 TO 7:PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <189> <187> <160> <160> <160> <189> < (187) <201]": R <013> <029> <150> <096> <a (117)="" 320="" 330="" 64="" 7:="" a="0" bereich:="" demon="" diesemispa="" down,="" for="" href="https://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</td></tr><tr><td>299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT" iclear,="" idown="" ispace="" j="" jgross="" juispace="" klein-schrift="" print"iblue="" print"idown2,="" provice="" right7]="" rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr<="" rvson,="" space="" td="" to="" white,=""><td><175> <125> <189> <187> <167> <1687> <1697> <1697> <1092> CE J J": R <013> <029> <150> XCVBN <187> <100BEN <246></td>	<175> <125> <189> <187> <167> <1687> <1697> <1697> <1092> CE J J": R <013> <029> <150> XCVBN <187> <100BEN <246>
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <187> <160> <167> <160> <187> <160> <189> <0992> CEJ]": R <013> <029> <150> <209> <150> <2096> XCVBN <187> <100BEN <246> <070> <031>	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] £ £ 0 V J C 6 4 D E M 0" 320 PRINT"[DOWN2, SPACE] JNISPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN—SCHRIFT[DOWN (117) 330 FOR A=0 TO 7: PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <187> <187> <160> <160> <160> <189> 092 CEI J": R <0013> <029> <150> <0296> <046> <046> <046> <046> <046> <046 <046 <0470 <048 <048 <048 <048 <048 <048 <048 <04	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <189> <160> <160> <160> <160> <160> <160> <1092> CEJ J": R <013> <029> <0150> <0296> <0246> <070> <031> <0310 <031> <0331> <0333> <0333	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 301 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7] £ £ 0 V J C 6 4 D E M 0" 320 PRINT"[DOWN2, SPACE] JNISPACE] DIESEMISPA BEREICH: [SPACE] GROSS/KLEIN—SCHRIFT[DOWN (117) 330 FOR A=0 TO 7: PRINT"[BLUE] RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	<175> <125> <187> <187> <160> <160> <160> <189> 092 CEI J": R <0013> <029> <150> <0296> <046> <046> <046> <046> <046> <046 <046 <0470 <048 <048 <048 <048 <048 <048 <048 <04	
299 REM 300 REM BILSCHIRM AUFBAUEN 301 REM 310 PRINT"[CLEAR, WHITE, DOWN, RVSON, RIGHT7]	<175> <125> <189> <189> <160> <160> <160> <160> <160> <160> <1092> CEJ J": R <013> <029> <0150> <0296> <0246> <070> <031> <0310 <031> <0331> <0333> <0333	

430 X=3+A/.08:Y=77-11*SIN(A)-9*CDS(A/.7)	<201>
440 AV=8192+320*INT(Y/8)+(Y AND 7)+8*INT(X/	A STATE OF THE PARTY OF THE PAR
	۵,
<016>	/BDE:
450 POKE AV, PEEK (AV) DR 21 (7-(X AND 7))	<025>
460 NEXT A	<144>
470 LA\$="***[SPACE]VON[SPACE]HAAS[SOFT[SPAC	
*** [SPACE2] FUER [SPACE] DAS [SPACE] 164 'ER [SP	
MAGAZINESPACE3]"	<072>
480 LA\$=LA\$+"***[SPACE3]P R Q V I C 6[SPA	
[SPACE3]"	<158>
490 LA\$=LA\$+LEFT\$(LA\$,25):R=53266	<074>
499 REM	<131>
500 REM DEMONSTRATINS-SCHLEIFE	<223>
501 REM	<133>
510 REM SPRITES SETZEN	<145>
7.7.7.1.1.1.1	<143>
520 FOR A=0 TO 7	<137>
530 POKE P1+2*A,30+24*A+7*RND(1)	
:PDKE P1+2*A+1,60+6*RND(1)	<156>
540 POKE P1+39+A,RND(1)*16:POKE P1+48+A,	VANCES SERVICES
13.5+RND(1)	<224>
550 POKE P2+2*A,30+24*A+7*RND(1)	- North Control of the Control of th
:POKE P2+2*A+1,110+6*RND(1)	<222>
560 POKE P2+39+A, RND(1)*16:POKE P2+48+A,	
13.5+RND(1)	(246)
570 POKE P3+2*A,30+24*A+7*RND(1)	
:POKE P3+2*A+1,160+6*RND(1)	(249)
580 POKE P3+39+A, RND(1)*16:POKE P3+48+A,	
13.5+RND(1)	<012>
590 POKE P4+2*A,30+24*A+7*RND(1)	
:POKE P4+2*A+1,207+6*RND(1)	<017>
600 POKE P4+39+A, RND(1) *16: POKE P4+48+A,	
13.5+RND(1)	<@34>
610 NEXT A	<039>
619 REM	(252)
620 REM LAUFSCHRIFT SETZEN	<017>
621 REM	(254)
625 FOR LP=1 TO LEN(LA\$)-25	<238>
630 LZ=LZ-1: IF LZ>0 THEN POKE P4+22,LZ OR 8	3
:FOR A=Ø TO 9:NEXT A:GOTO 630	<060>
640 PRINT TAB(6);: WAIT 53265,128: WAIT 53266	,64
	<125>
660 LZ=7:LF\$=MID\$(LA\$,LP,25)	<238>
670 NEXT LP	<190>
680 GET A\$: IF A\$=""THEN 500	<012>
690 SYS 52970: REM PROVIC-64 DESAKTIVIEREN	
999 REM	<121>
1000 REM SPRITE-DATEN	<234>
1001 REM	<123>
1002 DATA 0,0,0,0,126,0,1,255,128,7,255,224	
255,240,15,253,240,31,255,248	<008>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252	<008>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252 243,252,63,252,0,63,255,252,63	<008> ,63, <055>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15	<008> ,63, <055>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15 240,15,255,240,7,255,224,1,255	<008> ,63, <055> ,255, <099>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252, 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15, 240,15,255,240,7,255,224,1,255 1005 DATA 128,0,126,0,0,0,0,0,0,126,0,1,25	<008> ,63, <055> ,255, <099> 5,128,
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252, 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15, 240,15,255,240,7,255,224,1,255 1005 DATA 128,0,126,0,0,0,0,0,0,126,0,1,255,7,255,224,15,255,240,15,251	<008> ,63, <055> ,255, <099> 5,128, <197>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252, 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15 240,15,255,240,7,255,224,1,255 1005 DATA 128,0,126,0,0,0,0,0,0,126,0,1,253 7,255,224,15,255,240,15,251 1006 DATA 240,31,255,248,31,255,248,63,255	<008> ,63, <055> ,255, <099> 5,128, <197>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252, 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15, 240,15,255,240,7,255,224,1,255 1005 DATA 128,0,126,0,0,0,0,0,0,126,0,1,255, 7,255,224,15,255,240,15,251 1006 DATA 240,31,255,248,31,255,248,63,255,63,255,0,63,240,0,63,252,0,63	<pre><008> ,63, <055> ,255, <099> 5,128, <197> <197> ,240, <036></pre>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252, 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15 240,15,255,240,7,255,224,1,255 1005 DATA 128,0,126,0,0,0,0,0,0,126,0,1,253 7,255,224,15,255,240,15,251 1006 DATA 240,31,255,248,31,255,248,63,255	<pre><008> ,63, <055> ,255, <099> 5,128, <197> <197> ,240, <036></pre>
255,240,15,253,240,31,255,248 1003 DATA 31,255,248,63,255,252,63,255,252, 243,252,63,252,0,63,255,252,63 1004 DATA 255,252,31,255,248,31,255,248,15, 240,15,255,240,7,255,224,1,255 1005 DATA 128,0,126,0,0,0,0,0,0,126,0,1,255, 7,255,224,15,255,240,15,251 1006 DATA 240,31,255,248,31,255,248,63,255,63,255,0,63,240,0,63,252,0,63	<0008> ,63, <055> ,255, <099> 5,128, <197> ,240, <036> 55,