

VideoMagic LightPen

© 1984 Bernhard Engl

Reprint der Originalanleitung zum „VideoMagic LightPen“
Zur Konservierung von ATARI-Hardware und -Software durch den ABBUC e.V.
Mit freundlicher Genehmigung von Bernhard Engl

neu gesetzt von Florian Dingler
Version 1.00 - 10.03.2004

Mit dem VideoMagic Lightpen steht für die ATARI Heimcomputer erstmals ein preisgünstiger Lightpen zur Verfügung, der hohe Empfindlichkeit mit guter Auflösung vereint.

Jetzt können auch die Computerbesitzer mit kleinerem Geldbeutel die Vorzüge eines funktions-tüchtigen Lightpens genießen.

- Nur der Lightpen ermöglicht die direkte zweidimensionale Auswahl von Objekten am Bildschirm durch einfaches Daraufdeuten.
- Nur er ermöglicht die blitzschnelle Positionierung eines Cursors oder seine Bewegung in jede beliebige Richtung.
- Nur mit ihm kann man direkt am Bildschirm malen. Und das mit einfachster Software, sogar in BASIC!

All die „traditionellen“ Eingabegeräte wie Tastatur, Paddle, Joystick und Trakball können das nicht bieten. Sie sind indirekt, d. h. es ist eine Rückkopplung in Form eines Cursors o. ä. nötig. Das menschliche Gehirn sitzt dann in einem Regelkreis, dessen Kontrolle erst erlernt werden muß.

Der Lightpen verlagert den stupiden Teil in den Computer, dorthin, wo er hingehört, und setzt kreative Kapazität frei.

Von dieser Warte des „Human Engineering“ aus sind auch die vielgepriesenen „Mäuse“ ein Schritt in die falsche Richtung; sind sie doch nichts anderes als umgedrehte Trakballs.

Zweifellos gehört die Zukunft neben Spracherkennung und -synthese, berührungsempfindlichen Bildschirmen, Graphiktablets etc. auch dem Lightpen.

Grund genug, sich mit seiner Funktionsweise zu beschäftigen.

Das Fernsehbild wird von einem Elektronenstrahl erzeugt, der von oben links beginnend Zeile für Zeile die Leuchtschicht des Bildschirms anregt. Da pro Sekunde fünfzig komplette Bilder geschrieben werden und auch die Leuchtschicht etwas nachleuchtet, erscheint dem trägen menschlichen Auge ein stehendes, flimmerfreies Bild.

Der Lightpen enthält in seiner Spitze einen Phototransistor, der einen Impuls abgibt, wenn der Elektronenstrahl diese Spitze passiert. Der Impuls wird verstärkt und dem Computer zugeleitet. Im Videobaustein ANTIC befinden sich zwei Zähler, die die Horizontal- und Vertikalposition des Elektronenstrahls steuern. Der Impuls des Lightpen bewirkt die Übertragung der Zählerstände in die Lightpenregister PENH (\$D40C) und PENV (\$D40D). Durch ihre Auswertung kann die Position des Lightpens auf dem Schirm bestimmt werden.

Der Lightpen enthält noch einen Drucktaster, dessen Abfrage später besprochen wird.

Leider können die Hardwareregister PENH und PENV nur mit einer Interruptroutine sinnvoll abgefragt werden. Es könnte sonst sein, daß die Abfrage in dem Moment erfolgt, in dem sich die Register ändern. Fehlerhafte Resultate wären die Folge.

Das ATARI- Betriebssystem überträgt die Inhalte dieser Register während jedes VBLANK-Interrupts in die OS-Shadow-Register LPENH (\$234) und LPENV (\$235). Dort können die Werte auch ohne besondere Vorkehrungen mit BASIC abgerufen werden.

Vor dem ersten Beispielprogramm noch ein paar Worte zur erzielbaren Auflösung. Der Vertikalzähler (Wert in LPENV) wird alle zwei Zeilen inkrementiert. Bei normaler Bildschirmgröße (d.h. in jedem der OS-Graphikmodes) ist daher ein Y-Bereich von 0 bis 95 erzielbar.

Der Horizontalzähler (Wert in LPENH) zählt immer nur „ColorClocks“, wobei das letzte Bit laut Hardwaremanual „unsicher“ ist und ignoriert werden sollte. Es ergäbe sich eine Horizontalauflösung von 0 bis 79, was aber ein bißchen wenig wäre. Die Praxis hat aber gezeigt, daß die Unsicherheit beim „Malen“ auf dem Bildschirm ohnehin nicht auffällt, weil die führende Hand sowieso nicht zu exakten Geraden fähig ist. Daher ist die Ausnutzung des vollen Bereichs von 0 bis 159 durchaus sinnvoll.

Die Auflösung entspricht folglich, dem OS-Graphikmode 7 ohne Textfenster. Für viele Anwendungen, etwa Menüs, wird eine gröbere Unterteilung nötig sein. Man sollte bei solchen Anwendungen die menschliche Ungenauigkeit berücksichtigen und z.B. nicht die direkte Auswahl einzelner Zeichen in GRAPHICS 0 verlangen. Von der Hardware her wäre das problemlos möglich. Der Bediener jedoch muß sich auch bei einem recht großen Bildschirm zu sehr konzentrieren.

Wenn es um die genaue, gezielte Manipulation einzelner Bildpunkte geht, wird man aus dem selbigen Grund auf einen vergrößerten Ausschnitt nicht verzichten können, was übrigens auch für alle anderen Eingabegeräte gilt.

Ein weiterer Punkt, der zu berücksichtigen ist, ist die Wahl der Hintergrundhelligkeit. Der VideoMagic Lightpen ist im Gegensatz zu anderen Modellen sehr empfindlich und funktioniert daher schon bei recht niedriger Helligkeit. Um die volle Auflösung nutzen zu können, genügt in der Regel eine Helligkeitsstufe ab 8, so daß genügend Spielraum für kontrastreiche Darstellung bleibt.

Wenn man sich mit der Y-Koordinate zufrieden geben kann, wie bei eindimensionalen Menüs, genügt sogar der normale Texthintergrund.

Die Farbsättigungs-, Kontrast- und Helligkeitseinstellung des Monitors spielt dabei natürlich auch eine Rolle, so daß sie gegebenenfalls optimiert werden muß. Bei Monitoren mit Filterscheiben sollten diese abgenommen werden, wenn sie zu weit vom Bildschirm entfernt sind oder wenn sie das Bild zu stark verdunkeln (z.B. Rauchglas).

Für beste Ergebnisse sollte der Lightpen exakt senkrecht zum Bildschirm gehalten werden.

Die Bildschirmoberfläche sollte sauber gehalten werden, weil sich sonst Staub und Schmutz vor der Optik des Lightpens ansammeln. Auch diese sollte von Zeit zu Zeit gereinigt werden, um die Auflösung des Lightpens zu erhalten.

Nachdem das theoretische Rüstzeug vorhanden ist, folgt hier der erste praktische Versuch. Der Lightpen wird normalerweise an Joystickport 1 angeschlossen. Aus unerfindlichen Gründen wurde die Schaltung des ATARI 400 anders ausgelegt, so daß Joystickport 4 verwendet werden muß.

Das kleine BASIC-Programm tut nichts anderes als die Werte der Lightpenregister abzufragen und auszugeben:

```
10 REM LIGHTPEN ABFRAGE BEISPIEL 1
20 SETCOLOR 1,0,0:SETCOLOR 2,8,8
30 X=PEEK(564):Y=PEEK(565)
40 ? X,Y:GOTO 30
```

Wenn man jetzt mit dem Lightpen im Bild herumfährt stellt man fest, daß es so einfach doch nicht geht. Die Werte liegen nämlich nicht im gewünschten Bereich (siehe oben), denn die Zähler steuern den gesamten Bildaufbau und nicht nur das Textfenster. Der X-Wert springt sogar vor dem rechten Rand auf 0. Dies liegt an der Verzögerung der Signale im Computer und im Monitor. Im nächsten Beispiel wird dies durch Umrechnung kompensiert:

```
10 REM LIGHTPEN ABFRAGE BEISPIEL 2
20 SETCOLOR 1,0,0:SETCOLOR 2,8,8
30 LX=PEEK(564):LY=PEEK(565)
40 X=LX-76:IF X<0 THEN X=LX+151
50 Y=LY-16
70 IF X<0 OR X>159 OR Y<0 OR Y>95 THEN 30
80 ? X,Y:GOTO 30
```

Wenn jetzt noch total widersinnige Werte auftauchen sollten, ist das Bild zu dunkel oder zu kontrastarm. Abhilfe ist durch Verstellen der Regler am Monitor oder durch Änderung der Helligkeitswerte (Zeile 20) möglich. Wenn jetzt nur zwei Zeilen geändert werden, ist schon ein bißchen „Malen“ möglich:

```
20 GRAPHICS 23:SETCOLOR 4,8,8:COLOR 3
80 PLOT X,Y:GOTO 30
```

So recht geht es noch nicht, die Auflösung vor allem in X-Richtung ist noch nicht optimal. Der Grund liegt darin, daß jede Analogelektronik ein Signal verfälscht. Das fängt schon im Computer an, der Monitor trägt seinen Teil bei und der Verstärker im Lightpen ebenso wie die Entstörfilter am Lightpeneingang. Es gibt aber zum Glück eine Lösung. Wer genau hinsieht, kann erkennen, daß Abweichungen vom Soll praktisch nur in positiver Richtung (also nach rechts und nach unten) auftreten. Wenn man die Koordinaten mehrmals abfragt und dann die kleinsten Werte nimmt (Minimumbildung), läßt sich der Fehler in Grenzen halten.

BASIC wäre dazu aber doch ein wenig zu langsam. Mit einer kleinen Maschinensprach-Routine in Page 6 geht es bedeutend besser.

Die folgende universelle Lightpenroutine läßt sich an jedes BASIC-Programm anfügen und muß nur einmal mit `GOSUB 9080` initialisiert werden. Mit `GOSUB 9030` werden die Lightpenkoordinaten in die Variablen `X` und `Y` übernommen. Die Maschinensprach-Routine selbst ist verschiebbar und kann daher auch in einem String abgelegt werden.

Bei jedem Aufruf werden die Minima aus drei Registerabfragen gebildet. Bei Bedarf kann die „3“ auf Zeile 9110 entsprechend abgeändert werden. Es empfiehlt sich aber, den Wert nicht zu verkleinern, damit die geschriebene Software auch mit anderen Lightpens, wie z.B. den von ATARI funktioniert.

```

9000 REM
9010 REM LIGHTPEN-WERTE NACH X,Y
9020 REM
9030 A=USR(1536)
9040 Y=INT(A/256):X=A-Y*256:RETURN
9050 REM
9060 REM INIT LIGHTPEN-ROUTINE
9070 REM
9080 FOR A=1536 TO 1596:READ B
9090 POKE A,B:NEXT A:RETURN
9100 DATA 104,169,255,133,212,133,213
9110 DATA 162,3,165,20,197,20,240,252
9120 DATA 173,52,2,201,10,144,2,233
9130 DATA 228,24,105,154,197,212,176
9140 DATA 2,133,212,173,53,2,56,233
9150 DATA 16,197,213,176,2,133,213
9160 DATA 202,208,217,165,212,201,160
9170 DATA 176,203,165,213,201,96,176
9180 DATA 197,96

```

Alle folgenden Programmbeispiele verwenden diese Abfrageroutine und sind ohne sie nicht lauffähig! Aus Platzgründen wird die Routine nicht jedesmal mitausgelistet, also nicht vergessen, sie anzuhängen!

Damit sie nicht jedesmal neu eingegeben werden muß, kann sie mit `LIST` auf Cassette oder Disk gespeichert und mit `ENTER` an beliebige Programme angefügt werden.

Mit dem folgenden Malprogramm kann die Verbesserung direkt gesehen werden:

```

10 REM MALPROGRAMM 2
20 GRAPHICS 23:SETCOLOR 4,8,8:COLOR 3:GOSUB 9080
30 GOSUB 9030:PLOT X,Y:GOTO 30

```

Es stört nur, daß keine durchgehende Linie zu sehen ist, sondern nur einzelne Punkte. Das liegt nur zum kleinen Teil an der Langsamkeit von BASIC. Selbst bei einer Plotrate einem Punkt pro erzeugtem Bild wäre das Zeichnen einer Linie nur bei sehr langsamer Bewegung möglich.

Nur wenn man den `PLOT`-Befehl durch `DRAWTO` ersetzt, entsteht in jedem Fall eine durchgehende Linie, die natürlich umso runder wird, je langsamer eine Kurve gefahren wird.

Notwendig ist noch, eine Möglichkeit, die Linie zu unterbrechen, quasi den „Stift“ vom „Papier“ zu heben. Durch Abfrage des Tasters im Lightpen, der auf einem Joystickanschluß liegt, ist dies möglich. Bei Verwendung eines ATARI 400 muß `STICK(0)` auf `STICK(3)` geändert werden.

Leider halten sich nicht alle am Markt erhältlichen Lightpens an die „Norm“ und verwenden einen anderen Eingang. Es sollte daher nur abgefragt werden, ob `STICK(0)` gleich 15 ist (Taster losgelassen) oder nicht (Taster betätigt). Das folgende Beispiel zeigt dies:

```

10 REM MALPROGRAMM 3
20 GRAPHICS 23:SETCOLOR 4,8,8:GOSUB 9080
30 GOSUB 9030:COLOR 3
40 IF STICK(0)<>15 THEN DRAWTO X,Y:GOTO 30
50 LOCATE X,Y,A:COLOR A:PLOT X,Y:GOTO 30

```

Zeile 50 wird nur ausgeführt, wenn der Taster nicht gedrückt ist („Stift abgehoben“) und erscheint zunächst etwas widersinnig, da mit `LOCATE` die Farbe des Pixels abgefragt wird, so daß sich beim

anschließenden PLOT keine Änderung am Schirm ergibt. Notwendig ist's aber doch, weil sonst der nächste DRAWTO-Befehl nach Betätigung des Tasters wieder eine Verbindung zum letzten gesetzten Punkt herstellen würde.

Malprogramme sind damit zur Genüge besprochen. Am Schluß der Anleitung findet sich noch ein komfortableres Exemplar, das mit einem Menüfeld eine Farb- und Funktionsauswahl ermöglicht.

Um gleich bei den Menüs zu bleiben – hier zeigt sich die Überlegenheit des Lightpens besonders deutlich.

Normalerweise müssen Menüs über die Tastatur oder durch Bewegung eines Cursors mit dem Joystick bedient werden, was verschiedene Nachteile hat. Der Lightpen jedoch ermöglicht eine direkte, schnelle Auswahl durch einfaches „Deuten“ auf die gewünschte „Aktion“, die vorzugsweise durch Graphik versinnbildlicht wird.

Hier soll es nur um die mit dem Lightpen zusammenhängende Programmiertechnik gehen, daher die Beschränkung auf das Wesentliche:

```

10 REM MENUE-BEISPIEL
20 GRAPHICS 0:GOSUB 9080:?
30 ? "*"      AKTION 1":?
40 ? "*"      AKTION 2":?
50 ? "*"      AKTION 3":?
60 ? "*"      AKTION 4":?
70 IF STICK(0)<>15 THEN 70
80 IF STICK(0)=15 THEN 80
90 GOSUB 9030:IF X<8 OR X>15 OR Y<3 OR Y>30 THEN 70
100 Y=INT(Y/14)+1:A=INT(Y/2):IF A*2<>Y THEN 70
110 ON A GOSUB 120,130,140,150:GOTO 70
120 ? "UPRO 1",:RETURN
130'? "UPRO 2",:RETURN
140 ? "UPRO 3",:RETURN
150 ? "UPRO 4",:RETURN

```

Bei diesem Beispiel kann mit dem Lightpen eines von vier Unterprogrammen aufgerufen werden.

Die empfindlichen Auswahlfelder sind im Listing Sternchen, in der Praxis nimmt man aber besser inverse Leerzeichen, um die Sicherheit bei ungenauer Lightpenpositionierung zu erhöhen.

Auf Zeile 50 wird zunächst gewartet, bis der Taster im Lightpen losgelassen wird, in der nächsten Zeile auf eine erneute Betätigung. Ist diese erfolgt, werden die Koordinaten des Lightpens abgefragt und geprüft, ob sie im Bereich der Auswahlfelder liegen. Grob abweichende Koordinaten werden ignoriert. Da eine so grobe Abweichung nur willentlich hervorgerufen werden kann, wird auf eine neue Tasterbetätigung gewartet.

Zur Weiterverarbeitung genügt hier die Y-Koordinate. Aus ihr wird auf Zeile 100 die Nummer der Textzeile berechnet. Ungerade Zeilen werden ignoriert. Der Sprung auf Zeile 110 wird dadurch nur ausgeführt, wenn der Lightpen wenigstens in etwa auf ein Auswahlfeld zeigt.

Schon dieses einfache Beispiel kann man sinnvoll verwenden, wenn Programmnamen ausgegeben werden und in den Zeilen 120–150 RUN "D:NAME" eingesetzt wird. Das ergibt ein einfaches, lightpengesteuertes Programm-Menü.

Das nächste Beispiel, ein kleiner Zeicheneditor, treibt diese Technik noch weiter. Mit dem Lightpen kann damit ein Zeichen neu definiert werden, gleichzeitig sieht man an einer anderen Stelle des Bildschirms das Zeichen in natürlicher Größe.

Jeder, der schon einmal mühevoll auf Millimeterpapier Zeichen entworfen hat, die dann in der Praxis doch „anders“ aussahen, wird das zu schätzen wissen.

```

10 REM ZEICHENEDITOR
20 REM
30 GRAPHICS 0:GOSUB 9080:CH=12288:SC=57344
40 FOR I=0 TO 1023:POKE CH+I,PEEK(SC+I):NEXT I
50 FOR I=8 TO 15:POKE CH+I,0:NEXT I
60 POKE 756,48
70 DL=PEEK(566)+PEEK(561)*256:VR=PEEK(DL+4)+PEEK(DL+5)*256

```

```

80 ? : ? " ! " : ? : FOR I=1 TO 8 : ? " * * * * * * * * " : ? : NEXT I
90 IF STICK(0) <> 15 THEN 90
100 IF STICK(0)=15 THEN 100
110 GOSUB 9030 : X=INT(X/18)-6 : Y=INT(Y/8)-1
120 IF X<0 OR X>7 OR Y<0 OR Y>7 THEN 90
130 AD=VR+132+X*2+Y*80 : CA=CH+8+Y : CW=PEEK(CA) : CM=INT(2^(7-X)+0.5)
140 IF PEEK(AD)=10 THEN POKE AD,128 : POKE CA,CW+CM : GOTO 90
150 POKE AD,10 : POKE CA,CW-CM : GOTO 90

```

Im Prinzip handelt es sich um ein zweidimensionales Menü, nur werden hier X und Y sofort auf die Auswahlmatrix umgerechnet und erst dann auf erlaubte Werte geprüft (Zeile 110 und 120). Dann wird die Adresse des Auswahlzeichens im Videoram AD, die Adresse des umzuändernden Bytes im Zeichengenerator CA, der Inhalt dieser Adresse CW und der dezimale Wert des angewählten Bits CM berechnet. Die Verzweigung auf Zeile 140 ist für die Unterscheidung zwischen Punktsetzen und -löschen zuständig. Die Bitmanipulation im Zeichengenerator wird durch Addition oder Subtraktion des Bitwerts durchgeführt.

Damit dies überhaupt möglich ist, wurde der Zeichengenerator vorher vom OS ROM in den RAM kopiert (Zeile 40) und das zu ändernde Zeichen "!" auf Null gesetzt, weil in der Auswahlmatrix zunächst nur Sternchen stehen (alle Punkte gelöscht); andernfalls würde das Setz- und Löschverfahren durcheinander geraten.

Man beachte auch die Berechnung der Adresse der Displayliste DL und des VideoRAMs VR auf Zeile 70. Sie variieren nämlich je nach RAM-Kapazität des Systems.

Der neue Zeichengenerator wurde einfach ein gutes Stück „hinter“ dem BASIC-Programm in den RAM gelegt, ohne ihn besonders abzusichern. Das geht nur solange gut, bis das BASIC-Programm wesentlich erweitert wird! Also Vorsicht!

Damit ist dieser kurze Einblick in einige wenige der vielen Möglichkeiten des Lightpens und die notwendigen Programmieretechniken beendet.

Jeder durchschnittliche BASIC-Programmierer sollte jetzt in der Lage sein, den vielseitigen Lightpen in eigenen Programmen zu nutzen, was mit keinem anderen Eingabegerät mit so wenig Aufwand und Mühe möglich ist.

Für den Assembler-Programmierer bieten sich natürlich noch viel mehr Möglichkeiten. Es sei hier als Beispiel nur kurz angerissen, wie man ein Malprogramm mit beliebigen Farben, besonders sehr dunklen, realisieren könnte. Man könnte nämlich während der Lightpenabfrage, also wenn die Taste gedrückt ist, erst einmal alle Farbreister auf eine Minimalhelligkeit, z. B. 4 aufwärts, bringen. Selbst wenn das nur für einen Bildaufbau erfolgt, kann die Y-Koordinate schon sicher bestimmt werden. Beim nächsten Bild wird mittels Display List Interrupt ein „Streifen“ um diese Y-Koordinate hellgetastet, um die X-Koordinate sicher zu bestimmen. Zur Sicherheit kann dies noch einmal wiederholt werden, dann werden die Farbreister wieder auf die, ursprünglichen Werte zurückgesetzt.

Mit dieser Methode kann der Lightpen auf einem völlig dunklen Hintergrund arbeiten. Ein gewisses Flackern muß man dabei zwar in Kauf nehmen, aber bei wesentlich teureren professionellen Systemen blitzt und flackert es in diesem Fall auch.

Für alle, die nicht selbst programmieren können oder wollen, zum Abschluß noch ein kleines Programm zum Abtippen.

Viel Spaß!

```

10 REM MALPROGRAMM 4
20 DIM A$(158)
30 GRAPHICS 23:FOR I=0 TO 4:READ A,B:SETCOLOR I,A,B:NEXT I
40 DL=PEEK(560)+PEEK(561)*256:VR=PEEK(DL+4)+PEEK(DL+5)*256
50 ML=ADR(A$):TB=(INT(ML/256)+2)*256
60 COLOR 3:PLOT 0,0:DRAWTO 159,0:DRAWTO 159,95:DRAWTO 0,95:DRAWTO 0,0
70 PLOT 145,0:DRAWTO 145,95:PLOT 148,0:DRAWTO 148,88
80 FOR Y=0 TO 95 STEP 8:PLOT 145,Y:DRAWTO 159,Y:NEXT Y
90 FOR I=1 TO 3:COLOR I:FOR Y=I*8+2 TO I*8+6
100 PLOT 150,Y:DRAWTO 157,Y:NEXT Y:NEXT I:COLOR 3
110 READ X,Y:IF X<0 THEN PLOT ABS(X),Y:GOTO 110
120 IF Y>0 THEN DRAWTO X,Y:GOTO 110
130 IF LEN(A$)<>0 THEN 170
140 S=0:FOR A=1597 TO 1743:READ B:S=S+B:POKE A,B:NEXT A
150 FOR A=1 TO 158:READ B:S=S+B:A$(A,A)=CHR$(B):NEXT A:GOSUB 9080
160 IF S<>42692 THEN GRAPHICS 0:PRINT "DATA FEHLER":END
170 C=0:M=0:Y=1:GOSUB 300:Y=6:GOSUB 300
180 IF STICK(0)<>15 THEN 180
190 IF STICK(0)=15 THEN K=1:GOTO 190
200 GOSUB 9030:IF X>144 THEN 290
210 ON M=4 GOSUB 220,230,250,270:GOTO 190
220 PLOT X,Y:K=0:K1=0:RETURN
230 IF K=0 THEN DRAWTO X,Y:RETURN
240 GOTO 220
250 IF K1=0 THEN DRAWTO X,Y:RETURN
260 GOTO 220
270 LOCATE X,Y,A:IF A<>C THEN A=USR(ML,VR,TB,Y*256+X,A*256+C)
280 RETURN
290 Y=INT(Y/8):GOSUB 300:GOTO 180
300 COLOR 0:IF Y<4 THEN A=C:GOSUB 350:C=Y:A=C:GOTO 340
310 IF Y=4 OR Y=9 OR Y>10 THEN A=M:GOTO 340
320 IF Y=10 THEN RESTORE :GOTO 30
330 A=M:GOSUB 350:M=Y:A=M
340 COLOR 2
350 A=A*8+1:PLOT 146,A:DRAWTO 146,A+6:PLOT 147,A:DRAWTO 147,A+6
360 COLOR C:K1=1:RETURN
370 DATA 4,14,12,14,9,14,0,0,8,8
380 DATA -150,2,157,2,157,6,150,6,150,2
390 DATA -154,42,152,42,152,46,-153,44
400 DATA 154,44,-155,43,-154,50,152,50
410 DATA 152,54,154,54,-155,53,155,51
420 DATA -152,58,152,62,155,62,-155,66
430 DATA 152,66,152,70,-153,68,154,68
440 DATA -155,82,152,82,152,86,155,86,0,0
450 DATA 165,232,24,101,229,133,229
460 DATA 165,233,101,230,133,230,96
470 DATA 181,0,24,105,2,149,0,144
480 DATA 2,246,1,96,197,222,208,2
490 DATA 228,223,96,0,1,0,255,1,0
500 DATA 255,0,192,12,48,3,0,85,170
510 DATA 255,165,220,166,221,133,229
520 DATA 134,230,136,132,233,132,231
530 DATA 165,216,74,38,231,74,38,231
540 DATA 133,232,32,61,6,165,217,133
550 DATA 232,162,5,6,232,38,233,224
560 DATA 3,208,3,32,61,6,202,208,242
570 DATA 32,61,6,177,229,166,215,93
580 DATA 106,6,166,231,61,102,6,208
590 DATA 34,189,102,6,72,166,214,61
600 DATA 106,6,133,232,104,73,255
610 DATA 49,229,5,232,145,229,165
620 DATA 216,145,226,165,217,200,145

```

```
630 DATA 226,162,226,32,75,6,96
640 DATA 104,162,7,104,149,214,202
650 DATA 16,250,166,218,232,232,134
660 DATA 222,165,219,133,223,165,216
670 DATA 160,0,145,218,165,217,200
680 DATA 145,218,165,218,166,219,133
690 DATA 224,134,225,165,222,166,223
700 DATA 133,226,134,227,169,3,133
710 DATA 228,160,0,177,224,166,228
720 DATA 24,125,94,6,201,145,176,18
730 DATA 133,216,200,177,224,24,125
740 DATA 98,6,201,96,176,5,133,217
750 DATA 32,110,6,198,228,16,220,162
760 DATA 224,32,75,6,165,224,166,225
770 DATA 32,87,6,208,202,165,226,166
780 DATA 227,32,87,6,208,1,96,165,218
790 DATA 166,219,133,224,134,225,160
800 DATA 0,177,222,145,224,200,177
810 DATA 222,145,224,162,222,32,75
820 DATA 6,162,224,32,75,6,165,226
830 DATA 166,227,32,87,6,208,226,165
840 DATA 224,166,225,133,222,134,223
850 DATA 208,128
```

Alle Programmlistings sind auf der zugehörigen Diskette (LIGHTPEN.ATR) vorhanden. Zusätzlich noch zwei ebenfalls von Bernhard Engl stammende Assemblerlistings

- „Assembler Lightpen Driver“ und
- „Diamond Pattern Fill Routine“, ein LightPen Graphics Demo.