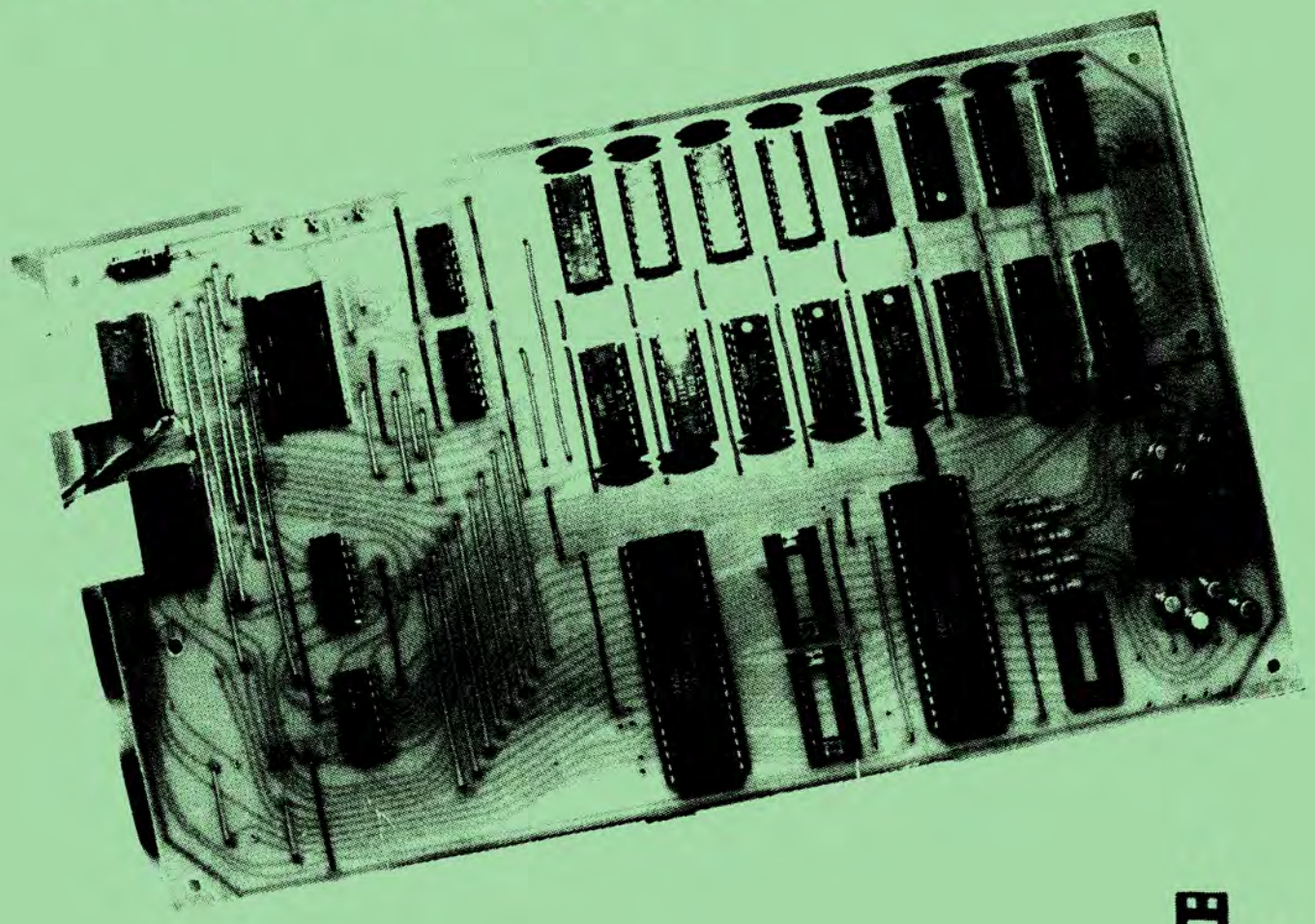
DREAMER Nö 9

MAY '81

Registered for posting as
a publication. Category B

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



04 0930 0

What a month April turned out to be. This issue is running behind schedule, due to the trials and tribulations of the Nelson Family over the Easter break. Garry went water-sking over Easter and blew the Outboard motor up on his ski-boat, then after fixing that, took a weeks holiday to take the kids out in the bush , where he blew two tyres on his Land-Rover, and was three days late getting home. Graeme had started sending up smoke signals by this time, wondering where he was and how were we going to get this issue to the printer in time. Then Garry's 9 year old son was rushed to hospital with appendicitis, resulting in further mass confusion.(He is O.K. now.) We got it all together eventually though, and here it is.

We have a WINNER FOR OUR COMPETITION - Mr. J. Gallagher, [redacted]., (This should REALLY put him in paradise) was the sender of the 100th survey card we received, so he will receive a FREE 6 MONTH SUBSCRIPTION to the DREAMER, with our compliments.

We have only received just over HALF the survey cards back so far, which is a little disappointing to us, as it seems to show a lack of intereest in whether we continue with the DREAMER or not, so please, if you haven't sent yours in yet, DO IT NOW. Even if you do not want to re-subscribe, just write 'no longer interested' across it and send it off, at least that way we know, and we can cross you off our mailing list.On a brighter note, many of you who have replied have said you enjoy our efforts and appreciate what we are trying to achieve, and for that we thank you. We have also received quite a few suggestions, and will implement them as we can.

We must apologise for the poor quality of Page 17-18 of issue No.7. We have spoken harsh words to our printer about it, and have included a replacement page with this issue. While on the point of copy quality, one of the requests on a survey card was for a better print quality, so the DREAMER would be easier to photo-copy. If this is for a file copy, or to keep the original intact, that is O.K., but if it is to distribute or sell copies, we must remind you that although we have never stressed the point, 'DREAMER' and all other publications are under copyright, and it is illegal to copy them for these purposes. (The main reason that programs for the TRS-80, Apple, etc, are so expensive is that relatively few copies are sold, due to the extensive copying that occurs.) Another request was for the P.C.B. layout of the J.R. Components Expansion board. This is also under copyright, and you can only get a board from J.R.Comp. (You might try asking him to sell you a board separately, if you do not need the full kit.)

Many people also asked for teaching articles on Machine Code programming. Unfortunately Graeme does not feel qualified to give these, and suggests that you either attend a Tech class on the subject if there is one in your area, or buy some books to teach yourself. (See the article on Assembly Language programming in this issue.)However, he will try to explain all M.C. Subroutines and programs used where possible.

If you have submitted a program, but it has not been used yet, don't worry, we will use them all eventually, but we just seem to run out of room each month. Last months newsletter cost almost double the normal cost for extra printing and postage, but we are trying to pack in as much as we can for you each issue. Starting with this issue, we will be putting in a couple of extra programs each month, which will help to clear up the backlog.

NEXT MONTH - We will have,
- An article on how to SELECT A PROGRAM ON TAPE
- A CHIP - 8 DISASSEMBLER, From the DREAMSOFT people.
- A CHIP - 8 BLOCK MOVE program
- GOLD FEVER, by Bruce Mitchell
- IAGO, (from the old board game, REVERSI.) (Two versions)
- A MOTOR CYCLE GAME
- A DICE GAME, A SPACE GAME CALLED ASTRO FIGHTER, AND A MATHS PROGRAM, ADDITION FOR BEGINNERS.

****

THE (FX29) INSTRUCTION

This will always set the Index to 0008, but it also stores the pattern required to display the Least Significant Byte of the Variable specified by X.
E.G. Variable 6 contains 09, and we want to display this. We use F629. This sets the Index to 0008 and stores the required pattern of dots from 0008 to 000C. (More on this later.)

And now, here it is, the instruction you have all been waiting for, DXYN.

The display instruction will show any desired pattern of dots, 8 dots wide, (always,) and N lines down. (1 to 16.)

X specifies the variable in which the X (Horizontal) Co-ordinate of the top left hand dot, (counting from the left hand edge of the screen,) of the pattern to be displayed is stored. This value will be in the range 0 - 3F.

Y specifies the variable in which the Y (Vertical) Co-ordinate of the top left hand dot, (counting from the top edge of the screen,) of the pattern to be displayed is stored. This value will be in the range 0 - 1F.

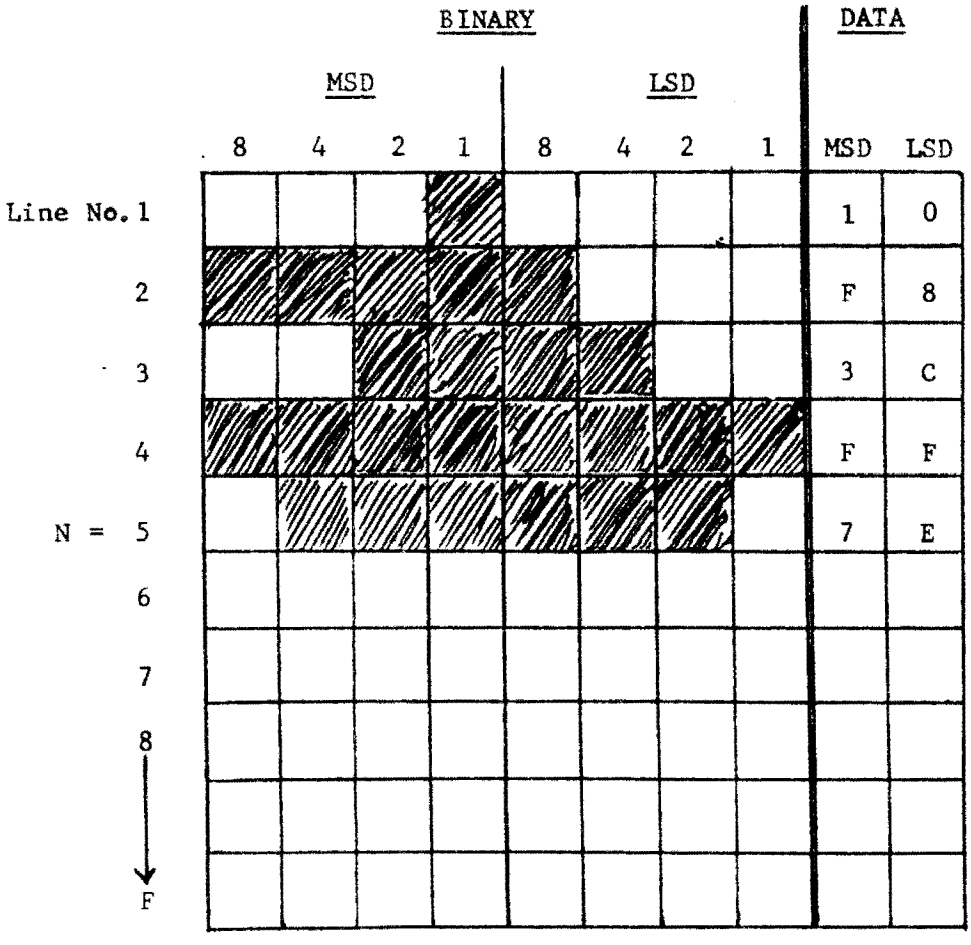
N is the number of lines to be displayed. If 0 is used for N, 10 (hex) (16 decimal), lines will be displayed.

The pattern is ALWAYS eight dots wide.

"But how do we define a pattern or find it, with only DXYN, as it only specifies the position and size?", I hope you asked.

The required pattern is stored in memory and is pointed to by the Index. You simply set the Index before the display instruction.

But how do you define a pattern? Easy, just use some graph paper, marked like this:-



(Form a grid ten squares wide and sixteen squares down.)

You then colour in the dots you want displayed on the paper, starting from the top left corner.

Then, count the minimum number of lines you need, down the left of the grid. This will be "N". In this case, N = 5.

Now for the fun part! You add up the dots in eaxh four dot group, (MSD or LSD), on each line and write the value in the MSD or LSD column at the end of the line. The way you add is in Hexadecimal, using the wighting shown above each column.

E.G. Line 1, MSD. (L1,MSD)

We only have one dot in the 1 column, so we write 1 in the MSD column of line 1.

Line 1, LSD. (L1,LSD)

We have no dots, so we write 0 in the LSD column of line 1.

L2, MSD.

We have a dot in all four spaces, so we add 8+4+2+1 = 15(dec) = F (Hex), so we put F in the MSD column of line 2.

L2, LSD.

We have one dot in the 8 column, so we put 8 in the LSD column of line 2.

Keep doing this until you come to the end of your pattern.

You may remember, back in the deep, distant past, (What, only last October?, it seems like centuries!) we included an expanded teaching program called 'Binary - Hexadecimal Quiz'. Well, if you were interested, and wanted to learn, you would have played around with that program, and would realise that what we have here is the same thing. You convert the four relevant Binary dots into the Hexadecimal equivalent. If you didn't, (tut, tut!), you will just have to go back now and give it a try.

Now that we have drawn it, how do we display our tank? (Yes, it is a tank, isn't it?) Just run the following program, and see.

```
0200  6A20  VA = 20
0202  6B10  VB = 10
0204  0000  NOP
0206  0000  NOP
0208  00E0  ERASE
020A  0000  NOP
020C  008C  NOP
020E  0000  NOP
0210  0000  NOP
0212  0000  NOP
0214  0000  NOP
0216  0000  NOP
0218  0000  NOP
021A  0000  NOP
021C  0000  NOP
021E  A280  I = 0280
0220  DAB5  SHOW 5 AT VA , VB
0222  6C10  VC = 10
0224  FC18  TONE = VC
0226  1200  GO TO 0200

0280   10F8 3CFF 7E00
```

Before 021E we set, the X co-ordinate (6A20)
the Y co-ordinate (6B10)
and erase the screen.

021E sets the Index to point to the Data for displaying the tank.
DAB5 displays the tank.
The rest produces a bleep and returns you to the start.

I strongly suggest you get some graph paper and mark it as shown, or draw a grid, then place a sheet of clear contact adhesive over the top. This will allow you to draw in your dots using water based texta colours on the grid, and erase them as desired with a damp cloth or tissue.

Draw some pictures on it , convert them to Hex and put it in from 0280 in the previous example program and run it again. (Remember to change 0220 DABN if your drawing is not five lines.

One other point is that the screen wraps around. (You will have seen this in some of the games.) It is the continuation of the displayed data off one side of the screen and back onto the other side, so if you displayed the tank over the edge, (Change 0200 from 6A20 to 6A3C), you will now see half the tank on one side of the screen and the other half on the other side. If you now change 0202 to 6B1D, you will get four pieces, one in each corner.

Now put in the following program.

```
0200  F80A  V8 = KEY
0202  FA0A  VA = KEY
0204  F90A  V9 = KEY
0206  FB0A  VB = KEY
0208  00E0  ERASE
020A  8884  V8 = V8 + V8
020C  8884  V8 = V8 + V8
020E  8884  V8 = V8 + V8
0210  8884  V9 = V9 + V9
0212  8A84  VA = VA + V8
0214  8994  V9 = V9 + V9
0216  8994  V9 = V9 + V9
0218  8994  V9 = V9 + V9
021A  8994  V9 = V9 + V9
021C  8B94  VB = VB + V9
021E  A280  I = 0280
0220  DAB5  SHOW 5 AT VA , VB
0222  6C10  VC = 10
0224  FC18  TONE = VC
0226  1200  GO TO 0200

0280   10F8 3CFF 7E00
```

This version allows you to enter your co-ordinates in the order, MSD X, LSD X, MSD Y, LSD Y. From 020A to 021E multiplies MSD X (8) and MSD Y (9), by 16,
I.E. (((X + X) + (X + X)) + ((X + X) + (X + X))) + (((X + X) + (X + X)) + ((X + X) + (X + X))), OR (((((X x 2) X 2) X 2) X 2) then adds it to the LSD X (A), and LSD Y (B), respectively.

Obviously you can enter any numbers from 00 to FF, but the first thing the Chip 8 interpreter does is AND the number entered with
3F (X) and 1F (Y)
I.E. 0011 1111 (Binary) 0001 1111 (Binary)
so no numbers over 3F (Hex) or 1F (Hex) get through. Therefore, numbers like 60 (Hex) become 20 (Hex) in the X position, or 00 (Hex) in the Y position. Try these and others to get the feel of it.

The other feature of the display instruction is the erasure of existing dots when they are to be displayed on, (and only these).
I.E. Try removing the 00E0, (Change 0208 to 0000) and put the same co-ordinates in twice. Now try adding one or two to the second X or Y co-ordinate. This is how balls tec. are moved. When displaying a pattern, if one or more dots are erased, Variable F is set to 01 (Hex), otherwise it is unchanged. To

detect a 'hit', first set F to 00, (6F00), display your pattern, then test if F is 00 or 01. Remember, you have to 'set' F BEFORE a display if you need to test for erasure of dots.

Change 0220 in the second program to:-

```
6F00    F = 00
DAB5    Display pattern
6C10    C = 10
4F00    If F ≠ 00, then Skip
FC18    Bleep
1200    Return to start
```

Now, if one or more dots is erased, you will hear a bleep.

Well, that is about all I have on the Chip - 8 instructions. I recommend you try writing short programs for yourself to try out the instructions. You will learn ten times as much by trying out programs for yourself than by re-reading all these articles. The worst thing that can happen if you get it wrong is that you will lose the contents of the memory and have to start from scratch. (Unless you store bits and pieces on tape.) It costs nothing if you make a mistake, and no permanent damage will be caused. If it does go wrong, just hit RESET and start again. You will soon learn!

Now that you all know just as much about the subject as I do, I hope to see lots of new programs soon.

Since no one has made any suggestions for future subjects in the teaching articles, I will explain the changing of key functions in the next article. This will be in two months, due to my University examinations. This time will also give you time to experiment for yourselves with all that we have covered to date. THIS IS VERY IMPORTANT!!!!

If you have a suggestion on the subject of future articles on teaching (or others) please send it in.

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.

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

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]

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

FOR SALE. DREAM 6802 IN CASE. Case has Digitran keyboard fitted and connectors for Joystick controller, Ext. Keyboard, Power, R.F.,Spare, and Tape IN/OUT. Joystick controller and Ext.keyboard supplied. Power Supply in case, 2 tapes of programs, Dreamer, issues 1 to 9, THE LOT FOR $220-00, O.N.O.
ALSO, Dick Smith Mini Scamp, Full kit, with case, working, $70-00. O.N.O.
JEFF BODNICK, [redacted]

****

ABSENT - MINDED DOG (0200 - 0300)

JAY MANN.

This is a simple graphics display, just for a giggle, which we have set out on the 'Chip-8 Disassembler.' It demonstrates the Chip - 8 Display and graphics well, and is easy to follow. We haven't added comments, but have left space for you to do it, as we think that you will benefit more from doing it yourselves.

The bottom section of the program (from 029C to 02E0) is 'Doggie' data. To analyse the display, look at each 'Set Index' instruction, (AMMM), then find the length of the Data from the next Display instruction. (Last digit.)

I.E. 0246 A29C
0248 DAB7

Then, you look at 029C (start of data) and draw the seven bytes under each other on graph paper in the manner shown below. (See 'How to Use Chip-8' in this issue also.)

BINARY	DATA	
84218421	MSD LSD	
****	0 F	
*****	1 F	
* ***	1 7	(Using graph paper gives
********	F F	correct spacing.)
********	F F	
*****	1 F	
********	F F	

This is obviously part of the dogs head. (Do this for all the data, then cut all the bits out and make a jigsaw puzzle.)

ADDR	INST	MNEMONIC	YOUR NOTES
0200	0000	NOP	
0202	00E0	ERASE	
0204	2230	DO SUB 0230	
0206	2240	DO SUB 0240	
0208	2268	DO SUB 0268	
020A	2268	DO SUB 0268	
020C	7AFF	VA = VA + FF	
020E	4A00	SKF VA NE 00	
0210	1216	GO TO 0216	
0212	00E0	ERASE	
0214	1206	GO TO 0206	
0216	228A	DO SUB 028A	
0218	2268	DO SUB 0268	
021A	0000	NOP	
021C	7CFF	VC = VC + FF	
021E	2268	DO SUB 0268	
0220	3C15	SKF VC = 15	
0222	121A	GO TO 021A	
0224	228A	DO SUB 028A	
0226	2268	DO SUB 0268	
0228	71FF	V1 = V1 + FF	
022A	3100	SKF V1 = 00	
022C	1226	GO TO 0226	
022E	1200	GO TO 0200	
0230	610A	V1 = 0A	
0232	620A	V2 = 0A	
0234	6C36	VC = 36	
0236	6D30	VD = 30	
0238	6A22	VA = 22	

ADDR	INST	MNEMONIC	YOUR NOTES
023A	6B2F	VB = 2F	
023C	00EE	RETURN	
023E	[redacted]	NOP	
0240	2246	DO SUB 0246	
0242	2254	DO SUB 0254	
0244	00EE	RETURN	
0246	A29C	I = 029C	
0248	DAB7	SHOW 7 AT VA , VB	
024A	A2A4	I = 02A4	
024C	7A08	VA = VA + 08	
024E	DAB4	SHOW 4 AT VA , VB	
0250	7AF8	VA = VA + F8	
0252	00EE	RETURN	
0254	A2AA	I = 02AA	
0256	87A0	V7 = VA	
0258	8B80	V8 = VB	
025A	7708	V7 = V7 + 08	
025C	7805	V8 = V8 + 05	
025E	D78A	SHOW A AT V7 , V8	
0260	7708	V7 = V7 + 08	
0262	A2B4	I = 02B4	
0264	D78A	SHOW A AT V7 , V8	
0266	00EE	RETURN	
0268	A2BE	I = 02BE	
026A	DCD4	SHOW 4 AT VC , VD	
026C	6902	V9 = 02	
026E	A2C2	I = 02C2	
0270	2280	DO SUB 0280	
0272	DCD4	SHOW 4 AT VC , VD	
0274	79FF	V9 = V9 + FF	
0276	3900	SKF V9 = 00	
0278	1270	GO TO 0270	
027A	A2BE	I = 02BE	
027C	DCD4	SHOW 4 AT VC , VD	
027E	00EE	RETURN	
0280	6440	V4 = 40	
0282	74FF	V4 = V4 + FF	
0284	3400	SKF V4 = 00	
0286	1282	GO TO 0282	
0288	00EE	RETURN	
028A	7A01	VA = VA + 01	
028C	2246	DO SUB 0246	
028E	A2C6	I = 02C6	
0290	7BFB	VB = VB + FB	
0292	7A08	VA = VA + 08	
0294	DABA	SHOW A AT VA , VB	
0296	7AF7	VA = VA + F7	
0298	7B05	VB = VB + 05	
029A	00EE	RETURN	

ADDR	DATA	SECTION OF DOG
029C	0F1F 17FF FF1F FF00	
02A4	E020 2020 AC00 [redacted]	
02AC	FFFF FFFF FFA0 A0A0 A0A0	
02B4	FFFF FFFF FF01 0101 0103	
02BE	8040 2010	
02C2	8142 2418	
02C6	8080 80F8 E8FF F8FF F0F0	

NOTE: If you want to know where the 'V' goes, remember what happens when you display something over something else.

6800 ASSEMBLY LANGUAGE PROGRAMS ON YOUR DREAM

TERRY MACKRELL,
[redacted]

There is no need for you to cough apologetically when you tell your friends that your home built computer is a DREAM 6800. Here is how to put it to some real work and give you the confidence you lack if, in fact, you feel that it is not "a real computer" - IT IS!

Get yourself the "6800 ASSEMBLY LANGUAGE PROGRAMMING" text by Lance A. Leventhal, (Osborne & Associates). Turn to Chapter 4, 'Simple Programs', and read; "...the only way to learn assembly language is to write assembly language programs. That is what we will do for the next six chapters."

This book is a gold mine if you want to examine typical microprocessor tasks. Forty of the programs are instant go-go's IF you amend them as explained below, and there are another 40 programs you can write to solve associated problems and prove them on your machine.

It is not necessary to go C000 mode but it does remove the VDU garbage.

Look first at the object program for 8-bit addition on page 4-3:

```
0000   9640   Re-writes as :   0200   9640      LDA A $0040
0002   9B41                   0202   9B41      ADD A $0041
0004   9742                   0204   9742      STA A $0042
0006   3F                     0206   7E C360   JMP   $C360
                                             RST
```

Now enter data in pseudo-accumulators. Say, 10 + 10 = ?

```
0040   0A
0041   0A
0042   00   RST
```

then GO. (0200, FN, 3.)

Did you notice the 00 wink at you? This is the nod "all done, examine the accumulator", so re-examine 0042 and you should find 14(hex).

Now turn to page 10-2 in Chapter 10. The purpose of this program is to convert a hex entry to ASCII Code.

```
0000   9F3E       0200   9F3E      STS   $003E
0002   8E00 58    0202   8E00 58   LDS   $0058
0005   9640       0205   9640      LDA A $0040
0007   BD00 20    0207   BD00 80   JSR   $0080
000A   9741       020A   9741      STA A $0041
000C   9E3E       020C   9E3E      LDS   $003E
000E   3F         020E   7E C360   JMP   $C360
                                   RST

0040               0040
0041               0041             RST

0020   8109       0080   8109      CMP A $#09
0022   2302       0082   2302      BLS   $ 02 (0086)
0024   8B07       0084   8B07      ADD A $#07
0026   8B03       0086   8B30      ADD A $#30
0028   39         0088   39 RST    RTS
```

Enter 0C at 0040, 00 at 0041, then GO, (0200, FN, 3.) Re-examine 0041 and you will find 43 which is ASCII Code for upper case letter C.

When you have worked through the programs now available you will have a sound working knowledge of Assembly Language. Do not say Machine Language .... but if you insist, this is the ML program for the 8-bit addition:

```
0200    1100  0110  0100  0000
0202    1100  1011  0100  0001
0204    1100  0111  0100  0010
0206    0111  1110
0207    1100  0011  0110  0000
```

The text also explains how to interface 7-segment LED's if you want lights.

Use the 6800 programmer's card in the CHIPOS manual, and you are in the ASSEMBLY GAME.

****

SUPER 8 - BUG (0600 - 0800)

GRAEME V. SAMWAYS.

This version of 8-Bug not only displays the contents of the variables, etc, it also saves the display ~~on the screen.~~

It is similar, but not exactly the same as the version in the March issue, as it is situated in the second K of memory, and saves the display.

I have used Chip-8 where possible, but I still had to use three M.C. subroutines.

1) Gets the Index value and the address of the next instruction.
2) Shifts the MSD of variable 0 Left 4 bits and stores the result in 1
3) Restores the original value of the Index.

Everything else is Chip-8. I kept the M.C. to a minimum, to allow easy disassembling of the program. I am sure you M.C. buffs will be able to write a much shorter version for yourselves.

The memory locations from 0700 - 07FF are used to store the contents of the screen.

```
SECTION     ADDR  INST  MNEMONIC          EXPLANATION

MAINLINE    0600  06BE  CALL M/C AT 06BE  Find & save I and next Instr.Address
Save Varia. 0602  A6EC  I = 06EC          Set I to variable storage
            0604  FF55  MI = V0 TO VF     Store variables
Save Screen 0606  6800  V8 = 00           Reset counter
            0608  A100  I = 0100          Point to start of screen
            060A  F81E  I = I + V8        Add counter to start of screen
            060C  F765  V0 TO V7 = MI     Load 8 Bytes from screen
            060E  A700  I = 0700          Point to start of Display 'Safe Place.'
            0610  F81E  I = I + V8        Add counter to start of 'Safe Place.'
            0612  F755  MI = V0 TO V7     Store 8 Bytes in 'Safe Place.'
            0614  7808  V8 = V8 + 08      Add 8 to counter
            0616  3800  SKF V8 = 00       If finished, Skip
            0618  1608  GO TO 0608        Re-do
            061A  00E0  ERASE             Clear Screen
            061C  6C00  VC = 00           Count = 0
            061E  6BFA  VB = FA           )
            0620  7B06  VB = VB + 06      ) Point to Line 1
            0622  6A00  VA = 00           )
            0624  FC29  I = DSP,VC        Set to display Counter
            0626  2698  DO SUB 0698      Display Counter
            0628  A6D0  I = 06D0          Set to display ':'
            062A  DAB4  SHOW 4 AT VA , VB Display ':'
            062C  7A02  VA = VA + 02      Move Right 2 dots
            062E  2682  DO SUB 0682       Display 2 digits (Variable)
```

SECTION	ADDR	INST	MNEMONIC	EXPLANATION
	0630	7A03	VA = VA + 03	Move Right 3 dots
	0632	4C04	SKF VC NE 04	If end of 1st line,
	0634	1620	GO TO 0620	Goto 'Move down 1 line'
	0636	4C08	SKF VC NE 08	If end of 2nd line,
	0638	1620	GO TO 0620	Goto 'Move down 1 line'
	063A	4C0C	SKF VC NE 0C	If end of 3rd line,
	063C	1620	GO TO 0620	Goto 'Move down 1 line'
	063E	3C10	SKF VC = 10	If not end of 4th line,
	0640	1624	GO TO 0624	Go Display the next variable
	0642	7B06	VB = VB + 06	Move down 1 line
	0644	6A00	VA = 00	Move to Left
	0646	A6D5	I = 06D5	Set I to show 'I'
	0648	DAB5	SHOW 5 AT VA , VB	Display 'I'
	064A	7A06	VA = VA + 06	Move Right 6 dots
	064C	2680	DO SUB 0680	Display 4 digits (I)
	064E	7A02	VA = VA + 02	Move Right 2 dots
	0650	A6DA	I = 06DA	Set I to show '8 B'
	0652	2698	DO SUB 0698	Display '8 B'
	0654	A6DF	I = 06DF	Set I to show 'UG'
	0656	2696	DO SUB 0696	Display 'UG'
	0658	A6E4	I = 06E4.	Set I to show 'P:'
	065A	2696	DO SUB 0696	Display 'P:'
	065C	7A05	VA = VA + 05	Move Right 5 dots
	065E	2680	DO SUB 0680	Display 4 digits (P)
	0660	F00A	V0 = KEY	Wait for key to be pressed
	0662	00E0	ERASE	Clear screen
Restore	0664	6800	V8 = 00	Reset counter
Screen	0666	A700	I = 0700	) Set I to start
	0668	F81E	I = I + V8	) of STORED screen
	066A	F765	V0 TO V7 = MI	Get 8 Bytes
	066C	A100	I = 0100	Set I to start of screen
	066E	F81E	I = I + V8	Add variable 8
	0670	F755	MI = V0 TO V7	Store 8 Bytes
	0672	7808	V8 = V8 + 08	Add 8 to counter to get the next 8 Bytes
	0674	3800	SKF V8 = 00	If not finished
	0676	1666	GO TO 0666	Redo
Restore	0678	A6EC	I = 06EC	Set I to start of stored variables
Variables	067A	FF65	V0 TO VF = MI	Get variables out of storage
	067C	06B0	CALL M/C AT 06B0	Goto Recall I
	067E	00EE	RETURN	Return to program
Chip-8 Subroutines				
Disp.4 dig.	0680	2682	DO SUB 0682	Display 2 digits.(Will get done twice.)
Disp.2 dig.	0682	A6EC	I = 06EC	Set I to start of variable storage
	0684	FC1E	I = I + VC	Add counter
	0686	F065	V0 TO V0 = MI	Get contents
	0688	06A0	CALL M/C AT 06A0	Do shift
	068A	F129	I = DSP, V1	Set I to display MSD variable
	068C	2698	DO SUB 0698	Display MSD
	068E	F029	I = DSP, V0	Set I to display LSD variable
	0690	2698	DO SUB 0698	Display LSD
	0692	7C01	VC = VC + 01	Increment Counter
	0694	00EE	RETURN	Return to program
	0696	7A04	VA = VA + 04	Move Right 4 dots
	0698	DAB5	SHOW 5 AT VA VB	Display 5 lines x 8 dots
	069A	7A04	VA = VA + 04	Move Right 4 dots
	069C	00EE	RETURN	Return to program

SECTION	ADDR	INST	MNEMONIC	EXPLANATION
M.C. Subroutines				
Shift	06A0	96 30	LDAA 30	Get digit from 0
	06A2	44	LSRA	)
	06A3	44	LSRA	) Shift MSD down to LSD
	06A4	44	LSRA	)
	06A5	44	LSRA	)
	06A6	97 31	STAA 31	Store at 1
	06A8	39	RTS	Return to program
Return I	06B0	FE 06 FC	LDX 06FC	Load X (M.C. Index) from 06FC
	06B3	DF 26	STX 0026	Store X at 0026
	06B5	39	RTS	Return to program
Store I	06BE	DE 26	LDX 0026	Load X from 0026
	06C0	FF 06 FC	STX 06FC	Store X at 02FC
Store P	06C3	30	TSX	Save stack pointer in X
	06C4	9E 24	LDS 0024	Load stack pointer from 0024
	06C6	32	PUL A	Pul Acc.A off stack (SP = SP + 1)
	06C7	B7 06 FE	STAA06FE	Store Acc.A at 06FE
	06CA	32	PUL A	Pul Acc.A off stack (SP = SP + 1)
	06CB	B7 06 FF	STAA06FF	Store Acc.A at 06FF
	06CE	35	TXS	Return stack pointer from X
	06CF	39	RTS	Return to program

		:					I:				8 B		
DATA	06D0	00	80	00	80	00	E0	48	40	48	E0	E6	A5	E6	A5	E6	57
	06E0	54	54	55	77	1C	15	1C	11	10	00	00	00				
		UG						P:									

06EC - 06FF: Storage for Variables, I & P.

0700 - 07FF: Storage for screen contents

****

DID YOU NOTICE?

Did you notice that all the Chip - 8 Mnemonics in 'How to Use Chip-8', 'Absent Minded Dog', Super 8-Bug', in this issue were generated by the computer, and printed out by the Centronics printer?

This routine was 'DREAMED UP' (sorry about that), by the DREAMSOFT people, and will appear in next months newsletter. It resides at 0400-0700, and needs a 'Dreamsoft' EPROM, a Teleprinter, and at least 2K RAM to run it.

****

ERRATTA

Not really , more an improvement to a previously published program. Mr. J. Panos has advised us that there were a couple of problems with his 'Three Reel Poker Machine' program, which was published in the November 1980 newsletter. (No.3.)

Firstly, after loading the program, you sometimes had 9's showing continuously for a while.

Secondly, their was an intermittent tendency for some cards to 'take their time' to show up.

Both these problems will be debugged if you make the following changes to the program. 0202-C80F, 0204-CD0F, 0206-CE0F, 027E-7201, 0286-3220.

****

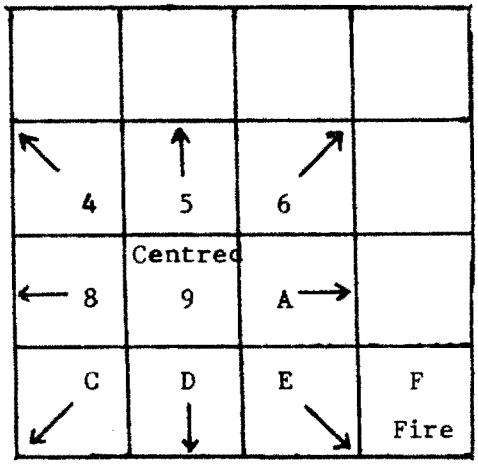
EIGHT DIRECTION JOYSTICK

GRAEME V. SAMWAYS.

Last month we showed you how to put a 4 directional 'paddle' across the keyboard. This months article gives you a subroutine that will decode the position of the paddle and will return with the corresponding key value for one of EIGHT positions, and also test if the 'F' (for 'Fire') button was held down.

The subroutine will decode the following keyboard layout:-

Each Key value corresponds to the Joystick direction indicated if wired as shown in last months diagram.



The number returned will be stored in the variable specified by the instruction at 0273, D73X. (Store Accumulator B at 003X.) I.E. Variable X, = 0 - F. The key number resides in the LSD of this Byte. I.E. 05, 0D, etc. If the 'F' button is held down, the number will appear as F5, FD, etc. This is easy to test for with an AND instruction, and is necessary because the joystick covers the operation of the 'F' button in the normal keypad routine. (Other keys can not be readily detected unless the joystick is centred.(9))

The test program displays the key value in the top left hand corner of the screen, followed by a '0' or 'F', depending on the status of the 'Fire' button. To test the program without the joystick, just push the keys 5, 8, A, D, and F. All others have no effect. (Try it.) To get diagonal movement, hold BOTH the keys adjacent to the direction required. I.E. To get 4, hold down 5 and 8. The corresponding number appears as the 1st digit. If the 'F' is down, an 'F' will appear in the 2nd digit, if not down, a '0' will appear.

To get the eight directions on the paddle very fine adjustment has to be made to the trim pots on each joystick pot. to get the right sectors. (If you lose control, you have too small a resistance on one or more pots.) When you have them adjusted correctly, put a drop of nail polish on the trim pot. movement so the arm can not move. (Do not put any on the carbon 'horse shoe' part.) When you have it right, put it all back together in the box, feed in the test program and have a play with it to get the feel of it.

Then, so that you will have a game to play with it straight away, here is a modified listing for 'SUB', which was included in our first issue. (Not 'Super Sub', which appeared in issue No.7.) If you do not have a copy of 'Sub', we still have a few copies of issue No.1 available at a cost of $4-00 each. The modified listing has a Machine Code subroutine from 023C to 02AD. It is similar to, but not the same as the general version in the test program, as it adds or subtracts the X and/or Y direction from C and D respectively, and also stops the sub wrapping around. It also sets 4 to 0, 1, 2, for firing torpedoes.

You use the 8 Directional paddle to play, moving the Sub in the same direction as the joystick. To fire torpedoes, push the 'F' button. The torpedoe will be fired in the direction the sub is moving horizontally. I.E., LEFT, ↖ STOP, ↑ RIGHT, ↗ . Vertical movement of the sub has no effect on the torpedoes.

****

MODIFIED SUB LISTING

```
0230    A3EE DE03 A3F1 DCD3   0230 128C CE80 106F
0240    0186 4AA7 0086 06A7   0186 FDA7 00A6 0084
0250    A081 A027 14D6 3D81   2027 07C1 0327 085A
0260    2005 C11C 2701 5CD7   3D86 BFA7 00A6 0084
0270    0581 0527 14D6 3C81   0127 07C1 0027 085A
0280    2005 C138 2701 5CD7   3C86 F7A7 005F A600
0290    8480 2616 86BF A700   A600 C60D 8405 8105
02A0    2708 8104 2702 CB02   C001 D734 3900 0000
02B0    0000 0000 0000 0000   0000 0000 DCD3 320D
```

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

EIGHT DIRECTION JOYSTICK TEST PROGRAM

ADDR	INST.	MNEMONIC	EXPLANATION
CHIP 8 MAINLINE (0200-022E)			
0200	6A00	VA = 00	Set X co-ordinate 00
0202	6B00	VB = 00	Set Y co-ordinate 00
0204	0230	CALL M/C AT 0230	Do 8 Direction Sub.Ret.with value in 4
0206	8040	V0 = V4	)
0208	610F	V1 = 0F	)Set up High & Low filters
020A	62F0	V2 = F0	)
020C	8012	V0 = V0 AND V1	Filter out MSD
020E	F029	I = DSP, V0	Set Index to show LSD
0210	DAB5	SHOW 5 AT VA , VB	Display LSD
0212	7A05	VA = VA + 05	Increase X co-ordinate
0214	8422	V4 = V4 AND V2	Filter out LSD
0216	44F0	SKF V4 NE F0	If 4 ≠ F0 (i.e.0) Skip next instruction
0218	640F	V4 = 0F	Set 4 to 0F (Pseudo MSD)
021A	F429	I = DSP, V4	Set Index to show pseudo MSD
021C	DAB5	SHOW 5 AT VA , VB	Display pseudo MSD
021E	6603	V6 = 03	)
0220	F615	TIME = V6	)
0222	F607	V6 = TIME	) Wait 60 ms.
0224	3600	SKF V6 = 00	)
0226	1222	GO TO 0222	)
0228	00E0	ERASE	Clear screen
022A	1200	GO TO 0200	Do again
022C	0000	NOP	
022E	0000	NOP	

EIGHT DIRECTION M.C. SUB.(022E-0275)

0230	C6 09	LDAB	#$09	Set centre value
0232	CE 80 10	LDX	#$8010	)
0235	6F 01	CLR	X+1	)
0237	86 4A	LDAA	#$4A	)
0239	A7 00	STAA	X	) Set up P.I.A.
023B	86 06	LDAA	#$06	)
023D	A7 01	STAA	X+1	)
023F	86 FD	LDAA	#$FD	)
0241	A7 00	STAA	X	) Put PA2 low
0243	A6 00	LDAA	X	Get keypad status
0245	84 A0	ANDA	#$A0	Filter out undesired values
0247	81 A0	CMPA	#$A0	See if either key closed
0249	27 08	BEQ	0253	If no key Skip 8 Bytes
024B	81 80	CMPA	#$80	See if D or 5 down
024D	27 02	BEQ	0251	If 5 down, Skip 2 Bytes
024F	CB 08	ADDB	#$08	Add twice 04 to value

EIGHT DIRECTION JOYSTICK TEST PROGRAM (Cont)

ADDR	INST	MNEMONIC		EXPLANATION
0251	C0 04	SUBB	#$04	Sub 04 from 9
0253	86 BF	LDAA	#$BF	) Put PA6 low
0255	A7 00	STAA	X	)
0257	A6 00	LDAA	X	Get keypad status
0259	84 05	ANDA	#$05	Filter out undesired values
025B	81 05	CMPA	#$05	See if either key closed
025D	27 08	BEQ	0267	If no key closed, Skip 8 Bytes
025F	81 04	CMPA	#$04	See if 8 or A closed
0261	27 02	BEQ	0265	If 8 closed, Skip 2 Bytes
0263	CB 02	ADDB	#$02	Add twice 01 to value
0265	C0 01	SUBB	#$01	Subtract 01 from value
0267	86 F7	LDAA	#$F7	) Set PA3 low
0269	A7 00	STAA	X	)
026B	A6 00	LDAA	X	Get keypad status
026D	84 80	ANDA	#$80	Filter out undesired values
026F	26 02	BNE	0273	If F not closed, Skip 2 Bytes
0271	CB F0	ADDB	#F0	Set Fire key indicator
0273	D7 34	STAB	0034	Store result in Variable 4
0275	39	RTS		Return from subroutine

****

DREAM RUMMY REVIEW.

The DREAM CARDS advertisement says that this program makes use of your DREAM'S intelligence, and it certainly does that.

All you require to play 'Dream Rummy' is 2K RAM, (and the tape and instructions from Dream Cards, 6/8 Elphin Street, Ivanhoe. 3079, of course!) This is the ONLY place you can get 'Dream Rummy'. (If it isn't, we believe that Mr. Ford, who wrote the program, is in the Law Business, and he may be interested in checking out how effective the Australian copyright laws are.)

Dream Rummy is similar to Gin Rummy, except that you only try to collect Three's and Four's of a kind, not 'runs' of cards. Even if you know nothing about the game of Rummy, you will very quickly learn, from watching the display, and reading the very detailed instructions which come with the game. These are of 'Dreamer' quality, simple and easy to understand.

There are two things you should look for when playing, but I will let you find them out for yourself. Once you have, you will have a chance of winning. The game is a real challenge, even after a long period of play, as the computer is and excellent opponent and never makes errors, which really keeps you on your toes.

The program also has a check sum routine on the end which you can run before playing 'Dream Rummy' to ensure that the program has loaded correctly, and therefore can not cheat on you.

'Dream Rummy' is fully Chip-8, and the optional explained listing is well worthwhile getting, as it is easy to follow and very informative.

That is all I have time to tell you about it, as I have to go and try to beat the computer. (The score is now 10 to 11, computer's favour.)

By the way, Dream Cards will soon have a 'Pontoon' or '21' game ready for release,which will certainly send you broke! (It requires 4K, so start building.) We will let you know when it is available.

GRAEME V. SAMWAYS.

****

VIDEO BINGO (0080 - 0400)

J. PANOS,
[redacted]

N.B. - VIDEO BINGO IS INTENDED TO BE PLAYED WITH BINGO GAME CARDS.

The program selects and displays random numbers, from 1 - 90, inclusive. Numbers may be selected automatically or manually at any time during the game. Numbers are selected once only during any game.

The selected numbers are automatically stored in RAM from 0080 and the program may be interupted at any time by using a key board code to display all the numbers selected to date. These are displayed over three video 'pages' of thirty numbers per page, and all in ascending numerical order. E.G. If No.1 has been selected to date, then it will be displayed as '01' at the top left hand corner of video page one. All numbers NOT selected to date will be displayed as '00'. The program can be changed to single random selections, or video page displays at any time you require.

KEY BOARD COMMAND CODES:-

0 : This key is normally used to break out of the routine of random number selections currently in progress. (Either manual or automatic selection modes.) It will also display a program identification and 'prompt'. (The word "BINGO" with a large "?" underneath.) The program will then be on standby to receive an input command code. Invalid code entries will be ignored.

1 : Page 1. Displays selected Bingo numbers from 1 to 30.

2 : Page 2. Displays selected Bingo numbers from 31 to 60.

3 : Page 3. Displays selected Bingo numbers from 61 to 90.

A : Automatic. This command automatically selects new Bingo random numbers and display each of them for six seconds. Keying '0' at any time within the first five seconds of the last number displayed on screen will cause a break out of this routine and display the identification, where it will wait for a new command code to be entered.

D : Delay. This key will allow manual selections to be made, using the 'F' key, with no time limit between selections. Keying '0' at any time will revert to the identification and wait for a new command.

CB : Clear Bingo. This command will reset Bingo Data to Zero for a new game re-start.

NOTE : Direct command codes may be entered after Video Pages 1, 2, or 3 are displayed.

```
00E0   0000 0000 1C12 DC12   1C48 4C4A 4948 9E90
00F0   9692 9E7E FFC3 C303   1F1F 1818 0000 1818

0200   6076 A227 F055 6080   A34D F055 603F A2AC
0210   F055 607F A2B4 F055   A382 FD65 6E0A A227
0220   2376 A227 F055 A0D0   237C F955 3A90 121C
0230   23C0 44CB 1200 6700   4401 22EA 4402 22E8
0240   4403 22E6 440A 125C   340D 1230 CB7F 6F0F
0250   EFA1 125C 6F00 EF9E   124C 1230 6F00 EFA1
0260   1230 435A 1230 2390   2346 7C01 7D01 3D0B
0270   1296 6D01 6E0A A34D   2376 A34D F055 603F
0280   A2AC F055 607F A2B4   F055 3C5B 1296 6C01
0290   6080 A34D F055 6E01   A2AC 2376 A2AC F055
02A0   A2B4 2376 A2B4 F055   234C F965 4000 7BFF
02B0   3B00 126A 7FC0 234C   F955 2352 7301 651C
02C0   660E 8EC0 00E0 2358   6F08 FF18 440A 6FD0
02D0   7F20 6E00 FF15 FF07   EEA1 1230 3F00 12D6
02E0   435A 6401 1236 771E   771E 771E A303 6080
02F0   F055 A307 6000 F055   236E 6A00 6B00 2346
```

(See bottom of Page 16 for 0300-0400.)

MORSE TEACHER (0200 - 0300)

W. BEYER,
VK3BHW,
[redacted]

This program generates random morse. The words are of random length, (one to eight letters), each word containing random letters and figures.

The initial speed is 5 words/minute, returning to the monitor after the last, and highest, speed group at 15 words/minute has been sent.

Address 0205 sets the number of words in a speed group, and is set to 05 (hex.), but of course this can be changed to suit your own requirements.

```
0200   6A60 6B98 6C05 C607  7601 A296 C31F F31E
0210   F065 6307 8302 8800  6000 6408 2242 73FF
0220   3300 122E 226C 76FF  3600 120A 1278 7002
0230   B230 6410 121C 6420  121C 6440 121C 6480
0240   121C 8482 3400 124C  125C 00EE 650F F518
0250   6505 F515 F507 3500  1254 124A 6505 F518
0260   6505 F515 F507 3500  1264 124A 85A0 F515
0270   F507 3500 1270 00EE  85B0 F515 F507 3500
0280   127C 1284 7CFF 3C00  1206 128C 7AF4 7BEC
0290   3A0C 1204 F000 120C  2C0B 0124 1B04 0274
02A0   2B14 1A0A 3B34 5C13  0309 2344 334C 6C1C
02B0   FDF5 E5C5 8D0D 1D3D  7DFD 120C 2C0B 0124
02C0   1B04 0274 2B14 1A0A  3B34 5C13 0309 2344
02D0   334C C61C FDFD
```

****

VIDEO BINGO (Cont.)

```
0300   7B01 A0DA F965 8900  A307 6E10 2376 A307
0310   F055 7A01 3A0A 132A  6A00 6E0A A303 2376
0320   A303 F055 6000 A307  F055 2350 2358 7503
0330   3542 1338 6500 7606  9B70 00EE 4B1E 236E
0340   4B3C 236E 12FE A0DA  F955 00EE A0BC 00EE
0350   8E90 A0DA F965 00EE  A0DA FE33 F265 F129
0360   2368 F229 2368 00EE  D565 7504 00EE 6500
0370   6600 00E0 00EE F065  80E4 00EE 6000 7A10
0380   00EE 0000 0000 0000  0000 0000 0000 0000
0390   607F 4300 23BA 603F  431B 23BA 601F 4339
03A0   23BA 600F 434B 23BA  6007 4353 23BA 6003
03B0   4357 23BA CB7F 7B01  00EE A3B5 F055 00EE
03C0   3700 13E8 00E0 6A0F  6B05 A0E4 DAB5 7A08
03D0   A0E9 DAB5 7A08 A0EE  DAB5 7A08 A3FA DAB5
03E0   6A1B 6B0E A0F3 DABD  6F20 FF18 F40A 340C
03F0   13F8 F40A 440B 64CB  00EE 7848 4B48 7800
```

****

RACE MEETING

(0080 - 0400)

B. N. HUSSEY,
[redacted]

You are at the Bucklands Beach Race Club and each race has three horses entered. Press any button to start the race and any button to start a new race.

The horses speeds are completely random, therefore any horse can win (or lose) any race. Each horse also has a 'handicap' in that there is a 10% chance that it may not be detected at the 'winning post' and will have to run a second lap.(If it has the time.)

Sorry, No Doubles or Trebles.

HAPPY PUNTING!

```
0080   6606 F615 F607 3600   1084 00EE 8000 8000
0090   2000 33A4 7878 4442   1000 1B52 3E3C 4488
00A0   00EA 8A8C 8AEA 8E8A   8E8A EAEC AAAA AAAC
00B0   6685 4725 C677 4567   4575 7545 4745 75EE
00C0   AAEE CAAA EE88 8C88   EE3E 2A2A 2A2A EE88
00D0   CC88 EEEE 4A4A 4A4A   B8A0 A0A8 BAEA 8A8E
00E0   8AEA A0BF DAB5 A0C4   DCB5 A0FB DDB5 00EE
00F0   EE8A CE8C EA00 EEAA   AAAA AE04 0C04 040E

0200   2300 6B11 22A0 2300   651B 22BE 22E0 22A0
0210   22BE 7BFF 75FF 22A0   22BE 6901 F918 3B00
0220   120E 2300 6B14 20E2   22E0 00E0 2300 6B02
0230   2310 2300 6B0A 2326   2300 6B12 23EA 22E0
0240   2300 6B1A 23B8 6908   F918 F00A 00E0 6A00
0250   6B01 A08C 7A01 DAB4   6901 F918 3A3F 1254
0260   6A00 6B06 6C00 6D0F   6E00 6118 2340 13D6
0270   3A38 4A39 1278 1280   00E0 2300 2310 133C
0280   3C38 4C39 1288 1290   00E0 2300 2326 133C
0290   3E38 4E39 1298 126E   00E0 2300 23EA 133C
02A0   A2EC DAB5 A0A1 DCB5   A0A6 DDB5 A0AB DEB5
02B0   A0B0 D1B5 A0B5 D2B5   A0DD D3B5 00EE A0BF
02C0   DA55 A0C4 DC55 A0C9   DD55 A0CE DE55 A0D3
02D0   D155 A0D8 D255 00EE   2080 2384 2352 1370
02E0   6650 F615 F607 3600   12E4 00EE CAAA EAAA
02F0   CE0E 020E 020E 0E02   0408 0E00 0000 0000

0300   6A03 6C0B 6D13 6E1B   6123 622B 6334 00EE
0310   A0F6 DAB5 A0FB DCB5   A2FB DDB5 A0DD DEB5
0320   A0F0 D1B5 00EE A0F6   DAB5 A2F6 DCB5 A2FB
0330   DDB5 A0A6 DEB5 A0C4   D1B5 00EE F00A 122A
0340   6815 692B 653F 6700   00EE 78FE 79FE 75FE
0350   00EE A3B3 D875 D975   D575 00EE 7AFF C202
0360   8A24 7CFF C303 8C34   7EFF C403 8E44 00EE
0370   235C 234A 0000 1270   A090 DAB8 DE18 A098
0380   DCD8 00EE A098 DAB8   DE18 A090 DCD8 00EE
0390   E8A8 E888 8EEE A8E8   A8AE E282 C381 E1BA
03A0   AAAA 2A3B B8A8 B8B0   A867 5476 5467 E648
03B0   4442 4C01 0001 0001   A390 DAB5 A395 DCB5
03C0   A39A DDB5 A39F DEB5   A3A4 D1B5 A3A9 D2B5
03D0   A3AE D3B5 00EE 2378   2352 2080 2378 2352
03E0   235C 234A 2384 2352   12D8 A0F6 DAB5 A2F1
03F0   DCB5 A2FB DDB5 A0C4   DEB5 A0AB D1B5 00EE
```

****

DAY OF THE WEEK CALCULATOR (0080 - 0400)

T. HILLCOAT,
[redacted]

This program calculates the day of the week for any twentieth century date. The user is prompted for a date, which should be entered as a six digit number, in the form DDMMYY. The date will then be displayed on the screen as it is entered. The day of the week will then be calculated and displayed alongside the date.

Press 'C' to restart the program and input another date, or 'F' to return control to the monitor.

The program is written in machine language and should be run by keying 0200, FN, 3.

```
0080   0104 0400 0205 0003  0601 0406 E785 E725
0090   E538 1010 1010 E585  E525 E738 2828 2828
00A0   F9A9 A9A9 A9CE 4A4A  4ACA E545 4545 4738
00B0   2038 2038 A9A9 A9A9  51CC 0ACA 0ACC E545
00C0   4745 4528 2828 2838  E785 E786 8538 1010
00D0   1038 0020 4080 0000  003C 0000 CEAA AEAA
00E0   CAEE 484E 484E 3808  1800 1000 0E51 01E1
00F0   0700 0451 03B4 00B4  0000 0000 0000 0000

0200   B0C0 794F 97F1 972E  96EC 972F 86DC 97F5
0210   C603 BD03 0FC6 E8BD  02FA BDC2 C481 092E
0220   F997 EDBD C079 96ED  BD02 C397 F086 D297
0230   F5C6 018D 030F C6FD  BD02 FABD 02BA 97F2
0240   86D2 97F5 C601 BD03  0FC6 FDBD 02FA BD02
0250   BA97 F396 F081 1F2E  0696 F281 0C2F 06BD
0260   034B 7E02 00DE F1A6  7F9B F0BD 0308 9BF3
0270   BD03 08D6 F354 541B  BD03 08D6 F2C1 022E
0280   0BD6 F3C4 0326 058B  06BD 0308 97F4 86D7
0290   97F5 C601 BD03 0F96  F448 4848 9BF4 9BF4
02A0   8B8C 97F5 C602 BD03  0FBD C2C4 8100 2707
02B0   810F 26F5 7EC3 607E  0200 BDC2 C481 092E
02C0   F997 EDBD C193 C605  BDC2 2406 ED48 4848
02D0   9BED 9BED 97ED BDC2  C481 092E F997 EE9B
02E0   ED97 ED96 EEC6 05BD  02FA BDC1 93C6 05BD
02F0   C224 C605 BD02 FA96  ED39 97EF 962E 1B97

0300   2E96 EC97 2F96 EF39  8007 2CFC 8B07 394F
0310   97F6 96F5 97F7 DEF6  8B05 97F5 97F7 A600
0320   3608 9CF6 26F8 CE00  0532 A707 098C 0000
0330   26F7 37C6 05CE 0008  BDC2 26C6 08BD 02FA
0340   33C0 012F 0596 F57E  0316 3986 3297 21C6
0350   40BD C2E5 7E02 0000
```

****

HANGMAN (0080 - 0400)

GRAEME SMITH,
[redacted]

This program is the popular 'Hangman" game, for two players. The first player keys in a word of up to ten letters, using the 'T.V. Typewriter' code. Key E is used to erase a previous letter and key F when the word is finished.

The second player then tries to guess the word, again using the 'T.V.Typewriter' code, with each wrong guess taking him (or her) one step closer to being 'hung'.

```
0080   F6CE B7DA E92E F492  B75A F248 B7FA B6DE
0090   F6DE 93DE 5EDE BBDE  C546 492E F6DA 56DA
00A0   BFDA B55A 4BDA F11E  A0BA 10AE A0BB DAB5
00B0   4501 120A 4E01 12F0  1240 F8A8 A8A8 A850

0200   6E00 6300 6400 6A02  6B18 6500 F00A 67FD
0210   8704 4F01 1280 0277  8100 F00A 4100 13BE
0220   4120 12C8 8101 8010  4E01 12D2 4016 10A8
0230   4020 10AC 0266 DAB5  4501 120A 4E01 12F0
0240   7A06 A0E0 F41E F055  7401 340A 120C F00A
0250   400E 128E 400F 12A0  6F10 FF18 124E 0000
0260   0000 0000 0000 9630  810F 2203 7EC1 9380
0270   10CE 007E 7EC1 9896  3048 4848 4897 3039
0280   400E 128E 400F 12A0  6F10 FF18 120C 4A02
0290   1288 7AFA 74FF 6501  A0E0 F41E F065 122C
02A0   00E0 6900 6C02 6D1E  A2BC DCD1 7901 7C06
02B0   5940 12A8 6E01 66F0  6800 1202 F800 67F6
02C0   8704 3F01 1288 1224  67FC 8704 4F01 1288
02D0   1224 8200 A0E0 F41E  F065 9020 122C 7A06
02E0   7401 5490 12D4 4301  1202 7610 8060 B310
02F0   6301 4F01 130A 7801  5890 12DE 6F10 FF18

0300   FF15 FF07 3F00 1302  12FC 78FF 122C 0000
0310   6C11 6D16 A34E DCD1  7C01 4C20 1202 1314
0320   6C18 6D15 A34E DCD1  7DFF 4DFF 1202 1324
0330   6C19 6D00 A34E DCD1  7C01 4C29 1202 1334
0340   6C19 6D01 A34A DCD5  1202 0810 2040 8000
0350   6C28 6D01 A34E DCD1  7D01 4D05 1202 1354
0360   6C27 6D05 A36A DCD3  1202 E0A0 E000 0000
0370   6C28 6D08 A34E DCD1  7D01 4D0D 1202 1374
0380   6C25 6D09 A38A DCD1  1202 EE00 0000 0000
0390   6C25 6D0D A39A DCD3  1202 2040 8000 0000
03A0   6C29 6D0D A3B6 DCD3  03BA 6F10 FF15 FF07
03B0   3F00 13AE 13A8 8040  2000 C610 D721 C640
03C0   7EC2 E500
```

****

SPACE INVADERS
(REQUIRES 2K)

(0200 - 0600)

FRED LEVER, (JNR)
[redacted]

At the start the words "SPACE INVADERS" are shown, misspelt, with a kind (?) invader fixing them. To go to the main program, press key '1'.

The object of the game is to shoot all of the invaders before they get YOU. As in the pinball parlour game, there are invaders, shields, your tank, your rockets, and of course, the invaders bombs.

The keys that control your tank are:-

Key C - LEFT
Key D - RIGHT
Key F - FIRE.

The invaders also have an invisible phasar gun which can only fire when it gets within a certain range of you. When they come close enough, the word 'DANGER' is shown at the top of the screen. When the invaders get back to the RHS of the screen, BLAMMO!!!...YOU'VE HAD IT! If you are hit FOUR times by their bombs, it is also goodbye.

If you lose, the program goes back to the words "SPACE INVADERS" at the start of the program. If you win, you will get a free game.

GOOD LUCK!

```
0200   6102 6A06 6B03 6000   22B0 3A1A 1208 6A00
0210   6B10 22B0 3A1C 1212   621C 6310 02BA D235
0220   6680 22CC A532 6C3B   6D12 DCD3 A532 DCD3
0230   7CFF DCD3 6605 22CC   3C1F 122C 6610 F618
0240   A532 DCD3 7C01 DCD3   02BA D235 7201 02BA
0250   D235 6605 22CC 3C3B   1240 A532 DCD3 02BA
0260   D235 6620 22CC A532   DCD3 7001 02BA D235
0270   A532 DCD3 7CFF DCD3   02BA D235 72FF 02BA
0280   D235 6605 22CC 321C   1270 A532 DCD3 7C01
0290   DCD3 6605 22CC 3C3B   128A A532 DCD3 6680
02A0   22CC 71FF 4100 12AC   02C3 1202 00E0 1200
02B0   02BA DAB5 7A04 7001   00EE 9630 CE05 56BD
02C0   C198 39CE 0100 86FF   BDC0 7D39 F615 F607
02D0   6E01 EEA1 12DC 3600   12CE 00EE 00E0 6000
02E0   6100 6200 6300 6400   6704 6818 6920 6A1D
02F0   6C1C 6D00 6E00 A550   6604 D784 7710 76FF

0300   3600 12FA 6704 A5FC   F455 A606 F155 A554
0310   D9A2 A606 F165 A557   C0FF 81E0 710F D011
0320   A606 F155 A5FC F465   8640 24DC 3F01 133A
0330   7403 A5FC F455 74FD   24E6 8630 24DC 3F01
0340   134C 7303 A5FC F355   73FD 24E6 8620 24DC
0350   3F01 135E 7203 A5FC   F255 72FD 24E6 8610
0360   24DC 3F01 1370 7103   A5FC F155 71FD 24E6
0370   8600 24DC 3F01 1382   7003 A5FC F055 70FD
0380   24E6 7DE2 A554 D9A2   660C E6A1 79FE 660E
0390   E6A1 7902 D9A2 660F   4802 13AE E69E 13D6
03A0   660A F618 A557 8B90   7B01 DBC1 6802 A557
03B0   DBC1 8C85 DBC1 3F01   13CA 4C1B 13CE 4C1A
03C0   13CE 4C19 13CE 4C18   13CE 3C00 13D6 A557
03D0   DBC1 6800 6C1C 340F   13EC 330F 13EC 320F
03E0   13EC 310F 13EC 300F   13EC 14B0 6602 F618
03F0   8640 24DC 8630 24DC   8620 24DC 8610 24DC
```

(See bottom of Page 21 for 0400-0600.)

MORE OPUS

FRANK REES,
[redacted]

Here is another song from Frank to play on the DREAM. The very popular, 'UP THERE CAZALY', which he tells us was done by his son,by ear.

```
0230          8050 0100  8055 8055 0100 8055
0240   5460 0100 5460 4466  0100 4466 FC80 3F00
0250   FC40 FC40 3F00 E444  B04C 0100 6C4C 3F00
0260   8055 3F00 8055 3F00  3F00 8055 0100 8055
0270   B04C 8055 7E00 2072  0100 4872 3F00 2072
0280   0100 4872 3F00 4466  0100 5166 2072 0100
0290   4872 FC00 8055 0100  8055 8055 0100 8055
02A0   5460 0100 5460 4466  0100 4466 FC80 3F00
02B0   FC40 FC40 3F00 E444  B04C 0100 6C4C 3F00
02C0   8055 3F00 8055 3F00  8055 0100 6055 B04C
02D0   8055 7E00 4466 0100  5166 3F00 2072 0100
02E0   4872 3F00 BD80 3F00  BD89 3F00 BD80 FF00
02F0   3F80 3F00 3F80 3F00  2072 0100 4872 3F00

0300   2072 0100 4872 3F00  4466 0100 5166 3F00
0310   4466 0100 5166 3F00  5460 0100 FF60 0000
```

****

SPACE INVADERS (0400 - 0600)

```
0400   8600 24DC 7DE2 4D00  6501 4D00 7E01 4D23
0410   65FF 4D23 7E01 8D54  3E0D 1452 600E 6214
0420   6300 02BA D235 7204  7001 322C 1422 3D21
0430   1452 24F4 6014 6A12  6B06 22B0 3A2E 143A
0440   6A12 6B12 22B0 3A32  1444 6650 F618 00E0
0450   1200 A606 F165 A557  D011 7101 D011 A606
0460   F155 3F01 1470 411D  147A 411E 147A 1312
0470   311F 1324 A557 D011  1312 6630 F618 77FF
0480   A557 D011 0000 0000  3700 1312 24F4 6023
0490   6A10 6B06 22B0 3A30  1494 6A14 6B0C 22B0
04A0   3A2C 149E 6A1A 6B12  22B0 3A26 14A8 144A
04B0   6630 F618 00E0 6034  6A10 6B00 22B0 3A30
04C0   14BC 6A08 6B06 22B0  3A38 14C6 6A0C 6B12
04D0   22B0 3A34 14D0 6680  22CC 12DC A532 F61E
04E0   DDEF 7D06 00EE A557  DBC1 6C1C 6800 660A
04F0   F618 00EE A554 D9A2  A556 D9A2 6010 A557

0500   8290 72FF 641D D241  A557 8390 7302 D341
0510   A557 D241 72FF D241  A557 D341 7301 D341
0520   70FF 6605 F618 3000  1510 6620 22CC 00E0
0530   00EE 2070 A820 70A8  2070 A820 70A8 2070
0540   A800 0000 0000 0000  0000 0000 0000 0000
0550   FFFF C381 40E0 6080  ⁽E7CE ᴾ93DE ᴬB7DE ᶜF24E
0560   ᵉF3CE ᴵE92E ᴺB6DE ᵛ56DA  ᴬB7DE ᴰD6DC ᵉF3CE ᴿBBDE
0570   ᶻF39E ᵗE7CE ᵈD6DC ᴬB7DE  ᴺB6DE ₒF64E ᶜF3CE ᴿBBDE
0580   ᴾ93DE ᴺB7DA ᴬB7DE ᵗE7CE  ᴬB7DE ᴿBBDE ⁽E7CE ᵠF64E
0590   ᵠF6DE ᵐ492E ᵎ0024 ᶜF24E  ᴺB7DA ᴬB7DE !8248 ᴵE92E
05A0   ᴺB6DE ᵛ56DA ᴬB7DE ᴰD6DC  ᴱF3CE ᴿBBDE ⁽E7CE D7DC
05B0   ₒF6DE B7FE D7DC ᵉF3CE  ᵇD6DC ᵞ4BDA ₒF6DE ᵘF6DA
05C0   ᵞ4BDA ₒF6DE ᵘF6DA 0000  ᴿB7DA ᴬB7DE ᵛ56DA ᵉF3CE
05D0   ⁽E7CE ᴬB7DE ᵛ56DA ᶜF3CE  ᵇD6DC 0000 ᵉF3CE ᴬB7DE
05E0   ᴿBBDE ᵀ492E ᴺB7DA !8248  ᴾ93DE ᶜF248 ᴬB7DE ᵞ4BDA
05F0   0000 ᴬB7DE ₆F64E ᴺB7DE  ᴵE92E ᴺB6DE 0006 0906
```

****

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.)

****