

Man kann aber auch einfach auf die Lautstärkeinstellung verzichten. Die Parameter für Frequenz, Dauer und Lautstärke entsprechen den in BASICODE vereinbarten.

Für die Tonhöhe sind die Argumente von 0 bis 127 zulässig, aber nur die Werte von ca. 20 bis 101 erscheinen sinnvoll. Unterhalb von 20 läßt die Tonqualität doch zu wünschen übrig, da ja nur Rechteck- und keine Sinusschwingungen erzeugt werden. Oberhalb 101 besteht in zunehmendem Maße nicht mehr für alle Argumente ein akustischer Unterschied in der Tonhöhe, da hier der gewählten Form der Tonerzeugungssoftware Grenzen durch die Taktfrequenz gesetzt sind. Diese Extremwerte dürften aber auch für Musikausgaben keine Bedeutung haben.

Beispiel:

```
50 BEEP 69,10,7
```

```
60 REM erzeugt 1 s den Kammerton a (440 Hz) in mittlerer Lautstärke
```

### Einige wichtige Adressen von BACOBAS

3000H	Kaltstart
3003H	Warmstart
3006H	E/A-Adresse zur Lautstärkeeinstellung
3007H	Sprung zur eigenen Druckerausgaberroutine; die Zeichenübergabe erfolgt im Akku, weitere Register dürfen nicht zerstört werden .
300AH	Sprung zur Initialisierung des eigenen Druckertreibers; es darf kein Register zerstört werden. Dieses Programm wird bei jedem LLIST und nach RUN vor den ersten LPRINT aufgerufen. Ist kein eigener Druckertreiber vorhanden, so brauchen diese beiden Sprungbefehle nicht verändert zu werden, sie rufen dann nur die Routine MS 30 des Monitorprogramms auf.

## 5.8. Bascoder für ATARI 800XL/XE 600XL 130XE

### Allgemeines

Da bislang für die ATARI-Rechner kein Bascoder zur Verfügung stand, waren bei der Entwicklung zwei Aspekte wesentlich:

Erstens sollte das Programm schnell entwickelt werden, um möglichst bald den ATARI-Nutzern die Mitarbeit bei BASICODE zu ermöglichen und den Software- und Datenbestand von BASICODE zugänglich zu machen.

Zweitens sollte die verwendete Hardware so einfach wie möglich und die Änderungen an Geräten so gering wie möglich sein.

Unter diesen Prioritäten hat an einigen Stellen die "Schönheit" des Programms etwas gelitten. An der Zuverlässigkeit und Portabilität wurden jedoch keine Abstriche gemacht. Auf zwei Variablen der Hardware-Änderungen wird am Ende eingegangen.

### Für ganz Ungeduldige

Wenn die Datasette wie hinten beschrieben umgebaut ist, kann man bereits mit BASICODE arbeiten. Dazu wird

die Kassette mit dem Bascoder eingelegt,  
 der Recorder auf "A" geschaltet,  
 der Recorder gestartet,  
 die START- und die OPTION-Taste gedrückt und  
 der Rechner eingeschaltet.

Nachdem sich BASICODE gemeldet hat, wird der Recorder auf "B" geschaltet, der Befehl

```
LOAD "B"
```

einggegeben und bei Ertönen des BASICODE-Vorlautons eine Taste gedrückt. Der Bildschirm muß viermal dunkel werden, nach der darauffolgenden READY-Meldung ist BASICODE betriebsbereit.

Jetzt kann von der Kassette ein BASICODE-Programm mit

```
CL.
```

eingelassen und danach mit RUN gestartet werden.

Im Rechner stehende BASICODE-Programme lassen sich mit

```
CS.
```

auf Kassette schreiben.

### Software

Für den ATARI-Rechner stehen vor allem das ATARI-BASIC und das TURBO-BASIC XL zur Verfügung. Beide BASIC-Dialekte weichen in mehreren Punkten stark von den üblichen MICROSOFT-BASIC und allen daraus abgeleiteten Varianten ab. Das Grundprinzip dieser Interpreter liegt jedoch der Struktur von BASICODE zugrunde. Deshalb kamen die o. g. ATARI-BASIC nicht für BASICODE in Betracht. Nun gibt es zwar für den ATARI auch ein MICROSOFT-BASIC, aber dieses ist so groß, so schwerfällig und teilweise fehlerbehaftet, daß sich das BASICODE-Testprogramm (17K) nicht mal mehr laden, geschweige denn abarbeiten ließ. Deshalb wurde für den ATARI ein bei anderen Rechnern unüblicher Weg beschritten.

Es wurde insgesamt für BASICODE ein spezieller BASIC-Interpreter geschrieben, der zugleich die Arbeitsweise von BASICODE (Lesen und Schreiben von BASICODE-Programmen, Bearbeiten von BASICODE-Datenbeständen usw.) als seinen normalen Standard ansieht.

Dieser Interpreter bearbeitet alle in BASICODE zugelassenen BASIC-Befehle und -Funktionen. Er kann aber auch als neuer BASIC-Interpreter selbständig benutzt werden. Dabei gibt es die folgenden Besonderheiten:

### AND; NOT; OR

Die Operationen werden bitweise ausgeführt.

### INPUT

Der Befehl darf nur im Programm benutzt werden, im Direktmodus ist die Zuweisung zu benutzen. INPUT läßt (erweitert gegenüber BASICODE-3) mehrere Variablen, die mit Komma getrennt sind, zu.

### RESTORE

Ebenfalls über BASICODE-3 hinaus, ist hinter RESTORE auch eine Zeilennummer zulässig. Weiterhin seien alle Funktionen und Kommandos vom neuen ATARI-Interpreter, die zusätzlich zu dem in diesem Buch beschriebenen BASICODE-3-Befehlsvorrat aufgenommen wurden oder abweichendes Verhalten zeigen, alphabetisch aufgeführt:

**C.** **C.(I)**  
Funktion zum Schließen einer Datei. Sie nimmt einen logischen Wert an, =0 falls o.k., < > 0 im Fehlerfall. I ist die logische Kanalnummer des Files (s. O.-Befehl).

**CL.** **CL.**  
Laden von BASICODE. Hier wird bei den folgenden Kommandos der Punkt Pflicht. Dieses Kommando führt folgende Schritte durch:

1. Das Programm ab Zeile 1000 wird gelöscht.
2. Die Datasette wird eingeschaltet. Bei Ertönen des BASICODE-Vorlauftons ist eine Taste zu drücken, danach wird das Programm eingelesen. Während dieser Zeit ist der Bildschirm abgeschaltet. Soll der Lesevorgang abgebrochen werden, ist START zu drücken.
3. Nach Beendigung des Lesens wird der Recorder vom Rechner angehalten, das Bild erscheint wieder. Jetzt setzt der Interpreter das eingelesene Programm in seine interne Darstellung um, dies kann bei langen Programmen schon einige -zig Sekunden dauern. Mit READY wird das Ende der Aktion angezeigt. Ein eventuell aufgetretener Prüfsummenfehler wird erst jetzt gemeldet. Dies gilt für den Normalfall, also sofern der Lesefehler nicht so folgenschwer war, daß die Grundstruktur des Programms zerstört wurde.

**CLR** **CLR**  
Alle numerischen Variablen werden auf Null und die Zeichenketten auf leer gestellt. Reservierter Speicher wird freigegeben.

**CONT** **CONT**  
Die Programmdurchführung wird fortgesetzt, sofern die Unterbrechung zuvor durch die BREAK-Taste oder eine STOP- bzw. END-Anweisung ausgelöst wurde. Sie wird an dem Punkt wieder aufgenommen, an dem die Unterbrechung erfolgt ist. Nach einem Fehler mit Halt kann ebenfalls die Bearbeitung fortgesetzt werden.

**CS.** **CS.**  
Das vorhandene Programm wird ab Zeile 1000 im BASICODE-Format auf Kassette gespeichert.

**D.** **D. I, J, K**  
Zeichnet im grafischen Betrieb eine Linie der Farbe K vom Punkt des letzten P.-Befehls oder dem Endpunkt des letzten D.-Befehls zum angegebenen Punkt I, J. Zur Bedeutung von I, J und K siehe den Befehl P.

**END** **END**  
Die Programmdurchführung wird beendet, alle Dateien werden geschlossen und zum Befehlsniveau zurückgekehrt.

**F** **F. (X)**  
Bringt die Anzahl von Bytes zurück, die nicht vom Programm genutzt werden. F. (X) verdichtet den Speicher mit der garbage collection, dies kann aber bis zu einer Minute dauern!

**G** **G.**  
Schaltet Grafikbetrieb ein. In ATARI-BASICODE wird nur ein Grafikmodus verwendet. Er entspricht Modus 24 in ATARI-BASIC. Es werden 192 Zeilen je 320 Bildpunkte dargestellt. Der Hintergrund ist weiß, der Vordergrund schwarz.



**INPUT** INPUT [ <"prompt"> ; ] <variablenliste >

Eingabe vom Terminal. Zur Kennzeichnung der Eingabe kann <"prompt"> angegeben werden, diese Kette wird als Bedienernachricht ausgegeben. Danach wird ein Fragezeichen ausgegeben. Die Eingabe wird erst beendet, wenn die <variablenliste > vollständig gefüllt ist. Ein vorheriger Abbruch oder Fehler (Typ, Länge usw.) wiederholen die Eingabeanforderung.

**LIST** LIST [ <zelle > [-] ]  
LIST [ <zeile1 > [- [ <zeile2 > ] ] ]

Angewendet im Befehlsniveau, listet das im Speicher befindliche Programm ab Zeile <zeile > oder <zeile1 > auf dem Display auf, bis zur <zeile2 >. Fehlende Angaben werden durch Ende des Programms bzw. Anfang des Programms ersetzt.

**LOAD** LOAD <name >

Lädt ein Programm mit dem Namen <name > in den Speicher. Die neuen Programmteile werden dabei in das sich im Speicher befindliche Programm eingefügt. Bei gleicher Zeilennummer wird die alte Zeile überschrieben. <name > muß eine Zeichenkettensvariable oder -konstante sein, die einer gültigen Filespezifikation entspricht. Möglich ist neben den im Betriebssystem üblichen Geräten (s. Handbuch) das Gerät "B:". Es entspricht der Kassette mit BASICODE-Dateiformat (nicht Programmformat!).

**LP.** LP. [ <liste der Ausdrücke > ] > ]

Druckt Daten auf dem Zeilendrucker, sonst wie PRINT.

**NEW** NEW

Das Programm im Speicher wird ab Zeile 1000 gelöscht, alle Variablen sind ungültig, alle Dateien geschlossen.

**O** O. <name > , <access > , <lun >

Eröffnen einer Datei. <name > ist eine Filespezifikation wie im LOAD-Befehl. <access > ist der Zugriffscode. Zulässig sind:

- 4 - lesen
- 6 - Directory lesen (nur für Diskette)
- 8 - neu anlegen und schreiben
- 9 - erweitern einer bestehenden Datei

<lun > ist eine logische Kanalnummer. Durch die Besonderheiten des BASICODE-Standards wird das niederwertige Bit ignoriert. 0 und 1, 2 und 3 usw. haben also gleiche Wirkung. Zulässig sind Zahlen von 0 bis 7, es können also maximal 4 Dateien gleichzeitig geöffnet sein. Unmittelbar nach Ausführung des O.-Befehls kann auf Zeile 32 der Status (= 0 - o.k., < > 0 - Fehler) und im Fehlerfall auf Zeile 35 der Fehlerstatus abgefragt werden.

**OSTR\$** OSTR\$ (X)

Bringt eine Kette zurück, welche dem Wert von X entspricht.

**P.** P. I, J, K

Schreibt im grafischen Betrieb einen Punkt in der Spalte I und der Zeile J mit der Farbe K. Links oben befindet sich der Punkt 0,0.  $0 < I < 319$ ,  $0 < J < 191$ ,  $K = 0$  entspricht weiß (Punkt löschen),  $K = 1$  entspricht schwarz (Punkt setzen).



**PEEK****PEEK (X)**

Bringt eine Dezimalzahl als Byte zurück, die auf dem Speicherplatz mit der Adresse X steht.

**POKE****POKE I, J**

Speichert ein Byte (dezimaler Wert des Ausdrucks J) auf die Speicherstelle mit der Adresse I.

**PRINT****PRINT [< ausdrucksliste >]**

Statt des PRINT kann auch ein Fragezeichen (?) verwendet werden.

**R.****R.(I)**

Funktion zum Lesen eines Satzes von der Datei. Die Funktion nimmt den Wert einer Zeichenkette an, die dem nächsten Satz entspricht. I ist die logische Kanalnummer (s. O.-Befehl). Unmittelbar nach der Funktion kann in Speicherzelle 32 der Status (=0 - o.k. < >0 - Fehler), in Zelle 35 der Fehlerstatus gelesen werden. Handelt es sich um eine BASICODE-Datei, ist der Status auf Zelle 6 zu finden. Diese Informationen gehen beim nächsten Ein-Ausgabe-Befehl, also auch bei PRINT oder LP., verloren. Die Zeile PRINT R.(I) ist also nicht sinnvoll, da danach kein Fehlercode mehr ausgewertet werden kann.

**RESTORE****RESTORE [< zeile >]**

Setzt den Zeiger, der die DATA-Anweisung liest, auf den Anfang zurück. < zeilenzahl > zeigt auf die DATA-Anweisung, fehlt diese Angabe, wird auf die erste DATA-Anweisung im Programm zurückgesetzt.

**RUN****RUN [< zeilenzahl >]**

Vom Befehlsniveau wird ein Programm ab der angegebenen Zeilenzahl bzw. ab Anfang abgearbeitet.

**S.****S. I, J, K**

Tonerzeugung. I ist die Dauer in Einheiten von 20 Millisekunden, J die Lautstärke von 0 bis 15. K ist die Tonhöhe von 8 bis 127 und berechnet sich folgendermaßen: 69 ist das mittlere A, also 440 Hz. Für jeden Halbton darüber wird 1 addiert, für jeden Halbton darunter 1 subtrahiert. Das mittlere C entspricht also der Zahl 60. Pitch-Werte unter 8 (sehr tiefer Ton) sind nicht mehr darstellbar und werden intern durch 8 ersetzt. Nach oben gibt es keine Beschränkung.

**SAVE****SAVE < name > [< zelle1 > - < zelle2 >]**

Es wird ein Programm vom Speicher auf Diskette oder Kassette unter dem angegebenen Namen abgespeichert. Die Angaben < zelle1 > und < zelle2 > haben dabei die gleiche Bedeutung wie beim LIST-Befehl. < name > wird wie beim LOAD-Befehl behandelt. Wichtige Anwendungen sind

SAVE "P:"

Drucken des Programms

SAVE "D:xxx.BC3",1000-

Abspeichern eines BASICODE-Programms auf Diskette

SAVE "D:ROOT.BC3",-999

Abspeichern des BASICODE-Anpassungsprogramms auf Diskette

SAVE "B",-999

Abspeichern des BASICODE-Anpassungsprogramms auf Kassette im BASICODE-Bestandsformat

(nicht Programmformat!)

**STOP****STOP**

Beendet ein Programm und führt zum Befehlsniveau zurück.

**T. T. I,J,X\$**

Schreibt den Text X\$ auf den grafischen Bildschirm. Dabei ist I die Adresse im Speicher. Wird angenommen, daß X und Y die Koordinaten des linken oberen Eckpunktes des Textbalkens sind, so berechnet sich I so:

$$I = \text{PEEK}(88) + 256 * \text{PEEK}(89) + 40 * Y + \text{INT}(X/8)$$

J gibt an, ob der Text schwarz oder weiß geschrieben wird. Ist J = 0, wird der Text schwarz, ist J = 255, dann weiß geschrieben. Grafikteile, die nicht von Buchstaben berührt werden, bleiben erhalten, auch wenn sie sich im Textbalken befinden.

**W. W.(X\$,I)**

Funktion zum Schreiben eines Satzes auf eine Datei. Sie nimmt einen logischen Wert zur Fehlererkennung an (= 0 - o.k., < > 0 - Fehler). X\$ ist der zu schreibende Satz, I die logische Kanalnummer (s. O.-Befehl).

**Aufruf von ATARI-BASICODE**

Vom Betriebssystem ATARI DOS-3 wird der BASIC-Interpreter unter dem Namen

BC.CMD

verwaltet. Unter diesem Namen wird das Programm geladen und der Befehlsmodus eingestellt. Damit kann der "Taschenrechnerbetrieb" eingestellt werden.

Bei Verwendung der Datasette sind die START- und OPTION-Tasten zu drücken, bei gedrückten Tasten ist der Rechner einzuschalten. Nach dem Piepton ist die Datasette einzuschalten und eine Taste zu drücken. Zuerst wird ein kurzes Ladeprogramm eingelesen, auf dem Bildschirm erscheint die Meldung:

LOADING BASICODE...

Danach wird der Interpreter gelesen. Dieses File ist identisch mit dem CMD-File der DOS-Version und mit 1200 Baud im TURBO-TAPE-Format aufgezeichnet. Tritt beim Laden ein Fehler auf, meldet sich das Test-Menü.

**Verlassen von BASICODE**

Die Befehlsfolge:

POKE 9,1 <CR>  
<RESET>

ist die einzige Möglichkeit, um in das Betriebssystem zurückzukehren. <RESET> allein kehrt nicht zum System zurück, sondern zum Befehlsniveau von ATARI-BASICODE.

**Eingabe-Editing**

Wenn ein falsches Zeichen getippt wurde, kann es mit DELETE gelöscht werden. Eine Zeile kann gelöscht werden, indem SHIFT/DELETE getastet wird. Wird eine Zeile korrigiert, die bereits im Speicher ist, so ist sie einfach unter der gleichen Zeilennummer nochmals richtig zu schreiben. Befindet sich die alte Zeile noch auf dem Bildschirm, kann sie durch die Pfeiltasten angesteuert und danach geändert werden. Bei Eingabe von RETURN wird jeweils die Zeile entgegengenommen, in der sich der Cursor gerade befindet. Die alte Zeile wird damit überschrieben. Soll ein Programm aus dem Speicher entfernt werden, so ist das Kommando NEW zu verwenden, ebenso, wenn ein neues Programm editiert werden soll.

### **Fehlermeldungen**

Wird während der Programmabarbeitung ein Fehler erkannt, so wird das Programm abgebrochen und eine Fehlermeldung ausgegeben. Nun kann dieser Befehl korrigiert werden.

### **Unterprogramme**

Unterprogramme in Assemblersprache sind unter ATARI-BASICODE nicht zugelassen.

### **Datelarbeit**

Es können unter ATARI-BASICODE nur Textdateien verarbeitet werden. Dazu gibt es den Befehl O. und die Funktionen R., W. und C. Die Verarbeitung von Binärdateien ist nicht vorgesehen.

### **Diskettenwechsel**

Disketten sollen nur im Befehlsniveau nach Drücken der RESET-Taste gewechselt werden. Um ganz sicher zu gehen, sollte man vor dem RESET explizit die Anweisung END eingeben oder einen Fehler provozieren, z. B. durch Eingabe eines unsinnigen Zeichens.

### **Hinweis:**

Unter BASICODE sind die Zeilen 0 bis 999 für systemgebundene Unterprogramme reserviert. Sie sollten auch von Programmierern, die diese Unterprogramme nicht benutzen, freigelassen werden. Eine Reihe von Befehlen verarbeiten nur Zeilen mit Zeilennummern ab 1000.

### **Wahl der veränderten Hardware bei der Datensette 1010**

Zur Verringerung des Hardware-Aufwandes sollte für BASICODE kein separates Kassettengerät angeschlossen werden. Es wurde statt dessen eine relativ leicht zu realisierende Modifikation der Datensette gewählt. Dafür gab es drei Gründe.

Erstens muß in jedem Fall Hardware gebaut werden, die Veränderung der Datensette machte den geringsten Aufwand. Zweitens existiert beim ATARI ohnehin eine übergroße Anzahl von Kabeln und Einzelgeräten. Sie sollte nicht weiter erhöht werden. Drittens ist die programmierte Motorschaltung der Datensette ein wertvolles Hilfsmittel, das so beibehalten werden konnte.

Der Umbau wurde an einer Datensette 1010 durchgeführt. Besitzer anderer Datensetten können bei ihren Geräten analoge Schritte durchführen.

BASICODE wird immer auf beide Kanäle der Kassette aufgezeichnet. Die Aufzeichnung mit beiden Kanälen erhöht die Sicherheit von Kassettenaufzeichnungen im ATARI-Format erheblich. Normal verwendet der ATARI nur den rechten Kanal, um Signale aufzuzeichnen. Der linke Kanal geht direkt zum Audiokanal des Fernsehers oder Monitors und ist für Zusatzinformationen, z. B. Sprache, gedacht. Das zum linken Kanal gehörige Kabel muß vom Tonkopf entfernt werden und beide Tonkopfköpfe phasenrichtig (!) in Reihe (nicht parallel) geschaltet werden. Damit existiert keine übliche Möglichkeit zum Mithören mehr. Um dennoch etwas zu hören, wird der Kollektor des zweiten Vorverstärker-Transistors Q102 über einen Kondensator von 22 nF direkt mit Pin 11 der seriellen Buchse verbunden.

Zur Aufzeichnung von BASICODE sind keine Hardware-Änderungen notwendig, alle Anpassungen sind mittels Software ausgeführt.

Für das Einlesen von BASICODE sind jedoch einige Probleme zu lösen. Die ATARI-Datasetten besitzen einen eingebauten Frequenzdecoder und geben ein bereits decodiertes bitserielles Signal ab. Dieser Decoder muß für die Wiedergabe von BASICODE-Aufzeichnungen außer Betrieb gesetzt werden. Unmittelbar vor dem Signalausgang (Pin 3 der seriellen Buchse) liegt ein Inverter mit offenem Kollektor (Transistor Q103). Die Drahtbrücke an seiner Basis ist aufzutrennen und durch einen einfachen Schalter zu ersetzen. Das noch nicht decodierte Signal wird an der gleichen Stelle (wie oben beschrieben Q102) abgegriffen und auf den Schleifer eines Regelwiderstandes von 68 kOhm gelegt. Das eine feste Ende des Regelwiderstandes liegt auf Masse, das andere an der Basis



von Q103. Der Regler muß bei geöffnetem Schalter so abgeglichen werden, daß beim Einlesen eines BASICODE-Programms Bit 4 des Ports 53775 mit gleicher Häufigkeit 0 und 1 annimmt, bei Kassettenrauschen jedoch immer auf 0 liegt. Das läßt sich sehr einfach mit folgendem Programm durchführen:

```
10 POKE 54018,52
20 PRINT PEEK(53775)AND16:GOTO 20
```

Wird nun der Schalter geöffnet, ist der Frequenzdecoder unwirksam, und der Recorder kann BASICODE-Kassetten verarbeiten. Dieses Regime wird im folgenden BASICODE-Modus oder einfach "B" genannt. Wird der Schalter geschlossen, so ist wieder der Frequenzdecoder wirksam. Die Widerstandsverhältnisse sind dabei so, daß der nicht dekodierte Pegel, der immer noch an der Basis von Q103 anliegt, unwirksam wird. In diesem Regime kann die Datasette Kassetten im ATARI-Format verarbeiten, es wird ATARI-Modus oder "A" genannt.

Wurde die Hardwareänderung an der Kassette (s. Einleitung) nicht durchgeführt, ergeben sich folgende Einschränkungen:

- Die Befehle CL. und CS. laufen nicht.
- Bei allen Dateibefehlen (LOAD, SAVE, O., C., R., W.) ist das Gerät "B:" (Kassette im BASICODE-Dateiformat) nicht benutzbar.

Alle übrigen Befehle laufen.

### Umbau der Datasette XC 12

Der folgende Vorschlag wurde uns inhaltlich als Ergänzung zum vorstehenden Umbau der Datasette 1010 von W. Koronowski übermittelt. Es wird ein Hardwarezusatz für die Datasette XC 12 mit 4800 Baud beschrieben, der schon mehrfach benutzt wird. Je nach der Phasenlage des Tonkopfes existieren 2 Varianten:

a) Mit den Schaltkreisen B 555 und DL 000

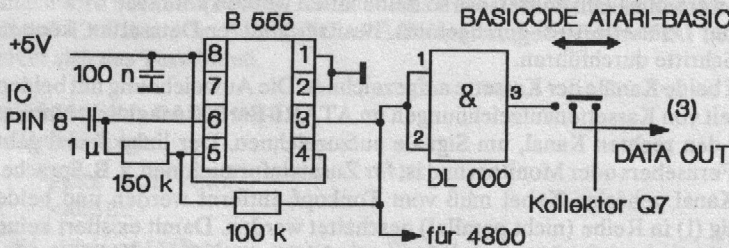


Abbildung 5.20.

b) Mit dem Schaltkreis B 555

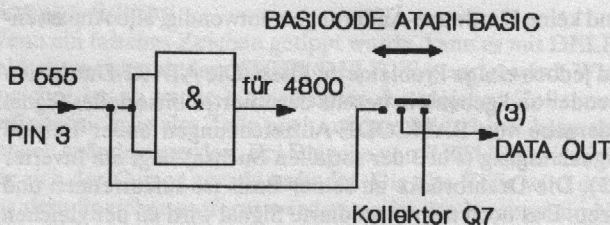


Abbildung 5.21.

Die Erweiterungen hierzu für BASICODE zeigen Abbildungen 5.17. und 5.18. DATA OUT (3) ist auf den jeweiligen Betriebszustand umzuschalten. Bei beiden Varianten braucht kein Leiterzug aufgetrennt zu werden.

Wenn schon der Umbau der Datasette auf 4800 Baud erfolgt ist, ist nur eine kleine Zusatzplatine in die Datasette einzubauen. Ein Abgleich ist nicht notwendig.

Die Änderung am Tonkopf ist wie bei der Datasette 1010 vorzunehmen. Auch der für die Mithörkontrolle benötigte Kondensator von 22 nF ist zu berücksichtigen. Für 4800 Baud ist dann noch eine Leitung zwischen Datasette und Computer zu realisieren, wenn die Audioverbindung für PROCEED benutzt wird. Die Abbildung 5.19. zeigt die Tonkopfanschlüsse.

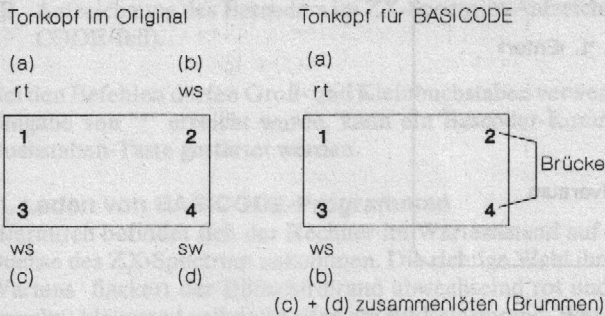


Abbildung 5.22.

## 5.9. Bascoder für den Sinclair ZX-Spectrum

### Verwendbare Rechnertypen

Der hier beschriebene Bascoder eignet sich für jeden zum Sinclair ZX-Spectrum mit 48 oder 80 kByte RAM kompatiblen Rechner mit dem originalen Sinclair-BASIC-Betriebssystem. Neben dem Standard-Kassetten-Interface des ZX-Spectrum sind Microdrives ohne große Probleme nutzbar, Disketteninterfaces je nach Typ jedoch nur mit gewissen Einschränkungen oder Problemen.

### Einfache Anleitung

Der Bascoder für den ZX-Spectrum besteht aus zwei Programmteilen:

- BASIC-Programm ("basiCODE 3") und
- Code-Block ("CODE BASICODE 3"CODE).

Beide Teile sind im Standard-Kassettenaufzeichnungsverfahren des ZX-Spectrum einzulesen. Von der Kassette wird der Bascoder folglich mit

```
LOAD"basiCODE 3" <ENTER> oder LOAD"" <ENTER>
```

geladen. Der zweite Programmteil wird also automatisch nachgeladen. Der Teil CODE ist als spezielles Token mit nur einem Tastendruck zu erzeugen. Der Bascoder meldet sich dann mit Titelbild und Versionsnummer.