EK-2A

Computer Drums Experimenter's Kit

Assembly Instructions

```
1008-   85 E9       0630        STA *PNTR    :ZERO SCORE POINTER
100A-   8D 20 08    0640        STA DSP      :AND DISPLA
100D-   AA          0650        TAX          :SAVE Z REG P
                                             :AND USE IT TO CLEA
1011-   E8          0670        INX
1012-   D0 FA       0680        BNE SLP0     :LOOP UNTIL DONE
1014-   20 53 10    0690  SLP1  JSR RDKY     :GO READ THE KEYBOA
1017-   B0 1F       0700        BCS ACTN     :AND IF NO NEW KEYS,
1019-   C9 10       0710  TSTS  CMP 10       :NEW KEY - A "CONTR
101B-   B0 1C       0720        BCS CTRL     :YES - BRANCH TO CO
101D-   C9 08       0730        CMP 08       :ONE OF "SPARE" KEY
101F-   B0 [illegible]          BCS SP       RANCH
                    0750  ;     **
1021-   A9 86       0760  CNTRL LDA 86       RUN ENTRY MODE, GE
1023-   85 39       0770        STA ACTN+01  :SET LINK
1025-   B9 D9 10    0780        LDA DSIG,Y   :GET DRUM SIGNATURE
1028-   A6 E9       0790        LDX *PNTR    :GET SCORE POINTER
102A-   9D 00 01    0800        STA SCOR,X   :SAVE DRUM SIG IN S
102D-   20 3E 10    0810        JSR PLAY     :PLAY THE DRUM BEAT
1030-   4C 14 10    0820        JMP SLP1     :LOOP FOR MORE
1033-   B9 D1 00    0830  CTRL  LDA STBL,Y   :GET COMMAND ADDRES
1036-   85 39       0840        STA ACTN+01  :AND SET LINK IN JS
1038-   20 86 00    0850        JSR [illegible]  :AND GO TO COMMAND
103B-   4C 14 10    0860        JMP SLP1     :THEN LOOP FOR MORE
                    0870  ;
                    0880  ;PLAY SUBROUTINE
                    0890  ;
103E-   A4 EB       0900  PLAY  LDY *EXP     :GET EXPRESSION VAR
1040-   8D 40 08    0910        STA OUTP     :OUTPUT CONTROL TO
1043-   29 7F       0920        AND 7F       :RESET STROBE BIT
1045-   88          0930  PLA0  DEY          :DELAY FOR THE EXP.
1046-   D0 FD       0940        BNE PLA0     :LOOP UNTIL DONE
1048-   8D 40 08    0950        STA OUTP     :AND TURN DRUM "OFF
104B-   E6 E9       0960        INC *PNTR    :INCREMENT SCORE PO
104D-   A6 E9       0970        LDX *PNTR    :PLACE IN X REGISTE
104F-   8E 20 08    0980        STX DSP      :AND SHOW IN DISPLA
1052-   60          0990        RTS          :THEN RETURN
                    1000  ;
                    1010  ;READ KEY-ALSO IMPORTANT TO TEMPO
                    1020  ;
1053-   20 00 FF    1030  RDKY  JSR DECD     :PIEBUG KEYBOARD SU
1056-   B0 05       1040        BCS DLY      :SAME KEY - JUST DE
1058-   A2 00       1050        LDX 0
105A-   86 EC       1060        STX *CNTR    :ZERO TEMPO COUNTER
105C-   60          1070        RTS
105D-   A2 20       1080  DLY   LDX 20       :SET X AND Y REGIST
                                             :DELAY PARAMETERS
                                             AND DO DELAY
1064-   CA          1120        DEX
```


PAiA Electronics, Inc.
1020 W. Wilshire, Oklahoma City, OK 73116 - (405)843-9626

COMPUTER DRUMS

by John S. Simonton, Jr.

Are you ready for a computer peripheral that's fun, instructive, low in cost and provides an impressive demonstration that even your least technically inclined friends can understand? How about this one - a computer controlled drum set.

Yes, there are a lot of automatic percussion units available and yes, they're pretty much a drag. Useful, under just the right set of circumstances, but even then their incessant BOOM-chick-chick can really turn into a BUMMER.

Now, think about computer drums. You're not forced into using what somebody else thinks is a cha-cha; you write your own rhythm pattern for the specific thing you're doing, even if it's in 7/16 time. and the real world niceties like bridges and intro's? They're simply not possible with conventional electronic drummers. But with drums tied to a computer - no sweat.

If you're interested, we've got a lot of ground to cover; drum circuits, interfacing and programming. Let's get on with it.

DRUM OSCILLATORS

Most drum units generate their percussion sounds using a simple active filter section like that shown in figure 1. There are two ways to think of this circuit. Either it is almost an oscillator which when excited by a pulse "rings" for a short period of time; or, it is a high Q filter section which extracts from a pulse excitation function a single sine wave component. Either of these explanations is valid. Either produces an accurate description of what happens. When you hit the circuit with a pulse it responds by generating a damped sinusoid, which is the electrical analog of the mechanical action of a drum head.

Table 1 shows a realization of this circuit when using one stage of an LM3900 type quad current differencing amplifier as well as the component values required to generate a number of different drum/percussion sounds.

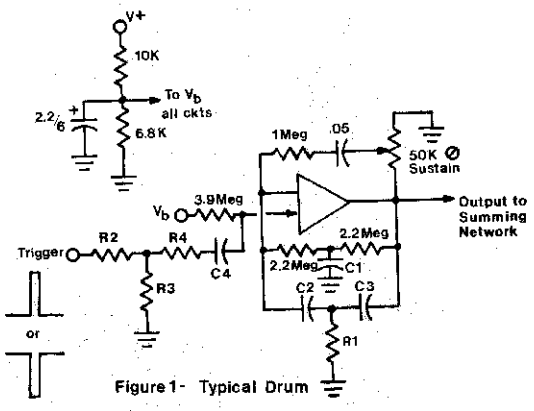


Figure 1- Typical Drum

DRUM	C1	C2,C3	C4	R1	R2	R3	R4
Wood Block	.01	.001	.005	10K	68K	10K	1Meg
Clave	500pf	500pf	.005	33K	68K	10K	330K
Tom-Tom	.01	.001	.005	39K	68K	10K	330K
Conga	.01	.001	.005	68K	68K	10K	330K
Bass [L]	.01	.005	.05	15K	47K	10K	220K
Bass [H]	"	"	"	"	22K	"	"

Table 1 - Component Values for Various Drums
L=light H=heavy

Snare drums and cymbals require a keyed noise source with filtering. Figure 2 shows a circuit that works well for snares - and don't forget that a drum circuit must be used at the same time to produce what is known as the "strike tone".

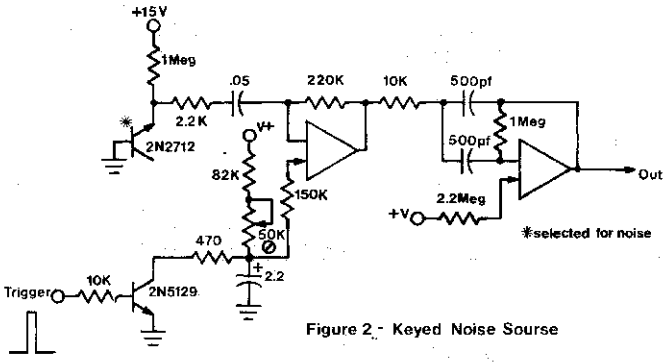


Figure 2 - Keyed Noise Sourse

Building a complete drum set is simply a matter of duplicating these circuits for as many different sounds as you require and resistively mixing them all together to a common audio buss.

INTERFACE

There is only one difficulty with using drum circuits like those shown as a peripheral to a computer and that is that the computer is just too fast. Our filter circuits like a trigger pulse that is at least several milli-seconds long whereas the average write cycle of a microprocessor (and we will be playing these drums by "writing" to them) is on the order of a few micro-seconds. The typical response of our drums to a micro-second long pulse looks like this:

As you can see, it doesn't strain even my non-existant artistic abilities.

If you own one of the computers that's all pretty and boxed up, you can refer to a section of your systems manuals entitled "slow peripherals". In the parlance of the field that's what the drums are.

I'm going to assume that you're going to be using someone's low cost "evaluation" unit. And further, that your manuals are typical of most of the ones that I've seen in that they tell you just enough to make sure that your level of confusion is appropriate to this "complex" field. My comments here are specificly for National's SC/MP evaluation kit, but with slight modifications are applicable to most others.

Figure 3 (shown on the following page) shows an extremely simple interface adapter that has performed well for me. As you can see, it's nothing but three packages of quad CMOS NOR gates. Here's what happens.

As I said before, playing the drums from a program standpoint will be a memory write operation from the processors accumulator to the memory location occupied by the drum set. When the program says write (actually STROBE) the group of ones and zeros which we will have loaded into the accumulator to produce the drum sounds that we want are put out on the machine's data lines while the address of the memory location into which we are going to write appears on the

processor's address lines. Very shortly after the valid address and data appear on their respective processor lines, the machine issues a WRITE command (in the SC/MP, the NWDS line goes to ground indicating a write). As we shall see shortly, when the machine issues both a write command and the address that we've selected for the drum card at the same time it causes the SEL (select) line on the interface to go high which is buffered by gates G1 and G2.

The output of G2 is coupled by the time delay circuit R1 and C1 to the input of G3. The output of G3, which was in a high state, switches low as soon as the SEL line is activated and produces two major results. First, any of the data lines which have a zero on them, when NORed with the output of G3, will cause the outputs of thier respective gates (G4-G10) to change state. These gates changing state cause the drum ocsilator attached to that gate to sound.

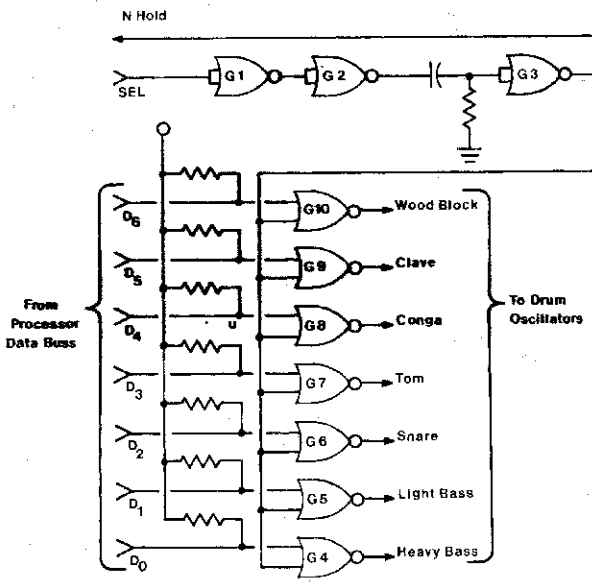


Figure 3 - Drum Interface

Also, the output of G3 is routed back to a pin on the processor called NHOLD. This is an interesting input to the processor, because when this pin is grounded it causes the machine to STOP with the data and address lines held in the same state they were when the hold command was issued. This has the effect of latching the address and data lines of the processor for as long as NHOLD remains low (a time determined by the R1, C1 time constant) and serves to stretch the micro-second write cycle out to the few milli-seconds required by the drums.

We need to discuss programming some, but before we do let's consider for a moment which memory location(s) we want the drum set to occupy. If you are using one of the larger systems by SWTP, IMSAI or MITS (or even some smaller systems such as the F-8), the problem is academic. These machines have provisions for output ports and you will use them - why fight the system. In these cases the SEL and NHOLD lines can be used for the handshaking "data ready" and "data accepted" lines respectively.

On smaller systems you will want to base your decision on what the cost (both financial and emotional) will be of using a specific location or group of locations. May I make a suggestion that might not occur to you otherwise? Use the same group of addresses that is occupied by whatever ROM your system has. This is not "ordinary" (you never write into the ROM locations with the usual programming because it won't do anything - they're READ ONLY MEMORY) but, there's nothing wrong with it and they have the tremendous advantage of being locations that are already decoded. On the SC/MP this is accomplished as shown in figure 4.

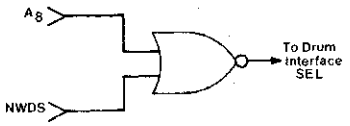


Figure 4 - SC/MP Address Decode

PROGRAMMING

There's another big plus to using the PROM locations as an output port, this one from a programming standpoint. And we're both going to have to go slowly here because it can be confusing otherwise.

ROM, even in minimal systems, represents a chunk of memory - ordinarily at least 256 bytes. This means that in a machine like the SC/MP which has 16 address "lines" (some are multiplexed onto the data bus, but forget that) only the most significant 8 bits are required to address the output port.

Most machines have a scheme of addressing memory which is refered to as "indexed" and they have, internally, one or more 16 bit wide "index" registers that can be used to "point" to a specific memory location. Since most of the processors used by hobbyist work on only 8 bits of information at a time it is obvious that it will take two "groups" of instructions to load the 16 bits into the index register - one group to load the "low order" 8 bits and a second group to load the"high order"8 bits. But if we're using 256 bytes of memory occupied by ROM as a single output port, we don't have to worry about the low order 8 bits, they can be anything and we will still be addressing the output port. Since they can be anything and still work, we don't have to worry about loading them, and as a result we've saved at least three (and probably much more in a complete program) lines of program code.

Programming for the drums can be as simple or elaborate as you like.Regrettably, there is sufficient space here for only a few tips. The instructions for the experimenter's kit goes into much greater detail with sample program listings for a variety of machines.

Let me point out a few hardware considerations that will affect programming. Each bit of data in the accumulator at the time of the write operation will determine whether a specific drum sounds or doesn't sound. A typical coding arrangement was shown in figure 3 and with this arrangement the drums would be coded like this (remember that a 0 sounds the drum).

drum sound	binary	hex	octal
heavy bass	11111110	FE	376
light bass	11111101	FD	375
snare	11111011	FB	373
tom-tom	11110111	F7	367
conga	11101111	EF	357
clave	11011111	DF	337
wood block	10111111	BF	277

This is confusing. It would be much easier if when we were programming we could just write a 1 for the drum we want to sound. We can do that if we write a program that reads a byte of drum data but before writing it to the drums does an Exclusive Or Immediate with FF. As many of you will realize, this has the effect of complementing every bit of data so that the drum sounds we want can be written in memory in this more logical form:

drum sound	binary	hex	octal
heavy bass	00000001	01	001
light bass	00000010	02	002
snare	00000100	04	004
tom-tom	00001000	08	010
conga	00010000	10	020
clave	00100000	20	040
wood block	01000000	40	100

This is particularly easy if we want to sound two drums simultaneously; for example a heavy bass down beat and a snare drum at the same time would be in binary 00000101.

For very simple repeating patterns, this complementing action even has an advantage in that we can use FF as a repeat indicator that is stored along with the drum data. Program flow would be 1) load drum data, 2) complement (XOR IMM, FF) 3) check for zero (11111111 complemented is 00000000) and if zero start again, 4) if not zero write accumulator to drums, 5) delay (tempo), 6) get next data and go to 2.

Various machines have an amazing variety of ways to test the accumulator (test for zero, not zero, pos., neg., carry, overflow, ect.) but one thing they all have is a test for zero.

Notice that the 8th data bit in the arrangements that I've shown does not have a drum associated with it. There are two reasons, first, we've got a bunch of drum sounds already and secondly, we can use the 8th bit in our programming as an indicator that the data we've loaded is not to be played as drums, but to be interpreted as an instruction. A simple example would be to suppose that we have a rhythm pattern that we wish to play 16 times and then stop. We can use a program based on the flow chart shown in figure 5 to accomplish this. Notice that in this case the program reads the drum data, tests to see if it is in fact an instruction, and if not plays it. When it does get the byte that is the instruction it decrements the "count", saves the data back in the location it got it from, and iterates the pattern. On the

last pass through the program the decrement operation on the "instruction data" results in the last seven bits being 0, which tells the machine that it is through (or to go to the next pattern, or whatever). There are, of course, as many ways to handle this as there are people to write programs.

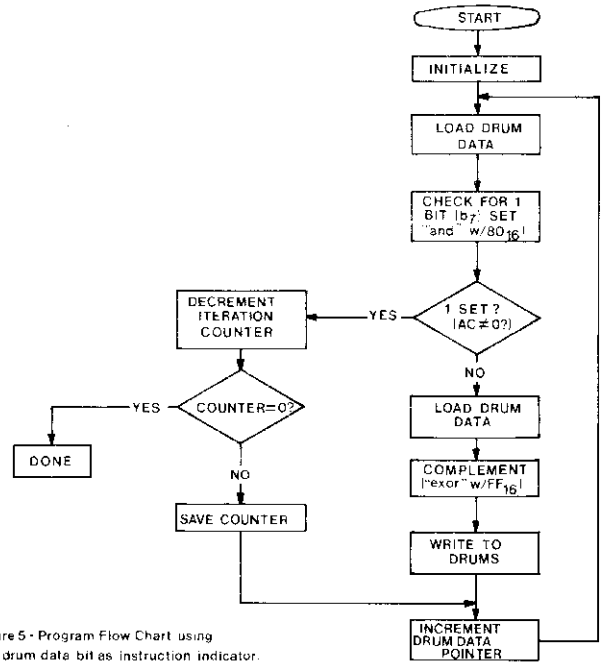
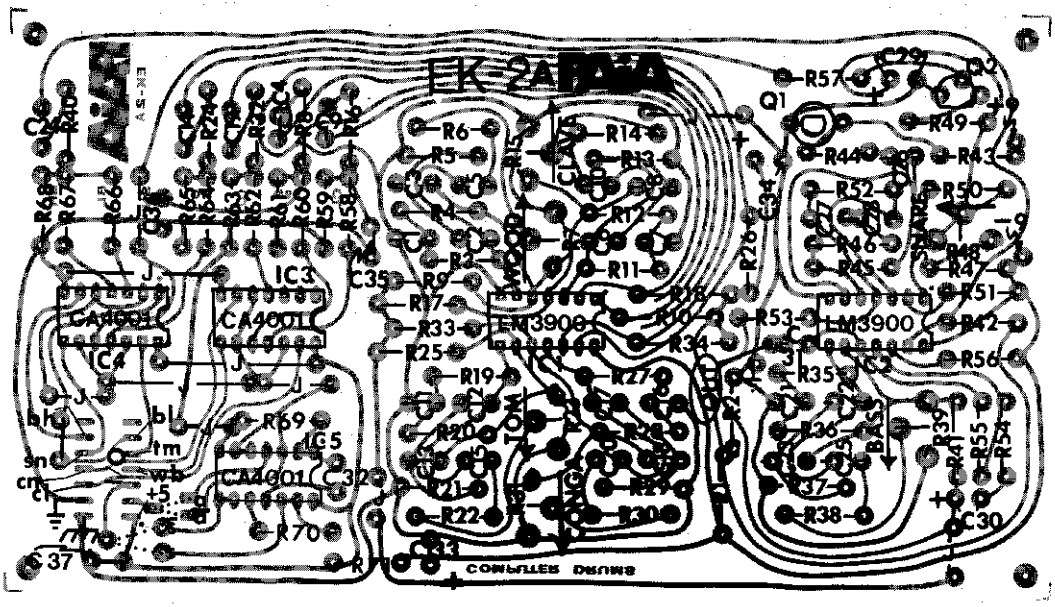


Figure 5 - Program Flow Chart using 8th drum data bit as instruction indicator.



PARTS LIST

QNTY	VALUE	DESC. (alternate marking)
		FIXED RESISTORS
1	47 ohm	yellow-violet-black
1	470 ohm	yellow-violet-brown
1	2200 ohm	red-red-red
1	6800 ohm	blue-grey-red
11	10K	brown-black-orange
1	15K	brown-green-orange
4	18K	brown-grey-orange
1	22K	red-red-orange
1	27K	red-violet-orange
1	33K	orange-orange-orange
1	39K	orange-white-orange
1	47K	yellow-violet-orange
5	68K	blue-grey-orange
1	82K	grey-red-orange
3	150K	brown-green-yellow
2	220K	red-red-yellow
5	330K	orange-orange-yellow
8	1 Megohm	brown-black-green
11	2.2 Megohm	red-red-green
5	3.9 Megohm	orange-white-green

CERAMIC DISK CAPACITORS

7	.001 MFD.	102
7	.005 MFD.	502
5	.01 MFD.	103
9	.05 MFD.	503

ELECTROLYTIC CAPACITORS

1	1 MFD./ 15V.	Greater voltage ratings acceptable.
2	2.2 MFD./ 15V.	
1	33 MFD./ 15V.	

SEMICONDUCTORS

2	LM3900 (3401)	QUAD "NORTON" AMPS
3	4001	QUAD NOR GATE PACKAGE
1	2N5129 OR 2N4124	TRANSISTOR
1	2N2712 OR 2N3391	NOISE TRANSISTOR

MISCELLANIOUS PARTS

6	50K ohm	TRIMMER POTENTIOMETERS
1	OUTPUT CABLE WITH RCA PHONO PLUG	
4 in.	ADHESIVE FOAM TAPE	
1	9 VOLT BATTERY CONNECTOR	

ASSEMBLY

Assembly of the EK-2A computer interfaced drum set is fairly straight-forward thanks to the fully imprinted circuit board. Begin assembly by thoroughly cleaning the circuit board with a steel wool or Scotch Bright (R) pad. The board must be bright and shiny to accept the solder and failure to clean the board will result in poor solder joints and will void the warranty on the kit.

Continue assembly by soldering all the resistors in place. Save the lead clippings for use as jumpers in later steps.

PART	VALUE	COLOR CODE
( ) R1	10K	brown-black-orange
( ) R5	10K	brown-black-orange
( ) R41	10K	brown-black-orange
( ) R46	10K	brown-black-orange
( ) R57	10K	brown-black-orange
( ) R59	10K	brown-black-orange
( ) R61	10K	brown-black-orange
( ) R63	10K	brown-black-orange
( ) R65	10K	brown-black-orange
( ) R68	10K	brown-black-orange
( ) R70	10K	brown-black-orange
( ) R2	6800 ohm	blue-grey-red

() R3	2.2 Meg	red-red-green
() R4	2.2 Meg	red-red-green
() R11	2.2 Meg	red-red-green
() R12	2.2 Meg	red-red-green
() R19	2.2 Meg	red-red-green
() R20	2.2 Meg	red-red-green
() R27	2.2 Meg	red-red-green
() R28	2.2 Meg	red-red-green
() R35	2.2 Meg	red-red-green
() R36	2.2 Meg	red-red-green
() R51	2.2 Meg	red-red-green
() R6	1 Meg	brown-black-green
() R8	1 Meg	brown-black-green
() R14	1 Meg	brown-black-green
() R22	1 Meg	brown-black-green
() R30	1 Meg	brown-black-green
() R38	1 Meg	brown-black-green
() R43	1 Meg	brown-black-green
() R52	1 Meg	brown-black-green
() R9	18K	brown-grey-orange
() R17	18K	brown-grey-orange
() R25	18K	brown-grey-orange
() R33	18K	brown-grey-orange
() R10	3.9 Meg	orange-white-green
() R18	3.9 Meg	orange-white-green
() R26	3.9 Meg	orange-white-green
() R34	3.9 Meg	orange-white-green
() R42	3.9 Meg	orange-white-green
() R13	33K	orange-orange-orange
() R16	330K	orange-orange-yellow
() R24	330K	orange-orange-yellow
() R32	330K	orange-orange-yellow
() R56	330K	orange-orange-yellow
() R69	330K	orange-orange-yellow
() R21	39K	orange-white-orange
() R29	68K	blue-grey-orange
() R58	68K	blue-grey-orange
() R60	68K	blue-grey-orange
() R62	68K	blue-grey-orange
() R64	68K	blue-grey-orange
() R37	15K	brown-green-orange
() R40	220K	red-red-yellow
() R45	220K	red-red-yellow
() R44	2200 ohm	red-red-red
() R47	82K	grey-red-orange
() R49	470 ohm	yellow-violet-brown

() R50	150K	brown-green-yellow
() R54	150K	brown-green-yellow
() R55	150K	brown-green-yellow
() R53	27K	red-violet-orange
() R66	47K	yellow-violet-orange
() R67	22K	red-red-orange
() R71	47 ohm	yellow-violet-black

Next install the ceramic disk capacitors. Values are in microfarad except picofarad as indicated.

PART	VALUE	ALTERNATE MARKINGS
() C1	.01	103
() C11	.01	103
() C16	.01	103
() C21	.01	103
() C36	.01	103
() C2	.001	102
() C3	.001	102
() C12	.001	102
() C13	.001	102
() C17	.001	102
() C18	.001	102
() C32	.001	102
() C4	.005	502
() C9	.005	502
() C14	.005	502
() C19	.005	502
() C22	.005	502
() C23	.005	502
() C35	.005	502
() C5	.05	503
() C10	.05	503
() C15	.05	503
() C20	.05	503
() C24	.05	503
() C25	.05	503
() C26	.05	503
() C31	.05	503
() C37	.05	503
() C6	500 PFD.	500
() C7	500 PFD.	500
() C8	500 PFD.	500
() C27	500 PFD.	500
() C28	500 PFD.	500

Next install the electrolytic capacitors. Note that these parts are polarized and that one lead will have either a "+" or a "-" symbol associated with it on the case. Note also that one of the pair holes in the circuit board is marked "+". If the "+" lead of the part is

marked, it should go through this hole. If the "-" lead is marked, it should go through the unmarked hole.

The capacitor must be installed with the proper orientation for the unit to operate properly.

Note that the voltage specified is the minimum acceptable and that parts supplied with individual kits may have higher working voltages and still be the appropriate part.

PART	VALUE
() C29	2.2 MFD./ 15 VOLT
() C34	2.2 MFD./ 15 VOLT
() C30	1 MFD./ 15 VOLT
() C33	33 MFD./ 15 VOLT

Install the six trimmer potentiometers by pushing their solder tabs through the holes provided in the circuit board

PART	VALUE
() R7	50K TRIMMER
() R15	50K TRIMMER
() R23	50K TRIMMER
() R31	50K TRIMMER
() R39	50K TRIMMER
() R48	50K TRIMMER

There are several wire jumpers on the circuit board which are indicated by a solid line broken with the letter "J". Using the resistor clippings saved from previous steps or the bare wire provided, form and install these jumpers.

The I/O configuration jumpers marked with the dashed lines will be installed in later steps.

QUANITY	ORIENTATION
() 7	HORIZONAL JUMPERS
() 4	VERTICAL JUMPERS

Install the two transistors Q1 and Q2 following the parts placement designators drawn on the circuit board.

Note that transistor Q1 has been selected for it's noise characteristics and is easilly identified by it's missing center lead.

All semiconductors, both transistors and the integrated circuits which follow, are heat sensitive and care must taken during their installation to prevent their being subjected to too high a temperature during soldering. The safest procedure is to grasp the lead being soldered with a pair of needle-nose pliers or hemostats during this operation.

PART	TYPE
() Q1	2N2712 OR2N3391 SELECTED FOR NOISE
() Q2	2N5129 OR 2N4124

Install the integrated circuits. Note that orientation of the IC's is keyed by an identifying notch which appears iether as one end of the body of the part or by a recessed dot which is adjacent to and identifies pin 1.

PART	TYPE
() IC1	LM3900 OR CA3401
() IC2	LM3900 OR CA3401
() IC3	4001 QUAD NOR GATE
() IC4	4001 QUAD NOR GATE
() IC5	4001 QUAD NOR GATE

() Install the output co-ax cable by stripping away aproximately 3/4" of the outer covering to expose the shielding wire or braid. Carefully unbraid the shielding wire and then twist the strands together. DO NOT TIN. Strip 1/4" of insulation from the inner conductor and twist and tin the exposed wire strands. Connect the center conductor of the co-ax to the PC hole marked "OUT" and the sheild to the adjacent hole marked with the ground symbol ( ).

() Solder the RED lead from the battery connector snap to the circuit board point marked "+9".

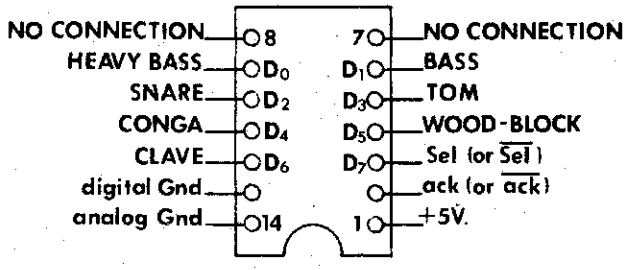
() Solder the BLACK wire from the battery snap to "-9".

If the drum set interface is to be by means of dip-header terminated cabling, as when connecting to the PAIA 8700 computer, install the 14 pin dip socket as the location provided in the lower left hand corner of the board.

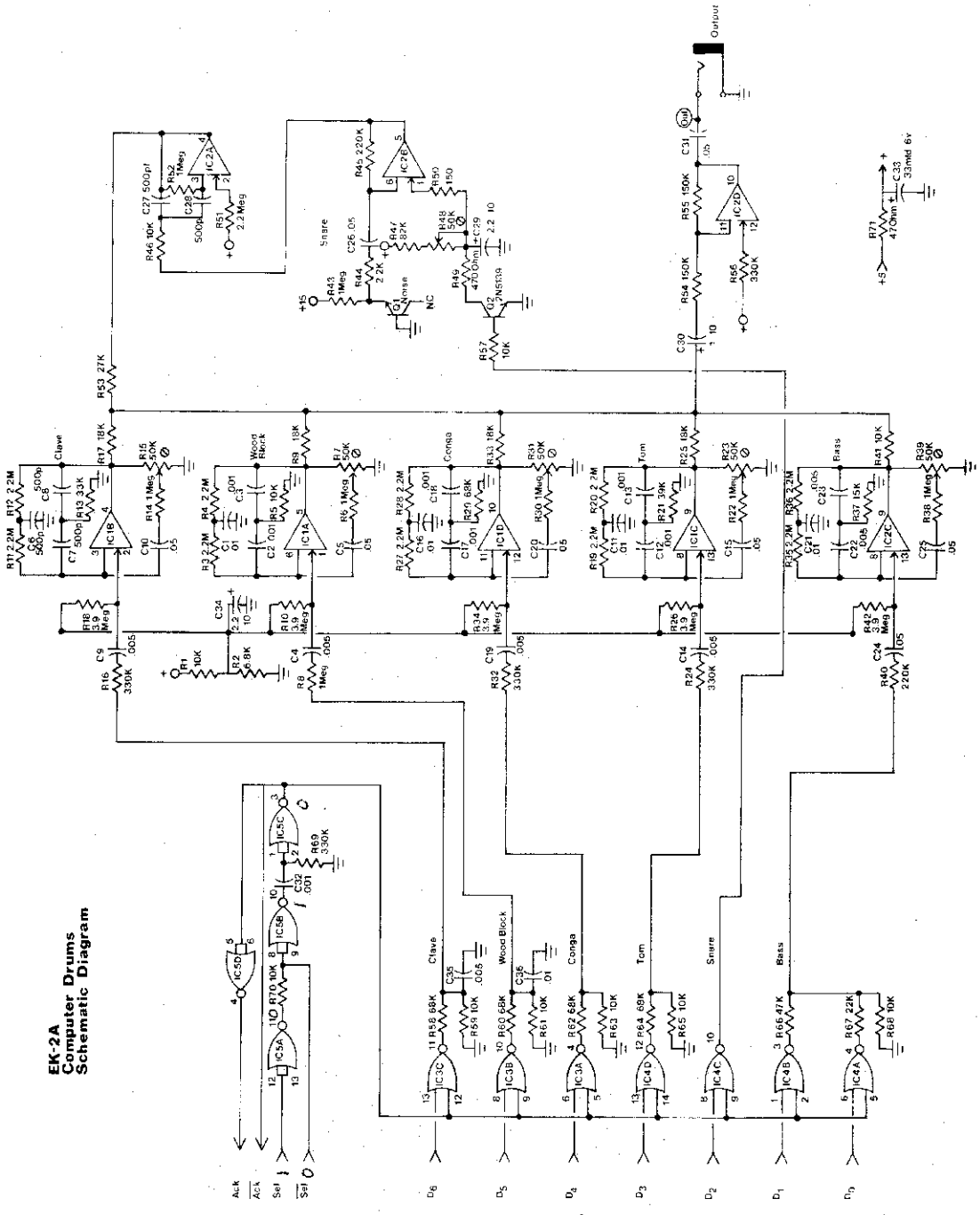
() Install the I/O connector socket.

THIS COMPLETES ASSEMBLY OF THE EK-2A COMPUTER DRUM KIT

I/O Connector



*NOTE: Adjust PC mount trim pots in direction of arrow printed on board until sustained oscillation occurs, then back off until oscillation just quits.



EK-2A
Computer Drums
Schematic Diagram

DRUMSYS 0.6
DEVELOPMENTAL VERSION
USER'S NOTES

Load DRUMSYS 0.6 into the PAIA 8700 Computer/Controller using the following entry sequence:

0-0-0-0-0-0-F-F-0-1-1-1-TAPE

When the program has loaded successfully the display will show "AA". There are at least two copies of the program on the tape. If for any reason the first one will not load, try the second.

When properly loaded, begin the program running form the starting location $0000 using this sequence:

0-0-0-0-RUN

With the program running, the control keys of the 8700 take on different meanings than that assigned by the PIEBUG Monitor, as outlined below:

DRUM SOUNDS ARE CONTROLLED BY THE KEYS 0-7

KEY#	DRUM SOUND
0	REST (NO DRUM)
1	LIGHT BASS
2	HEAVY BASS
3	SNARE
4	TOM-TOM
5	CONGA
6	WOOD BLOCK
7	CLAVE

Pressing any of the drum sound keys (0-7) causes termination of the current "mode" of operation and reversion to the "DRUM ENTRY" mode. Note that while touching drum sound keys, the corresponding drum sound is produced by the EK-2A and the displays count in hexadecimal. The number shown in the display is the "event number" of the drum sound produced.

In this version of the program, any of the keys 8-F cause the system to be reinitialized. Any score saved in memory when one of these keys is touched will be erased.

The various modes of operation for Drumsys are activated by touching one of the two rows of keys on the 8700 keyboard. Mode names and the corresponding keys that select them are as follows:

KEY NAME	MODE SELECTED
RUN	PLAY
DISP	SET TEMPO
BACK	BACK SPACE
ENTER	STOP/STEP
PCH	CONTINUE
PCL	DUMP SCORE
TAPE	LOAD SCORE
REL	STROBE DRUM

The actions produced by these various modes of operation are as follows:

PLAY - causes the drum score currently in memory to be played at the current tempo rate. Always starts at the beginning of the score (EVENT #0).

SET TEMPO - changes tempo value. When touched, this key causes a counter which will be the tempo value to begin counting. Counting is terminated by touching any other control key. Typically, this control would be used by touching first "TEMPO SET" then "PLAY". The time between touching these two keys is the time between events during playback.

BACK SPACE - causes the program to step through the current score backwards, for editing purposes. In all cases it is important to note that the number shown in the 8700's displays is the event number of the sound just produced.

STOP/STEP - when touched, produces a single step mode of operation. Using the BACK SPACE and STOP/STEP keys allows editing of individual drum sounds. Typical use would be to "STEP" through the score until the drum sound to be replaced (as indicated by sound and event number) is reached. At this point, touching the "BACK" key causes the same drum to sound again (Note that since this is the same "EVENT" as when stepping forward, the displays will not change). The old drum sound may now be replaced with the new simply by touching the proper drum sound key.

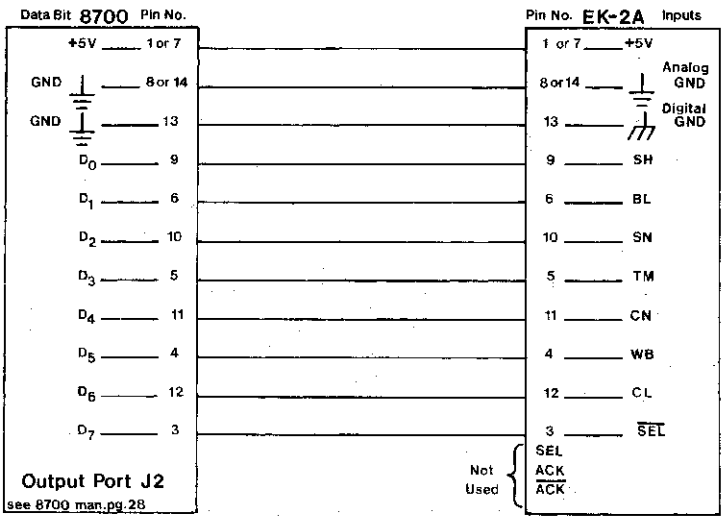
CONTINUE - very similar to the "PLAY" key except that the score will pick up from the event currently in the displays.

DRUM SCORE - this command key allows scores in memory to be saved on cassette tapes. When touched, there will be a couple of seconds of apparent inactivity followed by a counting of the displays as the score is transferred from computer memory to tape. Before touching the "DUMP" key, place your recorder in the record mode and allow it to run for several seconds to get beyond the sub-standard tape sections typically found at the beginning of tape cassettes. Note that relays for tape motion are controlled as outlined in the CS-87 manual.

LOAD SCORE - similar to DUMP SCORE except that the memory of the computer is loaded from the cassette tape. Make sure that there is a cassette to be loaded before touching this control as the computer will wait for data transfer completion before continuing with any further action. If this contro is inadvertently touched, you may recover by pressing the reset key and running the program again. There is a soft start location location of sorts at $014 which can often be used to start the program running without destroying the saved score.

STROBE DRUM - this special effect causes the score to be played at the current tempo rate, but essentially strikes each drum event many times rather than just once. The result is a very unusual bass instrument sounding voice.

EK-2A / 8700 Wiring Connections
For use with Drumsys Program



NOTES

```
0010  :
0020  :
0030  :************************************
0040  :*                                  *
0050  :*           DRUMSYS 0.6            *
0060  :*                                  *
0070  :*    8700/EK-2 DRUM OPERATING      *
0080  :*              SYSTEM              *
0090  :*                BY                *
0100  :*           JOHN SIMONTON          *
0110  :*                                  *
0120  :*(C) 1978 - PAIA ELECTRONICS, INC*
0130  :*                                  *
0140  :************************************
0150  :
0160  BUFF .DL 00F0
0170  CNTR .DL 00EC
0180  EXP  .DL 00EB
0190  TMPO .DL 00EA
0200  PNTR .DL 00E9
0210  DSP  .DL 0820
0220  DECD .DL FF00
0230  BEEP .DL 0F22
0240  SCOR .DL 0100
0250  DUMY .DL 0086
0260  OUTP .DL 0840
0270  SNBT .DL 0E25
0280  CASS .DL 0EAA
0290  :--------------------
0300  :00E8
0310  :    S-TABLE (CONTROL CODES)
0320  :00E1
0330  :--------------------
0340  STBL .DL 00D1
0350  :--------------------
0360  :00E0
0370  :    DRUM SIGNATURES
0380  :00D9
0390  :--------------------
0400  :************************************
0410        .OR 10D1
0420  :************************************
0430  :
0440  TAPE .HS FF00800100010001
0450  DSIG .HS FFFEFDF3F7EFDFBF
0460  CRL  .HS 6A9D8D877CB2B9CD000820
0470  :
0480        .OR 10ED
0490  PARM .HS FF
0500        .OR 10F6
0510  PAR1 .HS F6F7F8F9FAFBFCFDFF00
```

```
                        0520  :**********************************
                        0530        OR 1000
                        0540  :**********************************
                        0550  :
1000-   A9 86           0560  SPHK LDA 86       :SPARE HOOK KEYS 8-F
1002-   85 39           0570       STA *ACTN+01:USED ONLY TO RE-START
1004-   EA              0580       NOP         :SYSTEM.   IN LATER VERSIONS
1005-   EA              0590       NOP         :WILL PROVIDE ADDITTIONAL
                        0600  :                 FEATURES
                        0610  :
1006-   A9 00           0620  STAR LDA 0        :PREPARE ACCUMULATOR AND
1008-   85 E9           0630       STA *PNTR   :ZERO SCORE POINTER
100A-   8D 20 08        0640       STA DSP     :AND DISPLAYS
100D-   AA              0650       TAX         :PREPARE X REG AS POINTER
100E-   9D 00 01        0660  SLP0 STA SCOR,X  :AND USE IT TO CLEAR SCORE
1011-   E8              0670       INX
1012-   D0 FA           0680       BNE SLP0    :LOOP UNTIL DONE
1014-   20 53 10        0690  SLP1 JSR RDKY     :GO READ THE KEYBOARD, ETC.
1017-   B0 1F           0700       BCS ACTN    :AND IF NO NEW KEYS, BRANCH
1019-   C9 10           0710  TSTS CMP 10       :NEW KEY - A "CONTROL" KEY?
101B-   B0 16           0720       BCS CTRL    :YES - BRANCH TO CONTROL
101D-   C9 08           0730       CMP 08      :ONE OF "SPARE" KEYS?
101F-   B0 DF           0740       BCS SPHK    :YES- BRANCH
                        0750  :         **
1021-   A9 86           0760  NTRY LDA 86       :DRUM ENTRY MODE, GET LINK
1023-   85 39           0770       STA *ACTN+01:SET LINK
1025-   B9 D9 10        0780       LDA DSIG,Y  :GET DRUM SIGNATURE
1028-   A6 E9           0790       LDX *PNTR   :GET SCORE POINTER
102A-   9D 00 01        0800       STA SCOR,X  :SAVE DRUM SIG IN SCORE
102D-   20 3E 10        0810       JSR PLAY    :PLAY THE DRUM BEAT
1030-   4C 14 10        0820       JMP SLP1    :LOOP FOR MORE
1033-   B9 D1 00        0830  CTRL LDA STBL,Y   :GET COMMAND ADDRESS LINK
1036-   85 39           0840       STA *ACTN+01:AND SET LINK IN JSR DUMY
1038-   20 86 00        0850  ACTN JSR DUMY     :AND GO TO COMMAND SUBROUTINE
103B-   4C 14 10        0860       JMP SLP1    :THEN LOOP FOR MORE
                        0870  :
                        0880  :PLAY SUBROUTINE
                        0890  :
103E-   A4 EB           0900  PLAY LDY *EXP     :GET EXPRESSION VARIABLE
1040-   8D 40 08        0910       STA OUTP    :OUTPUT CONTROL TO EK-2
1043-   29 7F           0920       AND 7F      :RESET STROBE BIT
1045-   88              0930  PLA0 DEY          :DELAY FOR THE EXP.  TIME
1046-   D0 FD           0940       BNE PLA0    :LOOP UNTIL DONE
1048-   8D 40 08        0950       STA OUTP    :AND TURN DRUM "OFF"
104B-   E6 E9           0960       INC *PNTR   :INCREMENT SCORE POINTER
104D-   A6 E9           0970       LDX *PNTR   :PLACE IN X REGISTER
104F-   8E 20 08        0980       STX DSP     :AND SHOW IN DISPLAYS
1052-   60              0990       RTS         :THEN RETURN
                        1000  :
                        1010  :READ KEY-ALSO IMPORTANT TO TEMPO
                        1020  :
1053-   20 00 FF        1030  RDKY JSR DECD     :PIEBUG KEYBOARD SUBROUTINE
1056-   B0 05           1040       BCS DLY     :SAME KEY - JUST DELAY
1058-   A2 00           1050       LDX 0
105A-   86 EC           1060       STX *CNTR   :ZERO TEMPO COUNTER
105C-   60              1070       RTS
105D-   A2 20           1080  DLY  LDX 20       :SET X AND Y REGISTER
105F-   A0 3F           1090  NXTX LDY 3F       :DELAY PARAMETERS
1061-   88              1100  DELY DEY          :AND DO DELAY.
1062-   D0 FD           1110       BNE DELY
1064-   CA              1120       DEX
```

```
1065-   D0 F8     1130        BNE NXTX     :LOOP UNTIL DONE
1067-   E6 EC     1140        INC *CNTR    :INCREMENT TEMPO COUNTER
1069-   60        1150        RTS          :AND RETURN
                  1160  :
                  1170  :RUN SUBROUTINE
                  1180  :
                  1190  :         **
106A-   A9 7C     1200  RUN   LDA 7C       :COMMAND LINK TO "WAIT"
106C-   85 39     1210        STA *ACTN+01:SET COMMAND LINK
106E-   A9 00     1220  CYCL  LDA 00       :PREPARE AND SET
1070-   85 E9     1230        STA *PNTR    :SCORE POINTER TO 0
1072-   A6 E9     1240  CONT  LDX *PNTR    :GET CURRENT SCORE POINTER
1074-   BD 00 01  1250        LDA SCOR,X   :GET CURRENT DRUM SIGNATURE
1077-   F0 F5     1260        BEQ CYCL     :ZERO, END OF SCORE-BRANCH
1079-   20 3E 10  1270        JSR PLAY     :GO PLAY DRUM SOUND, ETC.
107C-   A5 EC     1280  WAIT  LDA *CNTR    :GET TEMPO COUNTER AND
107E-   45 EA     1290        EOR *TMPO    :COMPARE TO TEMPO VARIABLE
1080-   D0 04     1300        BNE RETN     :IF NOT TIMED OUT, RETURN
1082-   85 EC     1310  0CNT  STA *CNTR    :TIMED OUT - ZERO COUNTER
1084-   F0 EC     1320        BEQ CONT     :BRANCH ALWAYS TO PLAY, ETC.
1086-   60        1330  RETN  RTS          :RETURN
                  1340  :
                  1350  :SINGLE STEP SUBROUTINE
                  1360  :
                  1370  :         **
1087-   A9 86     1380  STEP  LDA 86       :COMMAND LINK TO "RETN"
1089-   85 39     1390        STA *ACTN+01:SET COMMAND LINK
108B-   D0 E5     1400        BNE CONT     :BRANCH ALWAYS TO PLAY, ETC.
                  1410  :
                  1420  :BACKSPACE SUBROUTINE
                  1430  :
                  1440  :         **
108D-   A9 96     1450  BACK  LDA 96       :COMMAND LINK TO "NEXT"
108F-   85 39     1460        STA *ACTN+01:SET COMMAND LINK
1091-   C6 E9     1470        DEC *PNTR    :SCORE POINTER BACK ONE
1093-   D0 DD     1480        BNE CONT     :GO PLAY SCORE, ETC.
1095-   60        1490        RTS          :AND RETURN
                  1500  :         **
1096-   A9 86     1510  NEXT  LDA 86       :COMMAND LINK TO "RETN"
1098-   85 39     1520        STA *ACTN+01:SET COMMAND LINK
109A-   C6 E9     1530        DEC *PNTR    :SCORE POINTER BACK ONE
109C-   60        1540        RTS          :RETURN
                  1550  :
                  1560  :TEMPO
                  1570  :
                  1580  :         **
109D-   A9 A5     1590  TMP   LDA 0A5      :COMMAND LINK TO "NXT2"
109F-   85 39     1600        STA *ACTN+01:SET COMMAND LINK
10A1-   A9 00     1610        LDA 00       :INITIALIZE TEMPO COUNTER
10A3-   85 EA     1620        STA *TMPO    :AND START COUNTIN
10A5-   E6 EA     1630  NXT2  INC *TMPO    :UNTIL NEXT COMMAND
10A7-   60        1640        RTS          :RETURN
                  1650  :
                  1660  :SET UP FOR TAPE TRANSFER
                  1670  :
10A8-   A2 07     1680  STTP  LDX 07       :TRANSFER SEVEN BYTES
10AA-   B5 D1     1690  STP   LDA *TAPE,X :GET PARAMETER
10AC-   95 F0     1700        STA *BUFF,X :PLACE PARAMETER
10AE-   CA        1710        DEX          :POINT TO NEXT
10AF-   D0 F9     1720        BNE STP      :LOOP UNTIL ALL TRANSFERED
```

```
10B1-   60          1730        RTS          :THEN RETURN
                    1740  :
                    1750  :TAPE IN AND OUT ROUTINES
                    1760  :
10B2-   20 A8 10    1770  TOUT  JSR STTP     :SET UP PARAMETERS
10B5-   A9 DD       1780        LDA 0DD      :SET DUMP "SWITCH"
10B7-   D0 05       1790        BNE DO       :BRANCH ALWAYS
10B9-   20 A8 10    1800  TIN   JSR STTP     :SET UP PARAMETERS
10BC-   A9 11       1810        LDA 11       :SET LOAD "SWITCH"
10BE-   20 25 0E    1820  DO    JSR SNBT     :TURN ON RELAYS
10C1-   20 AA 0E    1830        JSR CASS     :DO CASSETTE ROUTINE
                    1840  :         **
10C4-   A9 86       1850        LDA 86       :COMMAND LINK TO "RETN"
10C6-   85 39       1860        STA *ACTN+01:SET LINK
10C8-   18          1870        CLC          :PREPARE FOR BEEP
10C9-   20 22 0F    1880        JSR BEEP     :TURN OFF RELAYS AND BEEP
10CC-   60          1890        RTS          :AND RETURN
                    1900  :
                    1910  :STROBE DRUM EFFECT
                    1920  :
10CD-   C6 E9       1930  STRB  DEC *PNTR    :PREPARE TO GET SAME DRUM
10CF-   4C 72 10    1940        JMP CONT     :PLAY DRUM
                    1950  :
                    1960  :
                    1970  :
                    1980  :
                    1990  END   .EN
```

NOTES