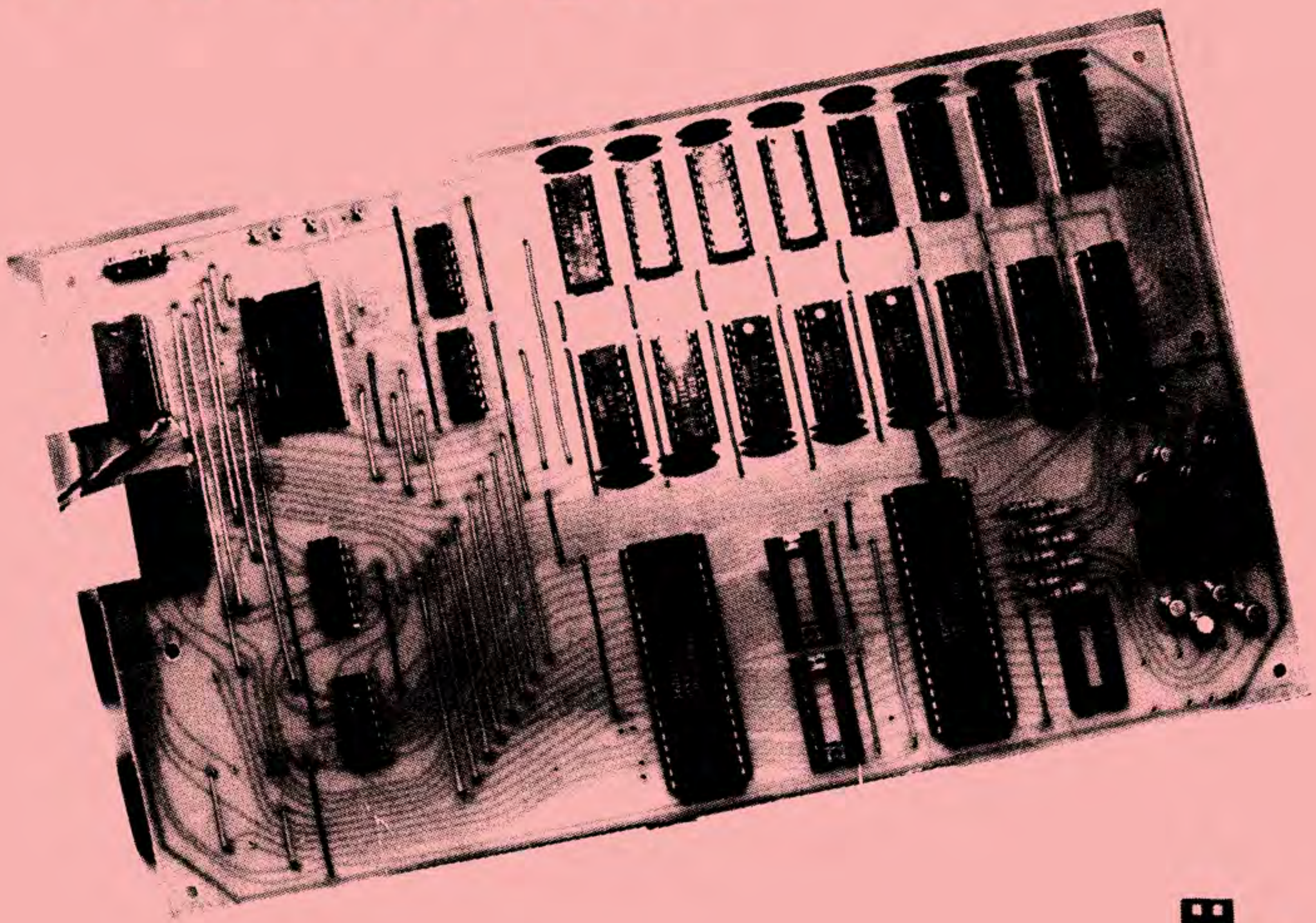
DREAMER Nö 8

Registered for posting as
a publication. Category B

APRIL '81

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



* DREAMSOFT *

2K of Non-volatile Dreamsoftware

(See Review in Dreamer No.6)

DREAMTEXT - Creates alphanumeric displays with full 64 character ASCII subset.

BLOCK MOVE (& Block Compare) - Copies any block of data to any other area of RAM.

TAPE LOAD & DUMP DISPLAY (& Tape Verify) - Simplifies tape handling.

BRANCH OFFSET CALCULATOR - Essential resident software for the Machine-code programmer.

SUPERTYPEDREAM - 4 formats including a disassembler provides hard copy via a disposals Baudot teleprinter.

SUBROUTINES - Many powerful User-callable subroutines.

All for $30.00 which includes a programmed 2716 EPROM and a 37 - page handbook containing Installation & Test instructions, interfacing information, data, list of subroutines and a fully commented listing.

The DREAMSOFT EPROM can, of course, be used without an expansion board - see Dreamer No.7. For those using an EA 4K RAM board, we can supply full instructions for incorporating our EPROM. These include circuit diagram, board layout, and pin-by-pin connections. No additional chips are required. Add $5.00.

Make Cheque or Money Order payable to :- DREAMSOFT
P.O. Box 139,
MITCHAM VIC. 3132

DREAM CARDS

REQUIRES 2k

* *DREAMCARDS*
6/8 ELPHIN STREET,
IVANHOE 3079 VIC.

LOOKING FOR A PROGRAMME THAT USES YOUR DREAM'S INTELLIGENCE, NOT JUST ITS DISPLAY?

"Dream Rummy" is an exciting and easily learned version of Gin Rummy for the Dream 6800. It has a powerful set of logic routines, a memory-mapped card deck and 2 levels of skill that give it realism and all the ability of a good human player.

Cassette & Instructions $10.00 (immediate delivery). Programme listing (fully commented) is $5.00 extra.

BONUS: Get "Strip Jack Naked" free with each "Dream Rummy" cassette bought.

DREAMER

No. 8 APRIL, 1981.

Hello again. What an issue this one is. You will notice it is bigger than usual, and even then, we could not get everything in! We had to leave 'SUPER 8 BUG' until next menth. M.J.B.'s SOUND EFFECTS GENERATOR is in though, and it is fantastic. Dream Invaders with sound effects is really something else. All components should be readily available from your normal supplier. The only thing we had trouble with was the SN76477 chip, until we discovered that it is available through SILICONE VALLEY suppliers. We will not be selling P.C. Boards, you will have to make your own up, either using the method described in the article, or tracing the design onto a piece of P.C.B. using carbon paper, then useing a DALO pen and etching it. (Dick Smith puts out a very good information sheet on how to make your own P.C. Boards in this manner, at a cost of about fifty cents.)

On a more sombre note, Michael Bauer has advised us that due to pressure of other business, he will shortly be closing down his 'DREAMWARE' organisation, so if you want a copy of DREAM INVADERS, I would suggest you get your order in quickly. (Before April 20th.)

We also have an article from Bruce Mitchell on playing music on your DREAM, and some more songs from Frank Rees for the OPUS program in last month's issue. Please note that these two programs are slightly different, and the keycodes for the notes are entered from different starting points and are assembled in a different manner. Read both articles carefully!

The wrappers we used to send the February newsletters out were an experiment to try and cut our costs a bit, but so many of you wrote to complain about not getting your newsletters in as good a condition that we have decided to go back to envelopes. We have also been granted registration of the DREAMER as a periodical, Category B, to try and reduce our postage costs.

While on the subject of costs, our printing and stationery costs have increased so much since we started, and production of the newsletter now takes up so much of our time, we have regretfully decided that there will have to be a price increase for the next subscription period, from July to December, 1981.

The new rates will be, $3-50 per single issue, and $18-00 for a six issue subscription, within Australia.

Overseas prices will be: N.Z. & Malasia $4-50 single & $24-00 sub.
Others, $5-00 single, $27-00 subscription.

In return for the increased cost, we will be able to produce a bigger, and we hope, better newsletter. The format over the next six months will concentrate more on programs, as we now have a good supply of games on hand, and we will include at least six programs in each issue. The new prices are still lower than our initial ones were in September last year, and with the increased content, we think that they still represent good value for all DREAM addicts.

In this issue you will find an envelope sized card, with a survey on the back. Please circle or tick the relevant places, add your comments if you wish, (constructive criticism is allowed), put a stamp on it, and drop it in the mail box. It will only take a minute or so, and the answers will let us plan a better newsletter for you, when we know what the most popular mods to the system are. It will also tell us if you wish to re-subscribe, and whether it is worthwhile carrying on with the DREAMER, or if all our effort is being wasted. Please return it early, so we know who to send labels out to in the May and June issues, for the next subscription period. There is no need to send any money yet, send that when you return your address labels, but PLEASE, fill in the survey card and send it back as soon as you can. In fact, why not do it now? We will publish the results of the survey as soon as we compile them.

We are sorry, but our volume of mail has grown so much, and Graeme's University studies now take up so much more of his time, (he reckons that third year is a lot harder than second year), that we have been forced to restrict the personal replies to your problems. In future, personal replies to problems will only be sent if the queries include a fee of $2-00, and a stamped, self addressed envelope. We will, however, continue to answer problems of general interest and publish solutions to problems sent in by readers, in the DREAMER.

We suggest that if you have a problem, you contact one of the following, as they have all indicated their willingness to help others.

N.S.W. - Fred Lever,Snr. [redacted]
VIC. - Frank Rees. [redacted]
S.A. - John Cranstone. [redacted]

If there is anyone else who is willing to help out in this way, please contact us, and we will print your name and address in the DREAMER, as the demand is now just too great for us to handle, on top of the work load we already have producing the best newsletter we can for you.

NEXT MONTH - We will have, another BUMPER ISSUE, containing,
- SUPER 8 BUG.
- An article on ASSEMBLY LANGUAGE PROGRAMMING.
- A DAY OF THE WEEK CALCULATOR for any 20th Century date. (Find out what day you were born)
- VIDEO BINGO. (Save money, play 'Housie' at home.)
- RACE MEETING. (A race horse game.)
- SPACE INVADERS. (By Fred Lever, Jnr.)
- A MORSE CODE TEACHING PROGRAM.
- HANGMAN.

Until next time,

HAPPY DREAMING,

GRAEME SAMWAYS AND GARRY NELSON,

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

****

STOP PRESS! STOP PRESS! STOP PRESS! STOP PRESS! STOP PRESS! STOP PRESS!

WE HAVE JUST DECIDED ON A NEW COMPETITION. TO HELP ENCOURAGE YOU TO FILL IN AND RETURN YOUR SURVEY CARDS EARLY, WE WILL GIVE A FREE SIX ISSUE SUBSCRIPTION, VALUED AT $18-00, TO THE SENDER OF THE 100th SURVEY CARD WE RECEIVE. WE COULD NOT HAVE A SIMPLER COMPETITION THAN THAT, JUST FILL IN YOUR CARD AND SEND IT BACK, EVERYONE CAN ENTER, AND REMEMBER, YOU HAVE TO BE IN IT TO WIN IT! WE WILL TELL YOU WHO THE WINNER IS IN THE JUNE ISSUE.

****

Have a look at the DREAM CARDS advertisement on the back of the front cover. We received a copy of the program just before this went to the printer, and first impressions are, it is GREAT. There will be a full review NEXT MONTH.

***********************************"

If you have submitted a program and have not yet received the tape back, it may be because you did not write your name and the program name and memory locations on it. We have several that have become separated from the letters they came with, and there is no way that we can load them to find out what is on them, without the start and finish locations. If you want them back, drop us a line and tell us the brand and length of tape, and what is on it, please.(We suggest you put a 'voice leader' on any tapes you send us.)

****

PRECOGNITIVE DREAMS

0200-030F

GRAHAM LEADBEATER,
[redacted]

In all modesty, I can claim this program to be something of a breakthrough, destined to have far-reaching effects on the structure of civilisation as we know it!

What it does is simply transfer information backwards in time! (Not quite the same thing as Predicting the Future, although that's what it amounts to).

A principle we take for granted in our lives is that of causality. Events always happen AFTER their causes, eg numbers may appear on a T.V. screen a few micro-seconds AFTER the key-closures which caused them - but never BEFORE. "Must it always be that way?" "Of course it must!, mustn't it?".

To answer that question, we must look at the nature of time itself and there is no shortage of esoteric theories about that. For example, there is the "Parallel Time Lines" concept which says that an infinite number of possible futures, dividing at the quantum level, co-exist "side-by-side" and a state of mind can influence the path we take. Ever heard of the "Power of Positive Thinking"? Well, it's a load of rubbish! Every Technician knows that Murphy's Law over-rides The Power of Positive Thinking!

Seriously though, my own research tends to support the conventional idea of Time as a steady stream flowing at a fixed rate from Past to Future. (There is some suggestion that events in the present are caused, not only by the "push" of the Past, but also by the "pull" of the Future. This need not concern us here, however).

Time travel (of the Science-fiction kind) is certainly possible according to General Relativity theory, all we need is the distortion of the space-time continuum caused by a rotating dense singularity - the popular Black hole. An astronaut crossing the event horizon associated with such a singularity will certainly experience time travel but where and when he ends up is anybody's guess and getting home again poses some problems.

A more practical proposition is not to attempt to move a physical object such as a person through time but rather to move information, which, being an abstract concept and not having any mass, is not so constrained by the laws of relativity.

All very well, but how do you do it?

This is a question a lot of people are asking these days and intensive research is going on on both sides of the Iron Curtain (Top Secret of course) with the certainty of World Domination as the prize for the winner! Heavy stuff, eh?

The most promising avenue at the moment is the Tachyon. This is a sub-atomic particle which travels faster than light and therefore (by relativity) backwards in time.

Contrary to popular opinion, faster than light travel is possible, it's just that light speed represents a barrier that cannot be crossed. Tachyons live on the "far side" of that barrier and cannot slow down to the speeds we are used to in our concept of the universe. They are elusive things to catch, since, because they travel backwards in time, they will be detected before they arrive! (Really heavy stuff!)

Scientists are attempting to marshall a stream of tachyons and use it to carry information, but it's a waste of time (pun intended)!

All we need to do is to write an algorithm for the behavior of a tachyon, as we would like it to be so we can simulate its behavior in software. The actual characteristics of the particles, or even their very existance, is irrelevant. The program behaves as the particles would and provides a convenient way of conveying the data to and from a human being.

PRECOGNITIVE DREAMS (CONT)

I have used the CHIP-8 Language for this program since speed is unimportant (we have all the time in the world) and I have set it up for a simple but very practical application - beating the Bookies. When a race is run you key in the number of the winning horse and the number appears on the screen 15 minutes before that to allow you to place your bets.

Sounds ridiculous, doesn't it? "What if I don't put it in?" is the obvious question. There is no paradox involved, and you can't trick it, if you don't put the number in it won't have appeared. It takes most people a while to adjust to this, since causality is basic building block of the model of the universe which we all carry in our minds.

Here is the program listing, it does not need any extra memory or even a DREAMSOFT EPROM.

```
0200   6A00 6B00 A260 DABE  7A08 A26E DABE 7A08
0210   A27C DAB8 7A08 A284  DABE 7A08 A292 DABE
0220   7A08 A2A0 DABE 7A08  A2AE DABE 7A08 6B0C
0230   A2BC DAB2 6A06 6B12  A2BE DABE 7A08 A2CC
0240   DABE 7A08 A2DA DABE  7A0C A2E8 DABE 7A08
0250   A2F6 DABE 6A32 6B1E  A304 DAB2 125C 0000
0260   1E3F 6100 C0C0 FFFF  C0C0 C0C0 C0C0 0303
0270   83C3 C3C3 C3C3 C3C3  C3C3 C3C3 FCFE 0703
0280   0307 FEFC 0F0F 0C0C  0C0C 0F0F 0D0C 0C0C
0290   0C0C F0F8 1C0C 0C1C  F8F0 C0E0 7038 1C0C
02A0   3F3F 0C0C 0C0C 0C0C  0C0C 0C0C 3F3F 0C0C
02B0   0C0C 0C0C 0C0C 0C0C  0C0C 0F0F FCFC FFFF
02C0   C0C0 C0C0 FEFE C0C0  C0C0 C0C0 C0C1 0303
02D0   0303 0303 0303 0303  0100 FCFE 8703 0303
02E0   0303 0303 0387 FEFC  3F7F E1C0 C0C0 C0C0
02F0   C0C0 C0E1 7F3F 0383  C3C3 C3C3 C3C3 C3C3

0300   C3C3 8303 FFFF 0000
```

If the code looks strange to you, don't worry, it's due to the cyclic nature of time. You've heard of "History repeating itself". The actual operation of the program is beyond the scope of this article since explaining the more esoteric aspects of relativity and particle physics to the average Dreamer is like explaining how dog food gets into cans, to the average Pekingese.

One word of warning: double check your program before running it, make sure all those OC's & CC's are exactly right.

References "By His Bootstraps" by Robert Heinlein

"All You Zombies" " "

"Timewarps" by John Gribbin

"Behold The Man" by Michael Moorcock

"Let's Do The Timewarp Again" "The Rocky Horror Show"

* * * * * * * * * * * * * *

Frankly, we were sceptical, to say the least, when we received this article - but we've tried the program and we can assure you that it does run as intended.

G.V.S.

HOW TO USE CHIP - 8. (Part 6)

The Chip-8 INDEX is a three digit (Hex) address that has many uses, but has two main applications.

The first is to point to the starting address of the data that you wish to display using the DXYN instruction. The second is to point to the address from which to load or store variables.

There are various methods available to set up the Index. The first is the absolute method. This is the AMMM instruction. It can point to any location from 0000 to OFFF. You simply select the location you want and put the A in front.

```
I.E. You want to point to 0278
     You use             A278
or,  You want to point to 0320
     You use             A320
```

The second instruction that directly changes the index is FX1E. This adds the value of the variable X to the value of the index.

```
I.E. You want to add the value of B to the Index.
                6B06        B = 06
                A300        I = 0300
                FB1E        I = 0306
or,  You want to add 8 to Index
                Start with 8 = A0
                A280        I = 0280
                F81E        I = 0320
```

The next two instructions store variables and recall variables. They are FX55, FX65 respectively. They also change the index in a secondary operation. Their main function is to store and recall the variables where you want them.

Firstly, to store variables. (FX55)

This instruction takes the variables from 0 up to and including the specified variable and stores them in the memory from the location pointed to by the Index. After the operation, the Index will point not to the first variable, but to the next byte after the stored variables.

E.G. We want to store variables 0 to 9 in memory from 0280, so we proceed as follows:-

```
                A280        A = 0280
                F955        Store 0 - 9 at I
```

	MEMORY	
Index before instruction →	0280	From 0
	0281	From 1
	0282	From 2
	0283	From 3
	0284	From 4
	0285	From 5
	0286	From 6
	0287	From 7
	0288	From 8
	0289	From 9
Index after F955 →	028A	-
	028B	-
	028C	-

You will see from this that the Index is automatically incremented. We could store variable 0 after variable 9 again, thus.

```
                A280        A = 0280
                F955        Store 0 - 9 at I
                F055        Store 0 at I
```

HOW TO USE CHIP - 8. (Cont)

MEMORY

	0289	From 9
Index before →	028A	From 0
Index after →	028B	-

Complications arise if you want to store say A and B, (i.e., X & Y locations of a ball etc.) To achieve this we do the following:-

```
A2FE     I = 02FE
80A0     0 = A
81B0     1 = B
F155     Store 0 & 1 at I
         (I will now be equal to 0300)
```

It is therefore important to try to leave 0, 1, 2, etc for calculations or other purposes which do not require the data to be stored permanently.

To retrieve variables from memory (FX65) you follow the same steps as for FX55, but you load the variables starting with 0, up to and including the variable specified by X, with the data from memory starting from where the Index was pointing. At the end the Index will be pointing to the next byte after the retrieved data.

E.G. To retrieve variables 0 to 9, followed by 0 again, as in the previous example;

```
A280     I = 0280
F965     Load 0 to 9 from Index
F065     Load 0 from Index
         (I will now = 028B)
```

MEMORY

Index at start →	0280	To 0
	0281	To 1
	0282	To 2
	0283	To 3
	0284	To 4
	0285	To 5
	0286	To 6
	0287	To 7
	0288	To 8
	0289	To 9
Index before F065 →	028A	To 0
Index after F065 →	028B	-
	028C	-

or, if we want to retrieve A and B, as in the second example:-

```
A2FE     I = 02FE
F165     Load 0 & 1 from Index
8A00     A = 0
8B10     B = 1
```

EXCHANGING TWO VARIABLES

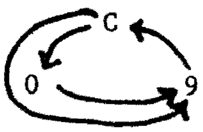
When you have to exchange say 9 and C, and F is available, use

```
8FC0     F = C
8C90     C = 9
89F0     9 = F
```

What happens if you do not have a spare variable? Easy, create one by using memory.

HOW TO USE CHIP - 8. (Cont)

```
A300      I = 0300     ) Hide 0
F055      Store 0 at I)
80C0      0 = C
8C90      C = 9
8900      9 = 0
A300      I = 0300
F065      Load 0 from I (Recall 0)
          (Don't forget to reset the Index)
```



Another instruction which stores a variable in memory is FX33, but this performs a very special operation.As you should know, each Chip-8 variable is Hexadecimal and can represent a number from 00 (decimal) to 255 (dec.) The FX33 instruction stores a three digit DECIMAL number, equivalent to the value specified by variable X, in three succesive bytes of memory pointed to by the Index. The Index is not affected at all by this instruction, so at its completion, the Index will still point to the first digit that was stored, NOT the location following the third digit.

The three digit decimal number can be recalled by a F265 instruction. This instruction is extremely useful for displaying scores, etc. The memory organisation is:-

	028F	-	
Index Before ⟶ and After FX33	0290	MSD	Most significant digit (100s)
	0291	MD	Mid digit (10s)
	0292	LSD	Least significant digit (1s)
	0293	-	

Well, I did say that I would complete the instructions this month, but I find that there is no room for the DISPLAY instruction, so I will feature it and FX29 next month, as it takes some explaining on how to use these.

Graeme V. Samways.

****

DIRECTIONAL PADDLE

Do you have trouble finding all four keys when playing Snake, and other fast moving games?

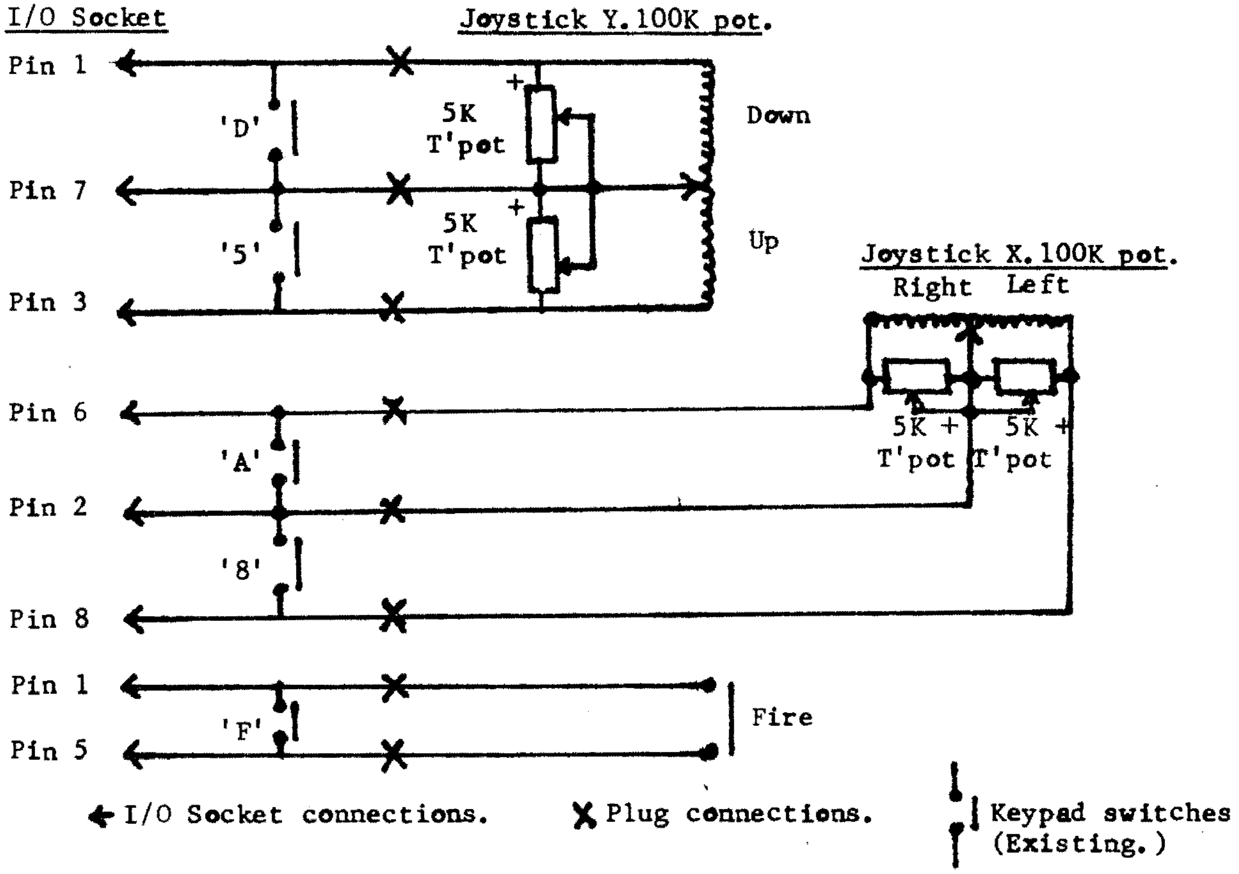
I did, so I decided to put a Joystick on the keyboard so it would sense one of four directions. I still wasn't satisfied with this however, as when I played 'Sub', I needed EIGHT directions. Various people had sent in suggestions for four directions, e.g., Joystick driving reed relays, or a home made joystick with switches. I knew none of this was necessary as any resistance under about 2K across any switch in the keypad would register as closed.

The problem was simple for four directions. Just wire a 5K Joystick across the keypad. First problem, D.S.E. doesn't supply them, and they are not generally available elsewhere, only 100K. Oh well, back to the drawing board! I did try the 100K version, but it needed to be thrown over hard in each direction. THEN IT HAPPENED. I realised that all that was needed was a 2.5K resistor across each side of the Joystick, so the computer was just about to see a closed state. When the Joystick moves about one third of the way across, the parrallel resistance shows the required 2K, and a closure is seen. SIMPLE! Now, what about eight directions? That was more complex. An op-amp, or comparators driving AND gates? No, too expensive. Then I decided that the general principle of LESS HARDWARE = MORE SOFTWARE would apply in this case, so I wrote a new keypad routine which returns with the direction (1 of 8) in a variable without wait, and using only the Joystick and resistors.

DIRECTIONAL PADDLE. (Cont)

This software will be included next month, but in the meantime, here is how to build the directional joystick.

The actual construction is very simple, just connect the Joystick as shown below. (I actually used +5K Trimpots instead of 2K7 resistors, as they allow for optimum adjustment of each direction.)



I suggest you put the Joystick in a zippy box the same as we did with M.J.B.'s Joystick, and include an extra button on the top (a fire button) wired across the 'F' switch. How you connect it to your DREAM is up to you, but the method used in the 'Invader Control' may be the best.

When you have completed the joystick, redirect 'The Snake' to suit your keyboard, (See 'Re-directed Snake' below,) it makes the game much easier.

Graeme V. Samways.

****

REDIRECTED SNAKE.

Well, we finally found out how to change the key directions for 'The Snake', so now you will all be able to change them to suit your own keyboard.

You must first select four keys for directions. I have chosen 5↑, 8←, A→, D↓. (Our standard key layout.) At present we have 1↑, 4←, 6→, 9↓.

To change directions, make the following alterations:-

DIRECTION	ADDRESS	FROM	TO	COMMENTS
SET UP	023C	6006	600A	Set up start direction. ⟶
DIR.⟶	024C	6B06	6B0A	BOTH SAME KEY!

REDIRECTED SNAKE. (Cont)

DIRECTION	ADDRESS	FROM	TO	COMMENTS
↑	0252	6A01	6A05	Input
	02F2	4001	4005	Erase end.
	0258	6A06	6A0A	Input
→	0272	4B06	4B0A	Move
	02FE	4006	400A	Erase end.
	0264	6A04	6A08	Input
←	026A	4B04	4B08	Move
	02F6	4004	4008	Erase end
↓	025E	6A09	6A0D	Input
	026E	4B09	4B0D	Move
	02FA	4009	400D	Erase end.

Graeme V. Samways

****

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.

+++++++++++++++

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]

++++++++++++++

FOR SALE. One only 6800 or 6802. (Take your choice, as I have both, but only need one.) Board complete, up and running, all I.C.'s in sockets. Price $125-00, including post and pack registered.
JOHN A. CRANSTONE, [redacted], [redacted]

****

DREAMSOFT EPROM

After further discussions with Graeme Leadbeater, we wish to advise the following corrections to our review of the DREAMSOFT EPROM.

The teletype ASCII - BAUDOT conversion is automatic and means that the same code is used for the printer and the VDU.

SUPERTYPE DREAM FORMATS.

A. Normal Block Dump - Address, then 16 separate bytes per line.

B. Machine Code Disassembler - Address, then 1, 2 or 3 bytes per line with Branch mnemonics and destinations calculated and printed.

C. CHIP - 8 BLOCK DUMP - Address, then 8 double byte instructions per line.

D. Address, then 1 double byte Chip-8 instruction per line.
Note that format 'D' is not a true Chip-8 Disassembler. This is, however, a current project at the Dreamsoft Lab and Graeme and Graham hope to publish it soon in the 'DREAMER.'

DREAM SOUND EFFECTS GENERATOR

by Michael Bauer
PO Box 343, Belmont, Vic. 3216

Here is a way of giving your computer a diverse range of sounds. It will add a new dimension to games and open up many possibilities for use in its own right; e.g. "music" generation. Many readers will be curious as to why I chose the T.I. 76477 rather than the General Instrument AY-8910chip, or some other scheme. The simplest way to generate sounds, from the hardware point of view, is to tack a digital-to-analog (D/A) converter onto an output port. This requires virtually continuous service from the microprocessor unit (MPU) to produce audio waveforms. Since many of our applications will require a lot of MPU power and speed (e.g: DREAM Invaders) we can't use this method. To take the load off the MPU, it becomes necessary to utilize external hardware for sound generation. Fortunately, there are a few ICs around which perform this function; the main two being the 76477 and the AY-8910.

The 76477 is like a micro-miniature 'Moog' synthesizer on a chip, but it is very very primitive; (see block digram enclosed within dotted lines in circuit). Since it is essentially an analog device (i.e. voltage controlled), it has a lot of external Rs and Cs to define time-constants, etc, and it is not readily able to be interfaced to the MPU bus. Conversely, the AY-8910 is entirely digital and is directly interfaced to the MPU bus. The chip contains a bank of registers which define the frequency of up to 3 oscillators and the noise and envelope generator characteristics. Since the AY-8910 is purely digital, it is highly accurate and stable.

So why choose the 76477 ? Does the author have rocks in his head? Read on for the explanation! The AY-8910 has a severe limitation for use as an effects generator. It is impossible to program the chip so that one of the 3 oscillators can modulate another (AM or FM). A modulation capability is essential to the production of many desired noises. The AY-8910 requires rapid periodic MPU intervention (e.g: using RTC interrupt) in order to produce frequency modulation (FM) effects. The DREAM-6800 already sacrifices a large time-slice (40%) of its processor power to the video display generator (VDG), and its 50Hz relative-time-clock (RTC) is too slow for use with the AY-8910, and so it must, regretably, be ruled out. For applications where the VDG can be turned off and 100% of the MPU power devoted to controlling the AY-8910, however, some fantastic things can be done, especially in the field of music generation where it leaves the T.I. 76477 for dead!

Therefore, like it or not, I was stuck with the task of interfacing the 76477 to the MPU somehow. Only one PIA could be justified, so to cut a long story short, the range of resistance (/capacitance) selectable on each analog-programmed terminal had to be severely compromised. Referring to the circuit, it can be seen that most pins have a fixed R or C, while a few offer a switched choice. Readers who are familiar with the 76477 might jump to the conclusion that I have oversimplified things and that the configuration finally chosen couldn't offer a sufficient diversity of programmable effects. (Wrong!)

Most of the things that you'll ever want to do with the 76477 can be done under MPU control using this configuration. It is quite unnecessary to have a wide choice of such things as: one-shot duration, attack and decay times, output amplitude or noise filter roll-off; so these are fixed. We do need a selection of 'SLF' osc. frequencies and a wide range of VCO freq's. The voltage-controlled osc. (VCO) is controlled by a 'digital-to-resistance' (D/R) converter, utilising half the PIA (port B) and a binary resistor network. Bit-7 selects the VCO timing capacitor, giving one of two ranges. There are 256 possible frequencies for the VCO. Referring to the graph of VCO freq. vs PIA value, we see that accuracy increases towards the high end of each range. The high range covers most of the audio spectrum (100Hz to 10kHz) while the low range gives better resolution in the middle band (250 to 1000 Hz) and goes right down to about 10Hz. In addition to being controlled by the MPU, via the D/R converter, the VCO may be simultaneously frequency modulated or amplitude-modulated by the 'SLF' oscillator.

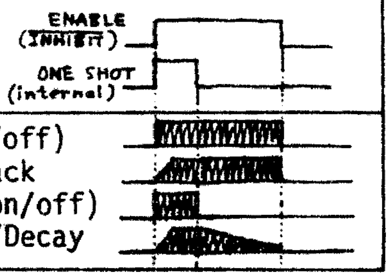
DREAM SOUND EFFECTS GENERATOR -------- PROGRAMMING CHART

'PATCH' Control Byte format (PIA port A, O/P reg.):

Bit:	7	6	5	4	3	2	1	0
	MC	MB	MA	ES	EE	VS	SC	SR

MIXER SELECT BITS			SIGNALS 'MIXED'
MC bit-7	MB bit-6	MA bit-5	
0	0	0	VCO
0	0	1	SLF
0	1	0	Noise
0	1	1	VCO & Noise
1	0	0	SLF & Noise
1	0	1	SLF & VCO & Noise
1	1	0	SLF & VCO
1	1	1	inhibit

ENVELOPE SEL.		*ENV. FUNCTION
ES bit-4	EE bit-3	
0	0	Enable only (on/off)
0	1	Enable with Attack
1	0	One-shot only (on/off)
1	1	One-shot w. Att/Decay



VCO Control Voltage Select (VS, bit-2)	
0	EXTERNAL preset adjustment
1	INTERNAL by 'SLF' oscillator

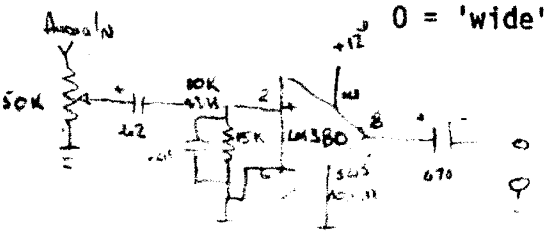
SLF Oscillator Frequency Select		
SC bit-1	SR bit-0	SLF Osc. Freq.
0	0	AUDIO-1 Preset
0	1	AUDIO-2 Fixed
1	0	LOW-1 Preset
1	1	LOW-2 Fixed

VCO FREQUENCY Control Byte format (PIA port B, DDReg.):

7	6	5	4	3	2	1	0
C	R6	R5	R4	R3	R2	R1	R0

RANGE
1 = 'low' (8Hz - 1kHz)
0 = 'wide' (80Hz - 10kHz)

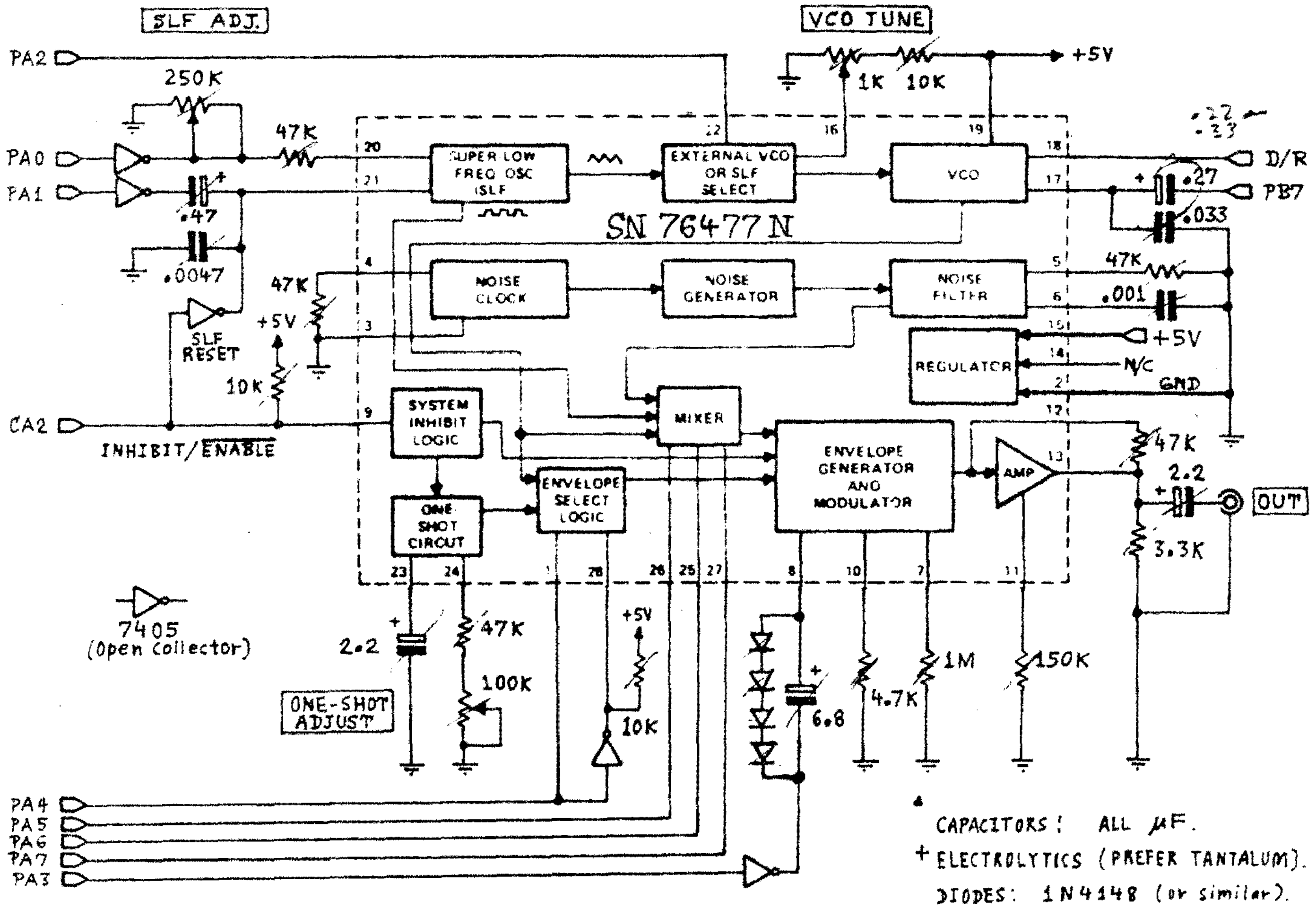
FREQUENCY
1 = Rx in circuit
0 = Rx not in cct



MJB '81

'DREAM' SOUND EFFECTS GENERATOR (M.J. Bauer 1980)

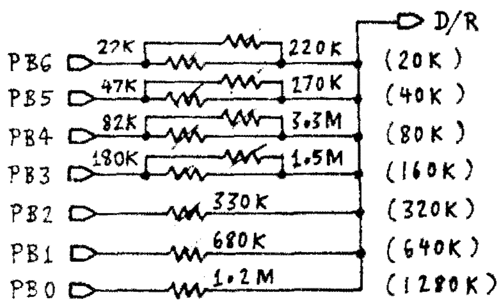
Circuit Diagram



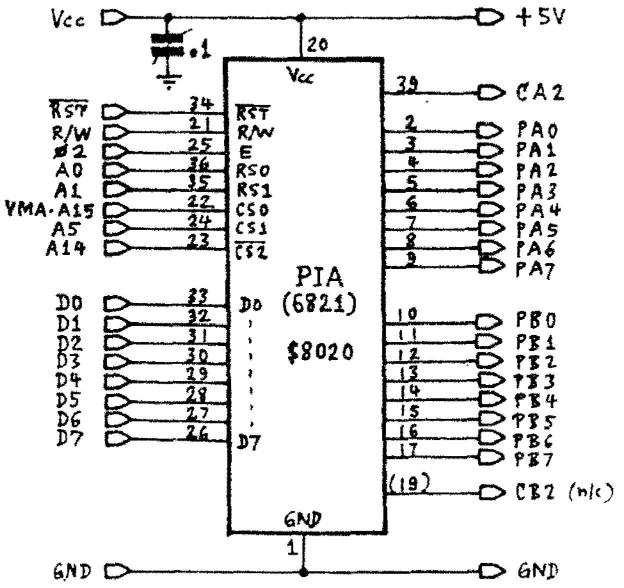
CAPACITORS: ALL µF.

+ ELECTROLYTICS (PREFER TANTALUM).

DIODES: 1N4148 (or similar).



'DIGITAL-TO-RESISTANCE' CONVERTER (D/R)



A couple of refinements have been made to the 76477, externally. Firstly, 3 diodes have been strapped across the envelope-generator capacitor to improve its otherwise abysmal performance. Secondly, an open-collector gate (SLF RESET) discharges the SLF osc. timing capacitor whenever the device is disabled. This forces the VCO to commence oscillating at the same freq. each time the device is triggered (assuming SLF controlling VCO). This feature is essential for many 'one-shot' sound effects.

Finally, this circuit will cost less than the AY-8910, including PIA and Rs and Cs. The 76477 is about $3 to $4, the AY-8910 would have been about $15 (if you can find one at all) and the going rate for a PIA is about $6.

PRACTICAL CONSIDERATIONS

By the time this issue is released, we hope there will be a PCB design available. If not, the circuit is simple enough to whip up on Vero DIP board, especially if you have an expansion board with a PIA already to go. This PIA should be located at $8020, else you will need to alter the software accordingly.

The circuit as it stands does not include an audio amplifier, because the output is designed to be fed into an external amplifier, for example the one in your T.V. set. The 2 transistor output stage given in the T.I. 76477 data sheet does not have enough guts! If you need a separate amplifier, an LM380 should do nicely. The use of 1% tolerance metal film resistors is recommended in the D/R converter, especially if you anticipate playing tunes with it. Set the trimpots initially to the half-way position; these can be tweaked for optimum effect later. By the way, try to get the data sheet and application notes with your 76477, for the useful info therein.

SOFTWARE

The PIA registers are programmed as indicated in the 'Programming Chart'. The PATCH byte is the A-side output register; the data direction register (DDR) being maintained at $FF (all outputs). The VCO-FREQ control byte is the B-side data-direction register (DDR); the output register being kept at 00. Writing a '1' into a bit position in the DDR makes the corresponding I/O line an output, thus grounding an external resistor (or C, if bit-7). Writing a '0' bit in the DDR makes the line an INPUT which is high impedance (floating), thereby effectively removing the resistor from circuit. Neat, huh?

To save you the bother of figuring out how to initialize the PIA, I've written 3 simple subroutines called 'low-level drivers' to handle this task. From the Programming Chart and VCO graph, you can work out the PATCH and VCO-FREQ data desired. Your program must call the subroutines DISAFX and INIZFX at the start, to initialize the PIA. To set up a new patch and enable the device, your program simply needs to load acc-A with the PATCH byte and call the subroutine ENABFX; thus:-

```
LDA A    PATCH
JSR      ENABFX
```

Thereafter, the patch may be altered without disabling the device, if desired, by writing to location $8020 the new patch data. The VCO frequency (and range) may be set or altered at any time simply by writing to location $8022. The low-level driver subroutines are relocatable.

Some unreal zany effects can be produced by sweeping the VCO under program control with any of the following patches (at least): 00, 60, C0, C1, 04, 07. Sweeping is accomplished simply by incrementing/decrementing the VCO freq. at periodic intervals.

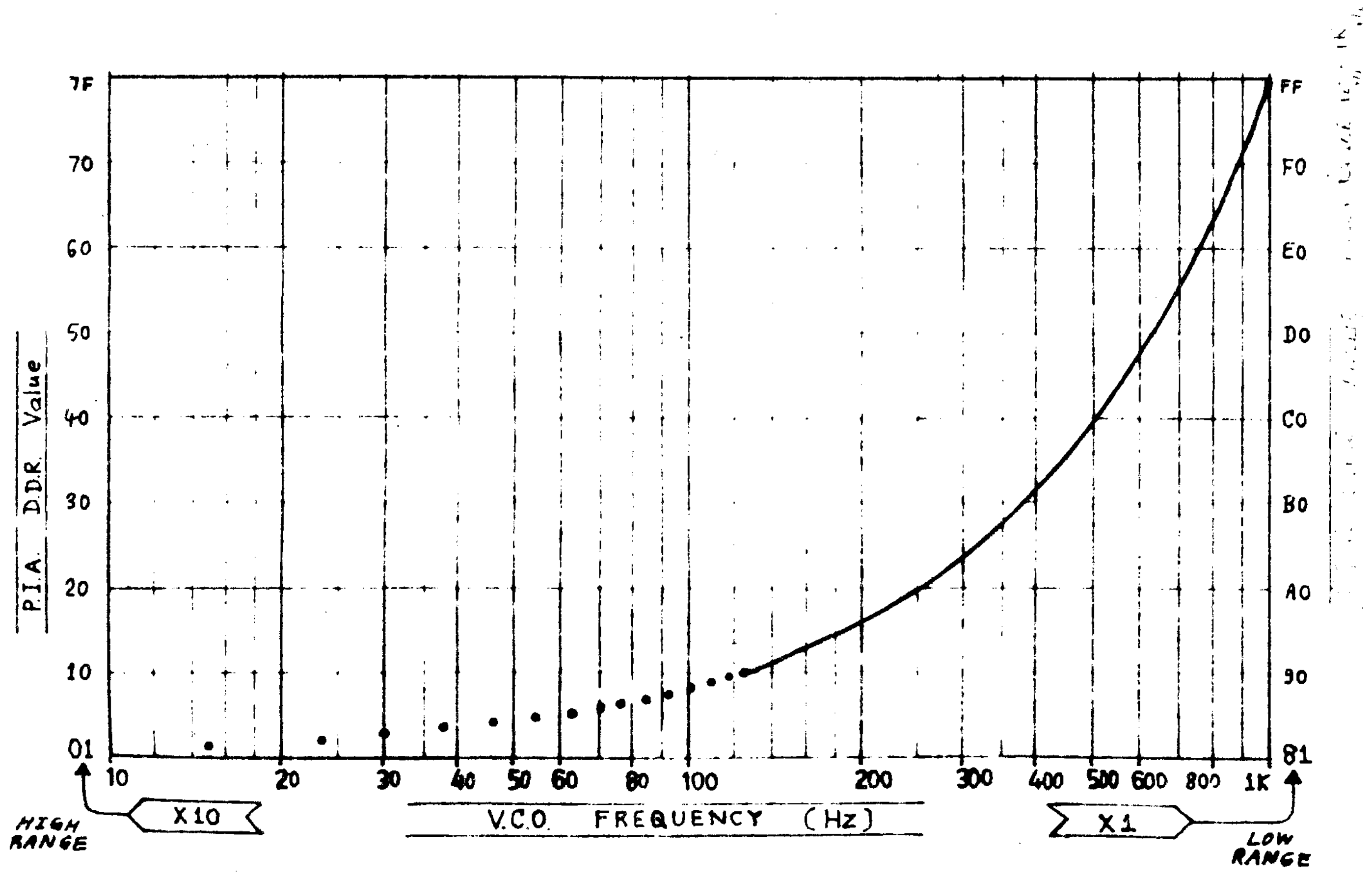
TEST & DEMO PROGRAM

The test program generates one of 16 pre-defined effects stored in a look-up table. The listing shows the PATCH and VCO values and the corresponding key to press to get each sound. You can easily replace any or all of the table entries with your own contrived effects. Note that the test program merely sets up the sound effects generator and enables it; the PATCH and VCO-FREQ remain constant until a different key is pressed. All of these effects are being produced by the 76477 on its own, without any MPU assistance whatsoever.

```
                         *
                         ***********************************************
                         *    DREAM-6800 SOUND-EFFECTS GENERATOR       *
                         *    TEST & DEMO PROGRAM +                    *
                         *    LOW-LEVEL DRIVER SUBROUTINES.            *
                         *         M.J. BAUER,   1981.                 *
                         ***********************************************
                         *
                                  TITLE    76477 SOUND FX GEN DRIVERS
                         *
                         FXPIA    EQU      $8020
                         GETKEY   EQU      $C2C4
                         KEYINP   EQU      $C297
                         BADRED   EQU      $0018
                         PAINZ    EQU      $C287             INIZ KEYPAD
                         ADDAI    EQU      $C189             16 BIT ADD A TO PTR (I)
                         I        EQU      $26               16 BIT POINTER
                         *
0200                              ORG      $0200
 0200   8D 54            TESTFX   BSR 0256 INIZFX
 0202   8D 4C                     BSR 0250 DISAFX
 0204   BD C2 87                  JSR      PAINZ
 0207   B6 80 11         WAIT1    LDA A    $8011             WAIT FOR KEYDOWN
 020A   2A FB                     BPL 0207 WAIT1
 020C   BD C2 97                  JSR      KEYINP            FETCH KEYCODE ==> A
 020F   48                        ASL A                      MULT BY 2
 0210   CE 02 30                  LDX      #TABLE            USE TO LOOK UP TABLE
 0213   DF 26                     STX      I
 0215   BD C1 89                  JSR      ADDAI
 0218   DE 26                     LDX      I
 021A   A6 01                     LDA A    1,X               SET VCO FREQ
 021C   B7 80 22                  STA A    FXPIA+2
 021F   A6 00                     LDA A    0,X               SET PATCH & ENABLE FX
 0221   8D 42                     BSR 0265 ENABFX
 0223   BD C2 97         WAIT2    JSR      KEYINP            WAIT FOR KEY RELEASE
 0226   7D 00 18                  TST      BADRED
 0229   27 F8                     BEQ 0223 WAIT2
 022B   20 D3                     BRA 0200 TESTFX            AGAIN...
                         *
0230                              ORG      $0230
                         *. LOOK-UP TABLE,   DATA FOR PATCH & FREQ (16 x 2):
                         *                   ***  KEY  ***  CONTINUOUS ENABLE:-
 0230   00FF             TABLE    FDB      $00FF   0   VCO, 1KHz
 0232   0040                      FDB      $0040   1   VCO, 5KHz
 0234   2000                      FDB      $2000   2   SLF, AUDIO-1 (Preset)
 0236   4000                      FDB      $4000   3   NOISE
 0238   0440                      FDB      $0440   4   FM, SLF AUDIO, VCO 5KHZ
 023A   06FF                      FDB      $06FF   5   FM, SLF LOW-1, VCO 1KHZ
 023C   C0FF                      FDB      $C0FF   6   AM, SLF AUDIO-1, VCO 1KHZ
 023E   C183                      FDB      $C183   7   AM, SLF AUDIO-2, VCO 23HZ
                         *                   ***  KEY  ***  ONE-SHOT ENVELOPE:-
 0240   161A                      FDB      $161A   8   FM, SLF LOW-1, VCO 2KHZ
 0242   1714                      FDB      $1714   9   FM, SLF LOW-2, VCO 1.5K
 0244   7084                      FDB      $7084   A   NOISE & VCO (30HZ)
 0246   D10D                      FDB      $D10D   B   AM, SLF AUDIO-2, VCO 1K
                         *                   ***  KEY  ***  ATTACK/DECAY ENVELOPE:-
 0248   780D                      FDB      $780D   C   NOISE & VCO (1KHZ)
 024A   1C70                      FDB      $1C70   D   FM, SLF AUDIO-1, VCO 9KHz
 024C   D987                      FDB      $D987   E   AM, SLF AUDIO-2, VCO 50HZ
 024E   DF20                      FDB      $DF20   F   EVERYTHING! (well, almost!)
```

```
                        * LOW-LEVEL I/O & SUBROUTINES
0250                          ORG     0250
                        *
                        * DISABLE SOUND EFFECTS GENERATOR:
0250   C6 3C            DISAFX  LDA B   #$3C
0252   F7 80 21                 STA B   FXPIA+1
0255   39                       RTS
                        *
                        * INITIALIZE SOUND-FX GEN (PIA-B DDR = VCO-FREQ.)
0256   C6 04            INIZFX  LDA B   #4              ACCESS O/P REG.
0258   F7 80 23                 STA B   FXPIA+3
025B   7F 80 22                 CLR     FXPIA+2         O/P LINES LOW
025E   7F 80 23                 CLR     FXPIA+3         ACCESS DDR
                        * SET VCO FREQ & RANGE (ACC-A ==> DDR)
0261   B7 80 22                 STA A   FXPIA+2
0264   39                       RTS
                        *
                        * ENABLE SOUND-FX GEN. (ACC-A ==> O/P REG)
0265   CE 80 20         ENABFX  LDX     #FXPIA          GET PORT ADRS
0268   C6 38                    LDA B   #$38            INHIBIT & SEL DDR
026A   E7 01                    STA B   1,X
026C   C6 FF                    LDA B   #$FF            WRITE DDR (ALL OUTPUTS
026E   E7 00                    STA B   0,X
0270   C6 3C                    LDA B   #$3C            SEL O/P REG
0272   E7 01                    STA B   1,X
0274   A7 00                    STA A   0,X             WRITE O/P REG
0276   C6 34                    LDA B   #$34            ENABLE FX
0278   E7 01                    STA B   1,X
027A   39                       RTS

                                END
```



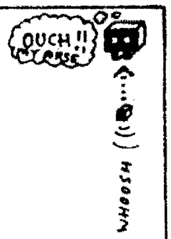
MELODY MAKING

The D/R converter should give adequate resolution over the top 2 octaves of the LOW range (250Hz to 1kHz, nominally) for musical purposes. Due to variations in resistor values and unknown 76477 anomalies, the VCO values for the musical scale cannot be accurately calculated. So far, I have not been able to investigate the musical potential of the sound generator due to lack of time, but here are some guidelines for experimenters who have a little music theory and like to dabble in machine language programming.

I suggest you write a program which accepts from the keyboard and displays 2-digit hex numbers, and 'plays' the corresponding VCO frequency. Then, with the aid of a piano, organ, guitar, bag-pipes, stylophone (or whatever), determine by trial-and-error those values which lie on the musical scale. These values can be used to construct a look-up table for use with your melody-making programs. Lets hear from you if you succeed in developing any useful software for the sound generator.

The sound generator described in this article is a very inexpensive enhancement to the DREAM-6800, which opens up many possibilities for experimentation. It should increase your motivation to learn about machine-code programming, and interfacing the digital and analog worlds.

SOUND EFFECTS for 'DREAM INVADERS'



```
Turret hit / Alien landed sound:
0700   86 7F BD 07 D6 86 06 BD 07 E5 C6 40 7F 00 20 7D
0710   00 20 27 FB 7A 80 22 7A 80 22 5A 26 EF 7E 07 D0

Initialization of I/O ports:
0720   BD C2 87 7E 07 D6

Alien descending sound:
0726   7C 00 9A CE DF 20 7E 07 51

Fire missile sound:
073D   7C 00 B3 CE 78 0E 7E 07 51

Alien hit sound:
0746   CE 16 14 BD 07 51 C6 03 7E 05 E4

Make sound specified in X-reg.:
0751   BD 07 D0 DF 26 96 27 B7 80 22 96 26 7E 07 E5

Essential modification :
0760   96 B3 91 B4 2D 09 96 21 84 78 26 03 7E 03 06 7E
0770   03 0C

Low-level drivers:  (see also 'Test & Demo' listing)
07D0   C6 3C F7 80 21 39 C6 04 F7 80 23 7F 80 22 7F 80
07E0   23 B7 80 22 39 CE 80 20 C6 38 E7 01 C6 FF E7 00
07F0   C6 3C E7 01 A7 00 C6 34 E7 01 39 -- -- -- -- --

Dream Invaders program changes:
0200   BD 07 20
0300   7E 07 60
0387   BD 07 26
0398   BD 07 00
03FB   BD 07 3D
045E   01 01 01 01 01
0468   BD 07 46
050D   01 01 01 01 BD 07 00
```

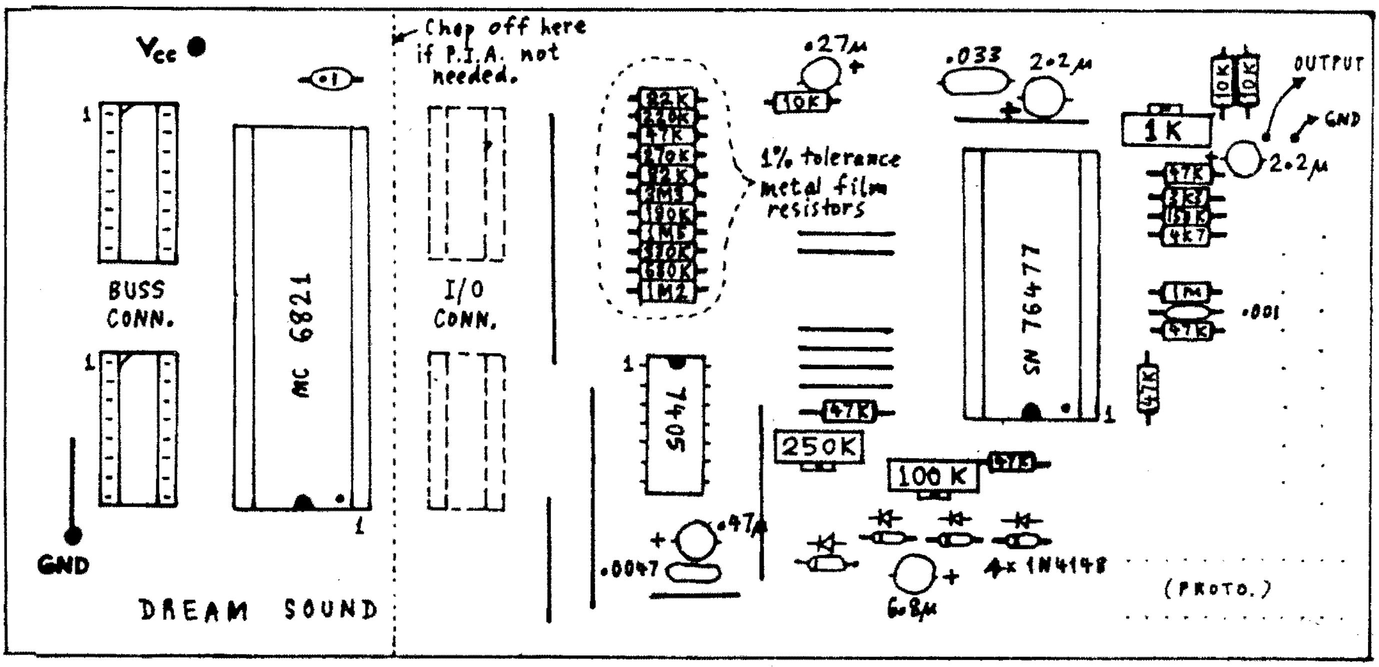
N.B: Sound generator I/O port addresses are marked thus: 80 XX.

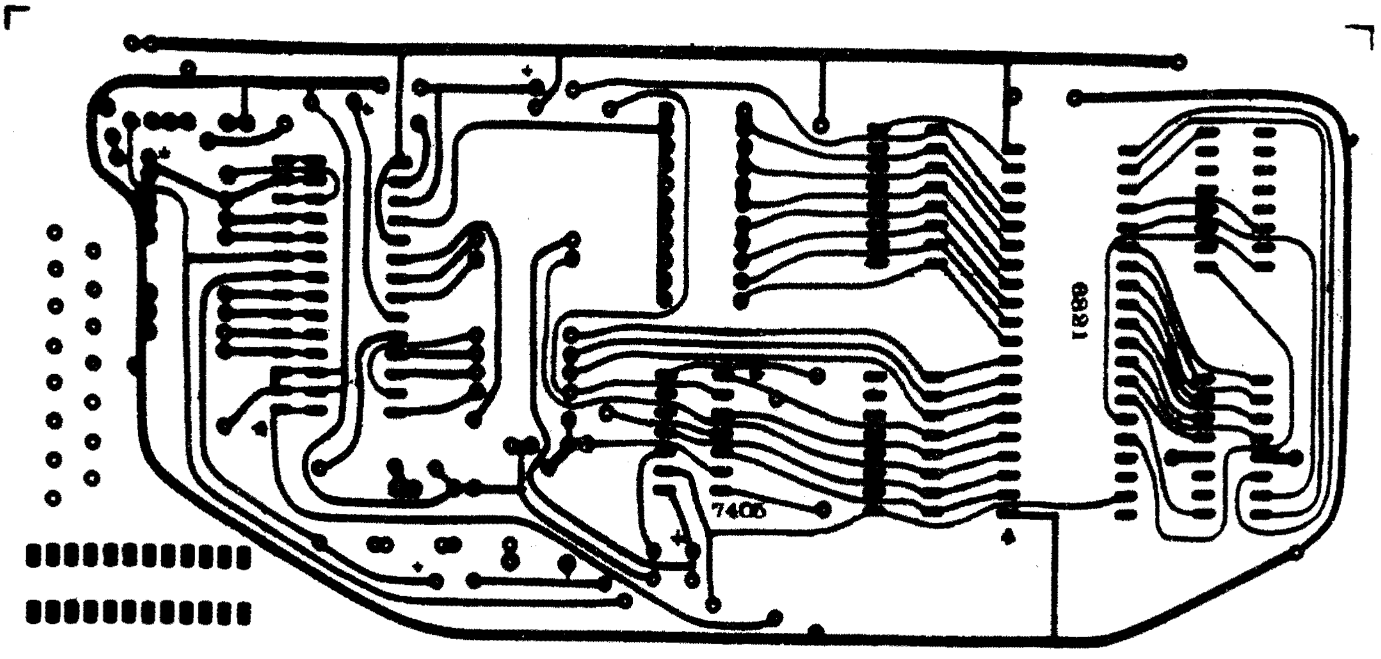
MTB

PRINTED CIRCUIT BOARD for the DREAM SOUND GENERATOR

Thanks to lan Combridge (of the Division of Electrical Engineering, Deakin University) we have a PCB layout for the sound generator. The board is designed to suit either those who need the extra PIA, or those who have a PIA on their existing expander board, in which case the PIA end of the sound board may be sawn off. A small proto area is included for experimenters to add extra circuitry (e.g; an LM380). Observe that the I/O sockets are configured the same as the J.R. expander board. The buss connectors are identical to the DREAM board, so that it is possible to "daisy chain" the mother board, a RAM expander board, and the sound board. Make sure you get the Vcc and GND connections right! Sockets are recommended for all IC's.

Note: There is as yet no commercial outlet for this PCB. Please do not write to the User Group or 'Dreamware' requesting PC boards. The layout is given here for those who are able to fabricate their own boards or who have access to facilities via friends in industry, etc. Certain PCB suppliers (e.g. 'RCS Radio') will produce "one-off" boards from your layout (at a price!). There is nothing printed on the back side of this page in the vicinity of the PCB pattern, so that it should be possible to produce a negative using the contact-print method.





SIMPLE SIMON (0200 - 0400)

I. J. CAMPBELL.

Press any key to start. A hex.digit will flash on the screen. You respond by pressing the key of the value flashed, then two digits will flash and you must press the keys in the correct order. If you do, three digits will flash, and you must press the three correct keys, and so on.

You win when you successfully reproduce eight digits in their correct order. You lose if you do not reproduce a digit correctly.

The upper row of digits is the computer's choice, the lower row is the digits you have selected.

SIMPLE SIMON restarts automatically - win or lose.

```
0200    FE0A 6A00 C00F C10F  C20F C30F C40F C50F
0210    C60F C70F 7A01 6803  6903 F029 D895 6E0B
0220    FE18 4A01 1276 680B  F129 D895 FE18 4A02
0230    1276 6813 F229 D895  FE18 4A03 1276 681B
0240    F329 D895 FE18 4A04  1276 6823 F429 D895
0250    FE18 4A05 1276 682B  F529 D895 FE18 4A06
0260    1276 6833 F629 D895  FE18 4A07 1276 683B
0270    F729 D895 FE18 6C0A  FC15 FC07 3C00 127A
0280    00E0 6807 6903 6806  FD0A FD29 D895 2250
0290    5D00 1364 4A01 1380  680E FD0A F129 D895
02A0    2250 5D10 1364 4A02  1380 6816 FD0A F229
02B0    D895 2250 5D20 1364  4A03 1380 681E FD0A
02C0    F329 D895 2250 5D30  1364 4A04 1380 6826
02D0    FD0A F429 D895 2250  5D40 1364 4A05 1380
02E0    682E FD0A F529 D895  2250 5D50 1364 4A06
02F0    1380 6836 FD0A F629  D895 2250 5D60 1364

0300    4A07 1380 683E FD0A  F729 D895 2250 5D70
0310    1364 2372 6300 F129  6200 6304 D235 A356
0320    D232 6000 6210 F029  [illegible] 6214 F029 D235
0330    A358 D231 6418 650C  A359 D455 641E F129
0340    D455 A35E 6420 D455  2372 6E20 FE15 1200
0350    FD29 D8B5 00EE E0E0  4088 8888 88D8 88C8
0360    A888 5800 [illegible] [illegible]  6414 6504 F129 D455
0370    1374 0050 [illegible] FC07  3C00 1376 00E0 00EE
0380    6C1F [illegible] [illegible] 7C00  1384 00E0 1214 [illegible]
0390    FE0A 6C10 2374 1202
```

TORTOISE   (0080 - 0400)

P. E. MARSTON.

Press key '8' 4 to set your racing car in motion. It moves very slowly, but beware, the speed increases after each lap completed.

Steer the car around the circuit using these keys:-

'8' 4 LEFT, '4' 6 RIGHT, '8' 9 UP, '5' 1 DOWN.

The game ends after 5 laps. (Lap score shown L.H.S. of screen.) If you crash, the car stops until you backup and then re-negotiate the corner. HINT : Do not cut corners. The game also ends if 5 crashes are recorded. (Crash total shown R.H.S. of screen.)

Your driving skill is assessed, and shown at the end of the game.

To increase difficulty, change 0271 to 05, and 032D to FF.

To change key functions, insert the key you wish to use for the direction indicated, at the following locations.

RIGHT 0289, LEFT 0295, UP 028F, DOWN 029B.

```
0080   6102 63EA 1364 619A  6302 1364 0000 0500
0090   F00A ~~00E0 1200 0000~~
            3002 1040 00D2 1200

0200   6A02 6B00 680F 2332  3800 1206 6A01 233C
0210   233C 233C 6A06 6B04  680D 2332 3800 121A
0220   6A05 233C 233C 6A39  6B05 DAB8 6A25 6B0D
0230   DAB8 6805 2332 3800  1234 6A3D 6B01 233C
0240   233C 6A29 6B11 DAB8  6805 2332 3800 124A
0250   6A05 6B14 6808 2332  3800 1256 6A01 6B18
0260   680A 2332 3800 1262  6600 6700 22D2 22DC
0270   680A 6A34 6B02 6C00  6D00 A34E DAB2 4605
0280   1352 4A38 12F0 DAB2  6E0A EEA1 22FE 6E05
0290   EEA1 2308 6E08 EEA1  2312 6E0D EEA1 231C
02A0   8AC4 8BD4 DAB2 8580  F515 F507 3500 12AA
02B0   4901 1286 4F00 127E  6901 F818 DAB2 6C00
02C0   6D00 22DC 7701 22DC  4705 1352 A34E DAB2
02D0   1286 A08C F633 632E  22E6 00EE A08C F733
02E0   6337 22E6 00EE 6414  F265 F229 D345 00EE
02F0   4B01 1326 4B02 1326  4B03 1326 1286 6900

0300   A34E 6D00 6C01 00EE  6900 A350 6C00 6DFF
0310   00EE 6900 A34E 6D00  6CFF 00EE 6900 A350
0320   6C00 6D01 00EE 22D2  7601 22D2 78FE A34E
0330   1286 A344 DAB1 78FF  7A04 00EE A346 DAB8
0340   7B08 00EE F000 8080  8080 8080 8080 C000
0350   8080 F818 3700 4701  1080 3702 4703 1086
0360   6172 639A 6003 6203  6401 65D0 03EA F818
0370   1090 0EFB BBAB 8100  0000 0AAA 922A 8100
0380   0000 0EAB 93AB 8100  0000 0AAA 922B 0000
0390   0000 0AAA 93BA 8100  0000 0AEE A0EE EC04
03A0   0000 0A8A A08A AA04  0000 0AEE E0AA AA04
03B0   0000 0A8C 40AA AA00  0000 04EA 40EE EC04
03C0   0000 0EAE FBBB B810  0000 08AA AA92 A810
03D0   0000 08EE AB92 A810  0000 08AA AA12 A800
03E0   0000 0EAA AA3B A810  0000 DE30 9032 270E
03F0   A600 08DF 30DE 34A7  0008 DF34 20EC 3900
```

RESCUE MISSION (0080 - 0400)

K. BOLCH,
[redacted]

In 2025 A.D., the main dome of the C.M.C. (Callisto Mining Corporation) mining complex blew open. Casualties were high, but a few managed to reach their life support cells. These were small plastic units capable of supporting life for 48 hours. The only available help was from an orbiting N.A.S.A. spacelab. This is your mission ; to pilot your two man landing module through the asteroid belt to the surface, land on a life support cell and shuttle the man and (collapsible) cell back to the mother ship.

This program utilises M.J.Bauer's Joystick Controller. To release from the mother ship press 'F', (Make sure the controls are centered first.) and guide your landing module onto a cell.

The HORIZONTAL pot controls your DIRECTION,
The VERTICAL pot controls your VERTICAL SPEED.

```
On the way DOWN, UP = STOP
                 CENTERED = DOWN, SLOW.
                 DOWN = DOWN, FAST.

On the way UP,   UP = UP FAST.
                 CENTERED = UP, SLOW.
                 DOWN = STOP.
```

When you have landed, press 'F' to launch your module. You start with 75 fuel units and 3 landing modules. Only by moving the vertical pot from the centered position will you use fuel, and when all your fuel is used you will not be able to regulate your speed.

You may be destroyed in the following ways; Landing or docking at full speed, (50% chance) ; Missing a cell or the mother ship ; A direct hit by an asteroid ; A glancing hit by an asteroid, (50% chance). You score one point per cell rescued, plus a bonus of two points if you have fuel left at the end of a layer. On the third layer you receive a bonus ship and the game is faster each layer.

If you are destroyed, or when you safely shuttle a cell to the mother ship, the number of modules, the amount of fuel you have left, and your score are displayed, (in that order).

For a real challenge, try changing the instruction at 00BC to C003.

```
0090   0000 0000 0000 0000  A08C FB55 A080 6800
00A0   C03F C10F 4100 6108  4101 6109 4102 6111
00B0   7801 F155 3804 10A0  C001 3201 0000 4000
00C0   60FF 4003 60FE F055  7801 3808 10B8 A080
00D0   FB65 A300 D01A A301  D23A A302 D45A A303
00E0   D67A A08C FB65 00EE  00E0 694B 6A00 7201
00F0   A2C8 601D 6E00 DE03  7E06 3E42 10F6 00EE

0200   0204 1264 C63E F780  1306 61F7 8012 C63F
0210   F780 130E 0219 DF00  397A 0020 7A00 217D
0220   8012 7C00 1696 1684  0327 013B 86FF 973C
0230   973D 7F80 1286 21B7  8012 C64A B680 1246
0240   4624 037C 003C 4624  037C 003D 5A26 ED96
0250   3C80 0A2C 014F 973C  963D 800A 2C01 4F47
0260   973D 3B10 6400 6200  6803 20E8 230C 6F0F
0270   EFA1 22A2 22CE 22A8  2326 4C00 66FF 4C3F
0280   6601 4900 126E 4D00  67FE 4D1F 6700 41FF
0290   129A 4D00 6700 4D1F  6702 3D00 4D1F 79FF
02A0   126E 4100 6101 00EE  4208 6207 6E08 8E25
02B0   FE15 FF07 3F00 12B2  00EE 0000 CE01 006F
02C0   0008 8C01 E626 F839  7F41 41F8 A8F8 A08C
```

(See bottom of page 21 for listing from 02D0 to 0400)

PING - PONG (0200 - 0400)

MARTIN HEAD,

[redacted]

Play PING - PONG against your computer. The first to score 21 wins.

Key 9 moves your bat UP, Key 1 moves it DOWN.

For a slightly more challenging opponent, change 030D from 03 to 04. Alternatively, changing it to 02 will give you an easier time.

```
0200   6400 6900 621F 6104  A34A D121 7108 313C
0210   120A 4206 121A 6206  1206 6E0A 229E 6E2B
0220   229E 6103 6211 6A3C  6B11 6703 6811 D123
0230   DAB3 F00A C502 4500  6501 C601 C001 3001
0240   22C2 8754 8864 D781  6001 E09E 1254 D123
0250   72EE D123 6009 E09E  1260 D123 7201 D123
0260   6302 F015 F007 3008  1264 481F 22C2 4806
0270   22C2 4702 22CA 4703  22CA 473D 22E4 473C
0280   22E4 D781 4301 1222  4300 1234 DAB3 A350
0290   FB55 0324 A350 FB65  A34B DAB3 1242 6C00
02A0   A350 3E0A 12BE F433  A351 F065 F029 DEC5
02B0   7E04 A352 F065 F029  DEC5 A34B 00EE F933
02C0   12A8 8060 6600 8605  00EE 8C20 2302 230A
02D0   3301 00EE 6E2B 229E  7901 6E2B 229E 4915
02E0   F000 00EE 8CB0 235C  230A 4301 12F2 6302
02F0   00EE 6E0A 229E 7401  6E0A 229E 4415 F000

0300   00EE 8050 6500 8505  00EE 6301 6D03 9C80
0310   6300 7DFF 7C01 3D00  130E 3301 00EE D123
0320   DAB3 00EE CE03 0086  3DA0 5781 0C2B 0EA6
0330   5B81 1127 042B 036A  5B39 6C5B 39A6 5B4C
0340   A158 2B03 6A5B 396C  5B39 FF80 8080 0000
0350   0000 0000 0000 0000  0000 0000 C502 4500
0360   6501 2302 C601 C001  3001 22C2 00EE 8282
```

0420 F418 [illegible]

****

RESCUE MISSION (Cont)

```
02D0   FB55 A080 FB65 A300  D01A 8084 D01A A301
02E0   D23A 8294 D23A A302  D45A 84A4 D45A A303
02F0   D67A 86B4 D67A A080  FB55 A08C FB65 00EE

0300   C0C0 0000 000C 0C00  0080 C0C0 02BC 6320
0310   6E00 A2C8 D3E3 C002  2098 6100 6523 6B01
0320   A3FA D5B2 70FF 3100  1340 A2C8 6E00 4300
0330   6001 4338 60FF D3E3  8304 D3E3 6700 8600
0340   A3FA 41FF A3FD D5B3  8564 8B74 45FF 7501
0350   453E 75FF D5B3 4F01  2370 4BFF 13B0 4B1F
0360   13B0 4B20 13B0 4B00  13B0 6600 8710 00EE
0370   4B00 7801 3B1C 13B0  D5B3 6E00 A2CB 601D
0380   DE03 A3FA D5B3 4F00  1394 D5B3 A2CB DE03
0390   7E0B 137C D5B3 A3FD  DE03 85E0 8B00 6100
03A0   22CE 22A8 6F0F EF9E  13A0 61FF 7A01 00EE
03B0   D5B3 78FF 23BE 4800  13B4 230A 00EE 4B00
03C0   7401 3A06 13D0 3900  7402 20E8 4203 7801
03D0   02BC 6E00 8180 23DE  8190 23DE 8140 6B1A
03E0   A035 F133 F529 DBE5  7B04 F629 DBE5 7B04
03F0   F729 DBE5 7E0B 22B0  00EE 40A0 0020 50F8
```

0080-0400

THE WELL-HEXADECIMALISED KEYBOARD or, IS YOUR BACH WORSE THAN YOUR BYTE ?

F. Rees, [redacted]

B. Mitchell, [redacted]

0080-00C5 is a machine code subroutine developed by Frank which looks at two consecutive bytes pointed to by the index register. The first byte gives the frequency at which PB∅ is switched, the second gives the duration. Connect an audio amp to PB∅ and experiment with filters to your ears' content. The subroutine is relocatable and can be entered at 0083 if your music listing begins at other than 0300. By changing 0095 to 40, you can dispense with the external audio amp and suffer a close approximation to bad punk complete with 1200 Hz intermodulation.

0200-02D7 is an aid to entering the music listing. It is mainly in CHIP 8 so enter in the normal way from C000. 00D0-00E3 is the lookup table for the key of G, but can be altered as outlined further on to any other key. The duration values in this table should give the duration for a quaver: for higher notes it is necessary to string together quavers (using the "1" command) to make crochets and minims as the duration bytes overflow when multiplied.

To enter a tune, clear the 0300-03FF workspace by entering "00". To play a previously entered tune, press "01". The current location of the memory pointer is displayed. Other commands are "1", which joins adjacent notes together without a pause; "2", which plays all the tune entered so far then waits at the end for further entries and "3", which is a backspace and erase function. Notes are entered in two keystrokes. The first denotes the note, the second the duration.

First keystroke

Key	4	5	6	7	8	9	A	B	C	D	E	F
Note	rest	C	D	E	F+	G	A	B	C'	D'	E'	F+'

Second keystroke

0 = ♪ (all keys)

1= ♩ (not with D' E' F+')

2 = ♩. (only with C D E and Rest.)

3= 𝅗𝅥 (only with C and Rest)

Other notes can be inserted into the lookup table to suit your needs from the following list (which is approximate when you consider the fun trying to work out $\sqrt[12]{A7}$ etc.).

Note	Freq.	Duration	Note	Freq.	Duration	Note	Freq.	Duration
C,	FF	20	C+	79	44	C+'	3C	88
C+,	F1	22	D	72	48	D'	39	90
D,	F4	24	D+	6C	4C	D+'	36	98
D+,	D8	26	E	66	51	E'	33	A1
E,	CD	28	F	60	55	F'	30	AB
F,	BE	2B	F+	5B	5B	F+'	2D	B6
F+,	B6	2D	G	55	60	G'	2B	BF
G,	AB	30	G+	51	66	G+'	28	CD
G+,	A1	33	A	4C	6C	A'	26	D8
A,	98	36	A+	48	72	A+'	24	F4
A+,	90	39	B	44	79	B'	22	F1
B,	89	3C	C'	40	80	C''	20	FF
C	80	40						

RANDOM OPUS

(0200 - 0300)
BRUCE MITCHELL.

Here is a fun one from Bruce, which uses the internal speaker of your DREAM, for those of you who haven't built an amplifier yet! (If you have, you can still run it with the amp., just change 022F to 02.)
NOTE: In both cases, the zeroes from 0268 to 026B are essential!

(By the way, Bruce tells us that he wrote this program especially for all those people who think that computers are useless. - Garry.)

```
0200   1260 C03F 4000 1202  C1FF 4100 1208 A030
0210   F165 A268 F155 021A  1200 CE02 68A6 0197
0220   194D 2603 0101 39E6  005D 2602 201C 8640
0230   B780 125A 26FD E600  4FB7 8012 5A26 FD7A
0240   0019 2704 E600 20E6  8610 0808 E601 C101
0250   2709 4A27 C85F 5C26  FD20 F708 0820 BE00
0260   A266 D011 1202 8000  0000 0000
```

****

THE WELL HEXADECIMALISED KEYBOARD.

(0080 - 0400)

PROGRAM LISTING

```
0080   CE03 00A6 0197 194D  2603 0101 39E6 005D
0090   2602 201C 8602 B780  125A 26FD E600 4FB7
00A0   8012 5A26 FD7A 0019  2704 E600 20E6 8610
00B0   0808 E601 C101 2709  4A27 C85F 5C26 FD20
00C0   F708 0820 BE00 0000  0000 0000 0000 0000
00D0   803F 7248 6651 505D  5560 4C6C 4479 407F
00E0   3990 33A1

0200   0000 6A00 6808 2278  FE0A 4E00 12CA 4E01
0210   1246 4E02 1254 4E03  125E 4E04 1262 A0C6
0220   8EE4 FE1E F165 226A  0000 FE0A 6B03 8BE5
0230   3F01 129C 8010 4E00  1240 8014 7EFF 1236
0240   226A 02B0 1206 6000  226A 6001 226A 1206
0250   0292 1202 6000 226A  0000 226A 0292 7AFE
0260   1296 6000 226A 613F  122A A300 FA1E F055
0270   7A01 4AF0 FA18 00EE  F818 6903 00E0 0282
0280   00EE BDC3 E0CE 0039  BDC3 C8CE 003A BDC3
0290   C839 7E00 8000 4AFE  6A00 1206 02A0 122A
02A0   C640 D721 BDC2 E539  0000 0000 0000 0000
02B0   DE39 0909 7E00 8300  6000 6100 A300 F11E
02C0   F055 7101 4100 1202  12BC FE0A 4E00 12B8
02D0   4E01 0292 1202
```

****

MORE OPUS SONGS

FRANK REES.

BEAUTIFUL BROWN EYES.
(0234 - 0300)

```
0230             A266 A266  A266 7E80 9072 A266
0240   AA60 5460 0100 FF60  A266 A266 A266 8055
0250   A266 D872 0100 2072  9072 A266 A266 A266
0260   7E80 9072 A266 AA60  5460 0100 5460 AA60
0270   C055 9072 A266 AA60  A266 9072 BD80 0100
0280   BD80 0000
```

BIG BEN. (0234 - 0260)

```
0230             E066 0F00  F080 0F00 EF72 0F00
0240   FFAB 0F00 E0AB 0F00  EF72 0F00 E066 0F00
0250   4080 0F00 E066 0F00  F080 0000
```

BEAUTIFUL DREAMER.
(0234 - 0280)

```
0230             FE40 F244  FE40 6060 0100 6060
0240   0100 6060 5166 0100  5166 0100 5166 9072
0250   A266 B04C 0100 B04C  0100 B04C C055 F244
0260   D84C D84C C055 AA60  AA60 A266 9072 F366
0270   0100 F366 0100 F366  0000
```

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,
[redacted]

(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.)

****