N.S.W. 6800 USERS GROUP

NEWSLETTER No. 1 SEPTEMBER, 1980.

Well, here it is, the first issue of our newsletter. There seems to have been some confusion caused by our item in E.A. Micronews, as to when it would appear, and we apologise if you thought it would be August. This is the first issue, and is September's. The second issue will be posted the first week in October.

To answer the question many of you asked in your letters, No, we do not have any connection with Michael Bauer, but we have had a letter from him wishing us well and promising to send us some bits and pieces for publication later on. Our purpose in starting a newsletter is to act as a clearing house for exchange of programs and information, thereby helping less experienced users to build up a collection of programs and learn more about their DREAM. There will also be some more technical items for those who can handle them. We will therefore welcome suggestions for improvements or inclusions in the newsletter, and in fact will need a constant supply of new programs to stay in existance. So, if you have written some programs, or can write an article on the uses or modifications to your DREAM, please send them in. (See section on how to submit programs for details.) All articles or programs printed will earn for the contributor that months newsletter and the next FREE OF CHARGE.

Judging from the letters we received, many of the people who purchased the "CHIPOS" manual from M.J. Bauer have been confused by it. Most inexperienced programmers seem to have tried to interpret the Machine Code Monitor and Sub-Routines with the CHIPOS language. We therefore suggest that unless you understand the CHIPOS and 6800 Machine Code fully, don't use that manual until you have more knowledge. The only part suitable for the novice are the pages starting with "Getting started with Chip-8" and ending with the Chip-8 instruction code table. The rest is for direct use by machine code programmers or people wishing to find out how the resident interpreter operates. This is not a task suited to the beginner. In this issue you will find two machine code sub-routines. Unless you are familiar with running machine code, don't use them. We will explain their use next month, and show you how to use machine code sub-routines in a short example program. In future issues we will be explaining how to use CHIPOS and working into machine code, eventually taking the monitor and the sub-routines apart to study their operation.

Next months issue should include an explanation of the differences between machine code and Chipos. There will also be six games programs, (including one contributed by a subscriber), and a machine code program to let you display the program currently in memory on the screen. Future plans include showing you how to interface a printer to your DREAM, and how to make it control external appliances, such as a Video Cassette Recorder, for automatic operation. There will also be some programs requiring more than 1K of RAM, so we have included plans for a memory expansion board in this issue. For those who cannot handle the wire-wrapping, don't despair, we believe there may shortly be available a P.C. board design for memory expansion, from J.R. Components in Sydney.

That's it for this month, we hope you like it. Don't forget, to order next months newsletter, just print your name and address on the enclosed label, and return it to us with your cheque or postal order for $4-00. (Interstate subscribers please add 10c for Stamp Duty.)

Happy DREAMing,

Graeme Samways and Garry Nelson,

N.S.W. 6800 USERS GROUP,

[redacted]

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 of inclusion in the newsletter. ALL CONTRIBUTORS OF ARTICLES AND PROGRAMS PRINTED WILL RECEIVE TWO MONTHS NEWSLETTERS FREE OF CHARGE. Along with a listing of the program submitted we will need a tape recording, with at least twenty seconds of "leader" tone on it. (This will be returned to you on request.) The electronic copy is 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 any correspondence, but will print corrections or improvements where possible.

We will not be dealing in tapes, but if you submit a program, and wish to sell tapes, just state this after your program explanation, and detail your charges etc.

Programs submitted for consideration must be typed on A4 or foolscap in black or blue. (NO RED) 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 omit your address.)

3) The program explanation. (Don't forget key functions.)

4) Details of cassette cost etc. (If applicable)

5) The program listing.

Accompanying all submissions we require a separate sheet as follows:-

a) Your name and address.

b) Date sent.

c) Program name and memory location, or name of article.

as well as the tape containing the program.

That's all there is to it, so send us in your favourites, don't forget, for each one we use, you get two months newsletters free of charge.

* *

BACK COPIES OF NEWSLETTERS

Copies of all newsletters from No.1, September 1980, are available at a cost of $5-00 each from:-

N.S.W. 6800 USERS GROUP,

[redacted]

BARRIER

0200 - 02D0

G.V.Samways,
[redacted]

The player directs a moving line so as not to hit any other part of the line or the wall. You set the speed. (0 Slow - E Fast) If you select F it starts at 0 and increases by one every ten dots until it reaches E. The other speeds remain constant. The line starts at a random position and direction.

To change direction push;

C to go left,
D to go up,
E to go down,
F to go right.

If you hit another part of the line or the edge the game is over and the number of dots you scored is displayed.

```
0200    00E0  FC0A  6D00  4C0F    6D01  4C0F  6C00  00E0
0210    6300  6400  6B00  22B0    6B1F  22B0  6A00  22C0
0220    6A3F  22C0  CA1F  7A08    CB0F  7B08  C503  A2AC
0230    0000  660C  E6A1  6500    660D  E6A1  6501  660E
0240    E6A1  6502  660F  E6A1    6503  4500  7AFF  4501
0250    7BFF  4502  7B01  4503    7A01  6F00  DAB1  3F01
0260    1288  00E0  6A18  6B0F    FA18  A2AD  F433  F265
0270    F029  DAB5  7A04  F129    DAB5  7A04  F229  DAB5
0280    7A04  F329  DAB5  1202    7301  430A  7401  3D01
0290    129A  430A  7C01  4C10    6C0F  430A  6300  6E14
02A0    8EC5  FE15  FE07  3E00    12A4  1232  8000  0000
02B0    6A00  A2BE  DAB1  7A08    3A40  12B4  00EE  FF80
02C0    6B01  A2BF  DAB1  7B01    3B1F  12C4  00EE  0000
```

* *

BOUNCE

0090 - 0300

G.V.Samways,
[redacted]

This game starts with a boundary around the edge, and a bat on the very bottom line. You move the bat by holding down:-
'C' Left fast, 'D' Left slow, 'E' Right slow, 'F' Right fast.

Two blocks of random dots are placed in the playing area to aid confusion. Push any key to serve the ball and as it hits the bottom line the ball rebounds but that dot disappears and you have another hole to protect with the bat. You have 20 balls and your score is the number of dots removed from the bottom line.

```
0090    0000  6A00  A0A0  DAB1    7A08  3A40  1094  00EE
00A0    FF80  6B01  A0A1  DAB1    7B01  3B1D  10A4  00EE
00B0    ****

0200    6B00  2092  6B1D  2092    6A00  20A2  6A3F  20A2
0210    6700  CA2F  CB0F  0000    7B06  A2EA  DAB1  7701
0220    3780  1212  123C  0000    6E0C  EEA1  6CFE  6E0F
0230    EEA1  6C02  6E0D  1268    0000  0000  6600  6714
0240    A2ED  6020  611E  D011    631D  623F  8202  77FF
0250    4700  12CA  FF0A  A2EB    D231  65FF  C401  3401
0260    64FF  A2ED  6C00  1228    EEA1  6CFF  6E0E  EEA1
0270    6C01  D011  80C4  D011    4F01  12B8  4200  6401
0280    423F  64FF  4300  6501    431F  12C4  A2EB  D231
0290    8244  8354  D231  3F01    1262  12F0  0000  6A02
02A0    FA18  0000  46FF  12CA    0000  C401  3401  64FF
02B0    C501  3501  65FF  1262    6A03  FA18  A2EB  D231
02C0    73FF  1256  A2EB  D231    1248  A2ED  D011  A310
02D0    F633  F265  6318  6418    F029  D345  7305  F129
02E0    D345  7305  F229  D345    12E8  0180  44FF  0000
02F0    331D  129E  D231  7601    12B8  0000  0000  0000
0300    ****
```

* *

FRAMED RANDOM PICTURE

0080 - 0260

G.V. Samways,

[redacted]

Mk 1 VERSION

This program displays a random movement of dots. You will notice a repetition in the pattern after a time.

Mk 2 VERSION

Gives a line display instead of dots.

Program common to both:-

```
0080   6B00 2092 6B1F 2092 6A00 20A2 6A3F 20A2
0090   00EE 6A00 A0A0 DAB1 7A08 3A40 1094 00EE
00A0   FF80 6B01 A0A1 DAB1 7B01 3B1F 10A4 00EE
```

Mk1

```
0200   2080 CC01 4C01 1216 CA3E 6B02 4A00 1202
0210   A0A1 DAB1 1224 CB1E 6A02 4B00 1202 A0A1
0220   DAB1 0000 CD03 4D00 7AFF 4A01 7A02 4D01
0230   7BFF 4B01 7B02 4D02 7A01 4A3E 7AFE 4D03
0240   7B01 4B1E 7BFE A0A1 DAB1 3F01 1224 600F
0250   E09E 1224 00E0 1200 0000 0000 0000 0000
```

Mk2

```
0200   2080 CC01 4C01 1216 CA3E 6B02 4A00 1202
0210   A0A1 DAB1 1224 CB1E 6A02 4B00 1202 A0A1
0220   DAB1 6800 CD03 4D00 7AFF 4A01 7A02 4D01
0230   7BFF 4B01 7B02 4D02 7A01 4A3E 7AFE 4D03
0240   7B01 4B1E 7BFE A0A1 DAB1 3F00 1256 78FF
0250   3800 1226 C803 600F E09E 1224 00E0 1200
```

* *

LIFE

0080 - 0240

G.V. Samways,
[redacted]

This is a display of cell growth, in accordance with the following rules:-

1. A cell is born if 3 cells are adjacent to an empty space.
2. A cell lives if 2 or 3 cells are adjacent, and dies otherwise.
3. All events take place simultaneously.

To start the game, you make a pattern by entering the cell coordinates, first the "Y" from 0-7 downwards, then the "X" from 0-F across. F initialises the program, and the number of scans is entered plus one, so that 1 gives 0 scans, to F giving 14, and 0 giving 255. Then sit back, and watch the colony live, or die.

```
0080    E0A0  E080  F10A  410F    10B0  6A08  8A12  4A00
0090    1096  FA18  1084  8114    8114  F00A  8004  8004
00A0    A080  D013  1084  6F00    D011  8CF4  D011  00EE
00B0    F20A  6000  6100  6E00    72FF  4200  1084  6C00
00C0    A083  D011  84F0  D011    70FC  20A6  7104  20A6
00D0    7004  20A6  7004  20A6    71FC  20A6  71FC  20A6
00E0    70FC  20A6  70FC  20A6    7004  7104  4C02  1202
00F0    4C03  10FA  3401  1202    122E  3401  122E  1202

0200    1084  7004  4040  7104    4040  6000  3120  10BA
0210    A238  FE1E  61FC  F155    6E00  A238  FE1E  F165
0220    A080  D013  7E02  31FC    121A  D013  10B2  A238
0230    FE1E  F155  7E02  1202    0000  0000  0000  0000
```

* *

SUB

0200 - 0400

G.V.Samways,
[redacted]

You are the commander of a submarine and your mission is to destroy enemy shipping, with a new miniature torpedo, which gives you unlimited fire power, but you still have to contend with depth charges, and you can only use one torpedo at a time.

To move your sub you hold down:-

'4' Surface left, '5' Surface, '6' Surface right,

'8' Left '9' Right

'C' Dive left 'D' Dive 'E' Dive right.

To fire torpedoes you push:-

'0' Left (up) '1' Up '2' Right (up)

If you are sunk your score is displayed, when you sink a ship its speed is added to your score (either 1 or 2) and it instantly reappears at a random position. The ships speed is selected randomly.

```
0080    DEPTH CHARGE DATA

0200    6000  611F  6400  6900    23E8  F155  7402  3440
0210    1208  6103  6200  A3F0    D011  7008  3040  1218
0220    6C1C  6D10  A3F1  DCD3    CE3F  C501  7501  6000
0230    A3EE  DE03  6403  660C    E6A1  6400  660D  E6A1
0240    6401  660E  E6A1  6402    6608  E6A1  6404  660A
0250    E6A1  6406  6604  E6A1    6408  6605  E6A1  6409
0260    6606  E6A1  640A  6600    E6A1  640C  6601  E6A1
0270    640D  6602  E6A1  640E    A3F1  DCD3  4C00  128C
0280    4400  7CFF  4404  7CFF    4408  7CFF  4C38  129C
0290    4402  7C01  4406  7C01    440A  7C01  4D03  12AC
02A0    4408  7DFF  4409  7DFF    440A  7DFF  4D1C  12BC
02B0    4400  7D01  4401  7D01    4402  7D01  DCD3  320D
02C0    12EA  A3F4  D783  78FF    D783  3802  1364  D783
02D0    6F00  D781  84F0  D781    6200  3401  1364  8954
02E0    A3EE  DE03  6A08  FA18    1228  320C  12FC  A3F6
02F0    D783  77FF  37FF  12C6    6200  1364  320E  131E

0300    A3F8  D783  7701  78FF    373E  1310  6200  1364
0310    D783  3802  1364  D783    A3F4  7702  12D0  340C
0320    1338  87C0  77FD  6A3F    8A71  3A3F  1364  620C
0330    88D0  A3F6  D783  1364    340E  1354  87C0  770A
0340    6A3F  8A71  3A3F  1364    77FE  620E  88D0  A3F8
0350    D783  1364  340D  1364    620D  A3F4  87C0  88D0
0360    7704  D783  6400  23E8    F165  311F  1376  7402
0370    4440  13FC  1366  A3F4    D011  6F00  7101  311F
0380    D011  4F01  13C2  23E8    F155  136E  83D2  3300
0390    13B0  6400  23E8  F165    7402  4440  13B0  311F
03A0    1394  74FE  23E8  80E0    6104  F155  A3F4  D011
03B0    A3EE  6000  DE03  8E54    4E3E  122A  4E3F  122A
03C0    122E  00E0  6A1A  FA18    A080  F933  F265  6B10
03D0    F029  DAB5  7A04  F129    DAB5  7A04  F229  DAB5
03E0    FA0A  FB18  00E0  1200    A080  F41E  00EE  287C
03F0    FF08  1CFF  8080  8040    2040  8000  C3FF  138C
0400    ****
```

* *

MEMORY DISPLAY

0080 - 0100

G.V.Samways,
[redacted]

This program is fully machine code and should be started as such, i.e. 0080, FN, 3. You then enter the four digit starting address and the program displays 32 bytes of memory on the screen, from the address entered. To continue, press any key and the next 32 bytes are displayed.

This program can be loaded anywhere without alterations, so you can use it say from 0380 - 0400 etc. It can be used to display the CHIPOS monitor or even the contents of the scratch pad, variables, etc.

```
0080    BDC3 9097 40BD C390   9741 BDC0 7986 185F
0090    9742 D743 972E D72F   CE00 408D 4008 8D3D
00A0    4FC6 0897 42D7 4397   2ED7 2F01 01DE 408D
00B0    2C86 019B 4197 4186   0099 4097 40D6 4286
00C0    4010 2EE7 4F97 4297   2EC6 06DB 43D7 43D7
00D0    2F86 2010 2ED4 BDC2   C420 AF01 01A6 0036
00E0    4444 4444 8D01 32DF   12BD C193 C605 BDC2
00F0    2486 049B 4297 4297   2E96 4397 2FDE 1239
```

* *

MACHINE CODE SUB ROUTINES

G. V. Samways,
[redacted]

1. SHIFT VARIABLE

This sub shifts the variable thus;
VX = OZ goes to VY = ZO, or VY = F * VX, VX can = VY

This sub is used for entering display coordinates. (0 - 3F, 0 - 1F) without a long loop.

```
B6003X  003X  Get X (hex)
48      Shift 1 bit left
48        "
48        "
48        "
B7003Y  003Y  Return to Y (hex). X can equal Y, but non compulsory.
39      Return from sub.
```

All you do is set a variable in question, call the sub, then add on the second digit, e.g.,

```
FC0A    Key = X, so C = OX
0300    C = XO
FD0A    Key = Y, so D = OY
8CD4    C = C + D, so C = XY
```

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

2. 1200 Hz TONE.

This sub bleeps for set or variable duration.

```
F6003X  003X (or C6mm, duration set permanently. At mm, get length.)
D721
C640    (For 2400Hz tone C641 (FX18) ).
7EC2E5  C2E5  Goto Chipos for remainder of sub.
```

Use F6 003X to get variable duration, or C6mm where mm is a set duration.

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

FINALLY why reduce the capabilities of a Hexadecimal system by limiting yourself to decimal as was done in the secret number game. To make the game hex, remove the filtering statements such;

```
0298    0000  C00F  0000  0000
02A0    0000  F055  etc.

02D8    12B4  6A00  12F4  0000
02E0    0000  0000  0000  0000
02E8    0000  F055  etc.
```

Your odds are now 4096 to 1 as opposed to 1000 to 1. That is over 4 times more interesting.

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

1 to 4K MEMORY EXPANSION FOR THE DREAM 6800

G. V. Samways,
[redacted]

Since E.A.'s fourth article on the DREAM I have been experimenting with CHIPOS, and have developed some programs requiring 2K RAM, so I developed this 1K to 4K memory expansion board. It only uses 2 IC's other than the 2114 RAM IC's. The first is a 3 to 8 line demultiplexer, the 74155 or 74LS155, the other is a quad 2 input AND gate, the 7408 or 74LS08.

CIRCUIT DESCRIPTION

The address lines A0 - A9, +5, GND, and R/$\overline{W}$, are connected to all RAM chips. The data lines D0 - D3 are connected to the first chip of each 1K block, and lines D4 - D7 to the second chip of that block. The address lines A10 - A12 are connected to one input on three of the AND gates, the other input on each is connected to BA. This returns the 'chip select' to the first 1K of RAM during the display sequence. The three outputs are fed into the data inputs on the 74155, CS is connected to both strobes. This is then decoded to 1 of 8 lines so ; K1 to pin 9, K2 to pin 10, K3 to pin 11, K4 to pin 12, K5 to pin 7, K6 to pin 6, K7 to pin 5, K8 to pin 4. These lines are connected to CS inputs on the two relevant RAM chips.

CONSTRUCTION

On my expansion board I used a piece of matrix board (no copper either side, DSE No. H-5310) and wire wrapped the connections daisy chain style. I then cut most of the wire wrap socket pins , except for the two expansion bus sockets and the 2114 on board, down to where the wire wrap ended. I arranged the board to plug into the main board via the expansion bus sockets and the right hand 2114 socket using the long pins on the wire wrap sockets on the expansion board.

With 5K running the current drawn on the +5v supply was approx. 1.0 amp for the whole board, less the modulator.

NOTE

The circuit can be used from 1 to 8K but if you have extra $\overline{CS}$ lines these could be used on PIA's etc.