NEWSLETTER Nö 3
NOV. `8O

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-couplers.

4K EXPANSION KIT: $99-00.

Consists of:Dream-sized fibreglass P.C.B.;4K RAM with 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 two amps.The P.C.B. is not sold separately.

3 AMP POWER SUPPLY KIT:$45-00.Now available separately.

Post,Packing and Insurance:$5-00 on all orders.

Phone Cash-On-Delivery orders are accepted.C.O.D.$2-00 extra.

Phone for details of Sydney counter sales.

- -

ORDER COUPON

Name:.................................

Address:...

..............................Postcode:..................

Signature:...........................Phone:(...)..................

( ) 4K Expansion Kit @ $99-00	$............
( ) 3 Amp Power Supply Kit @ $45-00	$............
Post,Packing,Insurance @ $5-00	$ $5-00
( ) Cheque/Money Order	$ 00-00
( ) Cash-On-Delivery (Optional)	$ ($2-00)
	$............

This order is subject to a 7 day Money-Back Guarantee.

J. R. COMPONENTS PTY LTD

P.O.Box 128,Eastwood,N.S.W.,2122,Telephone (02) 853385.

N.S.W. 6800 USERS GROUP

NEWSLETTER No.3 NOVEMBER, 1980.

Hello again, fellow DREAMers. By the time this reaches you, you will have realised that we are running a few days late. We hope that you will forgive us, as we had a very busy month putting all this together, but we feel sure you will agree it was worth waiting for.

We have received quite a lot of suggestions for improvements to the contents and format of the newsletter, and we will incorporate these as we go along, so keep them coming in.

Some of the games we have coming up for you in the next couple of months are, Lunar Lander, Blackjack, an improved version of Life, a Bio Rythm Plotter, a Lotto Number Selector, a Racing Car game, and some more "Space War" type programs, but we still need more to stay in operation, so keep sending them in to us.

The December issue will be slightly different in format to previous issues, as Graeme has University examinations in November, and Garry has a lot of pre-Christmas parties to get through. Bacause of this, December will consist mainly of games, and a very good "Surprise Package" from Michael Bauer, which you will love playing with. There will not be a chapter of "How to Use Chipos", as Graeme will be too tied up with his exams to write it, (and I don't know how to,) but this will continue in the January issue. (We will put some extra games in to keep you busy until then.)

And now for the really BIG NEWS. Because so many of you have requested it, commencing January, we will be offering you a SIX ISSUE SUBSCRIPTION FOR $15-00, which effectively reduces the cost per issue again, and will also save you postage etc. The single issue price will remain at $3-00. If you wish to take advantage of the subscription offer, please put a note to that effect in with your order for the December issue. Do not send us the money then, we will send you six labels with the December newsletter, please print your name and address on all six of them, and return them to us with a cheque or Postal Order for $15-00. The month before your subscription expires, we will send you a reminder note, and another six labels to complete. Please help us by not losing the labels, as it would take us ages to write out 200 odd every month.

Having got all that off my chest, I will answer some of your queries. The most common one is still the price, and subscriptions, so I hope we have fixed that one for you. We feel that the subscription price of $2-50 per issue is very reasonable when you consider the teaching programs you are getting, and at least four new games each issue. If you had to buy the games programs from a commercial software outlet, they alone would cost more than this. (If you could get them for the DREAM.) For those of you who cannot see why we have to charge $2-50 or $3-00 when Electronics Australia, a much bigger and more professional publication, is only $1-60, consider this. Their advertising rates range from approximately $150-00 for a small advertisement, up to about $680-00 for a full page. One night when you have nothing better to do, try counting the ads, and you will get an idea of where the main source of revenue is, and it is one which we cannot hope to compete with, in a specialised publication such as this.

Keep the feedback coming, we don't know what you want unless you tell us.

Please note that the extra -10c required on Interstate cheques is for Stamp Duty charged by the N.S.W. Government, not postage, and is only necessary on cheques, not Postal Orders.

If you do not have access to a typewriter, but wish to submit a program, yes, it is O.K. if you print it all out neatly in our standard format, I will type it for you.

It is very difficult for Graeme to analyse hardware faults by mail. We have had a few enquiries such as "my DREAM does not work. When I switch it on, all I get is six vertical white bars on the screen. Can you tell me what the problem is?" This sort of problem is almost impossible to diagnose without testing the actual board, and we are not in a position to be able to offer a repair service. We suggest that if you have this sort of a problem, you put an advertisement in the newsletter asking for somebody who lives near your area to help you. (Sydney people see the ad from Fred Lever in this issue.) We will print your advertisement in two issues for a fee of $1-00. Any solutions to common problems we receive, will be printed in the newsletter where ever possible. (See "The Problem of the Shifted Display" further on.)

Our thanks to those of you who have written in telling us you like what we are doing, and are happy with the newsletter. Don't forget, when you order your December issue, put a note in telling us if you want to save money by ordering six issues for $15-00. I cannot tell you everything that is in the December issue, because it is a surprise. Suffice to say, we think you will be happy with it.

Until then,

Happy DREAMing,

Garry Nelson and Graeme Samways,

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

****

IDEAS

CHEAP COLOUR

I made a cardboard frame to fit over my T.V. screen, and attached plain coloured plastic to it. To tighten up the plastic I placed it in a warm oven for a couple of minutes, with the door left open. The effect is quite stunning. Alternatively, coloured perspex can be used, but it costs a lot more.

Ray Leaper,
[redacted]

+ + + + + + + + + +

A HOT 6802?

It has been noticed by several DREAM 6802 owners that the 6802 running at full speed (1Mz) gets HOT.(approaching maximum power dissipation) After research through data sheets on the 6802 it was found that it does not have tri-state address lines, and in fact during DMA operations (Video display memory access) the 6802 bus displays the next current address which is then shorted because the DMA has relatively more powerful buffers.

The excessive dissapation can be reduced by inserting 4K7 resistors in the address lines A0 to A9 where they exit the 6802.(cut tracks, solder in) This should not affect bus drive capabilities, however full buffering is recommended for memory expansion etc.

D. A. Trabucco.

****

HOW TO USE CHIPOS. Part 2.

Last month you will remember we described the use of the GOTO and RETURN FROM instructions. This month we will be dealing with the use of the six conditional skip instructions.

All of these are similar in that, if certain conditions are met, (as specified by the program,) the instruction following the conditional skip instruction will not be executed. There is no limit on what instruction follows the skip instruction. In the following examples X (or Y) is a generalisation for any Hex variable.(i.e., 0 to F.) KK is a generalisation for any 2 digit Hex constant.(i.e., 00 to FF.)

5) SKIP NEXT INSTRUCTION IF X = KK (3XKK)

When this instruction is encountered in a program a test is carried out between the constant KK, (the last 2 digits in the instruction,) and the value of variable X, as specified in the instruction by the second digit. If the test finds that the two are equal, then it skips the following instruction. (see below for example)

6) SKIP NEXT INSTRUCTION IF X ≠ KK (4XKK)

This instruction is similar to No.5 (3XKK), except that if the test shows that the two are NOT equal, then the next instruction is skipped.

For example, you get a key, if it equals "F" you want to go to 0200, (i.e., restart the program), if it does not equal "F", you wait for another key to be pressed. (See examples below)

Example 1.

Address	Instruction	Explanation
0270	F00A	Get key, store in 0
0272	300F	Skip if 0 = 0F
0274	1270	Goto 0270 (Get key)
0276	1200	Goto 0200 (Restart the program)

Example 2.

0270	F00A	Get key, store in 0
0272	400F	Skip if 0 ≠ 0F
0274	1200	Goto 0200 (Restart the program)
0276	1270	Goto 0270 (Get key)

7) SKIP NEXT INSTRUCTION IF X = Y (5XY0)

The only difference between this instruction and No.5,(3XKK) is that the test is carried out on the two variables specified by X and Y.

8) SKIP NEXT INSTRUCTION IF X ≠ Y (9XY0)

Once again, the only diffence between this instruction and No.6 (4XKK) is that the test is carried out between the two variables specified by X and Y.

For example, the computer picks a random number between 0 and F, and you have to try to guess what it is.

0280	CE0F	Select a random number between 0 and F, and store in E
0282	F40A	Wait for key to be pressed and store in 4
0284	5E40	Skip next instruction if E = 4
0286	1MMM	Goto MMM to increment wrong score
0288	1NNN	Goto NNN to increment correct score

9) SKIP NEXT INSTRUCTION IF KEYDOWN = X (EX9E)

This instruction is similar to No.5 except it tests the keyboard to see if there is a key down. If there is, it tests to see if the key is equal to the value of X, and if it is, the next instruction is skipped. If

HOW TO USE CHIPOS. Part 2. (Cont)

the key was NOT equal to X, or, if there was no key down, it proceeds to the next instruction. This instruction does NOT WAIT for a key to be depressed! E.G. You want to hold a constantly moving dot still when key C is pressed.

```
0260        680C        Store OC in 8
0262        E89E        Skip next instruction if key pressed = 8
0264        7A01        Increase A by 01.
                        (When key C is depressed, you skip the instruction
                        that moves the dot.)
```

10) SKIP NEXT INSTRUCTION IF KEY ≠ X (EXA1)

This is almost the same as No.9 except the skip occurs when the key is not equal to X, or no key is depressed. The instruction does NOT WAIT for a key to be pressed.
E.G. You want to increment A by 1 when key C is held down, and to decrease by one if E is down.

```
0260        680C        Store OC in 8
0262        E8A1        Skip if key does not equal 8
0264        7A01        A = A + 01 (Increment A by 1.)
0266        680E        Store OE in 8
0268        E8A1        Skip if key does not equal 8
026A        7AFF        A = A + FF (Decrement A by 1.)
```

Next month I will deal with the Random number generator, and the logic instructions 8XY1 and 8XY2, so if you don't know much about logic, (especially AND and OR gates,) I suggest you study these before next month to gain the concepts of their functions.

G.V. Samways

****

ADVERTISING

Mr. G.CORNWELL, of [redacted] is "thoroughly obsessed" with the 6802 and its workings and applications, but does not know of any other 6800 users. He would like to correspond with other members of the group, to swap notes, and expand his horizons by talking to other people with similar interests and needs. If you are in a similar situation, why not drop him a line, or, if you live somewhere near, his telephone number is (0648) 22876.

+ + + + + + + + + +

FRED LEVER Snr, of [redacted], has had "heaps of experience" repairing DREAM's and getting them to run. If any members of the group are having difficulty building or debugging their DREAM, he would be happy to talk to them on the phone AFTER WORKING HOURS on Sydney [redacted]

****

KALEIDOSCOPE (0080 - 0400)

Graeme V. Samways,
[redacted]

This is the second of our "teaching" programs, and is a modified and improved version of the original Kaleidoscope program which appeared in Electronics Australia. It has been fully expanded so that you can follow the explanation column to see what each instruction does, and by so doing, learn how to write your own programs.

In this version, there are four quadrants which do not overlap. You can easily modify the program to allow intermingling quadrants, or a single dot on the screen (to form spirals) etc, etc. If you come up with any ideas for display types or key functions please send them in.

The program is set up to get key inputs and store them in memory from 0280. The keys each have a designated function.

i.e.	4	Up Left	5	Up	6	Up Right
	8	Left	9	Erase dot	A	Right
	C	Down Left	D	Down	E	Down Right

also
0 Run the instructions stored
1 Return Home. (Centre)
2 Do not display next key
3 Erase sereen
7)
B) Are not assigned but I am sure you will think of something!
F)

First you enter which mode you want, e.g., '0' Input (i.e.enter instructions) or '1' Run existing set of instructions (previously entered)

As each key is entered it is stored then its function is carried out and the next key is entered. If '0' is entered, or if you run out of memory, the index pointer is reset and each instruction is executed until '0' is again encountered, when the cycle restarts to form a continuous pattern.

The variables are assigned the following functions:-

0	Store key entered	8	Skip flag
1	X Co-ordinate position	9	Not used
2	Y Co-ordinate position	A	Display X
3	Index counter (Units)	B	Display Y
4	Index counter (100s Hex)	C	Display mirror of X
5	Calculations	D	Display mirror of Y
6	80 (i.e. 100 Hex/2 for 100s inc.)	E	Not used
7	Mode flag.(0 Set up,1 Repeat,2 Input)	F	Flag. (Not used)

ADRESS INSTR. EXPLANATION

SET UP (START) PROGRAM

```
0200   6009   0 = 09 (Set up pseudo key = 09)
0202   F70A   Get mode. ( Wait for key down then store in 7 )
0204   6501   5 = 01 (Set up filter) )  Filter out all but
0206   8752   7 = 7 AND 5            )  last bit. (0 or 1)
0208   611F   X = 1F )
                       Set co-ordinates to centre of screen
020A   620F   Y = 0F )
```

START OF MAINLINE

```
020C   6380   3 = 80  (Set starting index units.)
```

KALEIDOSCOPE (Cont)

```
ADRS.   INSTR.  EXPLANATION

020E    6400    4 = 00  (Reset index 100s Hex)
0210    6680    6 = 80  ( 2 x 80Hex = 100 Hex.)
0212    6800    8 = 00  Reset skip display flag.
0214    0000    No operation. (If you put 00E0 here the screen will be
                erased after each cycle.)
CALCULATE INDEX
(Normal re-entry point)
0216    A200    Index = 0200
0218    F31E    Index = Index + 3  (Add units)
021A    6500    5 = 00  (Reset loop counter)
021C    9540    Skip if 5 ≠ 4 ) If loop counter = Index 100s finish loop
021E    1228    Go to 0228    )
0220    F61E    Index = Index + 80  )Increases index by 100 Hex,i.e.0300 to 0400
0222    F61E    Index = Index + 80  )
0224    7501    5 = 5 + 01  (Increase counter)
0226    121C    Go to 021C

MODE DISCRIMINATION

0228    4700    Skip if 7 ≠ 00  (Setup)
022A    123A    Go to 023A  (Go to change mode)
022C    4702    Skip if 7 ≠ 02  (Input)
022E    1234    Go to 0234  (Go to GET KEY)
0230    F065    Recall instruction from Index location
0232    123E    Go to 023E
0234    F00A    0 = Key down (Get key function)
0236    123C    Go to 023C  (Store at Index)
0238    0000    No op.
023A    6702    7 = 02  (Change mode from Setup (0) to Input (2)
023C    F055    Store 0 at Index location

SET UP FOR JUMP TABLE

023E    8004    0 = 0 + 0 (Double key value for jump table.)
0240    B080    Go to 0080 + 0  (Go to Jump Table position for each key.)
                (You should also go to Jump Table ((from 0080)) if following
DISPLAY          program step by step.)

(Edge Limits - Upper Left hand quadrant.)
0242    41FF    Skip if 1 ≠ FF (Left edge limit)
0244    611F    1 = 1F         (Reset to middle)

0246    4120    Skip if 1 ≠ 20 (Right edge limit (Centre))
0248    6100    1 = 00         (Reset to left.)

024A    42FF    Skip if 2 ≠ FF (Top edge limit)
024C    620F    2 = 0F         (Reset to middle)

024E    4210    Skip if 2 ≠ 10 (Bottom edge limit (Centre))
0250    6200    2 = 00         (Reset to top)
(You can remove the limits to get overlapping displays.)

DISPLAY SKIP

0252    4800    Skip if 8 ≠ 00  (Skip if flag set.)
0254    125A    Go to 025A  (Go to 4 dot display)
0256    6800    8 = 00  (Reset skip flag.)
0258    1270    Go to 0270  (Increase index. i.e., Skip display)
```

KALEIDOSCOPE (Cont)

```
ADRS    INSTR.  EXPLANATION
```

4 DOT DISPLAY

```
025A    8A10    A = 1  (Set A = to X position 1st quadrant.)
025C    8B20    B = 2  (Set B = to Y position 1st quadrant,)
025E    6C3F    C = 3F (Set up C for mirror of X calculation.)
0260    6D1F    D = 1F (Set up D for mirror of Y calculation.)
0262    8CA5    C = C - A  (Calculate X mirror.)
0264    8DB5    D = D - B  (Calculate Y mirror.)
0266    A0CE    Index = 00CE (i.e. Position of data for dot (80)).
0268    DAB1    Display dot Top L.H.quadrant. (At AB)
026A    DAD1    Display dot Top R.H.quadrant. (At AD)
026C    DCB1    Display dot Bottom L.H.quadrant. (At CB)
026E    DCD1    Display dot Bottom R.H.quadrant. (At CD)
```

INCREASE INDEX

```
0270    7301    3 = 3 + 01  (Increase Index units)
0272    4300    Skip if 3 ≠ 00  )
                                ) Increase 100s if 3 = 00
0274    7401    4 = 4 + 01      )
0276    3402    Skip if 4 = 02 (End of RAM limit. 2=1K,6=2K,A=3K,E=4K.)
0278    1216    Go to 0216  (Get next instruction)
027A    6701    7 = 01  (Change to REPEAT mode)
027C    120C    Go to 020C  (Restart program)
027E    0000    No op.
```

DATA

```
0280    ****    Data stored by key inputs.
```

JUMP TABLE

```
0080    10A0    0 = 0 (Key 0)                          Jump Table
0082    10A4    0 = 2 ( "  1)                     i.e., 0080 + 0
0084    10AA    0 = 4 ( "  2)                     (0 = 2 X Key)
0086    10AE    0 = 6 ( "  3)
0088    10B2    0 = 8 ( "  4)
008A    10B4    0 = A ( "  5)                To change the key functions,
008C    10B8    0 = C ( "  6)                swap the GOTO instructions.
008E    1242    0 = E ( "  7) *              E.G. To change key 1 & 2,
0090    10BC    0 =10 ( "  8)                0082  10AA ,  0084  10A4.
0092    1242    0 =12 ( "  9)→Go to display, i.e., erase dot.
0094    10C0    0 =14 ( "  A)
0096    1242    0 =16 ( "  B) *
0098    10C4    0 =18 ( "  C)
009A    10C6    0 =1A ( "  D)
009C    10CA    0 =1C ( "  E)
009E    1242    0 =1E ( "  F) *              * Put in your own functions
                                               from 00D0
```

KEY FUNCTIONS

```
(0)  Start, Re-run
00A0    6701    7 = 01  Go from Input mode (2) to Repeat mode (1)
00A2    120C    Go to 020C  (Restart)

(1)  Go home. (Centre of screen.)
00A4    611F    1 = 1F  Reset X co-ordinate
00A6    620F    2 = 0F  Reset Y co-ordinate
00A8    1242    Go to 0242  (Display)
```

KALEIDOSCOPE (Cont)

```
ADRS.    INSTR.   EXPLANATION.

(2)  Skip next instruction
00AA     6801     8 = 01  Set skip flag
00AC     1270     Go to 0270  (Increase index)

(3)  Erase screen
00AE     00E0     Erase screen
00B0     1270     Go to 0270  (Increase index)

(4)       ↖
00B2     71FF     1 = 1 + FF  (I.E. 1 = 1 - 01)  Move LEFT 1 dot.

(5)       ↑
00B4     72FF←┐   2 = 2 + FF  (I.E. 2 = 2 - 01)  Move UP 1 dot.
00B6     1242 │   Go to 0242  (Go to Display)
              │
(6)       ↗   │
00B8     7101 │   1 = 1 + 01  Move RIGHT 1 dot.
00BA     10B4 ┘   Go to 00B4 for move up then display.

(8)       ←
00BC     71FF     1 = 1 + FF  (I.E. 1 = 1 - 01)  Move LEFT 1 dot.
00BE     1242     Go to 0242  (Go to Display)

(A)       →
00C0     7101     1 = 1 + 01  Move RIGHT 1 dot
00C2     1242     Go to 0242  (Go to Display)

(C)       ↙
00C4     71FF     1 = 1 + FF  (I.E. 1 = 1 - 01)  Move LEFT 1 dot

(D)       ↓
00C6     7201←┐   2 = 2 + 01  Move DOWN 1 dot
00C8     1242 │   Go to 0242  (Go to Display)
              │
(E)       ↘   │
00CA     7101 │   1 = 1 + 01  Move RIGHT 1 dot
00CC     10C6 ┘   Go to 00C6 for move down then display.

00CE     8000     DATA  (Dot)

00D0     ****     Room for user written functions. Always end with
 ↓       ****     Go to Display, (1242) or Go to Index increase, (1270)
0100     ****
```

* *

RADIO AMATEURS

Don't forget, if you would like to contact other DREAMers on the air, send in your particulars in the following format, and we will print a list next month, (December) and update it every three issues after that.

NAME   CALLSIGN   TIME AND FREQUENCY FOR CONTACT.

****

THE SNAKE

(0080 - 0400)

R. FAINT,
[redacted]

The idea of this game is that you, the player, direct the "snake" toward its "meal",(the larger square block). Every time the snake has a snack you receive 8 points for your trouble.

BEWARE, however, should the snake collide with,

1) The walls,
2) The smaller square,
3) Itself,

WELL......... THAT'S IT!!.

Also, after each meal, the snake grows a bit. This only goes to make your task even more difficult, as should the serpent get more than 48 bends in its body, it gets hopelessly knotted and you lose control.

KEY FUNCTIONS 9 UP, 4 LEFT, 6 RIGHT, [illegible] DOWN.

Any key restarts the game.

At the end of the game, the computer rewards you with a little chime, scores under 100 deserve no song at all, scores over 400 hear the lot.

```
0080      80E0   E0E0
                         PROGRAM WORKSPACE
00F0      6F30   10F6   6F0A   7FFF      3F00   10F6   00EE   0000
                        6FC5
0200      A080   6101   6200   2214      221E   6100   6200   221E
0210      2214   1228   D121   7101      313F   1214   00EE   D121
0220      7201   321F   121E   00EE      D121   A086   6000   6101
0230      F055   416A   123A   7101      1230   A090   6006   F055
0240      6120   6210   651F   6610      6708   6840   6B06   D121
0250      235A   6A01   EAA1   1278      6A06   EAA1   1278   6A09
0260      EAA1   1278   6A04   EAA1      1278   4B04   129A   4B09
0270  276 129E   4B06   12A2   12A6 [illegible] 9AB0   126A   8BA0   A086
0280      F065   7001   3030   128A      6000   A086   F055   A090
0290      F01E   80B0   F055   A080      126A   71FF   12A8   7201
02A0      12A8   7101   12A8   72FF      D121   3F01   12DC   8030
02B0      9010   12C0   7001   9010      12C0   7001   5010   13B8
02C0      8040   9020   12D2   7001      9020   12D2   7001   5020
02D0      13B8   A081   D343   2378      A080   670C   4701   2344
02E0      4700   12E8   77FF   1250      A087   F065   A090   F01E
02F0      F065   4001   76FF   4004      75FF   4009   7601   4006
                 4002                            4C01
0300      7501   A080   D561   A087      F065   8F00   A0C0   F01E
0310      F065   70FF   3000   132E      A0C0   FF1E   F055   7F01
0320      4F30   6F00   80F0   A087      F055   A080   1336   A0C0
0330      FF1E   F055   A080   78FF      3800   1250   A081   D343
0340      2344   1250   C33F   C41F      A081   D343   3F01   1354
0350      D343   1344   0000   6840      00EE   236C   F065   7001
0360      8F00   236C   80F0   F055      A080   00EE   A086   F065
0370      A0C0   F01E   00EE   0000      A088   F065   6F00   7001
0380      7F01   400A   1394   3F08      137E   A088   F055   6F08
0390      FF18   00EE   A089   F065      7001   400A   13A8   A089
03A0      F055   6000   A088   1386      6000   A089   F055   F065
03B0      A08A   7001   F055   13A2      00E0   A088   F265   6320
03C0      6408   4200   13E0   4201      13DE   4202   13DA   4203
03D0      13D6   F318   20F0   F318      20F0   F418   20F4   F318
03E0      6522   660B   F029   23F6      F129   23F6   F229   23F6
03F0      F00A   00E0   1200   D565      75FC   20F0   20F0   00EE
```

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

THREE REEL VIDEO POKER MACHINE (0080 - 0400)

Mr. J. PANOS,
[redacted]

This program uses graphics for the card displays. Keying "F" will "spin the reels", and decrement the score by one. It will not play if your score is 0. To alter your starting score, change 0201.(e.g., 32(Hex) will initialise the counter at 50.(Decimal)

When A,A,A, or 7,7,7, is obtained, the machine will not increment the jackpot value until it is followed by further keying. Keying "C" will cancel the jackpot, keying "A" will add the jackpot value to the score.This feature is used to confirm the presence of a jackpot, and also enables you to check first whether incrementation of the jackpot score will result in a final score greater than 255, as the counter resets to zero when a score of 256 is reached. All other winning combinations will automatically increment the counter score.

PAYS

9,-,-.	PAYS	2	J,J,J,	PAYS	10	K,K,K,	PAYS	18
9,9,-.	"	5	J,J,A	"	10	K,K,A	"	18
10,10,10.	"	10	Q,Q,Q,	"	14	A,A,A,	"	150
10,10,A	"	10	Q,Q,A	"	14	7,7,7,	"	150

```
0080   A366 D45B 7408 A371   D45B 12FA A37C D45B
0090   7408 A387 D45B 12FA   A392 D45B 7408 A39D
00A0   D45B 12FA A3A8 D45B   7408 A3B3 D45B 12FA
00B0   A3BE D45B 7408 A3C9   D45B 12FA A3D4 D45B
00C0   7408 A3DF D45B 12FA   A3EA D45B 7408 A3F5
00D0   D458 12FA 6900 6408   6503 1264 7301 6703
00E0   1224 D455 7405 00EE   F00A 400C 1248 300A
00F0   10E8 7996 79FF 6702   6610 1224 0000 FFFE

0200   6332 CA0F CB0F CC0F   6700 6900 6408 A0FE
0210   D491 7408 A0FF D491   7408 3438 120E 7901
0220   3911 120C A3FD F333   F265 6418 651B F029
0230   20E2 F129 20E2 F229   20E2 4700 124C 4701
0240   10D4 4702 10DC F618   3900 10F4 600F 4300
0250   6000 C21F E09E 124C   00E0 73FF 6701 120A
0260   6602 F618 6230 F215   F207 3200 1268 7901
0270   4901 8180 4902 81D0   4903 81E0 C21F 72FF
0280   7101 4119 6101 3200   127E 4901 12D6 4902
0290   12B2 8E10 8C10 4C09   108C 4C0E 1098 4C10
02A0   10A4 4C14 10B0 4C16   10BC 4C18 10C8 7C01
02B0   1296 8D10 8B10 4B06   1080 4B0A 108C 4B0E
02C0   1098 4B13 10A4 4B16   10B0 4B17 10BC 4B18
02D0   10C8 7B01 12B6 8810   8A10 4A03 1080 4A06
02E0   108C 4A0B 1098 4A10   10A4 4A14 10B0 4A16
02F0   10BC 4A18 10C8 7A01   12DA 7408 3438 1260

0300   6900 3A03 130C 4B06   7903 7902 4C16 1314
0310   3C09 131C 3B0A 131C   4A06 790A 4C16 1324
0320   3C0E 132C 3B0E 132C   4A0B 790A 4C16 1334
0330   3C10 133C 3B13 133C   4A10 790E 4C16 1344
0340   3C14 134C 3B16 134C   4A14 7912 3C16 1358
0350   3B17 1358 4A16 10E8   3C18 1364 3B18 1364
0360   4A18 10E8 1246 3F3F   3838 3F3F 0000 383F
0370   3FF8 F838 38F8 F838   3838 F8F8 3333 3333
0380   3333 3333 3333 33F8   F818 1818 1818 1818
0390   F8F8 0000 0000 0000   0038 383F 3FF8 F870
03A0   7070 7070 7070 F0F0   3F3F 3838 383B 3B39
```

(See bottom of "ALIEN" page for 03B0 - 0400)

ALIEN

(0080 - 0300)

K. BOLCH,

[redacted]

You have three bases with which you must destroy as many aliens as possible. An extra base is awarded if you shoot ten or more aliens. You can "steer" your bullets by moving your base.If you are hit, or if you hit an alien, your score is flashed on the right, and the number of bases you have on the left.

To move your base you press:-

"0" LEFT "2" RIGHT

To fire you press 1.

(If you wish to use C, D & E, Change 021F to ~~0B~~,0225 to ~~0C~~, 022B to ~~0E~~.)
4 ,F , 6 0F 04 06

```
0080    6120  8A15  3F00  1082    8A14  8A85  4F00  10D2
0090    72FF  C401  4400  A0B4    4401  A0C3  6603  12CE
00A0    EE8A  8AAA  EEE2  4243    4242  A8AD  AFAA  AAA0
00B0    A0A0  80A0  CEAA  EAAA    CEE8  ADAF  AAEA  9090
00C0    9080  9097  A5C7  A595    7755  7545  4745  4555
00D0    2829  7301  A0A0  430A    7201  6604  12CE  8AB0
00E0    6816  1080  8AE0  681B    1080  00E0  3200  1204
00F0    F00A  1200  0000  10BA    EEFE  9224  FFDB  3C24

0200    6203  6300  A0FB  6B00    22BE  DAB5  A0F6  601C
0210    611B  D015  661B  D061    690B  DDE1  6500  640F
0220    E4A1  6501  6404  E4A1    1252  6406  E4A1  1262
0230    4501  123A  1282  1272    121E  D061  8655  4600
0240    124A  D061  4F01  10D2    1234  661B  D061  6500
0250    1234  D061  D015  70FF    4000  7001  D015  D061
0260    122A  D015  D061  7001    403A  70FF  D015  D061
0270    1230  DDE1  7E01  4E20    22BE  DDE1  4F01  10E4
0280    1238  A0FB  DAB5  490B    22B0  7901  8A84  8BC4
0290    4B20  6B00  6414  8B45    4F01  6C01  8B44  DAB5
02A0    4F01  10DE  A0F6  3C01    1236  3800  FC18  1236
02B0    C803  78FE  48FE  12B0    6900  CC01  00EE  8DA0
02C0    8EB0  6720  7E07  8E75    3F01  7E20  00EE  00E0
02D0    6705  6410  6500  D455    76FF  7408  F71E  3600
02E0    12D6  6410  6510  F229    D455  A02F  6420  F333
02F0    F029  D455  7404  F129    D455  6440  F418  10EA
```

* *

THREE REEL VIDEO POKER MACHINE (CONT)

```
03B0    3F3F  00F8  F838  3838    38B8  F8F8  F870  3838
03C0    393B  3F3F  3F3B  3938    3870  F0E0  C080  0080
03D0    C0E0  F070  0307  070E    0E1C  1C3F  3F38  3880
03E0    C0C0  E0E0  7070  F8F8    3838  3F3F  0000  0001
03F0    0307  0F1E  3CF8  F838    78F0  E0C0  8000  0000
```

* *

TIC-TAC-TOE

0200-0352

R. Schmidt,
[redacted]

The program randomly selects the first player. Players then take it in turns to play by pressing the key related to the square they want their symbol to appear in.

A win is detected and the winning symbol flashed.

Pressing key F starts a new game.

Key pad layout is

```
C D E F
8 9 A B
4 5 6 7
0 1 2 3
```

If your key layout is different, change the following locations :- 8-0249, 9-024D, A-0251, 4-023D, 5-0241, 6-0245, 0-0231, 1-0235, 2-0239.

```
0200      A100  FF65  6A1A  6B25  A31E  DAC1  DBC1  7C01
0210      3C20  120A  6A10  6B0A  6C15  DAB1  DAC1  7A01
0220      3A30  121A  C001  4000  22F6  4001  2302  FD0A
0230      4D00  1256  4D01  125E  4D02  1266  4D04  126E
0240      4D05  1276  4D06  127E  4D08  1286  4D09  128E
0250      4D0A  1296  122E  87E0  6A11  6B17  229E  88E0
0260      6A1C  6E17  229E  89E0  6A27  6B17  229E  84E0
0270      6A11  6B0C  229F  85E0  6A1C  6B0C  229E  86E0
0280      6A27  6B0C  229E  81E0  6A11  6B01  229E  82E0
0290      6A1C  6BC1  229F  83F0  6A27  6B01  229E  DAB8
02A0      6F00  8F14  8F24  8F34  2320  8F44  8F54  8F64
02B0      2320  8F74  8F84  8F94  2320  8F14  8F44  8F74
02C0      2320  8F24  8F54  8F84  2320  8F34  8F64  8F94
02D0      2320  8F14  8F54  8F94  2320  8F34  8F54  8F74
02E0      2320  4000  12EE  2302  22F6  6000  122E  22F6
02F0      2302  6001  122E  A30E  6E01  6A00  6B0C  DAB8

0300      00EE  A316  6F10  6A37  6B0C  DAB8  00EE  1824
0310      4281  8142  2418  8142  2418  1824  4281  8000
0320      4F03  132C  4F30  1338  6F00  00EE  22F6  6D0F
0330      EDA1  0350  2344  132C  2302  6D0F  EDA1  0350
0340      2344  1338  6E02  FE15  FE07  3E00  1348  00EE
0350      7E C000
```

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 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 20sec High tone, then 20sec 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 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 in BLACK 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) Details of cassette cost etc. (If applicable)

5) 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, and saves us a great deal of work if they do not have to be retyped before printing.

That's all there is to it, so send us in your favourites, and 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 $4-00 each, from:-

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

****

JOYSTICK CONTROLLER

M.J. Bauer

[redacted]

Adding a joystick to your DREAM-6800 is a very simple job. The circuit uses only one chip, a CMOS 4013 dual D-type flip-flop, and plugs into the 'Extended I/O' socket on the DREAM board; i.e. it uses the spare lines on the existing PIA.

A machine code routine is provided which is designed to be incorporated into any CHIP-8 program that utilizes the joystick. Two variables, VC and VD, are then dedicated to be the X and Y values of the joystick position. These variables are updated 12½ times a second by a special RTC interrupt routine. Thus, the CHIP-8 programmer can be oblivious to the fact that the joystick is being serviced; he merely reads the variables VC and VD at any point in the program where the joystick position is required.

CONSTRUCTION

All the components, including the joystick are conveniently located on a chunk of Veroboard, approx 2 x 3½ inches (0.1" pitch). The prototype used a Dick Smith (Cat # R-1976) joystick pot costing $3.50. The unit may be housed in a small plastic "Jiffy" box, with a ribbon cable terminated by a 16 pin dual-inline (DIL) plug (or DIP header, or whatever you call it). The extended I/O socket on the DREAM board should be wired with Vcc/+5V (pin 16) and GND/OV (pin 11), to supply power to the joystick controller; pins 11 and 16 are not allocated otherwise.

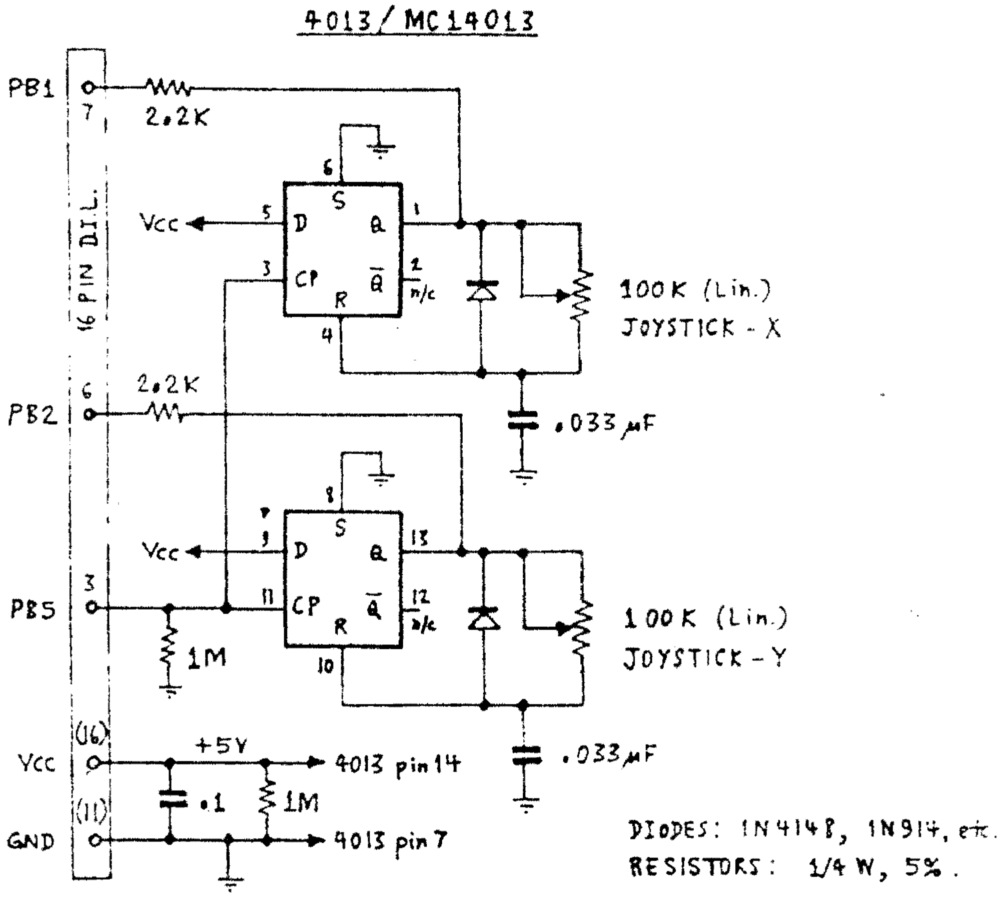
THEORY OF OPERATION

(This section may be skipped by those offended by technical jargon, abbreviations, statements requiring prerequisite knowledge or experience to comprehend, or any matter suggestive of higher-level thought processes.)

The flip-flops are set up to operate as one-shots (monostable multivibrators). Output pulse width depends on RC time constant (t=.7RC approx). Since R is one joystick pot, the output pulse width depends on joystick position. The one-shots are both triggered together by the PIA line PB5. A software counting loop is used to determine the pulse durations on input lines PB1 and PB2.

The 1M resistors are for electrostatic protection of the CMOS chip should the unit be disconnected and left sitting on a Van der Graf generator. The 2k2 resistors prevent logic contention in case PB1 and PB2 are set up as outputs by another program (e.g. the tape load and dump routines).

'DREAM-6800' JOYSTICK CONTROLLER CIRCUIT DIAGRAM



JOYSTICK SOFTWARE   (Computer Dump)

```
0200   02 04 12 64 C6 3B F7 80 13 C6 61 F7 80 12 C6 3F
0210   F7 80 13 CE 02 19 DF 00 39 7A 00 20 7A 00 21 7D
0220   80 12 7C 00 16 96 16 84 03 27 01 3B 86 FF 97 3C
0230   97 3D 7F 80 12 86 21 B7 80 12 C6 4A B6 80 12 46
0240   46 24 03 7C 00 3C 46 24 03 7C 00 3D 5A 26 ED 96
0250   3C 80 0A 2C 01 4F 97 3C 96 3D 80 0A 2C 01 4F 47
0260   97 3D 3B 10 00 E0 86 C0 87 D0 A2 9A D6 71 A3 00
0270   F6 33 6A 08 6B 04 22 8A A3 00 F7 33 6A 14 22 8A
0280   56 C0 12 64 57 D0 12 64 12 80 A3 00 F2 65 F1 29
0290   DA B5 7A 04 F2 29 DA B5 00 EE 80 00
```

The one-shot pulse width is measured by triggering the devices, then entering a loop in which the outputs are sensed, and as long as the output under test is HIGH, a counter (memory location) is incremented. When an output is LOW, its corresponding counter is inhibited.

If we want a count of between 0 and N, depending on joystick position, then the time to go round the loop N times must be equal to the maximum output pulse width of the one-shots. The timing capacitors (.033μ) are selected on this basis.

The joystick service routines given here have N = 74 to produce values in the range 0 to 63 (00 to 3F hex.). Why N = 74 ?? After offsetting by 10, we get a value in the range -10 to +63. If we suppress values less than 0, we end up with a number in the range 0 to 63 (dec.), as required. But, there will be a 'dead-band' (an area where nothing happens) at extreme stick positions (hard left, hard right, etc), which nicely clears up problems associated with variances in component values (we hope)!

The looping takes about 3 milliseconds to complete. This is a substantial time overhead, so it was decided to service the joystick once every <u>fourth</u> RTC interrupt, instead of every RTC interrupt, to allow more time for normal program execution. A sampling rate of 12½ times a second is ample for a manual input device of this nature. Using interrupts is very convenient from the user's programming point-of-view, since the joystick servicing is 'transparent'; i.e. it happens without you having to worry about calling joystick subroutines which might disturb the timing of a game or simulation.

The user program merely has to treat two variables, VC and VD, as the joystick X and Y position values (resp.). These variables are automatically updated every 80 mSec (12½ times/sec) by the new RTC interrupt service routine at $0219 (that's its entry address, not its price). However, care must be taken not to use VC or VD for any other purpose when writing a joystick program.

There are two important notes about the joystick service routine:

(1) The program as it stands gives joystick resolution of 00 ≤ VC ≤ 3F (hex) horizontally, and 00 ≤ VD ≤ 1F (hex) vertically, to suit the screen format. To obtain a vertical resolution of 00 ≤ VD ≤ 3F for any application, simply change the byte at 025F to 01 (NOP).

(2) The user CHIP-8 program begins proper at 0264, hence the GOTO 264 at 0202. Run using C000 (FN)(3) as usual.

```
                  ***  JOYSTICK  SERVICE  ROUTINES  ***
                  ***  for use with CHIP-8 programs

                  *  Scratch locations:-
                  TVFC  EQU   $0016      TV frame counter
                  TIME  EQU   $0020      RTC timer 'register'
                  TONE  EQU   $0021
                  VC    EQU   $003C      CHIP-8 variable VC
                  VD    EQU   $003D      CHIP-8 variable VD
                  PIAB  EQU   $8012      PIA port B

0200  0204              CALL  204        M/C call to initialize new setup
0202  1264              GOTO  264        jump to CHIP-8 program proper

0204  C6 3B       JINZ  LDA B #$3B       setup PIA
0206  F7 8013           STA B PIAB+1
0209  C6 61             LDA B #$61       write DDR
020B  F7 8012           STA B PIAB
020E  C6 3F             LDA B #$3F       write ctrl reg.
0210  F7 8013           STA B PIAB+1
0213  CE 0219           LDX   #$0219     setup new IRQ vector
0216  DF 00             STX   $0000
0218  39                RTS

0219  7A 0020     JISR  DEC   TIME       **  new IRQ service routine  **
021C  7A 0021           DEC   TONE
021F  7D 8012           TST   PIAB
0222  7C 0016           INC   TVFC       incr. frame counter
0225  96 16             LDA   TVFC
0227  84 03             AND A #$03       want every 4th frame only
0229  27 01             BEQ   *+3        service joystick if zero
022B  3B                RTI
022C  86 FF             LDA A #$FF       'reset' X & Y counters (VC & VD)
022E  97 3C             STA A VC
0230  97 3D             STA A VD
0232  7F 8012           CLR   PIAB       trigger the one-shots
0235  86 21             LDA A #$21
0237  B7 8012           STA A PIAB
023A  C6 4A             LDA B #74        do 74 times.........................

023C  B6 8012     J1    LDA A PIAB       test bit-1  (x)
023F  46                ROR A
0240  46                ROR A
0241  24 03             BCC   *+5        skip if X time-out
0243  7C 003C           INC   VC         incr. X counter
0246  46                ROR A            test bit-2  (y)
0247  24 03             BCC   *+5        skip if Y time-out
0249  7C 003D           INC   VD         incr. Y counter
024C  5A                DEC B
024D  26 ED             BNE   J1         continue loop.......................

024F  96 3C             LDA A VC         offset and clip VC
0251  80 0A             SUB A #10
0253  2C 01             BGE   *+3
0255  4F                CLR A            if VC<0, then VC=0
0256  97 3C             STA A VC
0258  96 3D             LDA A VD         offset and clip VD
025A  80 0A             SUB A #10
025C  2C 01             BGE   *+3
025E  4F                CLR A            if VD<0, then VD=0
025F  47                ASR A            VD = VD/2  (range 00 to 31, see note)
0260  97 3D             STA A VD
0262  3B                RTI              return
```

The following CHIP-8 program is designed to test operation of your joystick. Key in and save the joystick service routine and test program, from the computer-generated hex dump, not the listings, to avoid possible typing errors. Run the program (go from C000) and check that your joystick covers the full range of X (00 to 63) and Y (00 to 31). If not, you might try fiddling with the timer capacitor value (.033 nominally) or the software.

JOYSTICK TEST PROGRAM (in CHIP-8)

```
0264  00E0    ERASE             clear screen
0266  86C0    V6=VC             read joystick coords into V6, V7
0268  87D0    V7=VD
026A  A29A    I=29A             show spot at this location
026C  D671    SHOW 1@V6,V7
026E  A300    I=300             store decimal value of V6 at 0300
0270  F633    MI=DEQ;V6
0272  6A08    VA=08             setup coords to show numbers (VA, VB)
0274  6B04    VB=04
0276  228A    DO 28A            do subroutine to show value of V6
0278  A300    I=300             store decimal value of V7 at 0300
027A  F733    MI=DEQ;V7
027C  6A14    VA=14             setup coords to show V7
027E  228A    DO 28A            do subroutine to show value of V7
0280  56C0    SKF V6=VC         wait for any change in VC or VD........
0282  1264    GOTO 264          if change detected, show new pos'n
0284  57D0    SKF V7=VD
0286  1264    GOTO 264
0288  1280    GOTO 280          else, keep looping.......

028A  A300    I=300             subroutine to show a 2-digit decimal
028C  F265    LOAD V0:V2            number stored at 0300........
028E  F129    I=DSP;V1
0290  DAB5    SHOW 5@VA,VB
0292  7A04    VA=VA+04
0294  F229    I=DSP;V2
0296  DAB5    SHOW 5@VA,VB
0298  00EE    RETURN

029A  80xx    DATA              data for spot

0300  xxxx    xxxx              workspace (3 bytes)
```

A CHALLENGE !

By doctoring the test program above, you can easily make an object of your own derivation move about the screen. But here's a more exciting challenge!

Write a program to move an object about the screen, using the joystick to control the THRUST (or acceleration) of the object in any direction, instead of controlling its position directly. Then you'll have some idea what it must be like to be in a space vehicle with only manual controls on its rocket engines!

(Hint: You don't need fancy trig. functions, or integration/differentiation routines, or even complex math. Just remember that velocity (speed in the X or Y direction) is the rate of change of position and acceleration (thrust) is the rate of change of velocity. To all intents and purposes, 'rate of change' is merely the magnitude of a variable increment or decrement.)

THE MYSTERY OF THE MISSING DISPLAY

By G.V.Samways.

QUESTION : Why doesn't the DREAM use all of the display screen, and, can the DREAM be converted to use all of the available display?

ANSWER : The DREAM uses only one quarter on the screen, (the centre), because the Microprocessor unit (the 6800 or 6802, hereafter referred to as the "Micro") and the display have to share the same memory locations (I.E., 0100 - 0200.)

In the DREAM the Micro works along happily until it is time to show what it has in the memory allocated to the display. At this time a signal is sent to the Micro to tell it to finish the instruction it is doing, and stop. (i.e. a HALT signal.) Acting on this instruction the Micro sets all the address and data lines into tristate. (See Note 1) When the Micro has completed everything it has to do when told to Halt, it sets the Buss Available signal (BA) high. This turns on the hardware section, (Left hand side of the circuit diagram) and sends the information out to the T.V. monitor. This takes up approximately 40% of the total time. When this is complete the HALT signal is removed and the Micro recommences operation from where it left off. This takes place 50 times per second. You can see from this that the more times the Micro is stopped and the display sequence is running, the slower your DREAM becomes. In fact it is necessary to turn off the display sequence when tape load or dump is occuring, so that the Micro is not interrupted. This is done by the Micro via the P.I.A. and DMA - ENABLE, i.e., it stops the HALT signal from occuring. For more technical information you should re-read the first article from E.A. on the DREAM. (May '79.)

NOTE 1. Tristate is a third state (the other two being High, +5v, or Low, 0v.) which is built into the outputs of some I.C.s. What it effectively does is to isolate the two sides of the device so if you were looking (electrically) at the output of a chip which was in tristate, you would not see +5v or 0v, you would see a very high impedance approximately equal to an input. As the output is high impedance it places no load on other devices connected to that line or Buss. This effectively removes the device from the buss and allows other outputs to use the same buss.

The reason you cannot modify your DREAM to give full screen graphics is that this would necessitate halting the Micro for the full frame, which would only allow it to operate during the vertical sync pulse. This would probably amount to about 1 or 2 CHIPOS instructions being executed every 2 frames, i.e.,25 CHIPOS instructions per second, and even then, the TONE and TIMER operations would be stopped, as they have to be decreased every frame.

You can see from this that the DREAM would be very, very, slow with the full display. By the way we believe that Michael Bauer is developing a finer set of graphics for the DREAM. and we hope that this modification will be available to us soon.

****

ERRATA

Well I did it again. After promising that I would not make any typeing errors last month, I made TWO. Graeme sacked me on the spot when he found out, but fortunately he had to reinstate me, because he can't type!

No.1 In "Mine Field Small", 0256 should be 9340, not 9E40. (Our thanks to B. Ritter for pointing this one out.)

No.2 In "Binary - Hexadecimal Quiz" 0214 should be FD33, not F033. Also, the + signs in the explanation column at 021C,0224,022C and 0234, should be DOTS, as dots mean Boolean AND, and + means Boolean OR. (I didn't know that, so Graeme will explain when he deals with the logic instructions in "How to use CHIPOS, Part 3.")

Garry.

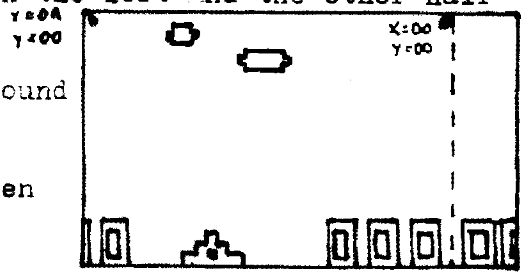
****

THE PROBLEM OF THE SHIFTED DISPLAY

J. Farnan,
[redacted]

I discovered this problem when I ran the U.F.O. Intercept program. Everything printed on the screen was shifted to the left. Half the score was printed on the left and the other half on the right.

After a bit of fiddling around with programs I worked out that the whole display area had been moved ten dot positions to the left.



Because of the wrap-around feature in CHIP-8 anything off the screen on the left appears on the right of the screen. That's why half the score on the U.F.O. Intrcept program was appearing on the wrong side of the screen.

I could display data on the far left by using the co-ordinates x - OA , y - 00 instead of x - 00 , y - 00. So for every program I used, I had to change the display co-ordinate variables before the program could be run.

I put up with this for a couple of months before I decided something had to be done. I was reading the second DREAM 6800 article in 'Electronics Australia' and on page 87 in the last paragraph it said: "... a note for perfectionists, the width of the first and last dot on every line is controlled by the delay network on E64 (120 ohms, 220 ohms, .0033uF)... if R.H.S. dots are too narrow then try increasing the capacitor to .0047uF." I had not realised that my problem might be a severe case of having too narrow dots on the right.

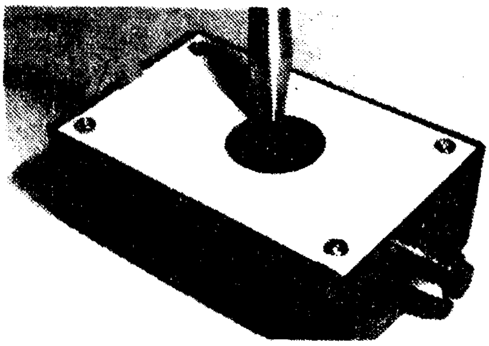
The capacitor involved is located between I.C.'s 17 and 18 . I did not have any .0047uF capacitors so I went looking in my junkbox for anything close to that.

I got about five capacitors and then put one in place of the .0033uF with a display on the screen. I was watching the screen as I put it in and the display shifted!! This looked promising so I tried a few more values. I ended up using a .0039uF 3KV cap. out of an old T.V. set. At last my problem was solved.

If any other readers had the same problem then try what I did, it just might work!!

JOYSTICK CONTROLLER - HOW WE BUILT OURS

G. V. SAMWAYS.



THE FINISHED UNIT

I started by designing a P.C.Board which holds all the components except the joystick. It was designed to fit into a Dick Smith "Zippy Box" No.H-2755. The zippy box has internal ribs so I made the board to fit between two of these. (See Figure 1.) In Figure 1 you can also see some of the components and the external connections.

The joystick I used is a Dick Smith type R-1976. This is a new version and has a plastic housing and smaller than usual pots. The housing has four holes in the corners, so I tapped these out to one eigth of an inch and mounted the joystick centrally into the bottom of the box with counter sunk screws. (See Figure 2)

Ribbon cable was used to connect the joystick to the computer, and the P.C.B. is connected to the pots with hookup wire. The pots are wired between the wiper and the end, which gives zero resistance when the joystick is in the top left position. I also put a push button in the top and wired it in parallel with the F button, so it can be used as a fire button. A hole was then cut in the cover plate for the joystick knob, and a slot for the ribbon cable.

FIGURE 1.

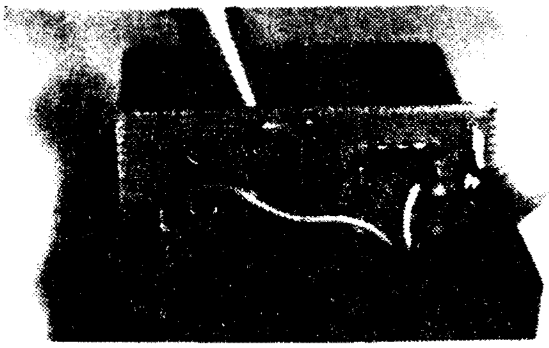
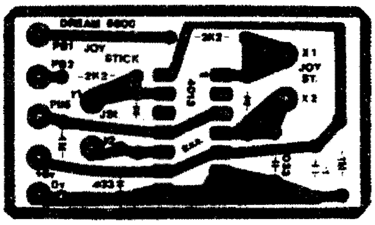
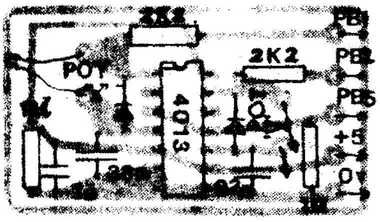


FIGURE 2

This is the P.C.B. layout and the component overlay for the joystick.





JOYSTICK CONTROLLER - HOW WE BUILT OURS CONT.

To make the P.C. Board, you trace the layout onto a piece of blank copper board, using carbon paper. Ink the carbon trace in with a "DALO" etch resist pen, allow to dry, then etch.

If you do not wish to make up your own board, we will be making P.C.B's available at a cost of $2-00 each. When you order your next newsletter, put a note in telling us how many you would like, add the appropriate amount to your cheque, and we will enclose them with your newsletter. If you want one sooner than that, send us a stamped, self addressed envelope as well, and we will post it straight back to you.

That's all there is to it, so get busy, the sooner you build it the sooner you can play with it, and then, for all you budding software geniuses, we have a competition for you. See below for details.

****

COMPETITION

The best game program we receive for publication over the next three months, which uses the Joystick Controller, will win a SIX ISSUE SUBSCRIPTION TO THE NEWSLETTER, VALUED AT $15-00.

The rules are simple. The program must be original, your own work, and be controlled by the joystick. Entries will close on the 15th February, 1981 and the winner will be announced in the March newsletter.

****

DIGITRAN KEYBOARD

We mentioned last month that all programs would be modified to suit the layout of the Digitran keyboard, and a few people have asked why, as there are a lot of different keyboards in use.

We have chosen to use the Digitran as it is high quality, arranged logically, designed for Hexadecimal and readily available. From J R Components.) We also feel that it is the most widely used.

It is laid out as follows:-

0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

For uniformity, we will convert all programs in which keys represent directions to the following form:-

↖4	↑5	6↗
←8	9	A→
↙C	↓D	E↘

When all the CHIPOS instructions have been explained in "How to Use Chipos", Graeme will explain how to find and change key functions.

****