

Registered for posting as
a publication. Category B

MAR '81

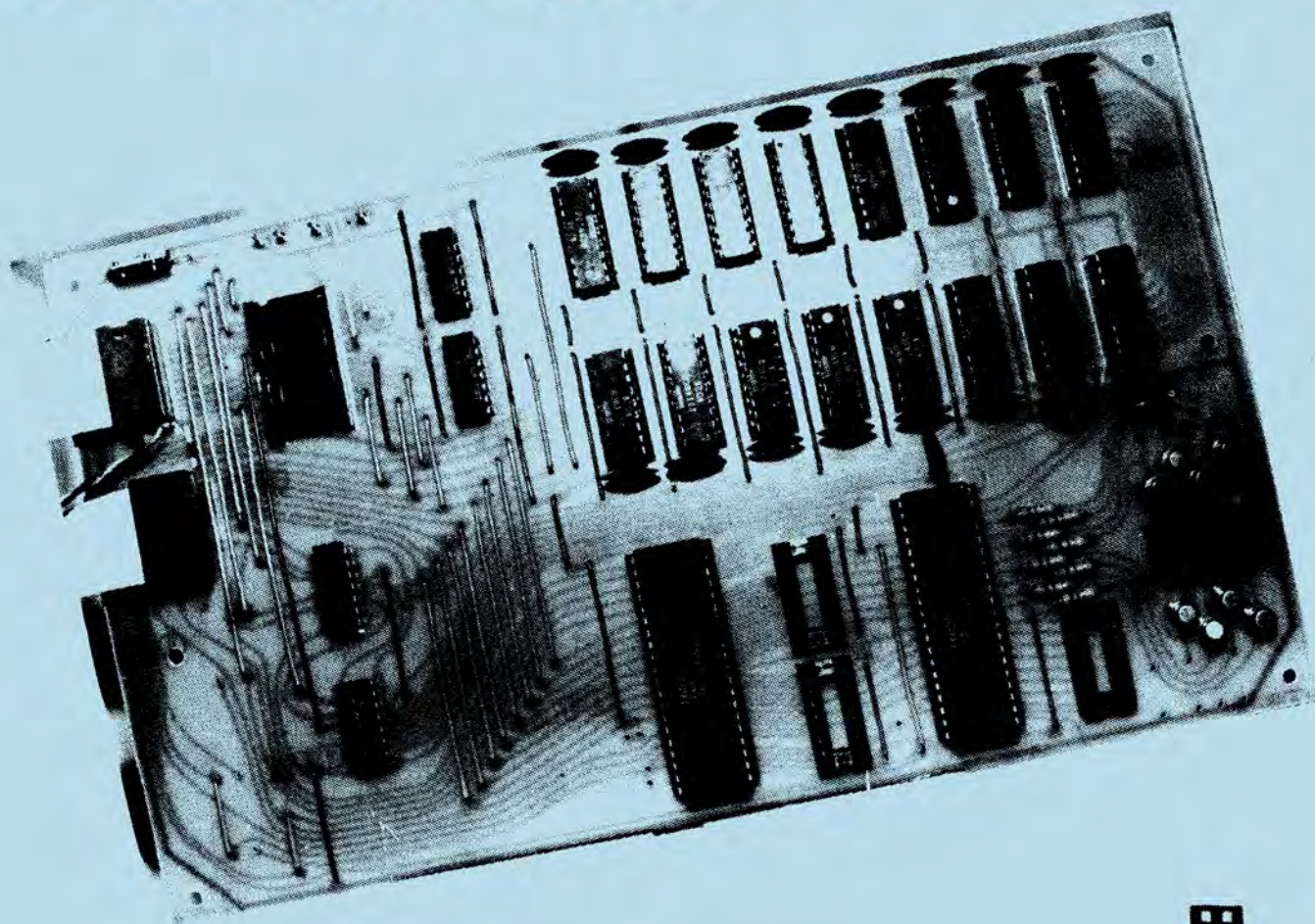
'DREAM 6800'
NSW 6800 USERS
GROUP
G. SAMWAYS
G. NELSON

⌘ ⌘ ⌘ ⌘
⌘ ⌘
9 ⌘ ⌘ ⌘

⌘ ⌘
⌘ ⌘
⌘

.⌘

9 . .
06 1270 0 ⌘



⌘
⌘
⌘

04 0930 0 ⌘

D. R. E. A. M. EXPANSION KIT

DESIGNED ESPECIALLY FOR THE DREAM 6800 AND 6802 !!!

The printed circuit board in the kit has provision for:

* 8K RAM * TWO PIA's * ONE EPROM (2708 or 2716) * ADDRESS BUFFERS *
* SELECT LOGIC * DRIVE TRANSISTORS FOR OFF CARD OPTO-COUPERS *

4K EXPANSION KIT : \$99-00.

Consists of; DREAM sized fibreglass P.C.B.; 4K RAM ; All sockets ;
Address Buffers ; Select Logic ; Connectors and Instructions.

The 1K on the DREAM board is transferred to this board, making 5K in total,
expandable to 8K. The EPROM, if used, connects to one of the RAM addresses.
A "fully populated" board draws less than 2 AMPS. The P.C.B. is not sold
separately.

THE COMPLETE KIT, LESS RAM, IS AVAILABLE FOR \$59-00 IF RAM IS NOT REQUIRED.

3 AMP POWER SUPPLY KIT : \$45-00. Now available separately. (Post FREE if
you mention this advertisement.)

Post, Packing and Insurance : \$3-00 on all orders. Phone Cash-on-Delivery
orders accepted. C.O.D. \$2-00 extra. Phone for details of Sydney counter sales.

Orders subject to a 7 day Money Back Guarantee. Use separate order coupon
in Jan & Feb newsletters so you don't have to mutilate your copy. Available
only from:

J. R. COMPONENTS PTY LTD

P.O. BOX 128, EASTWOOD. N.S.W. 2122. TELEPHONE (02) 853385.

We have to start this issue off with an apology, as the contents are not exactly as we said last month they would be.

We have had to hold Michael Bauers 'SOUND EFFECTS GENERATOR' over, for two reasons. 1) We just ran out of space this issue, and 2) We haven't finished building ours yet, as we had a problem finding a supplier for the SN76477 chip used. We think we have that sorted out now though, and the article should appear next month. (Touch wood!) For those of you who wrote asking did you HAVE to have a J.R. Expansion board to run the S.E.G., don't panic, the P.C.B. design for the project includes provision for the extra P.I.A. you need to run it, so you can use it even with a basic DREAM.

The other change is that the PING PONG game is not in this issue, as someone, (who shall remain nameless, but his initials are G.N.), typed up the wrong program. PING PONG will definately be in next month.

COMPETITION. - YES, WE HAVE A WINNER. Mr. K. BOLCH, [REDACTED], sent in a game which we judged to be the best, (and the most fun to play,) from several others we received. It is called RESCUE MISSION, and will appear next month. Congratulations Mr. Bolch, a voucher for a SIX MONTH SUBSCRIPTION is on its way to you. The other programs will all appear in the DREAMER over the next few months.

RADIO AMATEURS - We have not received any new names to add to the list which appeared in the December issue, so we plan to drop this segment. Should the demand return, we will print an updated list at irregular intervals.

MUSICAL DREAMS? - To keep you busy until the Sound Effects Generator appears, have a look at the 'OPUS' article from Frank Rees in this issue. It actually makes the DREAM PLAY MUSIC.

NEXT MONTH - We will have,

- Four new programs, PING PONG (I hope), TORTOISE (A racing car game), SIMPLE SIMON (Test your memory against the computer's), and RESCUE MISSION (Our competition winner)
- The next instalment of 'How to Use Chip-8.'
- SOUND EFFECTS GENERATOR, by Michael J. Bauer.
- Some more songs to play using Frank Rees' OPUS program.

Until next month,

Happy DREAMing,

GARRY NELSON AND GRAEME SAMWAYS,

N.S.W. 6800 USERS GROUP,
[REDACTED]

WANTED

Here is a selection of things that people have requested appear in the DREAMER. If you would like to try your hand at writing a program, or an article, but can not think of a subject, why not try one of the following?

- A CHESS program.
- A FLIGHT SIMULATOR game.
- DRAUGHTS
- Interfacing the DREAM to external devices.

This subroutine displays the contents of all sixteen variables, the Index Pointer, (I) and the location of the next instruction to be executed on the screen. When any key is pressed, the screen is erased, and the program recommences. All variables are saved, as well as the Index Pointer. The only thing that is lost is the display. This is erased.

Next month we will include a version which stores the display, then recalls it, but this will be for 2K (or more) DREAMs, due to the excessive RAM needed.

8 BUG is run as a CHIP-8 subroutine. Its intended use is as an aid when programming. If you want to see what effect a certain instruction has on the program, you just insert a 'Go to Chip-8 Subroutine' instruction, (2MMM) at the appropriate place, and run the program. By doing this at different points, you can compare the effect that the instruction has had on the variables etc.

There are three Machine Code subroutines contained in 8 BUG. The first gets the variables and I, and puts them in a 'safe place'. The second shifts variable 0 four bits right, and stores the value in 1. The third returns the variables and I from the 'safe place' to their original position.

Apart from these subs the program is fully Chip-8. We have expanded it into a fully commented teaching program so you can follow it through and see how it works.

SECTION	ADDR.	INSTR.	EXPLANATION.
Main Line	0300	0371087	Do M.C. Sub. at 0371. (Save)
(Chip-8)	0302	6C00	<u>C</u> = 00 (Count = 00)
	0304	6BFA	<u>E</u> = FA) <u>B</u> = 00. Start,
New line	0306	7B06	<u>B</u> = <u>B</u> + 6) increment 6 each line
	0308	6A00	<u>A</u> = 00
Next	030A	FC29	Set Index pointer (I) to show <u>C</u>
	030C	234C284	Go to Sub at 034C (Display)
	030E	A395AB95	I = 0395 (:)
	0310	DAB4	Display (:)
	0312	7A02	<u>A</u> = <u>A</u> + 02
	0314	23542854	Go to Sub at 0354 (Display Byte)
	0316	7A03	<u>A</u> = <u>A</u> + 03
	0318	4C04	If <u>C</u> ≠ 4, skip next instruction
	031A	13061806	Go to New line
	031C	4C08	If <u>C</u> ≠ 8, skip next instruction
	031E	13061806	Go to New line
	0320	4C0C	If <u>C</u> ≠ C, skip next instruction
	0322	13061806	Go to New line
	0324	3C10	If C = 10, Skip next instruction
	0326	130A180A	Go to Next
	0328	7B06	<u>B</u> = <u>B</u> + 06
	032A	6A00	<u>A</u> = 00
	032C	A39A189A	I = 039A (I:)
	032E	DAB5	Display (I:)
	0330	7A06	<u>A</u> = <u>A</u> + 06
	0332	23521852	Do Sub at 0352 (Display I)
	0334	7A02	<u>A</u> = <u>A</u> + 02
	0336	A39F189F	I = 039F (8 B)
	0338	234C184C	Do Sub at 034C
	033A	A3A418A4	I = 03A4 (UG)
	033C	234A184A	Do Sub at 034A (Message)
	033E	A3A918A9	I = 03A9 (P:)
	0340	234A184A	Do Sub at 034A (Message)
	0342	7A05	<u>A</u> = <u>A</u> + 05

SECTION	ADDR.	INSTR.	EXPLANATION.
	0344	2352 2352	Do Sub at 0352 (Display I)
	0346	F00A	Get key
	0348	13C4 13C4	Go to 03C4
Message	034A	7A04	$A = A + 04$
Display	034C	DAB5	Display digit or message 5 lines high
	034E	7A04	$A = A + 04$
	0350	00EE	Return
Display I	0352	2354 2354	Do Sub at 0354 (Next line, so does Disp.Byte twice)
Display	0354	A3B0 A3B0	I = 03B0 (Safe place)
Byte	0356	FC1E	$I = I + C$
	0358	F065	Load 0 with contents of Byte pointed to by I
	035A	0368 0368	Do M.C. Sub at 0368 (Shift)
	035C	F129	Set I to show least significant digit <u>1</u>
	035E	234C 234C	Do Sub at 034C (Display)
	0360	F029	Set I to show l.s.d. <u>0</u>
	0362	234C 234C	Do Sub at 034C (Display)
	0364	7C01	$C = C + 01$
	0366	00EE	Return
Shift	0368	96 30	LDAA Load Accumulator A from 30 (<u>0</u>)
(M.C.)	036A	44	LSRA)
	036B	44	LSRA)
	036C	44	LSRA) Shift right 4 bits
	036D	44	LSRA)
	036E	97 31	STAA Store shifted value in 31 (<u>1</u>)
	0370	39	RTS Return
Save	0371	DE 26	LDX Load X from 0026
(M.C.)	0373	08C0 FF 03 C0	STX Store X at 03C0
	0376	08C0 CE 03 B0	LDX X = 03B0
	0379	DF 26	STX Store X at 0026
	037B	86 0F	LDAA Accumulator A = 0F
	037D	97 2B	STAA Store Accumulator A at 002B (X)
	037F	BD C1 FA	JSR Do Sub at C1FA
	0382	30	TSX Stack pointer + 01 \rightarrow X
	0383	9E 24	LDS Load Stack pointer from 0024
	0385	32	PULA Acc. A loaded from S.p. S.p. = S.p. + 1.
	0386	08C2 B7 03 C2	STAA Store Accumulator A at 03C2
	0389	32	PULA Acc. A loaded from S.p. S.p. = S.p. + 1.
	038A	08C3 B7 03 C3	STAA Store Accumulator A at 03C3
	038D	34	DES Stack pointer = S.p. - 1.
	038E	34	DES Stack pointer = S.p. - 1.
	038F	9F 24	STS Stack pointer stored at 0024
	0391	35	TXS X - 1 \rightarrow S.p.
	0392	7E C0 79	JMP Go to C079 (Erase)
Data	0395	00 80 00 80 00 E0 48 40 48 E0 E6	(Data for :, I:, 8 B, UG, P:.)
	03A0	A5 E6 A5 E6 57 54 54 55 77 1C 15 1C 11 10 00 00	
Safe place	03B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	(Saved variables)
	03C0	00 00 00 00	(Saved I and Next Instruction)
Finish	03C4	03C8 03C8	Do M.C. Sub at 03C8 (Finish)
	03C6	00EE	Return
Restore	03C8	08C0 CE 03 B0	LDX X = 03B0
(M.C.)	03CB	DF 26	STX Store X at 0026
	03CD	86 0F	LDAA Accumulator A = 0F
	03CF	97 2B	STAA Store Acc. A at 002B
	03D1	BD C2 04	JSR Do Sub at C204

SECTION	ADDR.	INSTR.	EXPLANATION.
	03D4	03 C0 LDX	Load X from 03C0
	03D7	DF 26 STX	Store X at 0026
	03D9	7E C0 79 JMP	Go to C079 (Erase)
	03DC	0000	
	03DE	****	

NOTES : 0379 to 0380 actually does FF65.
 0382 to 0394 detects position of Next to be executed after routine
 03C8 to 03D3 restores variables
 03D4 to 03D8 restores I

ADVERTISING

If you would like some help, can offer some help, have something to sell, or would like to buy something, send it in to us with a fee of \$1-00, and we will print it in two newsletters. THIS OFFER ONLY APPLIES TO PRIVATE ADVERTISERS, and we would ask you to keep them reasonably short, something like the ones below. Commercial enterprises who wish to advertise in the DREAMER are invited to contact us for details of rates etc.

+++++

DREAMER would like to contact other learner beginners with broad Micro-computer interest, to share information, routines and programs. If you would like some useful routines, send S.S.A.E. to FRANK REES, [REDACTED]
 [REDACTED] (See Frank's articles in this and last months issues - Garry)

+++++

AUTOMATIC RELOCATION OF CHIP - 8 INSTRUCTIONS when you perform block moves using EDIT - 8. Also five line display of double-byte memory, scrolling forwards and backwards, zeroing blocks, double-byte memory modifications, branch calculations. Program resides at either 0400 - 07A0 or 0C00 - 0FA0. Ten Dollars for cassette, instructions and source code, from, JAY MANN, [REDACTED]

+++++

SERVICE FOR DREAM SYSTEMS.

FRED LEVER, (SNR), is happy to discuss problems after hours on [REDACTED] or send letter with S.S.A.E., tale of woe and symptoms to [REDACTED]
 [REDACTED]

If repair is required on DREAM send (P.C.B. only preferred) to above address. With faulty expansion boards, send BOTH P.C.B.'s, as the problem may be on the main board. A fee of \$33-00 will cover post and normal repairs, exceptional repairs may cost more, but the owner will be advised BEFORE proceeding.

If you have a DREAM or EXPANSION KIT partly built, I will complete it for you, price will be quoted on individual jobs.

This month I will deal with the remaining instructions, with the exception of those requiring the index pointer. Don't forget to write and tell us what facets of programming you would like to see discussed after we finish all the instructions.

GET KEY, STORE IN X (FX0A)

This instruction waits for you to push a key, then it stores the value of that key in the variable specified by X.

i.e., you push A, it stores 0A in X

you push 0, it stores 00 in X

Also, if you push (FN), it stores 8C in X. This can be useful if you wish to use all 16 Hex keys, and use FN to escape, etc.

No variables are affected except X. (If you wish to input 2 digit values, refer to the Machine Code subroutine in issue No.1.)

TONE. (FX18)

This instruction outputs a 1200 Hz tone from the speaker for a duration calculated from a value which has been previously stored in the variable specified by X. The duration of the tone is 20 milliseconds multiplied by the value in variable X. To calculate how long you want the tone on, either calculate how many 20 mSec periods are in the time you want, (50 dec. = 1 Sec, etc.) then convert this value to Hexadecimal (See M.J.B.'s "Chipos Manual" for a Hexadecimal to Decimal converter,) or just try some different numbers, hit or miss. (The tone produced with each key stroke is 80 mSec long.) The maximum duration is approximately five seconds.

e.g. To produce a tone from 00 to 300 mSec, as specified by inputting a key, then redo, with value stored in 8.

0200	F80A	Get key and store in <u>8</u> .
0202	F818	Bleep for 20ms X <u>8</u> .
0204	1200	Go to 0200. (Redo)

The value stored in the variable used is not lost in the tone instruction, so it can still be used for other things later in the program.

ERASE SCREEN (00E0)

This instruction erases the screen. No variables are affected.

NO OP (0000)

This instruction is just like a space. It does nothing in the program. You can put them in to remove an instruction, or to leave room for instructions you may need later on, (a good idea when programming,) or to make the length of time in a clock the same as for another branch of the program.

STOP (F000)

This instruction is the same as hitting the RST key. I use it when programming to find out when the program enters a particular part of a program, i.e., it returns to the monitor.

CALL MACHINE CODE SUBROUTINE (OMM)

This instruction sends the program to a user written subroutine at the specified location, (OMM) and returns when a 39(hex) is encountered. (See issues 1 and 2.)

TIMER INSTRUCTIONS (FX15) and (FX07)

The timer function can be used to create delays etc in the following way. Firstly you must load the time duration into a variable, then store this

HOW TO USE CHIP - 8. (Part 5) Cont.

in the timer. (FX15) This value is automatically decreased by 01 every 20 mSec. NOTE: When it reaches 00, it is then decreased to FF. It DOES NOT STOP at 00, it keeps on counting down, so I recommend you don't set the time at 00 or 01 as instead of 00 or 20 mSec wait you may get five seconds wait.

After the timer has been set up you can read the current value in the timer with FX07, and store it in X. (X does not have to be the same variable as the X you used in the FX15. Remember that the timer value is continuously being decreased.

To program time delays, you

First,	Set a value	6XKK
Second,	Store in timer	FX15
Then,	Read timer value	FX07
	Test if 00	3X00
If not,	Jump back and get timer value	1MMM

This way you go into a loop until the timer reaches 00. (It doesn't have to be 00.)

If you have a long delay, you can go away and do other things, as long as they don't take too long. e.g., display the time each second.

EXAMPLE: Get key, bleep for this time, wait the sametime, go back and bleep again. If 'F' is down, go back to get a new key.

This program will give you an idea of what times to use as an alarm type tone in clocks and other programs.

0200	F10A	Get key, store in <u>1</u>
0202	F118	Bleep for 20mSec X value of <u>1</u>
0204	F115	Set timer to value of <u>1</u>
0206	F007	Get timer value, store in <u>0</u>
0208	3000	Skip next instruction if <u>0</u> = 00
020A	1206	↪ <u>0</u> ≠ 00, Go to 0206, get timer value again.
020C	6B0F	↪ <u>0</u> = 00, <u>B</u> = 0F
020E	EB9E	Skip next instruction if key = <u>B</u>
0210	1202	Go to 0202 (Tone)
0212	FB18	Bleep on exit. (Bleep for 20mSec X value of <u>B</u>
0214	1200	Go to 0200 (Get key)

When you push 0, you will most likely not hear anything, due to the short time involved.

You may notice that it is sometimes hard to Escape from the loop after 0 has been pressed. This is due to the timer missing 00 the first time through and having to wait another five seconds. (FF X 20 mSec.)

Sometimes the same thing happens with the tone (i.e. a very long bleep).

You do not have to wait until the timer reaches 00.

E.G.

0200	6A20	<u>A</u> = 20
0202	FA15	Set timer to value stored in <u>A</u>
0204	FA07	Load <u>A</u> with timer value
0206	3A10	If <u>A</u> (timer value) ≠ 10, skip
0208	1204	Go to 0204 (Get timer value)
020A	FA18	Bleep 20mSec X value of <u>A</u>
020C	1200	Go to 0200

Graeme V. Samways.

FRANK REES,

One of the best and most reasonably priced computer books I have found is RADIO SHACK TRS-80 ASSEMBLY LANGUAGE PROGRAMMING at \$3-95 from Tandy. The following program is a translation to 6800 of music 'Opus 1', using only the abovementioned book and Bishop's BASIC MICROPROCESSORS AND THE 6800. As much as was possible it is a word for word translation. The main point of interest is the use of only three registers, A, B, and Index, where five were used in the original. In locations 0203-4, 'A' is loaded with duration and stored in memory 0233. This move has replaced 'C' register with a memory location, and 'C' does not have to be loaded into 'A' for '0' test. The 6800 uses only one more byte, the reserve memory byte to replace the 'C' register in original. This can be compensated for by changing 0230, now 7Exxxx (Jump to absolute address) to 26xx (Branch to relative address.)

Connect an audio amplifier to PB1, (Pin 7 of the PIAB input/output socket), via a 0.1 uF capacitor to protect the computer from shorts. An old pocket radio will do, with the output from the DREAM to the top end of the volume control, and 0v from the computer to GND. (Negative side of battery.) Key in the following program, and run 0200, FN, 3. (NOT C000, FN, 3.)

```

0200  CE 0234  START  LDX TABLE  Start of music table
0203  A6 00    CONT 1  LDA A 0,X  Duration
0205  B7 0233          STA A      Store duration count
0208  8A 00          ORA          Test for 00 (Terminator)
020A  26 02          BNE
020C  20 FE    LOOP  BRA LOOP    Loop here on done
020E  E6 01    CONT 2  LDA B 1,X  Get delay count
0210  86 02          LDA A $02
0212  B7 8012          STA A PIAB Turn on ½ cycle
0215  5A          LOOP 1  DEC B
0216  26 FD          BNE LOOP 1 Delay for frequency
0218  E6 01          LDA B 1,X  Get delay again
021A  86 00          LDA A $00
021C  B7 8012          STA A PIAB Turn on other ½ cycle
021F  5A          LOOP 2  DEC B
0220  26 FD          BNE LOOP 2 Delay for frequency
0222  7A 0233  LOOP 3  DEC      DECrement duration
0225  26 E7          BNE CONT 2 Go if not done
0227  08          INX          Point to next note
0228  08          INX
0229  C6 FF          LDA B $FF  Increment value
022B  86 30          LDA A $30  Initialise delay value
022D  1B          LOOP 4  ABA      Delay for interval
022E  26 FD          BNE LOOP 4 between notes
0230  7E 0203          JMP CONT 1 Continue
0233          RMB 1

```

MUSIC TABLE

```

0234  A090 3FA2 5CAC 6090  A090 4090 7080 F090
0244  5DA2 5BAD 6090 E06B  485F FF54 00(Terminator)

```

NOTE: Location 020D is now FE for once through, for continued play, change to F2

COMMENTS: We have asked Frank to send us some more tunes, and a follow up article on how to write the tunes, and we will print these as soon as they come. Also, when we tested the program, we used the 'Basic Amplifier' kit

from the 'Project Electronics' series from E.T.I., which Garry had built when he first started playing around with electronics, and it worked very well, so now he has a use for another one of these 'things' that he built. The kits are available from Dick Smith and several other suppliers for about \$5-00.

Graeme.

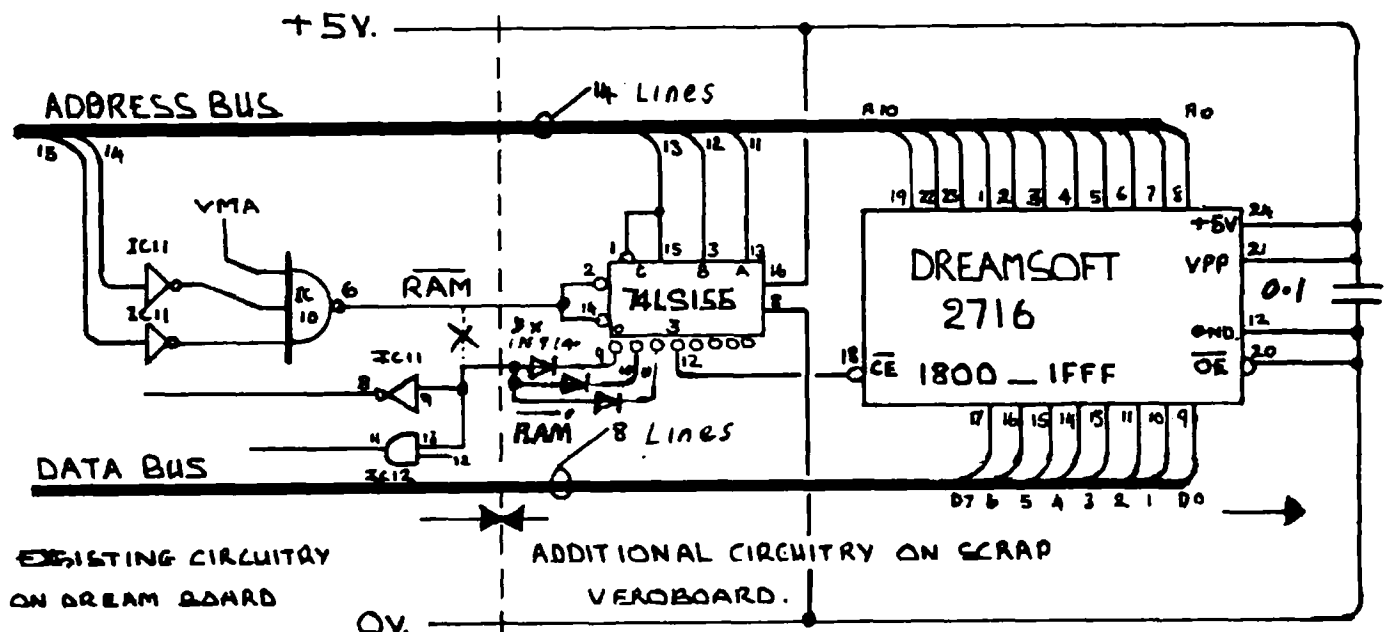
OPERATING THE DREAMSOFT EPROM
WITHOUT A "JR COMPONENTS EXPANSION BOARD"

GRAHAM LEADBEATER,

The DREAMSOFT package has proved to be very popular with users of the JR COMPONENTS Expansion Board. There must be, however, a lot of people who would like the resident software package in EPROM, but who don't need or cannot afford an expansion board, or who may have built a memory expansion board from either the design published in Electronics Australia, or the one featured in the first issue of the "DREAMER".

If this describes you, then help is at hand. A scrap of veroboard, a 74LS155, a 24-pin IC socket and some wire are all you need to add the DREAMSOFT EPROM to your system.

Here is how it is done:



How it works

The decoding in the DREAM enables the RAM over the address range 0000 - 3FFF (The entire bottom 16K). Now the DREAMSOFT package is intended to reside 1800 - 1FFF and this would collide with RAM images unless we modify the decoding.

The $\overline{\text{RAM}}$ signal in the DREAM goes low for any address in the range 0000 - 3FFF, this signal is used to enable a 74LS155 (1 of 8 decoder). This decodes address lines 11, 12 & 13, dividing the bottom 16K into 8 x 2K blocks.

OPERATING THE DREAMSOFT EPROM (Cont)

The lowest 3, (0000 - 17FF) is designated $\overline{\text{RAM}}$ and is taken back to where $\overline{\text{RAM}}$ used to go. This means that up to 6K of RAM at 0000 - 17FF will be able to be used. The decoder output for the address range 1800 - 1FFF is simply connected to the enable pin of the DREAMSOFT EPROM.

How to build it

The circuit may be built on a piece of 0.1" grid veroboard, 17 strips x 24 holes, as shown. Most connections to the DREAM board may be made via the bus expansion sockets. Some minor surgery will be required to intercept the $\overline{\text{RAM}}$ line on the main board. (See overlay below.)

Cut the tracks where marked 'x' in the overlay shown below, and mount the new board as close as possible to the DREAM expansion sockets. Try to keep the connecting leads less than 100 mm.

X Cuts in P.C. tracks



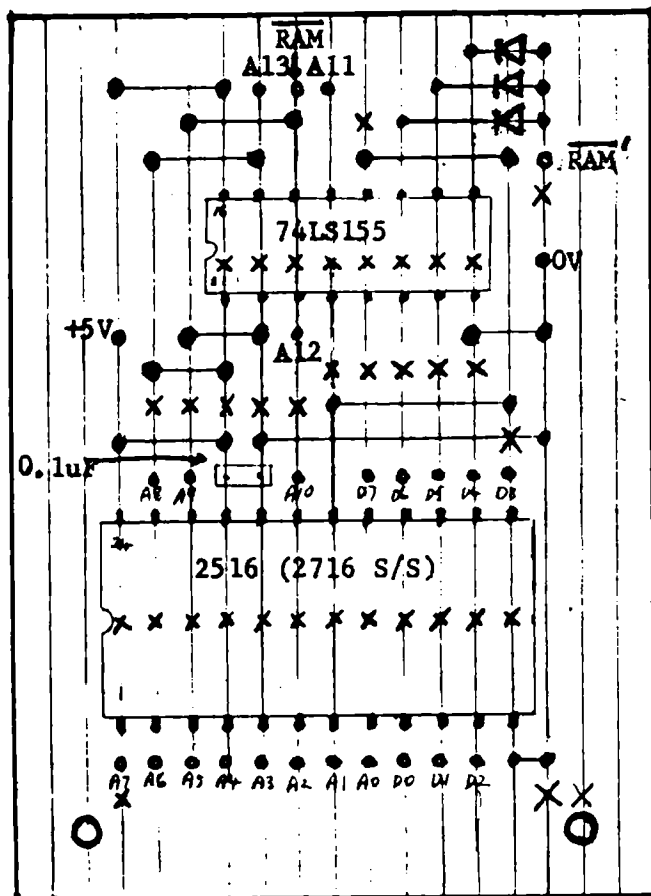
Links



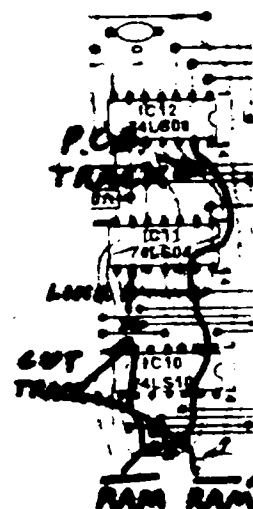
Mounting holes

● External connections (to DREAM board)

Diodes are IN914's (3 required)



Modifications to DREAM board.
Make 2 cuts & insert 1 link as shown.
Take $\overline{\text{RAM}}$ & $\overline{\text{RAM}}$ from here to new board.



ERRATTA

I did it again. Mr. G. Bidgood has advised us that there are TWO typing errors in his EDITOR program, in the January issue. Address 00F3 should be 4A, not 45, and 013F should be 00, not 0D. Sorry about that.

Garry.

DEPTH CHARGE

(0200 - 0400)

P. E. MARSTON.

Search for and destroy the submarine hidden in a 3D grid 16 x 16 x 16 in FOUR tries.

SURFACE OF WATER - 16 x 16 grid shown L.H.S., top is NORTH, left is WEST, origin (0,0) = top L.H. corner.

DEPTH OF WATER - 16 dots shown R.H.S., top = 0, bottom = F.

The computer selects the position for the submarine, within the grid. Enter your first guess by keying in the co-ordinates. e.g., 5, A, 7. A sonar report will then appear on the screen, e.g., S E H. This means your depth charge was SOUTH EAST of the sub. and too HIGH in the water. You must therefore make your second guess NORTH WEST and LOWER in the water. e.g. 3,8,B.

Keying the correct co-ordinates will destroy the submarine, but you only have FOUR tries. If you miss after four, the correct position of the submarine is revealed and your ship is torpedoed.

Keep a pencil and paper handy to record sonar reports. GOOD HUNTING.

```

0200 2330 6D22 6E06 A349 DDE1 7D08 3D3A 1208
0210 233A 6200 6100 A35E 3220 121E 122A D121
0220 7102 3120 121E 7202 1214 603C 6200 D021
0230 7202 3220 122E C80F C80F C00F 6600 7601
0240 F70A F80A F90A 6E01 6500 6418 8070 8180
0250 8290 8004 8114 8224 A35E D101 603C D021
0260 6324 5A70 22BA 5880 22CE 5C90 22E4 3E00
0270 12FA 4500 1302 3604 123E 6324 FA29 D345
0280 7308 FB29 D345 7308 FC29 D345 6D2F 6E0D
0290 A35E DDE2 22AE DDE2 7EFF 3E05 1292 2330
02A0 7E01 DDE3 22AE DDE3 3E09 12A0 1304 651F
02B0 F515 F507 3500 12B2 00EE 6501 8A75 3F01
02C0 12CA A34F D345 8A74 00EE A34A 12C4 7308
02D0 6501 8885 4F01 12E0 A354 D345 8884 00EE
02E0 A359 12DA 7308 6501 8C95 3F01 12F6 A363
02F0 D345 8C94 00EE A35E 12F0 7EFF 65FF 22B0

```

```

0300 1260 233A 650C F518 DDE3 F518 DDE3 F518
0310 A368 DDE4 F518 DDE4 F518 A36C DDE5 F518
0320 DDE5 651F F518 6D35 6E0E F629 DDE5 132E
0330 6D2B 6E03 A344 DDE3 00EE 6D2B 6E0F A347
0340 DDE3 00EE 287C FF08 1CFF 6080 4020 C088
0350 C8A8 9888 E080 C080 E088 A8A8 A870 8080
0360 8080 E090 90F0 9090 0014 0014 2200 0000
0370 2200 0000 0000 0000 0000 0000 0000 0000

```

BLACKJACK

(0080 - 0400)

G. J. BIDGOOD.

You start the game with a credit of \$200. Your cards are displayed on the RIGHT hand side of the screen, the computer's on the LEFT. You input your bet when announced by a flashing \$ sign approximately centre of screen. Your bet is regarded as invalid if the amount you input when added to or subtracted from your current credit is greater than \$500 or less than \$0. The amount you input will be erased and a new entry is required. If more than three numbers are entered, your bet will be erased and new bet input. A white square dealt to the player indicates an Ace. You make the decision if you want a card value of '1' or '11'. If an Ace is dealt to the computer it makes its own decision.

KEYS USED.

- If an Ace is dealt, Key 1 = value of 1. Any other key gives value of 11.
- You input your bet as a decimal number using keys 0 - 9. Up to three digits may be input for any bet.
- Key 'A' acknowledges your bet and next card is dealt, provided bet is valid.
- Key 'C' allows bet to be changed while bet is being input.
- Key 'F' will start a new game with a starting credit of \$200.
- Key 'O' transfers play to the computer.
- To call another card press any key other than '0'

On completion of game: * F

- Any key other than '0' will continue play with current credit.
- Key '0' will start a new game with a credit of \$200.

```
0080  A00C F265 6401 108A 6411 631D F029 4000
0090  1096 D345 7304 F129 3100 10A0 4000 10A4
00A0  D345 7304 F229 D345 00EE 0000 0000 0000
00B0  50F8 80F8 08F8 50FE FEFE FEFE FEFE FEFE
00C0  B7D4 F39E E79E 3EC8 E7CE F70C 24DE F7DE
00D0  67DE 9248 F6DE ABDE BD5E AB8E 0200 0022
00E0  0200 0030 130A 0200 0008 1305 0000 0002
00F0  0000 0000 0001 0000 0000 0000 0100 003E
```

```
0200  A00C 6BC8 FB33 A0F0 6000 610F F055 71FF
0210  3100 120C A0B0 6316 6400 D347 2080 A0E6
0220  6300 6417 6500 F555 A0E0 6338 F555 6E00
0230  4E05 1250 6D00 A0E0 F565 231C A0E0 F555
0240  4B00 128A 4E00 1252 7E01 FF0A 3F00 1230
0250  6E01 6D01 7E01 A0E6 F565 231C A0E6 F555
0260  4B00 1294 4E01 12CA 4E05 1280 6611 8750
0270  8765 3F00 1280 3510 1254 0806 4800 1254
0280  A0E5 F065 8505 4F00 1294 6940 F918 2080
0290  A0E6 12A8 A0E0 F265 6A04 2088 6908 22BA
02A0  3A00 129A 2080 A0EC F265 A00C F255 2080
02B0  FF0A 00E0 3F00 1206 1200 F918 6908 F915
02C0  F907 3900 1200 7AFF 00EE 6A05 6316 6410
02D0  A0B0 D347 6902 22BA 3A00 12D2 6000 6100
02E0  6200 6800 6411 631D FA0A 4A0C 1312 4A0F
02F0  12B2 4A0A 130C 3000 1312 FA29 D345 7304
```

(See bottom of Page 14 for listing from 0300 - 0400.)

MATH TABLES

(0200 - 0400)

D. WOOLNOUGH,


This program is utilitarian rather than a game, the idea being to interest primary age children in learning their maths tables. The multiplier is random select between the limits 0 - 12.

After entering the program and calling up C000, FN, 3, as usual, you should get a blank screen. The table to be learned is then entered via the keyboard, and the computer immediately replies by putting a problem on the screen. The child enters the answer and a RIGHT or WRONG is displayed. After three wrongs, the correct answer is shown for a brief period and another random multiplication problem for the same math table is displayed.

Leading zeroes are suppressed and the child enters the answer as he/she would say it. To change tables, simply press key 'F' and enter the new table.

Should anyone prefer sequential multipliers rather than random select, change the following locations.

0202 to 6900, 0204 to 6803, 0206 to 8490, 0208 to 1212. If more or less tries are to be given before the correct answer is shown, change the data at 0202, or 0204 in the alternate (sequential) program.

0200	F30A	6803	640C	001F	8405	4F00	1204	8940
0210	8490	6500	6A03	6B06	6C0C	6D16	6E50	A360
0220	F333	F265	3100	122A	122E	F129	DAB5	7A07
0230	F229	DAB5	7A07	A20A	DAB5	7A07	A360	F433
0240	F265	3100	1248	124E	F129	DAB5	7A07	F229
0250	DAB5	7A07	A2D0	DAB5	7A07	8534	74FF	3400
0260	125A	A360	F533	F265	3000	1272	3100	1280
0270	128E	F60A	360F	127C	00E0	1200	5600	12D6
0280	F60A	360F	128A	00E0	1200	5610	12D6	F60A
0290	360F	1298	00E0	1200	5620	12D6	12F2	3000
02A0	12A8	3100	12AC	12B0	F029	22C4	F129	22C4
02B0	F229	22C4	FE15	FE07	3E00	12B6	6E10	FE18
02C0	00E0	1202	DAB5	7A07	00EE	8850	2050	8800
02D0	00F8	00F8	0000	A324	22EC	A32A	22EC	A330
02E0	22EC	A336	22EC	A33C	22EC	130E	DCD5	7C07
02F0	00EE	A32A	2308	A342	2308	A33C	2308	A348
0300	2308	A34E	2308	129E	DCD5	7C07	00EE	FE15
0310	FE07	3E00	1310	6E10	FE18	00E0	78FF	3800
0320	1210	129E	8888	A8A8	F800	F888	F890	9000
0330	F888	8888	F800	88C8	A898	8800	F880	9888
0340	F800	2020	2020	2000	8888	F888	8800	F820
0350	2020	2000	7901	1204				

CAMPBELL FETHERS,
[REDACTED]

This is a much modified and improved version of Graeme's original 'Sub' game, which appeared in the first issue of the 'Dreamer.' For those of you who haven't played before, I will recap the instructions.

You are the commander of a submarine, and your mission is to destroy enemy shipping. You have an unlimited number of torpedoes, and the ship has an unlimited number of depth charges.

To move your submarine, hold down: -

'4'	Surface Left	'5'	Surface	'6'	Surface Right
'8'	Left			'A'	Right
'6'	Dive Left	'7'	Dive	'Z'	Dive Right

To fire torpedoes, you push: -

'0'	Left Up	'1'	Up	'2'	Right Up
-----	---------	-----	----	-----	----------

- NOTES:
1. The ship drops depth charges at random, which explode at various depths, and also when they hit something.
 2. If a torpedo hits a depth charge, both explode.
 3. All explosions are accompanied by exploding graphics and various noises.
 4. When a ship is hit, it sinks, then reappears at random.
 5. Two torpedoes may be in operation at any one time.
 6. If a depth charge explodes close to a submarine, without necessarily scoring a direct hit, the submarine may sink.
 7. After the submarine is sunk, the score is shown. The game recommences after five seconds, or when key 'F' is pressed.

To change key allocations.

Bytes 03CD to 03DC form a look-up table defining the meaning of each key to the keyboard scanning routine. Bytes may be swapped around within this table to change the meanings of the keys, to suit your keyboard. Byte 02B7 defines key 'F' as the restart key. This may be changed also.

0200	03F6	A3FF	6203	6000	D021	7008	3040	1208
0210	6E00	6D0F	6C1C	6A00	6903	22C2	620F	A3F3
0220	DCD3	68FF	6700	03EB	1246	7801	5890	1260
0230	68FF	3E3F	123A	6EFF	C903	22C2	0000	0000
0240	7E01	DE03	1260	A2CA	4998	1254	DE98	7901
0250	DE98	1260	DE98	C901	CE3F	22C2	0000	68FF
0260	03A8	0390	1270	A3F3	DCD3	037C	DCD3	1274
0270	3000	035F	A0C0	0319	22EC	A0C3	0319	22EC
0280	C007	A080	0400	A080	04A6	37FC	1224	6640
0290	F618	6B0B	F715	A008	FA33	F265	6A18	F029
02A0	00E0	3000	DAB5	7A04	F129	4000	3100	DAB5
02B0	7A04	F229	DAB5	620F	F107	E29E	4100	1200
02C0	12B8	6000	A3FD	DE03	00EE	080C	1C3C	1C3C
02D0	100C	C602	B680	1288	40B7	8012	4F4C	375A
02E0	26FD	3311	26F7	5CC1	1826	E939	6F80	8F92
02F0	3700	3F00	00EE	A3FD	7A01	DEF3	6980	A2CA

SUPER SUB. (Continued.)

0300	DE98	02D2	14CA	D63D	E701	963C	E602	2B05
0310	8B04	5A2A	FBA7	0020	20DE	26A6	012B	E727
0320	3D8D	16DE	26A6	0181	0327	25E6	0027	2BC1
0330	3F27	276A	01EB	02E7	00A6	0097	2EA6	0197
0340	2FE6	025C	58CE	0397	8D03	DDC6	027E	C226
0350	E600	D03E	C107	2202	9737	6F00	6F01	39D6
0360	3B26	0E53	D73B	D631	96C1	2606	4397	C1D7
0370	C239	96C4	2605	4397	C4D7	C539	963C	9B30
0380	2B06	8138	2E02	973C	D63D	D831	C103	2F06
0390	C11D	2E02	D73D	3980	4080	8040	8096	3081
03A0	F827	049A	3126	4939	BDC2	97D6	1826	1516
03B0	2B12	CE03	CD8D	26A6	0016	C403	4747	4747
03C0	4A5A	2004	4F5F	973B	9730	D731	3990	9192
03D0	1100	1020	1101	1121	1102	1222	11DF	14D8
03E0	15D7	1524	037C	0015	DE14	3996	392A	0139
03F0	7EC0	D008	1CFF	CE00	804F	7EC0	7D28	7CFF
0400	DE26	A601	2B71	2614	D630	265F	D639	2B5B
0410	963E	A700	C604	D730	E701	200A	8D6B	E601
0420	C11F	2755	6C01	7F00	3F8D	5E96	3F26	07BD
0430	C132	843F	2635	DE26	A600	903C	810A	2E14
0440	81FE	2D10	A601	903D	8104	2E08	81FE	2D04
0450	C6FC	D737	CE00	C0DF	128D	737C	0013	7C00
0460	138D	6BDE	26C6	80EA	01E7	01DE	2608	08DF
0470	268C	00C0	268A	398D	1D6F	016F	0020	EC00
0480	A600	972E	A601	972F	398D	F5CE	0466	C601
0490	BDC2	26DE	2639	8DE8	CE04	9FC6	0720	F110
04A0	387C	FE7C	3810	DE26	A601	2A13	8DD8	E600
04B0	A601	C003	8003	E700	A701	8DDA	BD02	D208
04C0	08DF	268C	00C0	26DE	3900	F218	00EE	DE12
04D0	E601	272A	A600	DE26	A000	E001	81FD	2D1E
04E0	8102	2E1A	C1FD	2D16	C102	2E12	8602	9721
04F0	C640	BDC2	E5DE	12BD	0339	DE12	6F01	3900

BLACKJACK. (Continued.)

0300	8010	8120	82A0	A0E0	F255	12E8	03B8	4800
0310	1230	2088	4A0C	12DC	6A06	12CC	6C01	C70F
0320	4700	131E	470B	131E	470F	131E	A0EF	F055
0330	F71E	F065	4004	1316	A0F0	F71E	7001	F055
0340	8A70	4A0E	670A	4A0C	670A	4A0D	670A	A0B7
0350	D349	4A0A	6C02	4A01	2394	8630	7402	7301
0360	3A0A	7301	03B0	D345	7A01	7CFF	3C00	135E
0370	8360	74FA	3D00	7330	3330	1380	7308	1382
0380	73F8	3D00	73D0	6615	8574	8655	0BF0	A0EF
0390	F065	00EE	3D01	13A6	6908	6616	8655	8695
03A0	3F00	670B	13AC	F70A	3701	670B	00EE	0000
03B0	CE00	BC96	3A7E	C198	CE00	DE0C	A600	A904
03C0	198B	F084	0FA7	108D	2A26	F125	2D81	052D
03D0	0A22	27A6	0C26	23A6	0D26	1F08	0808	0CA6
03E0	00A2	0497	EF8B	0A25	0296	EF8D	0626	F025
03F0	0920	0AA7	0A09	8C00	DB39	7C00	3839	0101

QUESTIONS AND ANSWERS

One of our readers, Mr. Nick Pyner, of [REDACTED] N.S.W. asked us the following question:-

"In your regular article 'How to submit programs' you request 20 secs. of High and Low leader, saying '....We need a leader to align our tape heads, and tune the DREAM's input port....' I would be glad if you would elaborate on this, as there could be a secret therein that could solve some problems with loading from tape."

Well Nick, here is the reason:-

The method I use to set up to load a tape which has been recorded on a DREAM other than my own, and therefore is recorded at different frequencies, is as follows. Firstly, I rewind the tape, and start it playing. Then, I wait for the leader tone to begin and adjust the tape head position until the maximum volume is obtained. This is done with a small screwdriver inserted through a small hole drilled in the case of the tape player above the tape head adjustment screw. This can be done during either the High or Low tone, or the program itself. Now I rewind and wait for the tone to start again, then insert the screwdriver through a hole in the case of my DREAM and adjust the 5K Pot. on the DREAM board until the GREEN LED glows continuously during High tone, or RED LED during Low tone. (See the article 'Ideas, A Bright One' in the October issue for more information on these LEDs.)

Tuning the DREAM in this manner allows us to load tapes from almost any source, no matter how far out the frequencies are, but if we do not have both the High and Low leader tones it is very hard to make any adjustment. This applies to all submitted tapes, even if your tones are 1200.0000 Hz and 2400.0000 Hz, BECAUSE MINE IS NOT.

Graeme V. Samways.

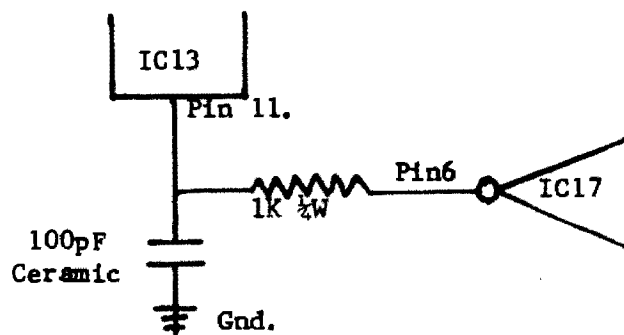
VERTICAL SYNC PROBLEM

When firing up my DREAM, I had a vertical sync problem. i.e., Picture would not lock, and multiple vertical pictures.

Replacing IC13, IC16 & IC17 or any rearrangement would not fix it. All the tracks and soldering was double checked. In the trouble shooting section of the DREAM project Michael Bauer suggests there may be problems resetting IC13.

The reset pulse is very short and can not be seen on cheap CRO's. I decided to lengthen these pulses with a delay network. (See diagram.) PRESTO, it worked.

The resistor may be placed on the board in place of the link between Pin 11 of IC13 and Pin 6 of IC17, and the capacitor wired on the back of the board between IC13 and Ground.



Lee De Vries.

HOW TO SUBMIT PROGRAMS

To remain in operation, we need a constant supply of new programs, and articles about the DREAM 6800. If you can write an article on modifications you have made to your DREAM, or the use you are making of it, or if you have written any games, or utility programs, we invite you to submit them to us for consideration. ALL CONTRIBUTORS OF PROGRAMS PRINTED WILL RECEIVE VOUCHERS FOR TWO FREE NEWSLETTERS. CONTRIBUTORS OF ARTICLES AND IDEAS PRINTED WILL RECEIVE FROM ONE TO THREE VOUCHERS, BASED ON THE GENERAL INTEREST CONTENT OF THE ARTICLE, AND THE AMOUNT OF WORK THAT HAS GONE INTO IT. Along with the listing for all programs submitted, we will need a tape recording, with at least twenty seconds of High and Low "leader" on it. We need a leader to align our tape heads, and tune the DREAM input port. To do this you first must record 20 Sec High tone, then 20 Sec Low tone. The High tone is normal leader, and can be recorded normally. To get the Low tone, load in the following Machine Code program.

```
0200      8640  Accumulator A = 40
0202      B78012 Store in PIA output port.
0205      20FE  Branch back 2 bytes from 0207
0207      0000
```

This will produce a continuous Low tone when run 0200, FN, 3. After 20 seconds press RESET to return to normal. Then load your program. We need the electronic copy so we can test the program and verify the listing BEFORE printing, to eliminate program errors and increase the enjoyment of other users.

We will not be able to enter into correspondence, but will print corrections or improvements where necessary. We will not be selling tapes.

Programs submitted for consideration should be typed, for clarity, and set out in the following format:-

- 1) Program name and memory location.
- 2) Your name and address. (If you do not wish to receive any correspondence from other users, omit your address.)
- 3) The program explanation. (Don't forget key functions)
- 4) The program listing, typed single space. (If in doubt, have a look at the way the programs in this issue have been typed, and copy the format)

Following the guidelines set out above lets us check out the programs submitted quickly and easily. If you do not have access to a typewriter, we will accept a handwritten listing, providing it is LEGIBLE, and accompanied by a tape. However, if we cannot read your writing, and the tape will not load, or has 'bugs' in it, there will be no way we can check the program, and it will not be considered.

That's all there is to it, so send us in your favourites, and don't forget, for each one we use, you get vouchers for two newsletters free of charge. Should you be a prolific programmer, and accumulate some surplus vouchers, or have already paid a subscription to the newsletter, we will redeem the vouchers at a rate of six vouchers for \$15-00.

PRICE STRUCTURE

The cost of this newsletter is \$3-00 per issue. An advance subscription is available at reduced cost. Please write for details of cost and length of time remaining in current subscription period.

BACK ISSUES. Copies of all newsletters from No.1, September, 1980, are available at a cost of \$4-00 each, from:-

N.S.W. 6800 USERS GROUP,



(Please add -10c to all CHEQUES sent from outside N.S.W., to cover Stamp Duty charged by N.S.W. Government. This is only required on cheques and does not apply to Money Orders etc.)
