

Hardcopy leicht gemacht

Hardcopy-Programme konnten wir Ihnen schon viele vorstellen. Allerdings meist in Form von undurchsichtigen Maschinenroutinen. Hier zeigen wir Ihnen, wie man ein Hardcopy-Programm in Basic schreibt.

Eine Hardcopy ist die Kopie dessen, was auf dem Bildschirm zu sehen ist, auf Papier. Dazu wird Zeichen für Zeichen aus dem Bildschirmspeicher geholt und an den Drucker gesandt, sofern ein Textbildschirm kopiert werden soll. Der C 64 kennt zwei verschiedene Arten der Bildschirmdarstellung, die nicht ohne weiteres miteinander gemischt werden können: Text und hochauflösende Grafik.

Im folgenden soll die Hardcopy des Textbildschirmes näher betrachtet werden. Dazu soll ein einfaches Basic-Listing (Listing 1, Bild 1) dienen, das auf jedem Drucker den momentan am Bildschirm stehenden Text ausdruckt. Vielleicht kennen Sie das Listing in ähnlicher Form aus dem Handbuch zum Commodore MPS801. Es ist nicht ganz perfekt, aber einfach zu verstehen. Damit das Programm auch mit anderen Druckern arbeitet, müssen Sie die jeweils angegebenen Änderungen daran vornehmen.

Doch vor der Praxis etwas Theorie über die Organisation des Textbildschirmes des C 64. Der C 64 kann höchstens 1000 Zeichen am Bildschirm darstellen. Und zwar in 25 Zeilen zu je 40 Spalten. Für die 1000 Zeichen besitzt der C 64 einen reservierten Schreib-/Lese-Speicherbereich, den Bildschirmspeicher. Der Bildschirmspeicher liegt zwischen den Speicheradressen 1024 und 2023. Probieren Sie mal POKE 1024,1 aus: Ein »A« erscheint links oben am Bildschirm. POKE 1025,2 läßt ein »B« um eine Stelle nach rechts versetzt erscheinen. Mit dem Befehl PRINT PEEK(1024) erhalten

Sie als Ergebnis »1«. »1« ist der Bildschirmcode des Zeichens »A«. Wenn Sie 129 anstelle von 1 in die Speicherzelle 1024 POKEn, erscheint ein reverses »A« an der ersten Bildschirmstelle. Mit den Codes 0 bis 127 können Sie alle normalen Zeichen des C 64 darstellen, mit den Codes 128 bis 255 die reversen. Diese Codes werden als »Bildschirmcodes« bezeichnet. Das folgende kleine Programm erzeugt den gesamten C 64-Zeichensatz am Bildschirm.

```
10 print chr$(147)
20 for i=0 to 255
30 poke 1024+i,i
40 next i
```

Die Bildschirmcodes dürfen nicht mit den ASCII-Codes verwechselt werden. Die ASCII-Codes gelten nämlich nur für den CHR\$(x)-Befehl, mit dem Sie Zeichen über die PRINT-Anweisung drucken können. So kann das gleiche »A«, das Sie mit POKE 1024,1 auf den Bildschirm gebracht haben, auch durch PRINT CHR\$(64) gedruckt werden. Nicht aber mit CHR\$(1). Vergleichen Sie dazu einmal die Bildschirm- und ASCII-Tabelle im C 64-Handbuch. Sie werden feststellen, daß nicht nur die Codes verschieden sind, sondern auch, daß die ASCII-Tabelle keine Reverszeichen kennt.

Prinzip der Hardcopy

Eine Hardcopy macht im Prinzip nun nichts weiter, als ab Adresse 1024 nacheinander alle Zeichen aus dem Bildschirmspeicher zu lesen

und in der gleichen Reihenfolge an den Drucker zu schicken. Da aber der Drucker keine Bildschirmcodes versteht, müssen Sie ins ASCII-Format übersetzt werden. Die Vorschrift für eine Bildschirmcode-ASCII-Wandlung:

Bildschirmcode	ASCII-Korrekturfaktor
0 ... 31	+64
32 ... 63	±0
64 ... 95	+32
96 ... 127	+64
128...255	1.-128

2. danach nochmalige Wandlung und Zeichen revers drucken

Um Bildschirmcode in ASCII-Code zu wandeln, müssen Sie die angegebenen Korrekturfaktoren zu den Bildschirmcodes addieren. Bildschirmcodes über 127 müssen erst um 128 verringert und dann ins ASCII-Format übersetzt werden. Aus den Bildschirmcodes der reversen Zeichen werden nämlich erst die Codes der entsprechenden »normalen« Zeichen gemacht. Die reversen Zeichen wurden normal gedruckt. Aber mit einem kleinen Trick schafft man auch die reverse Darstellung. Vorausgesetzt man hat einen Commodore MPS801 oder einen anderen Drucker, bei dem CHR\$(18) und CHR\$(146) den Reversdruck ein- und ausschalten. Ein reverses »A«, zum Beispiel, wird bei diesen Druckern über die Kommandofolge: PRINT CHR\$(18)+ "A" + CHR\$(146) ausgegeben.

Das ganze Problem der Bildschirmcode-/ASCII-Wandlung kann nur mit Maschinensprache vollends gelöst oder besser umgangen werden, indem man den Textbildschirm

```

60000 REM TEXTBILDSCHIRM-HARDCOPY          <230>
60010 VIC=13*4096:MO=VIC+24                <044>
60020 REM ANFUERUNGSZ. DEF.                <089>
60030 QU$=CHR$(34): REM ANFUERUNGSZ.      <146>
60040 REM REVERS ON,REVERS OFF             <031>
60050 RV$=CHR$(18):RO$=CHR$(146)          <137>
60060 BA=PEEK(648)*256                      <047>
60070 IF (PEEK(MO)AND 6)=6 THEN 60100      <134>
60080 OPEN 4,4:PRINT#4: REM GROSS/GRAFIK   <224>
60090 GOTO 60110                             <243>
60100 OPEN 4,4,7:PRINT#4:REM GROSS/KLEIN   <193>
60110 REM HARDCOPY                           <135>
60120 FOR Z=0 TO 24 : REM ZEILEN 1-25      <042>
60130 QF=0 : REM NOCH KEIN >>"<<        <170>
60140 A$="" : REM A$ LOESCHEN               <026>
60150 FOR S=0 TO 39 : REM SPALTEN 1-40     <164>
60160 REM BILDSCHIRMCODE HOLEN             <074>
60170 BC=PEEK(BA+40*Z+S)                   <042>
60180 REM 1. ODER 2. ANFUERUNGSZEICHEN    <018>
60190 IF BC=34 THEN QF=1-QF                <155>
60200 REM REVERSES ANFUERUNGSZEICHEN
      DURCH REVERSES >'< ERSETZEN         <195>
60210 IF BC=162 THEN BC=167                <087>
60220 REM BILDSCHIRMCODE/ASCII-WANDLUNG    <084>
60230 REM FUER COMMODORE MPS 801          <222>
60240 IF QF=1 AND (SC>=128) THEN BA=BA-128:
      GOTO 60260                             <159>
60250 IF BC>127 THEN BC=BC-128:RF=1:A$=A$+
      RV$:REM REVERSMODUS EIN              <186>
60260 IF BC<32 OR BC>95 THEN AS=BC+64:
      GOTO 60290                             <238>
60270 IF BC>31 AND BC<64 THEN AS=BC:
      GOTO 60290                             <090>
60280 IF BC>63 AND BC<96 THEN AS=BC+32    <085>
60290 A$=A$+CHR$(AS)                       <120>
60300 IF RF=1 THEN A$=A$+RO$:RF=0:
      REM REVERSMODUS ENDE                 <206>
60310 NEXT S                                <017>
60320 IF QF=0 THEN PRINT#4,A$:GOTO 60340   <111>
60330 PRINT#4,A$QU$                         <188>
60340 NEXT Z                                <103>
60350 CLOSE 4                               <184>
    
```

Listing 1. Basic-Listing des Hardcopy-Programms

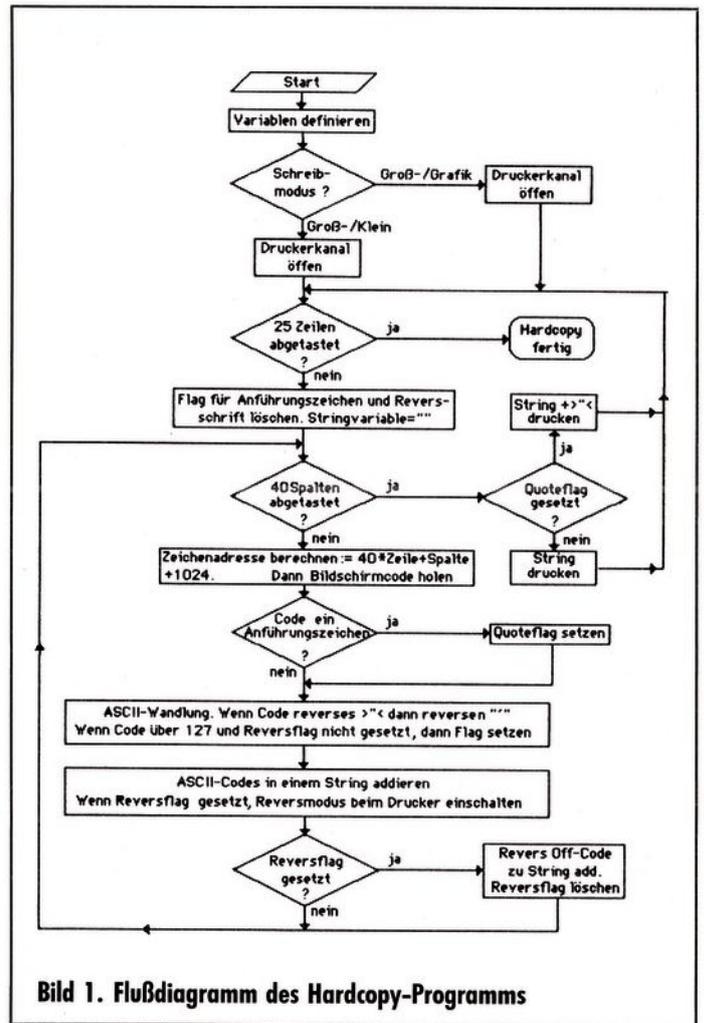


Bild 1. Flußdiagramm des Hardcopy-Programms

wie einen Grafikbildschirm behandelt. Wie das geht, erfahren Sie in der nächsten Ausgabe, in der wir Ihnen ein Maschinenprogramm vorstellen, das auf Tastendruck eine Hardcopy vom Textbildschirm druckt. Daß man nicht auf Maschinensprache verzichten kann, liegt einfach daran, daß Basic-Befehle nur auf festgelegte Speicherebenen zugreifen können.

Doch nun zurück zu der kleinen Basic-Routine, die eine Hardcopy eines Textbildschirmes druckt. In Zeile 60010 wird die Adresse des VIC-Registers 24 berechnet, das für die Schriftart verantwortlich ist: Groß-/Grafik- oder der Groß-/Kleinmodus. Die Adresse des Registers wird an die Variable MO (Modus) übergeben um mit dem PEEK-Befehl das Register auszulesen. Der Registerinhalt entscheidet in Zeile 60070 darüber, welche Sekundäradresse für den Drucker ausgewählt wird. Entweder keine Sekundäradresse für Groß-/Grafik (Zeile 60080) oder »7« für Klein/Groß (60100). Ist Bit 2 und 3 von MO gesetzt, dann liegt Groß-/Kleinmodus vor.

In den Zeilen 60050 und 60060 werden einige bestimmte Steuerco-

des für den Drucker Variablen zugeordnet. Im Einzelnen:
 QU\$=CHR\$(34): ASCII-Codes eines Anführungszeichen
 RV\$=CHR\$(18): reverse Schrift einschalten
 RO\$=CHR\$(146): reverse Schrift ausschalten

Wie Sie sehen, sind die genannten SteuerCodes für den Commodore Drucker MPS 801 mit denen des C 64 identisch. Haben Sie keinen MPS 801-kompatiblen Drucker, entnehmen Sie bitte die SteuerCodes Ihres Druckers dem Drucker- oder Interface-Handbuch. Sollte Ihr Drucker keine Reverszeichen drucken können, bleibt Ihnen nichts weiter übrig, als reverse Zeichen in normaler Darstellung ausgeben zu lassen. Definieren Sie dazu RO\$ und RV\$ als CHR\$(0) oder als einen anderen ASCII-Code, den Ihr Drucker ignoriert. Auf diese Weise ersparen Sie sich das Durchsuchen des Listings nach SteuerCodes und deren Entfernung.

In Zeile 60050 stellt das Programm die Bildschirm-Anfangsadresse fest, die nicht grundsätzlich immer bei 1024 liegen muß. Das High-Byte dieser Adresse steht in der Spei-

cherzelle 648, es muß mit 256 multipliziert werden. Ein Low-Byte existiert nicht, da der Bildschirm nur um ganze KByte verschoben sein kann. In Zeile 60080 wird schließlich der Druckerkanal für Groß-/Grafikmodus geöffnet, entsprechend in Zeile 60100 für Groß-/Kleinmodus. Diese Sekundäradressen müssen Sie, so wie die SteuerCodes, auch auf Ihren Drucker anpassen.

Hardcopy in Basic

Ab Zeile 60110 beginnt schließlich die eigentliche Hardcopy. Die ganze Routine besteht im Prinzip aus zwei FOR-NEXT-Schleifen, die jede Zeile spaltenweise »abtasten«. Die erste Schleife (60120...60340) erhöht die Zeilenzahl (Variable Z), die zweite (60150...60310) die Spaltenzahl (Variable S). Vor Eintritt in die Spalten-schleife wird die später zu druckende Variable A\$ gelöscht. Denn wenn nichts in einer Zeile steht, soll auch nichts gedruckt werden. Ebenso wird QF, die Flag-Variable für den Quote-Modus, zurückgesetzt.

Quote-Modus bedeutet, daß alle folgenden Zeichen innerhalb von

Anführungszeichen stehen. Diese Flagge wird auf »1« gesetzt, sobald ein Anführungszeichen gedruckt werden soll. Nach dem zweiten Anführungszeichen wird QF wieder auf »0« gesetzt. Innerhalb von zwei Anführungszeichen bedeutet beispielsweise CHR\$(18) ein reverses »R«, sonst aber, daß alle folgenden Zeichen revers gedruckt werden. Drücken Sie mal »Cursor rechts«. Der Cursor bewegt sich nach rechts über den Bildschirm. Drücken Sie dann die »"«-Taste und dann »Cursor rechts«. Es erscheinen lauter reverse eckige Klammern, die Steuerzeichen für Cursor rechts. Sie befinden sich im Quote-Modus. Es ist sehr wichtig zwischen dem normalen und dem Quote-Modus zu unterscheiden. Auf den Code CHR\$(18) hin druckt auch der Drucker alle folgenden Zeichen revers, während die Sequenz Anführungszeichen, CHR\$(18) nur ein reverses »R« erzeugt, genauso wie am Bildschirm.

In Zeile 60170 werden die Zeichencodes aus dem Bildschirm-speicher mit PEEK geholt. Die Zeichenadresse berechnet sich aus der Startadresse BA+40*Zeile+Spalte. Danach wird als erstes untersucht, ob der gelesene Code ein An-

führungszeichen ist. Wenn ja, wird, wie schon erwähnt, das Quoteflag QF gesetzt. Ist BC der Code für ein reverses Anführungszeichen (Bildschirmcode 162), wird BC der Code für einen reversen Apostroph zugewiesen. Diese »Umgehung« ist nötig, weil ein reverses Anführungszeichen nicht ohne weiteres gedruckt werden kann.

Bildschirmcode-/ASCII-Wandlung mit Tricks

Die Bildschirmcodes werden ab Zeile 60220 in ASCII-Format gewandelt. Eine besondere Erklärung bedarf die Zeile 60250. Da alle Bildschirmcodes größer als 127 die reversen Darstellungen der Codes 0 bis 127 sind, werden diese Codes um 128 verringert und das Reversflag RF gesetzt. Ist RF=1, dann wird vor dem eigentlichen Zeichen ein CHR\$(18) (Revers ein) an den Drucker geschickt und im Anschluß an das Zeichen ein CHR\$(146) (Revers aus). In Zeile 60290 werden schließlich die ASCII-Codes einer ganzen Bildschirmzeile zusammen-

gesetzt, komplett mit allen Druckersteuerzeichen. Wurden alle Spalten einer Zeile »abgetastet« und die ASCII-Codes in A\$ addiert, wird A\$ an den Drucker geschickt. Der Druck einer Zeile erfolgt schließlich in 60320 oder 60330 (wenn QF gesetzt ist, also im Quotemodus gedruckt wurde). Am Ende der Hardcopy wird in Zeile 60330 der Druckerkanal geschlossen.

Diese Basic-Hardcopy-Routine können Sie eigenen Programmen anhängen und mit GOSUB 60000 aufrufen. Vergessen Sie dann aber nicht in Zeile 60340 ein RETURN einzugeben.

In den nächsten Ausgaben des 64'er Magazins setzen wir diesen Artikel fort. Da erfahren Sie, wie man in Maschinensprache eine Textbildschirm-Hardcopy-Routine programmiert. Diese Routine druckt auf Tastendruck hin eine Text-Hardcopy. Mit allen Zeichen die der C 64 darstellen kann.

Wir zeigen Ihnen dann auch, wie man eine Hardcopy des Grafikbildschirms zu Papier bringt.
(hm)

Vergleichstest: Grafikerweiterungen

Grafikprogrammierung auf einem Computer ist eine sehr reizvolle Aufgabe. Gut gemachte Grafiken können nicht nur den Programmierer, sondern auch ein Publikum, das wenig mit Computer zu schaffen hat, faszinieren.

Der Commodore 64 ist von seiner Hardware her gesehen ein äußerst leistungsfähiger Computer. Leider werden die außerordentlichen Grafik- und Soundfähigkeiten vom eingebauten Basic V2 in keiner Weise unterstützt. Will man zum Beispiel eine einfache Grafik programmieren, so steht einem eine ermüdende Programmierarbeit mit einer Unmenge von POKEs bevor. Um diesem Mißstand abzuhelpen,

werden von verschiedenen Softwarefirmen Befehlserweiterungen angeboten, die eine bequeme Programmierung von Grafiken, Sprites und Sounds ermöglichen. Die Befehlserweiterungen der verschiedenen Hersteller sind in ihrem Umfang recht unterschiedlich. So gibt es Erweiterungen, die sich nur auf Grafik-, Sprite- und Soundbefehle beschränken, andere wiederum bieten eine ganze Palette von nützli-

chen Zusatzbefehlen wie Tools (zum Beispiel Find, Auto, Trace etc.), Befehle zur Unterstützung der strukturierten Programmierung, Befehle zum leichteren Arbeiten mit der Floppy, etc. Wir haben sieben Erweiterungen unter die Lupe genommen und werden im folgenden näher darauf eingehen, wobei von dem jeweiligen Befehlsvorrat nur die grafikspezifischen Befehle näher betrachtet werden (Tabelle 1).