

Electronic

MUSICIAN

A MIX PUBLICATION

DEPARTMENTS

- 6** Editor's Note
- 8** Letters
- 15** Operation Help
- 16** What's New
- 134** Re:Views by Robert Carlberg
- 135** Classifieds
- 138** Advertiser Index

COVER

All musicians want their music heard in the best possible light—and the optical CD medium is the ultimate showcase. Now you can bring your music into the digital age on a medium with no hiss, no wow, no flutter, and razor-flat frequency response. It's easier than ever to make your own CDs; the sound quality has never been better, the manufacturing process more predictable, or the prices lower. Our cover represents the process of music turning into CDs, which we suspect more and more electronic musicians will be doing in the months ahead. (Artwork courtesy of Discronics Inc.)

Electronic Musician is published at 2608 Ninth Street, Berkeley, CA 94710 and is © 1988 by Mix Publications, Inc. This is Volume Four, Number Two, February 1988. *Electronic Musician* (ISSN: 0884-4720) is published monthly. Second Class postage paid at Berkeley, CA and additional mailing offices. All rights reserved. This publication may not be reproduced, quoted in whole or in part by mimeograph or any other manner without written permission of the publishers.

Subscriptions are available for \$22.00 per year (12 issues). Single or back issue price is \$3.50. Subscription rates outside the U.S. are \$34.00 per year.

Send subscription applications, subscription inquiries and changes of address to *Electronic Musician*, PO Box 3747, Escondido, CA 92025 or call one of these toll-free numbers: in CA, 1-(800) 255-3302; outside CA, 1-(800) 334-8152.

Address all other correspondence to *Electronic Musician*, 2608 Ninth Street, Berkeley, CA 94710, (415) 843-7901, FAX: (415) 843-9540.

We are interested in receiving unsolicited manuscripts but cannot be responsible for them and cannot return them unless they are accompanied by a stamped, self-addressed envelope. We urge you to send for our "How to Write for EM" guidelines.

Display advertising rates, specs and closing dates are available upon request.

To the best of our knowledge the information contained herein is correct. However, Mix Publications, Inc., its editors and writers cannot be held responsible for the use of the information or any damages which may result.

ARTICLES



RECORDING

Compact Disc Production for the Independent Musician by Dr. Toby Mountain **52**

Just about everyone loves the great sound of CDs. Now you can immortalize your own music in this format for less than you might think.

Three-Dimensional Textures in Electronic Music by Bob Mithoff **72**

Why settle for two-dimensional sound: three dimensions are within your grasp?

The Alignment Caper by Mike Babbitt **80**

Follow ace detective Larry Cheeseman as he tracks down the tracks, and finds there's more to recorder alignment than meets the (private) eye.

Synchronized Recording with Virtual MIDI Tracks by Gary Osteen **62**

Here's a case history of how MIDI enhanced a multi-track recording project.



MIDI

MIDI by the Numbers by Dick Valenti **22**

If you're working with MIDI codes, you'll find this handy reference chart invaluable.



SPECIAL FEATURES

Making Musical Instruments Magical by Craig Anderton **44**

It's the chemistry between you and your instrument that inspires new creative wonders.

MIDI For Horn Players: The Akai Electronic Woodwind Instrument by Tim Tully **118**

Give me your tired, your hungry, your huddled wind instrumentalists searching for new means of expression.



CAREERS

Breaking into Commercial Music by Hal Goodtree **100**

It's not be so hard to make a living from your music if you understand the market.



SOFTWARE

Drumbox: The ST/CZ Connection by David Snow **32**

This ingenious Atari ST program turns the CZ-101 (or any other multi-timbral synth) into an algorithmic drummer.



DO-IT-YOURSELF

Amiga 1000 MIDI Interface by Mark D. Braunstein **90**

As the Amiga computer gains momentum, you'll want to add a MIDI interface—so we present one you can build for a song.

Budget Box Stocks Sloppy Floppies by Steve Tischer **96**

Simple, inexpensive filing for 3.5-inch disks population explosion.

Service Clinic by Alan Gary Campbell **107**

Every do-it-yourselfer needs a good reference library, so here are the recommendations of a seasoned professional.



REVIEWS

First Take **112**

Roland GP-8; Coda MacDrums; TX81Z Pro Editor/Librarian for the Mac; Atari ST Public Domain Patch Editors; Book: FM Theory and Applications; Korg DSM-1

The Roland VP-70 Voice Processor by Paul Lehrman **124**

Battle of the Budget Synchronizers: The JL Cooper PPS-1 vs. the Harmony Systems MTS-1 by Craig Anderton **130**



*You too can
crank out persuasive
percussive patterns
by the bucketful—
or rather, your
computer can.*

DRUMBOX

The CZ/ST Connection

BY DAVID SNOW

A vision of things to come:

It is morning, sweet and glorious, as sunlight filters through the pines and spreads across the rolling landscape of my estate. I am taking breakfast on the piazza, my only companions a copy of the Enquirer and a French poodle named Sasha. Perusing the "Arts and Leisure" section, I refrain from devouring my last bite of Twinkie long enough to scan the paper's review of my latest symphonic creation, "Poem of Near-Ecstasy, or At Least Feeling Real Good About Myself." The familiar hyperbole virtually leaps from the page: "...brilliant...magnificent...a work of profound, almost shocking originality and significance...worthy of Liberace."

Meanwhile, the PC in the music room is chugging out another masterpiece, running under my favorite algorithmic composer software, Opus 1 2 3.

I put down the paper and sigh. Ah, the good life...

While life may not be all poodles and Twinkies for most composers, the idea of

David Snow studied electronic music at Yale University with Jacob Druckman. His composition credits include: awards and grants from BMI, ASCAP, and the National Endowment for the Arts; several commissions; and performances in Washington, New York, and Tokyo. His recording of "The Passion and Transfiguration of a Post-Apocalyptic Eunuch" with the NEW HIPPIES! is on the Opus One label.

cranking out chart busters by the bale is appealing, especially if it makes you rich. After all, music has as much to do with process as it does with sound, and any compositional process, even an automated one, is fine if it works for you.

The use of computers to generate musical material has been covered before in **EM**, usually with an apologetic note about randomness being no substitute for talent (sez who?). But I present you here with an unrepentant opportunity to explore the netherworld of creativity: Drumbox is a program written in ST BASIC that generates, plays, and sequences random rhythmic patterns on your CZ synthesizer, and—if you're not a purist—allows you to edit them to your heart's content. It owes a debt to Tim Ebling's Random Rhythms program (April '87 **EM**), and to Lucky Westfall's June '87 article on using the CZ-101 as a drum machine. Thanks, guys. This thing works! It's fun! It's musically useful...and it's free!

ABOUT THE PROGRAM

Drumbox has three main modules:

- ✓ The *Randomizer* generates four-part polyphonic rhythmic patterns. Each pattern is divided into two to 16 pulses (a pulse is a beat or sub-beat, depending on how you hear it). Up to 20 different patterns can be generated and stored.
- ✓ The *Editor* allows alteration of the rhythmic or pitch content of each pattern.

✓ The *Sequencer* links patterns into sequences, and chains sequences into larger units; sequences can contain up to 16 patterns, and up to 20 sequences can be stored; chains can be up to 85 sequences long.

The degree of randomness for each of the four voices in each pattern is selectable and input as a percentage; only the rhythmic placement of notes or "hits" in each pattern is randomized. The default pitch for each hit is middle "C." Once a pattern is generated, it is displayed on the screen as a grid that can be edited and played back. Once modified to suit your taste, the pattern can be stored and incorporated into sequences.

BASIC isn't the language best suited to real-time tasks like MIDI sequencers since it can't deliver lightning speed. Even so, Drumbox performs quite well, since it limits its commands to note on/off commands on only four channels. Although unison hits won't be perfectly simultaneous, the effect is a natural one given the imprecision of live ensembles.

HOW IT WORKS

Referring to the program listing in **Fig. 1**, here's the blow-by-blow description:

Lines 80 through 130 adjust the size of the output window, initialize variables, and set up arrays that store rhythmic patterns and sequences.

Lines 160 through 290 set up the

List of DRUMBOX.BAS

```

10  '*** ST DRUMBOX by David Snow ***
20  '
30  '   THIS PROGRAM GENERATES AND EDITS
40  ' SEQUENCES OF RANDOM RHYTHMIC PATTERNS,
50  ' AND PLAYS THEM ON MIDI CHANNELS 1 TO 4
60  '
70  '*** INITIALIZE VARIABLES ***
80  option base 1: randomize (0)
90  dim array(20,4,16): dim seq(20,16)
100 a#=gb: gintin=peek(a#+8): gintout=peek(a#+12)
110 addrin=peek(a#+16): b#=addrin: c#=gintin+4
120 tempo=150
130 width(80): fullw 2
140 '
150 '*** MAIN MENU ***
160 title$=" ST DRUMBOX ": gosub 2100
170 clearw 2: gotoxy 0,3
180 ? tab(34): "1. RANDOMIZE"
190 ? ? tab(34): "2. EDIT
200 ? ? tab(34): "3. SEQUENCE"
210 ? ? tab(34): "4. SET TEMPO:"
220 ? ? tab(38): "(1-50)"
230 ? ? tab(38): "[":int(50-(tempo-50)/10):"]"
240 gotoxy 17,16: input "OPTION: ",option$
250 if option$="1" then 380
260 if option$="2" then 1540
270 if option$="3" then 710
280 if option$="4" then 310
290 goto 170
300 '
310 '*** SET TEMPO ***
320 gotoxy 18,13: ? " __ "
330 gotoxy 17,13: input tempo$
340 if len(tempo$)=0 then 170
350 if val(tempo$)>50 then 170
360 tempo=10*(50-(val(tempo$)))+50: goto 170
370 '
380 '*** RANDOMIZE RHYTHMIC PATTERN ***
390 title$="RANDOMIZER": gosub 2100
400 clearw 2: gotoxy 6,8
410 input "STARTING AT WHICH PATTERN NUMBER (1-20)? ",pattern
420 if pattern<1 or pattern>20 then 400
430 for pattern=pattern to 20
440 title$="RANDOMIZER": gosub 2100
450 clearw 2:gotoxy 15,1: ? "PATTERN NUMBER";pattern
460 ?:" " NUMBER OF PULSES (2-16) ":
470 input "HIT 'RETURN' FOR MENU": ",n$
480 if len(n$)=0 then 150
490 n=val(n$): if n<2 or n>16 then 450
500 ?:" " PROBABILITY OF HIT (1-100):"
510 ?:" " for track=1 to 4
520 ? tab(32): "TRACK";track:
530 input "= ",prob
540 if prob>100 then 520
550 if n<16 then array(pattern,1,n+1)=129
560 for pulse=1 to n
570 rand=int(rnd*100)+1
580 if rand<prob then array(pattern,track,pulse)=60
590 if rand>prob then array(pattern,track,pulse)=128
600 next pulse
610 next track
620 gosub 1720
630 '
640 '*** EDIT/SAVE DIALOG BOX ***
650 poke gintin,1: option$="[2]:[Edit pattern?]:[EDIT]:[SAVE]:[RETRY]"
660 poke b#,varptr(option$): gemsys(52): p=peek(gintout)
670 if p=1 then gosub 1970: goto 640
680 if p=3 then 440
690 next pattern
700 '
710 '*** SEQUENCER ***
720 on error goto 2150
730 title$="SEQUENCER": gosub 2100
740 clearw 2: gotoxy 0,0
750 '
760 '*** SEQUENCER DIALOG BOX 1 ***
770 poke gintin,3
780 option$="[2]:[A: CREATE SEQUENCES]:[B: CREATE SEQUENCE CHAINS"
790 option$=option$ + ":[C: QUIT TO MENU]:[ A : B : C ]"
800 poke b#,varptr(option$): gemsys(52): p=peek(gintout)
810 if p=2 then 1190
820 if p=3 then 150
830 clearw 2: gotoxy 8,8

```

FIG. 1: Program listing for "Drumbox."

menu from which all program functions are accessible.

The tempo-setting routine in lines 320 through 360 calculates a variable which is used in a delay loop in the playback routine.

The randomizer starting at line 410 initially asks for an identifier number from 1 to 20 to give the first pattern to be generated. As each pattern is produced, edited, and stored, the number increments automatically, up to 20. Lines 460 to 470 ask for the number of pulses the pattern will have, then lines 510 to 610 ask for the probability (0 to 100%) of a hit for each of the four parts. A random number from 1 to 100 is generated for each pulse of each track, and if that number is less than or equal to the probability entered for that track, the pulse is recorded as a hit, and given the default pitch value of 60 (middle "C"). If the random number is greater than the probability, it is given the value of 128, which the playback routine interprets as a rest. These values are used so that you can edit the pitch to any valid MIDI note from 0 to 127.

Starting at 840, the sequencer asks for an initial sequence number (1-20) which is incremented as each sequence is stored. In line 950 you input a string containing the numbers of the patterns you want in your sequence. To make the string simple for the program to interpret and to keep it legible, each number is entered as two digits separated by a space or any other character (e.g., 02 04 04 11 12 . . .). Each sequence string can contain up to 16 patterns and can either be stored or written over, but not edited.

At line 1270 you enter a string containing the numbers of the sequences that you wish to chain together. The sequence chain can consist of up to 85 sequences, since a string has a maximum length of 255 characters.

Lines 1430 to 1520 comprise the playback routine. For each pulse in the pattern to be played, the array element corresponding to that pulse in each track is examined. If the value of that element is less than 128, it is valid MIDI pitch data and is output to the synthesizer in line 1460. If it equals 128, then it is a rest; if it equals 129, it is an end-of-line marker indicating that the pattern has fewer than 16 pulses and is finished. After the pulse data is examined, a delay loop in line 1500 creates a pause dependent upon the value of the tempo variable. After the

pause, the next pulse is examined and so on, until the end of the pattern.

Line 1470 is an optional note-off command for each hit. Percussive synthesizer patches with no sustain don't require it, and since we want to speed things up as much as possible, it's desirable to omit it.

The commands in lines 980, 1300, and 1920 turn off the ST's GEM interface. This is done to speed up data output during playback, and to enable the keyboard buffer-checking routine in lines 1020, 1350, and 1940. As the program is configured, if a key is pressed during playback, playback will be terminated at the end of the current pattern. If GEM were left on while the program was running, it would "steal" the data in the buffer before the program could get to it, and the program would never know that a key had been pressed. GEM must be turned on again in lines 1050, 1380, and 1950 to resume normal operation.

A grid display of each rhythmic pattern is constructed in lines 1760 to 1900, showing the position and pitch value of each hit. Rests are depicted as blank spaces.

Lines 2110 to 2130 identify the current program function in the output window's title bar so you know what's going on.

Lines 2160 to 2180 catch sequencer-string entry errors so that the program doesn't come to a grinding halt if you mis-type.

TYPING IT IN

This is a long program, and the possibility of entering a typo is great. Save the program to disk before you run it. The worst that can happen with an ordinary BASIC program with a bug is that it just stops or won't run if you attempt to run it. Unfortunately, Drumbox uses AES (Applications Environment Services) routines to draw dialog boxes which, if mis-typed, can cause crashes and force you to reset the machine.

The following comments should clarify most typing questions:

- ✓ Line 320: The characters between the quote marks are a space, two underlines (shifted dash), and two spaces.
- ✓ Line 650: The characters between the quotes must be typed in exactly as written. Note the use of square brackets. The vertical lines around the words (solid

```

840 input "STARTING AT WHICH SEQUENCE NUMBER? (1-20) ",m
850 if m<1 or m>20 then 830
860 clearw 2: gotoxy 0,0
870 ? : ? " EACH SEQUENCE MAY CONTAIN NO MORE THAN 16 PATTERNS."
880 ? : ? " ENTER PATTERNS BY NUMBER (1-20)";
890 ? " [HIT 'RETURN' FOR MENU]:"
900 ? : ? " (Use 2-digit format with spaces between ";
910 ? "numbers [e.g. 02 02 11 08...])"
920 n=77: border$=string$(n,"_"): ? border$
930 for m=m to 20
940 ? : ? " SEQUENCE NUMBER";m
950 ? : input sequence$
960 if len(sequence$)=0 then 150
970 if len(sequence$)>48 then sequence$=left$(sequence$,48)
980 poke systab+24,1
990 for e=1 to len(sequence$) step 3
1000 pattern=val(mid$(sequence$,e,2))
1010 gosub 1430
1020 if inp(-2)<>0 then 1040
1030 next e: goto 1050
1040 j=inp(2): if inp(-2)<>0 then 1040
1050 poke systab+24,0
1060 '
1070 '*** SEQUENCER DIALOG BOX 2 ***
1080 poke gintin,1
1090 option$="[2][ Save sequence? ][SAVE][RETRY]"
1100 poke b#,varptr(option$): gemsys(52): p=peek(gintout)
1110 if p=2 then 950
1120 e=1
1130 for f=1 to len(sequence$) step 3
1140 seq(m,e)=val(mid$(sequence$,f,2))
1150 e=e+1: next f
1160 if e<17 then seq(m,e)=0
1170 next m
1180 '
1190 '*** CHAIN SEQUENCES ***
1200 clearw 2: gotoxy 0,0
1210 ? : ? " ENTER SEQUENCES BY NUMBER (1-20) ";
1220 ? " [HIT 'RETURN' FOR MENU]:"
1230 ? : ? " (Use 2-digit format with spaces between ";
1240 ? "numbers [e.g. 02 02 11 08...])"
1250 ? : ? " (Enter '*' to repeat previous chain of sequences.)"
1260 n=77: border$=string$(n,"_"): ? border$
1270 ? : input chain$: if len(chain$)=0 then 150
1280 if chain$="*" then chain$=old$
1290 old$=chain$
1300 poke systab+24,1
1310 for g=1 to len(chain$) step 3
1320 for e=1 to 16
1330 pattern=seq(val(mid$(chain$,g,2)),e)
1340 if pattern>0 then gosub 1420 else goto 1370
1350 if inp(-2)<>0 then 1380
1360 next e
1370 next g: goto 1390
1380 j=inp(2): if inp(-2)<>0 then 1380
1390 poke systab+24,0
1400 goto 1270
1410 '
1420 '*** PLAY RHYTHMIC PATTERN ***
1430 for pulse=1 to 16
1440 for track=1 to 4
1450 if array(pattern,track,pulse)>127 then 1480
1460 out(3),track+143: out(3),array(pattern,track,pulse): out(3),64
1470 '(OPTIONAL NOTE OFF)out(3),array(pattern,track,pulse): out(3),0
1480 if array(pattern,track,pulse)=129 then 1520
1490 next track
1500 for delay=1 to tempo: next delay
1510 next pulse
1520 return
1530 '
1540 '*** EDIT ***
1550 title$="EDITOR": gosub 2100
1560 clearw 2: gotoxy 0,8
1570 ? " ENTER PATTERN NUMBER TO EDIT (1-20) ";
1580 input "[HIT 'RETURN' FOR MENU]:" ,pattern$
1590 if len(pattern$)=0 then 150
1600 if val(pattern$)>20 then 1560 else pattern=val(pattern$)
1610 for n=1 to 16
1620 if array(pattern,1,n)=129 then 1640
1630 next n
1640 n=n-1: gosub 1720
1650 '
1660 '*** EDITOR DIALOG BOX ***
1670 poke gintin,1: option$="[2][ Edit pattern? ][EDIT][SAVE]"

```



```

1680 poke b#,varptr(option%): gemsys(52): p=peek(gintout)
1690 if p=1 then gosub 1770: goto 1660
1700 goto 1560
1710 '
1720 '*** DISPLAY RHYTHM GRID ***
1730 title$="EDITOR": gosub 2100
1740 clearw 2: gotoxy 17,1: ? "PATTERN";pattern
1750 ?:" " PULSE:";tab(14);
1760 for pulse=1 to n
1770 hit$=right$(str$(pulse),2)
1780 ? hit$:" " :; next pulse
1790 border$=string$(13+n*4,"_")
1800 ?:" ? border$
1810 for track=1 to 4
1820 ? " TRACK";track;tab(13);"!";
1830 for pulse=1 to n
1840 hit$=right$(str$(array(pattern,track,pulse)),3)
1850 if val(hit$)>127 then hit$=" "
1860 if len(hit$)<3 then hit$=" "+hit$
1870 ? hit$:" ?";
1880 next pulse
1890 ?:" next track
1900 ? border$
1910 gotoxy 0,11: ? " [HIT 'RETURN' TO STOP PLAY]"
1920 poke systab+24,1
1930 gosub 1420
1940 if inp(-2)<>-1 then 1930
1950 j=inp(2): if inp(-2)<>0 then 1950
1960 poke systab+24,0: return
1970 gotoxy 0,11: ? " TRACK TO EDIT (1-4) ";
1980 input "[HIT 'RETURN' TO START/STOP PLAY]: ",track$
1990 if track$<>" " then 2010
2000 goto 1920
2010 if val(track$)>4 or val(track$)<1 then 1920
2020 ?:" " PULSE TO EDIT ( 1 -;n;)" :; input pulse
2030 if pulse>n or pulse<1 then 2020
2040 ?:" input " ENTER PITCH NUMBER (0-127) OR REST (*): ",pitch$
2050 if pitch$="*" then pitch$="128"
2060 if val(pitch$)>128 then 2040
2070 array(pattern,val(track$),pulse)=val(pitch$)
2080 goto 1730
2090 '
2100 '*** OUTPUT-WINDOW TITLE ***
2110 title$=title$+chr$(0)
2120 poke gintin,peek(systab+8): poke gintin+2,2
2130 poke c#,varptr(title%): gemsys(105): return
2140 '
2150 '*** ERROR TRAP ***
2160 poke systab+24,0
2170 ? " >>>ERROR IN SEQUENCER ENTRY<<<<"
2180 resume 950

```

lines on the screen, broken when printed) "Edit Pattern?," and between the words "EDIT|SAVE|RETRY" are typed by shifting the back slash key, which is to the right of the RETURN key.

✓ Lines 780 to 790 and line 1090: See the comments for line 650 above.

✓ Line 1470: This line is printed as a REMark statement, and will not be executed when the program is run unless the apostrophe and the characters "(OPTIONAL NOTE OFF)" are deleted. Type it in as written for now.

✓ Line 1670: See the comment for line 650 above.

✓ Line 1750: There are three spaces before the word "PULSE."

✓ Line 1780: There are two spaces between quotes.

✓ Line 1820: There are two spaces before the word "TRACK." The vertical line

in quotes at the end of the line is a shifted back slash.

✓ Line 1850: There are three spaces between quotes.

✓ Line 1870: The vertical line in quotes is a shifted back slash.

USING DRUMBOX

Typing the listing is the hard part, but the program is simple to use. It was written on a 1040ST monochrome system, though there shouldn't be any resolution problems running it in color. I don't know whether it is necessary to turn off buffered graphics on a 520ST with 512K of RAM in order to conserve memory.

The first thing to do is to set your synthesizer to Mono mode on channels 1 to 4 (refer to your owner's manual, or re-read "Casio's New Drum Machine—

The CZ-101?" in June's EM). Assign patches with contrasting timbres and no sustain to the four channels. Fig. 2 includes a good selection of patches by a variety of authors to get you started.

Load BASIC from your language disk and run Drumbox. You first need to generate some patterns, so enter option 1, *Randomize*. You will be asked to enter a number from 1 to 20 to assign the first pattern. If you're starting a session, enter 1. If you have already generated a few patterns and then quit to the main menu in order to edit, enter the number where you left off. You will then be asked to enter the number of pulses you want for the pattern, and the probability of a hit for each of the four tracks. The pattern is then generated and the screen displays a grid that shows the relative position of each hit with its MIDI pitch value. The synth will play the pattern over and over until you hit a key, and a dialog box asks if you wish to edit, save, or re-try. If you click the mouse pointer on SAVE, the pattern is stored and the entry process repeats for the next pattern. If you click on RE-TRY, a new pattern will be written over the current one.

To edit, click on the EDIT box, or hit the RETURN or ENTER keys. You will be asked for the track and pulse you wish to change, and the new pitch value. Enter a value from 0 to 127 for a hit, or "*" for a rest. The grid will be redrawn and the pattern played until a key is hit, and the dialog box again asks for your next move.

Once you have saved some patterns to your liking, return to the main menu and select option 3: SEQUENCER. Click the mouse on box "A," type the numbers of the patterns you wish to link together in the format indicated, and hit ENTER or RETURN. The sequence will play and you will be given the option to save it or try again.

When you have assembled some sequences, return to the menu and select option 3 again, but click on box "B" to create chains of sequences. Type the numbers of the sequences you wish to chain and hit ENTER or RETURN.

Don't move the mouse while patterns are playing, since the mouse interface generates interrupts that slow down the program.

Playback can be interrupted at any time by hitting a key. It will terminate at the end of the current pattern.

To set your tempo, select option 4

from the menu and enter any value from 1 (slow) to 50 (fast).

FINAL MUSINGS


You may find that in order to get the most satisfying results, the more pulses you assign to a pattern, the lower the overall range of hit probability should be. A 16-pulse pattern seems to work best with probabilities in the range of 15 to 45, while an eight-pulse pattern works better in a 35 to 65 range. You can assign 100% probability to a track in order to get hits on every pulse, or 0 to leave a track blank. Patterns longer than 16 pulses can be made by sequencing two or more shorter patterns together. Changing the patches assigned to each voice can alter the character of a rhythm radically. There are more possibilities than you've got days to live. Experiment!

The implications of randomness are fascinating. If you put 100 monkeys on 100 STs, the chances are that one of them would come up with Varese's "Ionization." I wonder how Edgar would have felt about that?

EM

Note:

The following pages contain patch sheets of some tested and recommended drum sounds for the Casio CZ-101. These can be used as they are, or can provide excellent starting points for developing distinctive drum sounds for your own CZ percussion library. Blank, full-sized versions of these patch sheets were provided in the November, 1987 issue of *Electronic Musician*. While the author designed the Drumbox program with the CZ-101 in mind, any synthesizer with multi-timbral capability and at least four voices could be used in place of the Casio, offering an endless number of musical variations.

CZ101		TONE NAME		CART # TONE #												
SOUND DATA		Hi-Hat														
PARAMETERS																
LINE SEL	MODULATION	DETUNE			VIBRATO		OCT RANGE									
1-2	RING NOISE	+/-	OCTAVE	NOTE	FINE	WAVE	DELAY	RATE	DEPTH							
	OFF ON	+	3	11	60	4	05	99	20	+1						
LINE 1						LINE 2										
WAVE FORM	1	0	DC01 PITCH ENV					5	4	DC02 PITCH ENV						
STEP	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
RATE	80								45	97	74					
LEVEL	00								53	65	00					
S/E	E										E					
KEY FOL	9	DCW1 WAVE ENV					9	DCW2 WAVE ENV								
STEP	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
RATE	90								99	97	86					
LEVEL	00								99	61	00					
S/E	E										E					
KEY FOL	9	DCA1 AMP ENV					0	DCA2 AMP ENV								
STEP	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
RATE	90								99	99	61					
LEVEL	00								99	99	00					
S/E	E										E					
										NOTES						
RECOMMENDED KEYBOARD RANGE										By Alan Bezozi						
Playing different keys produces different hi-hat sounds.																

E!

for the

YAMAHA

DX7II

series

DX7IIFD • DX7IID • DX7S

We added a few things

- The Octal mode** • Play 8 single voices at a time!
- 16 note polyphony with Dynamic Voice Allocation
 - Octal Stereo Pan and Octal Random Pan
 - Separate Volumes/Velocity Processing for each voice

- SEI quencer** • 16 Tracks/16 MIDI channels
- Tracks play any internal DX voice or any MIDI channel
 - 220,000 event memory on DX7IIFD
 - 22,000 event memory for DX7IID/DX7S
 - Real-time and Step recording • Quantize notes

- ENGINE** • 16 Track DX/MIDI event processor
- 16 Track MIDI channel mapping
 - Transmits/receives on 16 MIDI channels simultaneously
 - Auto-accompaniment modes: Player and Chords

- Expanded memory** • 256 internal single voices
- 128 internal Performances

Suggested retail price: \$399

15916 Haven Avenue Tinley Park, Illinois 60477

GREY MATTER

CZ101
SOUND DATA

TONE NAME	CART #	TONE #
Hand Claps		

PARAMETERS

LINE SEL	MODULATION	RING	NOISE	DETUNE	WAVE	DELAY	RATE	DEPTH	OCT RANGE
1+2'	OFF	ON	+	1	00	00	1	00	22

LINE 1

WAVE FORM	8	0	DC01 PITCH ENV
STEP	1	2	3 4 5 6 7 8
RATE	99	99	
LEVEL	82	00	
S/E	E		

LINE 2

WAVE FORM	1	0	DC02 PITCH ENV
STEP	1	2	3 4 5 6 7 8
RATE	99		
LEVEL	00		
S/E	E		

DCW1 WAVE ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	99 28
LEVEL	99 00
S/E	E

DCW2 WAVE ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	99 33
LEVEL	86 00
S/E	E


DCA1 AMP ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	99 99
LEVEL	99 00
S/E	E

DCA2 AMP ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	99 87 99 67
LEVEL	99 00 99 00
S/E	E

NOTES
By Craig Anderton



RECOMMENDED KEYBOARD RANGE

Play multiple notes, slightly out of sync, in the lower octaves. Strike several keys at once for best results.

CZ101
SOUND DATA

TONE NAME	CART #	TONE #
"Roto" Toms		

PARAMETERS

LINE SEL	MODULATION	RING	NOISE	DETUNE	WAVE	DELAY	RATE	DEPTH	OCT RANGE
1+2'	ON	OFF	-	0	00	16	1	00	52

LINE 1

WAVE FORM	5	2	DC01 PITCH ENV
STEP	1	2	3 4 5 6 7 8
RATE	99	40	
LEVEL	48	00	
S/E	E		

LINE 2

WAVE FORM	1	0	DC02 PITCH ENV
STEP	1	2	3 4 5 6 7 8
RATE	99	40	
LEVEL	48	00	
S/E	E		

DCW1 WAVE ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	99 74
LEVEL	99 00
S/E	E

DCW2 WAVE ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	00
LEVEL	00
S/E	E


DCA1 AMP ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	99 48
LEVEL	99 00
S/E	E

DCA2 AMP ENV

KEY FOL	0
STEP	1 2 3 4 5 6 7 8
RATE	75 75
LEVEL	99 00
S/E	E

NOTES
By Craig Anderton



RECOMMENDED KEYBOARD RANGE

Vibrato adds a slight inconsistency between notes; turn off if desired.

The International MIDI Association

The International MIDI Association is an international MIDI information network

We are the exclusive distributors of the MIDI 1.0 Detailed Specification - the official MIDI technical document written by the MMA and JMSC

Membership in the IMA includes a subscription to our monthly newsletter, *the IMA Bulletin*, as well as access to our 'MIDI Hot-line' where we can answer any and all questions regarding the MIDI interface and it's applications

Members also receive access to our library of System Exclusive codes and 50% off the subscription fee for an international MIDI-related telenetwork as well as other services

IMA Individual Membership \$40/year

MIDI 1.0 Detailed Specification:
Members \$25
Non-members \$35

Complete collection of IMA Bulletin Backissues from Nov. '84 \$30

Call for our catalog of MIDI-related publications

IMA

5316 W. 57th St.
Los Angeles, CA 90056
213-649-MIDI