nascom

NEWSLETTER

Volume 3, No. 1
April 1983
£1.25

THE NASCOM NETWORK HAS BEEN UP AND RUNNING SINCE LAST SUMMER

Lucas Nascom

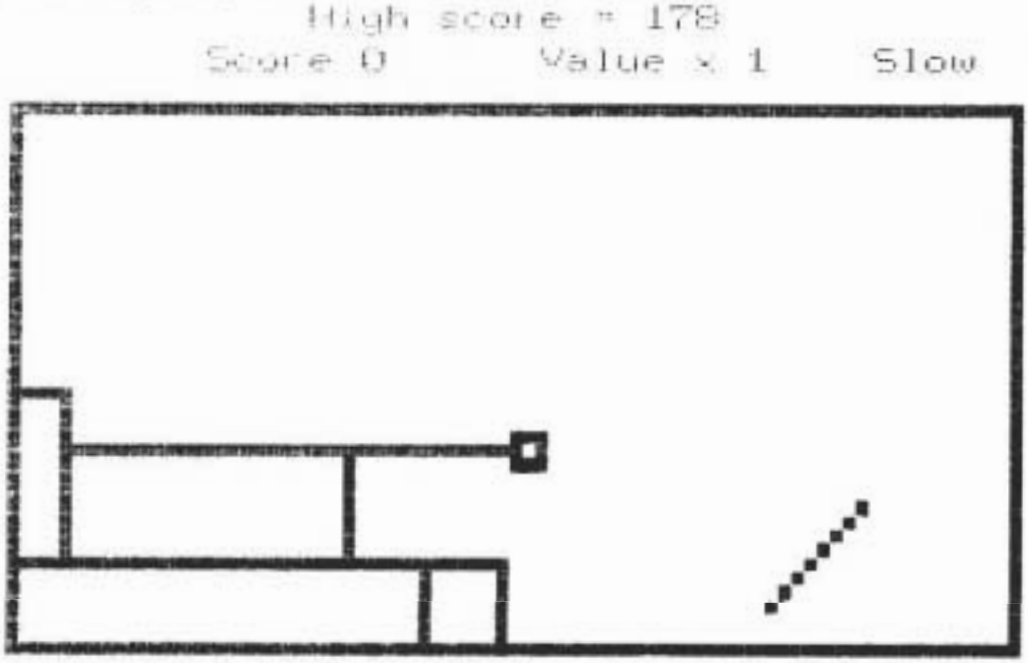
NEW PROGRAMS
from
MICRO POWER

Centipede £6.95

A very good version of the popular arcade game. You have control of a spray can of DDT and your task is to kill the bugs before they kill you. The bugs range from centipedes to spiders, all equally deadly. The game was written using a compiling BASIC so is effectively machine-code which leads to fast and smooth movement. A good game for all the family.

Dart £6.95

Another game from the author of our Invasion Earth. Written using compiling BASIC, it is very fast and addictive. Based on the arcade game 'The Qix', the object is to 'box' in the Dart by drawing round areas of the screen. In the first phase you only have to avoid being hit by the Dart when you are in the process of drawing a line but in the second phase onwards you must avoid being trapped by the Fuse which burns along the lines as you draw them. A different type of game to the standard 'Invaders' type but equally if not more addictive.



Naspic £9.95

A very useful tool for all BASIC programmers. Supplied either in EPROM or on tape, this relocateable, 2K utility allows you to add graphics and text displays to your BASIC programs easily. Using the standard screen editing facilities of Nas-sys simply design the screen display you want (graphics and/or text) and then Naspic will convert it to a BASIC subroutine and add it to the end of your program. It makes the designing and programming of screen displays and layouts easy.

MICRO POWER LTD. - 8/8a Regent Street, Chapel Allerton,
Leeds LS7 4PE

NASCOM NEWSLETTER Vol 3, No 1 April, 1983

Contents

Editorial	Page 1
Corn Corner	Page 2
Nascom Graphics ROM Editor	Page 5
Disassembler - Part II	Page 9
Letters	Page 12
Software Standards	Page 13
SYS-EX - A 1K Extension for Nas-sys Monitors	Page 16
Micro Power 64K RAM Board Review	Page 22
News from Nascom	Page 26

Editor - Ian J Clemmett

Published by - Micro Power Ltd., 8/8a Regent Street, Chapel Allerton, Leeds LS7 4PE

Printed by - Dataform Press

Editorial

Not much to tell you about this issue. I'm hoping to have reviews and comparisons of a number of graphics boards in the near future. I've also got a number of pieces of software for review (when I get some free time) so I will be able to give you my opinions on SPEX, MANOR and NASCALC.

Due to the fact that the issue of the magazine with the competition details in went out later than planned, I have decided to extend the deadline until the end of this month (30.4.83). Keep the entries coming.

Thanks to the people who responded to my request for circuit diagram drawing. I'll be in contact with you in the near future. It should make the magazine look better and there should be less errors.

Well, that's it for another editorial. Happy Nascoming...

IJC

Corn Corner

by Cresby

Random Static

A friend was telling me that at work they had trouble with CMOS chips. Careful handling, reappraisal of design, change of suppliers all failed to reduce the failure rate. One day, as all good stories go, they realised that after ignoring the problem, it had gone away. Good engineering philosophy dictated a post mortem but little transpired. Could the wet weather have reduced static build up? Did chaining workers to benches help? Then one wag pointed out (I deny even the remotest hint of authorship) that the charge hand (pause!) no longer worked there. There was no suggestion of his being discharged, he was a down to earth chap don't you know. As it happens, the offending chips were bus drivers so you would expect to see no failures for ages and then they would all arrive together.

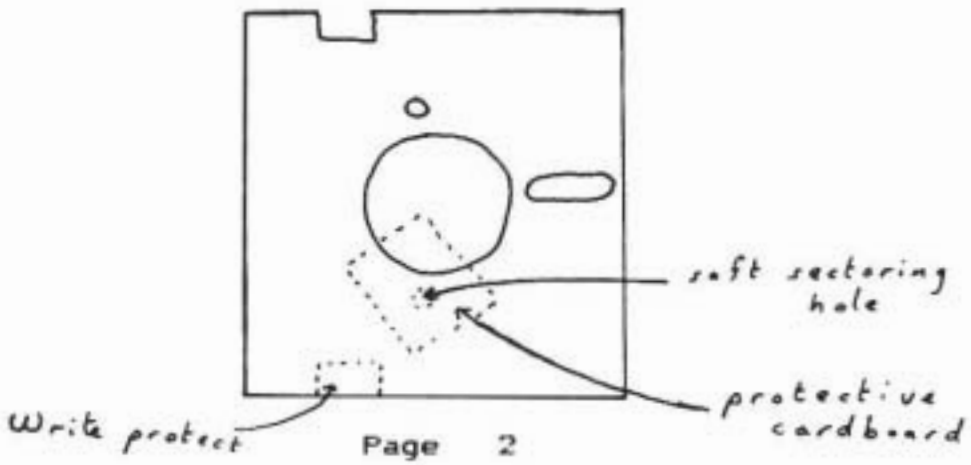
Keeping Tabs

Has anyone ever tried Tabulating nothink? Not RAM but PROM nothink. In Nas-sys 3 with the alpha field on, you get all manor of wierd displays. It only occurs if XROM is linked to a memory address with no ROM (or pin 24 of it if you bend pins). The reason is related to indeterminate states on the data bus but ignore it, don't send one of your Nascoms back like I did. (Thanks again to B&L at Kenilworth - nice guys).

NB. This is worth repeating, if you haven't seen it or forgot: any Nas-sys command defaults to whatever is in the ARG1 to ARG9 locations which means with regular taks you don't always have to repeat the length, width or alpha operands. Subsequent Modify or Execute commands pick up the same starting address if you omit their operands, which can be handy to know.

Flip Floppies

There is a company selling reversible floppy discs and I was about to buy one when I was distracted by special offers on normal discs. Most discs are double sided whether they are guaranteed as such or not. To make flip-floppies, all it needs is a soft-sectoring hole on 'tother side and a notch for write protect tabs. I would advise experimentation on older discs, a card slipped inside the jacket and the use of small sharp scissors (not a knife) to cut the plastic.



A bit of a Clock-up

There seems to be a surplus of real-time Nas-clocks based on the MM58174. The one I used was designed-in on a GM816 but this one dropped bit 4 at random. Having tried the obvious bus pull-ups and finger capacitance I still failed to modify its erractic behaviour. So, I wrote a fault tolerant algorithm. Basically, instead of reading and checking for an F nibble which signifies a 'change since last read' situation, I tried reading twice and comparing. Thus, intended F's and erratic bits both caused a re-read of all pertinent data.

Broken but not interrupted

With Gemini, the sign of twins, there was no suprise in a second problem - a broken IEI track. The sympton to that was a totally upredictable interrupt vector, not easily distinguished. By filling an INT RAM block with 10, 10, 11, 11, 12, 12, 13, 13, etc, then filling all lower RAM with E7 (Break point - BRKPT) and the I register with the INT block address, it soon became obvious that the INT vector was coming from anywhere in the block including odd addresses. With pull-ups on the data bus, the vector at (I reg) + FFH and (I reg) + 10H was repeatable and clearly illegal. A voltmeter along the IEI line confirmed whether the interrupt was being serviced because it does not reset until a RETI is executed and because of the breakpoint this awaited re-execution under Nas-sys. Of course there are many software hurdles to confuse INT inaction but if you suspect hardware, the above tricks do not need a scope.

The problem also showed how to break a program without 'Resetting' and re-entering any start up data like registers. In Basic you can escape but in assembler what can you do? Add fancy routines which take as long to debug as the program itself. Or, you could hit the NMI. The result is the same as a Breakpoint - the registers are displayed but to continue, only 'E' need to be typed. To invoke NMI there is a debounced input on pin 6 of the Nasbus or pin 4 of the ribbon connector to the keyboard. A 'push-to-make' switch has to earth one of these pins. My Break switch is physically on the keyboard. Don't hit it twice because on resuming execution it will lock out. There are cards which make use of the NMI but if you have got that far you will have already have figured out this trick. Happy breaking.

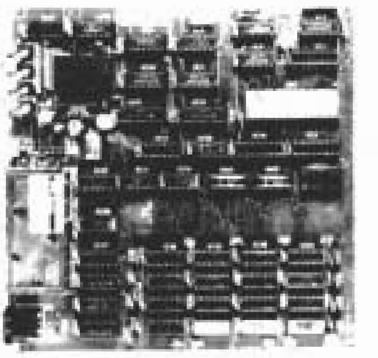
Nas-Ass you like it

In the Nas-sembler the PHASE pseudo-op appears to be two-phased! or am I anti-phase? Guess who read DEPH as depth? (revealing the depth of my mis-understanding). Even after using DEPH (dephase) properly, it still produced a correct listing but an object file displaced by the length of the code. The solution is logical; using PHASE is like using several ORG's which requires MODE 0, MODE 1 is default and the manual omits explanation so pencil this one in the margin (under phase saving). This phase-ability study cannot omit reminding the reader that MODE 0 object code loads from the assembler only, using the '/ ' command. (Don't forget the space).

HIGH RESOLUTION COLOUR GRAPHICS NEED NOT COST MORE THAN £199

Introducing the new **Microvector MV256** adding a high Resolution Colour Graphics capability to your *NASCOM* or *GEMINI* computer system.

The **MV256** uses the *Thomson EF9365* graphics display processor to provide high level graphics functions in hardware.



Both the **MV256A** and **MV256B** are supplied built and tested with comprehensive documentation and programming examples. Postage & Packing free.

MV256A **£199** +VAT

MV256B **£220** +VAT

Colour Video monitor 10 inch 0.6 mm pitch tube 9MHz bandwidth, suitable for use with MV256B. Also has PAL decoder enabling direct connection to domestic video recorder.

£199 +VAT & CARR

Arfon Light pen, Plug in, compatible with Micro Vector board.

£50 +VAT

Cable Accessories available.

For further information please contact

CLIMAX COMPUTERS LTD

17a Broad Street South Molton
Devon EX36 3AQ England
Tel South Molton (07695) 2314

Nascom 1 & 2 Graphics ROM Editor

by Z.R. Lesiakowski

For those of you with an EPROM blower wanting to create your own set of characters, the accompanying program will allow you to easily create, change and amend each character cell. It is written in machine code and will run under Nas-Sys.

As most of you are already aware, each character cell for the N1 is made up of 16 rows of 8 dots whereas for the N2 it is 14 rows of 8 dots, the last two rows being unused by the N2. As there are 128 characters in a character EPROM, the EPROM must be 16 x 128 bytes, ie. 2K. On executing the program you will be asked whether you are programming an N1 or N2 EPROM followed by the origin of the 2K of RAM where the characters will be stored. Characters are produced by using the commands listed below and storing them on tape ready for blowing into EPROM. (NB the origin must be above 1700H).

Commands.

CTRL/F - will Fill the character RAM area with FFH. Thus, unprogrammed locations will remain at FFH enabling you to overprogram the characters at a later date.

I - will Increment the location number displayed in the corner of the screen. This number corresponds to the alphanumeric address given in your Nascom manual. For the Graphics characters simply add 80H to the location number.

D - will Decrement the location number.

G - will Get a character from RAM specified by the location number and place it into the character cell.

C - will Complement a character within the cell.

B - will Blank or clear the character cell.

R - will Reflect a character in the cell. This is very useful when two characters are used to produce a symetrical shape such as a pawn from a chess set.

S - will Save the character from the character cell into the specified RAM location number.

V - will Invert a character in the cell.

u - will roll the character in the cell up.
d - will roll the character in the cell down.
l - will roll the character in the cell left.
r - will roll the character in the cell right.

E - calls the Edit mode. A particular dot may be set or cleared with the 1 and 0 keys respectively, the current position being indicated by a flashing cursor. The cursor will automatically advance to the next location when a dot is set or reset. However, this auto cursor advance facility may be disabled and enabled by pressing the A key in the Edit mode. This allows lines and other shapes to be formed easily without having to backspace the cursor, the cursor keys may be used. When in Edit mode other commands may not be entered. ESC exits the Edit mode.

M - will enter Nas-Sys. A Write command with the appropriate values will be displayed on the screen allowing all the characters to be stored on tape.

For anyone wanting the above program blown into EPROM, please send £6 for a 2716 or £8 for 2708's to:-

Z.R. Lesiakowski,
6, Westbourne Road,
Trowbridge,
Wilts.
BA14 0AJ

stating the required memory loactions of the EPROMs.

```
1000 3E 0C F7 21 C5 E0 AF 32 2A 16 77 06 12 86 23 10   >.w!E'/2*.w...£.
1010 FC FE C7 28 20 21 96 09 22 29 0C EF 42 41 53 49   !~G( !.."").oBASI
1020 43 20 52 4F 4D 20 52 45 51 55 49 52 45 44 20 21   C ROM REQUIRED !
1030 00 DF 5D DF 5D 3E 0C F7 21 96 09 22 29 0C EF 4E   ._]_]>.w!..").oN
1040 41 53 43 4F 4D 20 31 20 6F 72 20 32 20 28 31 2F   ASCOM 1 or 2 (1/
1050 32 29 3F 00 CF 32 29 16 FE 31 FD 21 75 13 28 0A   2)?.O2).~1)!u.(.
1060 FE 32 FD 21 78 13 28 02 18 CB 3E 0C F7 21 90 09   ~2)!x.(..K>.w!..
1070 E5 22 29 0C EF 4F 72 69 67 69 6E 20 6F 66 20 67   e").oOrigin of g
1080 72 61 70 68 69 63 73 20 63 68 61 72 61 63 74 65   raphics characte
1090 72 73 20 3A 2D 00 DF 63 E1 22 29 0C 06 20 CD 21   rs :-._ca").. M!
10A0 16 DF 79 38 C5 21 0B 0C 7E FE 01 20 BD 2A 0C 0C   ._y8E!..~~. =*..
10B0 11 00 E9 AF ED 5A 30 B2 3E 0C F7 01 28 00 11 CE   ..i/mZO2>.w.(..N
10C0 0B 21 4D 13 ED B0 3A 29 16 32 D5 0B 21 1C 08 22   .!M.mO:).2U.!.."
10D0 29 0C EF 4F 72 67 3A 2D 20 00 2A 0C 0C DF 66 EF   ).oOrg:- .*.._fo
10E0 11 48 00 AF 32 28 16 21 2D 08 22 29 0C EF 4C 6F   .H./2(.!-.").oLo
10F0 63 6E 3A 2D 20 30 30 48 00 2E 6B 22 29 0C EF 43   cn:- OOH..k").oC
1100 68 61 72 61 63 74 65 72 20 43 65 6C 6C 00 11 40   haracter Cell..@
1110 00 21 BC 08 22 29 0C EF 75 2C 64 2C 6C 26 72 20   .!..").ou,d,l&r
1120 52 6F 6C 6C 20 63 68 61 72 61 63 74 65 72 20 55   Roll character U
1130 50 2C 44 4F 57 4E 00 CD D1 11 EF 20 20 20 4C 45   P,DOWN.MQ.o   LE
1140 46 54 20 26 20 52 49 47 48 54 20 69 6E 20 63 65   FT & RIGHT in ce
1150 6C 6C 2E 00 D7 7B EF 49 26 44 20 49 4E 43 20 26   ll..W{oI&D INC &
1160 20 44 45 43 20 4C 6F 63 6E 2E 20 63 6F 75 6E 74    DEC Locn. count
1170 65 72 2E 00 D7 5B EF 47 20 20 47 45 54 20 63 68   er..W[oG  GET ch
1180 61 72 61 63 74 65 72 20 66 72 6F 6D 20 52 41 4D   aracter from RAM
1190 2E 00 D7 3D EF 52 20 20 52 45 46 4C 45 43 54 20   ..W=oR  REFLECT
11A0 63 68 61 72 61 63 74 65 72 20 69 6E 20 63 65 6C   character in cel
11B0 6C 2E 00 D7 1C EF 43 20 20 43 4F 4D 50 4C 45 4D   l..W.oC  COMPLEM
11C0 45 4E 54 20 63 68 61 72 61 63 74 65 72 2E 00 18   ENT character...
11D0 05 19 22 29 0C C9 D7 F9 EF 42 20 20 42 4C 41 4E   ..").IWyoB  BLAN
11E0 4B 20 63 65 6C 6C 2E 00 D7 E7 EF 56 20 20 49 4E   K cell..WgoV  IN
11F0 56 45 52 54 20 63 68 61 72 61 63 74 65 72 20 69   VERT character i
1200 6E 20 63 65 6C 6C 2E 00 D7 C7 EF 53 20 20 53 41   n cell..WGoS  SA
1210 56 45 20 63 68 61 72 61 63 74 65 72 20 69 6E 20   VE character in
1220 52 41 4D 2E 00 D7 AA EF 4D 20 20 4D 4F 4E 49 54   RAM..W*oM  MONIT
1230 4F 52 20 72 65 74 75 72 6E 2E 00 D7 94 EF 45 20   OR return..W.oE
1240 20 45 44 49 54 20 6D 6F 64 65 3A 2D 31 20 6F 72    EDIT mode:-1 or
1250 20 30 20 74 6F 20 73 65 74 20 6F 72 00 CD D1 11    0 to set or.MQ.
1260 EF 20 20 20 72 65 73 65 74 20 64 6F 74 73 2C 20   o   reset dots,
1270 00 2A 29 0C 36 09 23 36 0B 23 36 0D 23 36 5E 23   .*).6.£6.£6.£6^£
1280 22 29 0C EF 20 74 6F 20 6D 6F 76 65 00 21 BF 0B   ").o to move.!..
1290 22 29 0C EF 63 75 72 73 6F 72 2C 41 20 3D 41 55   ").ocursor,A =AU
12A0 54 4F 2C 45 53 43 20 74 6F 20 65 78 69 74 2E 00   TO,ESC to exit..
12B0 1E 25 CD 6C 15 FD 5E 01 CD 6C 15 16 41 CD 77 15   .%Ml.}^.Ml..AMw.
12C0 16 5C CD 77 15 DF 62 30 FC 11 23 44 FE 53 CA 49   .\Mw._bO|.£D~SJI
12D0 15 08 21 5F 08 7E FE 20 28 08 22 29 0C 06 05 CD   ..!_.~~ (.")...M
12E0 21 16 08 FD 46 00 FE 45 CA 22 14 FE 52 CA 94 13   !..}F.~EJ".~RJ..
12F0 FE 43 CA AA 13 FE 42 CA 11 15 FE 47 CA 36 15 FE   ~CJ*.~BJ..~GJ6.~
1300 64 CA D1 13 FE 75 CA ED 13 FE 6C CA B7 13 FE 72   dJQ.~uJm.~lJ7.~r
1310 CA C4 13 FE 56 CA 0A 14 FE 06 CA EF 14 21 28 16   JD.~VJ..~.Jo.!(.
```

```
1320 FE 49 20 03 34 18 54 FE 44 20 03 35 18 4D FE 4D   ~I .4.T~D .5.M~M
1330 20 93 3E 0C F7 EF 14 57 20 00 2A 0C 0C DF 66 11    .>.wo.W .*.._f.
1340 00 08 19 DF 66 21 0A 08 22 29 0C DF 5B 4E 41 53   ...._f!.")._[NAS
1350 43 4F 4D 20 20 20 47 52 41 50 48 49 43 53 20 50   COM   GRAPHICS P
1360 52 4F 4D 20 45 44 49 54 4F 52 20 20 20 5A 52 4C   ROM EDITOR   ZRL
1370 20 31 39 38 33 10 02 05 0E 06 09 7E FA 81 13 18    1983......~z...
1380 08 FE FF 3E 7F 28 01 AF 77 21 34 08 22 29 0C DF   .~.>.(./w!4.")._
1390 68 C3 C5 12 C5 CD 06 16 06 08 17 CB 19 10 FB 79   hCE.EM.....K..{y
13A0 CD EE 15 C1 1D 1D 10 EC 18 E7 CD 06 16 2F CD EE   Mn.A...l.gM../Mn
13B0 15 1D 1D 10 F5 18 DA CD 06 16 07 CD EE 15 1D 1D   ....u.ZM...Mn...
13C0 10 F5 18 CD CD 06 16 0F CD EE 15 1D 1D 10 F5 18   .u.MM...Mn....u.
13D0 C0 D5 05 CD 06 16 08 AF CD EE 15 1D 1D CD 06 16   @U.M.../Mn...M..
13E0 08 CD EE 15 10 F5 D1 08 CD EE 15 18 A4 CD 06 16   .Mn..uQ.Mn..$M..
13F0 05 08 1D 1D CD 06 16 1C 1C CD EE 15 1D 1D 1D 1D   ....M....Mn.....
1400 10 F2 08 1C 1C CD EE 15 18 87 48 CD 06 16 F5 1D   .r....Mn...HM..u.
1410 1D 10 F8 41 11 23 44 F1 CD EE 15 1D 1D 10 F8 C3   ..xA.£DqMn....xC
1420 C5 12 DF 62 21 27 16 38 10 35 20 F6 1D CD 9F 15   E._b!'.8.5 v.M..
1430 C4 83 15 CC 91 15 1C 18 E9 36 01 FE 1B CA E0 14   D..L....i6.~.J'.
1440 FE 31 20 05 CD C4 15 18 07 FE 30 20 0E CD D9 15   ~1 .MD...~0 .MY.
1450 3A 2A 16 B7 20 63 14 14 14 18 5E FE 41 20 25 21   :*.7 c....^~A %!
1460 2A 16 7E 2F B7 77 21 50 08 E5 22 29 0C F5 06 08   *.~/7w!P.e").u..
1470 CD 21 16 F1 E1 28 0D 22 29 0C EF 41 55 54 4F 20   M!.qa(.").oAUTO
1480 4F 46 46 00 D5 DD E1 FE 12 20 05 14 14 14 18 18   OFF.U]a~. ......
1490 FE 11 20 05 15 15 15 18 0F FE 14 20 04 1D 1D 18   ~. ......~. ....
14A0 07 FE 13 C2 22 14 1C 1C D5 DD E5 D1 CD 9F 15 F5   .~.B"...U]eQM..u
14B0 1D F1 CC 83 15 C4 91 15 D1 3E 5C BA 20 04 1D 1D   .qL...D..Q>\: ...
14C0 16 44 3E 41 BA 20 04 1C 1C 16 59 FD 7E 01 3C BB   .D>A: ....Y}~.<;
14D0 20 02 1E 23 3E 25 BB C2 22 14 FD 5E 02 C3 22 14    ..£>%;B".}^.C".
14E0 CD 9F 15 F5 1D F1 CC 83 15 C4 91 15 C3 C5 12 2A   M..u.qL...D..CE.*
14F0 0C 0C 11 00 08 0E FF AF 71 23 1B B2 B3 20 F8 21   ......../q£.23 x!
1500 50 08 22 29 0C EF 46 49 4C 4C 45 44 20 20 00 18   P.").oFILLED  ..
1510 DB AF CD EE 15 1D 1D 10 F9 18 D1 3A 28 16 2A 0C   [/Mn....y.Q:(.*.
1520 0C F5 E6 F0 0F 0F 0F 0F 57 F1 E6 0F 07 07 07 07   .ufp....Wqf.....
1530 5F 19 E5 DD E1 C9 D7 E3 11 23 44 DD 7E 00 DD 23   _.e]aIWc.£D]~.]£
1540 CD EE 15 1D 1D 10 F4 18 A3 21 5F 08 22 29 0C EF   Mn....t.£!_.").o
1550 53 41 56 45 44 00 D7 C3 FD 46 00 11 23 44 CD 06   SAVED.WC}F..£DM.
1560 16 DD 77 00 DD 23 1D 1D 10 F4 18 80 16 41 D7 13   .]w.]£...t...AW.
1570 14 3E 5D BA 20 F8 C9 1E 25 D7 08 FD 7E 01 BB C8   .>]: xI.%W.)~.;H
1580 1D 18 F6 D5 C5 F5 D7 2D D7 22 CD 43 FF F1 C1 D1   ..vUEuW-W"MC.qAQ
1590 C9 D5 C5 F5 D7 1F D7 14 CD 58 FF F1 C1 D1 C9 D5   IUEuW.W.MX.qAQIU
15A0 C5 D7 12 D7 07 46 CD ED FF C1 D1 C9 6A 26 00 E5   EW.W.FMm.AQIj&.e
15B0 16 00 C3 31 FF 7B FE 2D 30 05 3E 2C 93 5F C9 3E   ..C1.{~-0.>,._I>
15C0 5C 93 5F C9 D5 D7 BC 14 D7 B9 15 15 D7 B5 1D D7   \._IUW<.W9..W5.W
15D0 B2 14 D7 AF 14 D7 AC D1 C9 D5 D7 B5 14 D7 B2 15   2.W/.W,QIUW5.W2.
15E0 15 D7 AE 1D D7 AB 14 D7 A8 14 D7 A5 D1 C9 C5 06   .W..W+.W(.W%QIE.
15F0 08 16 44 CB 07 F5 DC C4 15 F1 F5 D4 D9 15 F1 14   ..DK.u\D.quTY.q.
1600 14 14 10 EF C1 C9 C5 AF 4F 16 44 06 08 D7 90 20   ...oAIE/O.D..W.
1610 05 37 CB 11 18 03 AF CB 11 14 14 14 10 EF 79 C1   .7K.../K.....oyA
1620 C9 3E 20 F7 10 FD C9 00 00 00 00 00 00 00 00 00   I> w.)I.........
```

Disassembler Part II

by M. Reed

Welcome to Part II - I hope that you managed to struggle through Part I alright, but don't worry, it can only get worse. By now you should have your print routine working to your satisfaction - called PRDIS.

Now there are 9 subroutines to write, yes 9! and also a lot of data. One of these, is a sort of controller, and calls one of the remainder depending upon the data it is fed. A great deal of the disassembler is a great deal of data, organised into tables.

I will define the control routine;

i) On entry HL' contains a pointer to some data. This holds the routine number to which control will pass. This routine number is held in the least 3 significant bits (0, 1, 2) so ANDing the contents of HL' [ie (HL')] with 7 will give this number. Don't forget to increment HL' to point to the next piece of valuable data.

ii) Bits 3, 4 and 5 may contain data that the called subroutine will need, though not the only data it may need. Consequently the data byte [ie (HL') before incrementing!!] must be put somewhere - just in case.

iii) Bit 7 of this data byte is used as a sort of 'stop' flag. If it is not set the CONTROL will loop back on itself in order to call the next routine. If it is set then CONTROL will break out of the loop, call PRDIS and then move onto the next op-code to be disassembled.

Obviously HL' must contain the correct address. This is achieved by creating a table of twelve addresses, corresponding to the 12 combinations of TYPE and F. Let me explain, TYPE represents whether the instruction to be disassembled began with £ED, £CB or neither. It can only have 3 possible values - use 0, 1 and 2 because life is so much easier. F will hold bits 6 and 7 of the instruction being disassembled. Therefore, it has 4 possible values. Hence 12 combinations. There should be an address in your table for each possible case. Get it into HL'.

This should cause no problems if you order your table correctly. For example: (and may I recommend this method)

```
TABST   TYPE = 0    F = 0
        TYPE = 0    F = 1
        TYPE = 0    F = 2
              "
        TYPE = 2    F = 3
```

Now the first byte of each address is given by TABST + (2 x

((TYPE x 4) + F)). Store the most significant byte first. Use a couple of shift lefts to multiply by 2 and 4, and don't corrupt anything!!. You may now send your Nascom off to CONTROL. But wait!! The table of routine addresses is, as yet, unconstructed.

This table will start with the address of routine 0 then 1, 2 etc, finishing with that of routine 7. CONTROL will work along similar lines to the last routine described. You now want, OK need, to know about the other routines. They are subroutines, and as such MUST end with a RET (£C9) instruction. This applies to all of the routines 0 to 7.

Routine 0 - This is, in effect, a jump but depending on the value held in H (not of HL fame). H can be 0 through 7. So a table will need to be built directly after the £00 byte calling this routine. This table will be 16 bytes long and contain 8 addresses, corresponding to H=0, H=1 etc. Work out which address is needed and then put it into HL'. This will trick CONTROL into effecting a very complex 8 way jump. Note that bit 7 of the calling code for this routine must NEVER be set else when CONTROL is returned to, it will think that has just finished the last routine and not loop back to do our carefully prepared jump. For some unknown reason this routine played havoc with my disassembler. That is, until I remembered that Routine 0 is the first routine - Not Routine 1.

If I might digress for a while to give a short analogy as to what I see CONTROL as. CONTROL seems to be a microprocessor itself. It has HL' as its program counter. Bits 0, 1 and 2 represent the opcode, bits 3, 4 and 5 the data (if required). Routines 0 to 7 are just units, black boxes if you like (you haven't heard of the Black-Box Theory!!!), within the processor. Routine 0 is just a complex jump instruction. Some of the other routines are for handling lists. I hope this analogy is useful to you. I know - Clear as mud!!!

And now a description of Routine 1.
Routine 1 - This routine will add a string of characters to STRDIS. Don't forget the pointer/marker system has to be updated. The first character will be pionted to by HL' and when the routine is finished HL' must point to the byte after the last charecter that was added. Confused? I was, but if you can do this properly then a lot of work (and RAM) is saved later on. This routine can be compared to STRDIS=STRDIS+"xxxxxx" in BASIC. Where xxxxxx is the literal string to be added. May I suggest setting bit 7 of the last character of your literal? Thus, when this is sensed your routine will know the last charecter. Remember to reset bit 7 before adding to STRDIS.

For example

```
01 41 42 43 C4 nn
```

will cause ABCD to be added to STRDIS, leaving HL' pointing to nn. C4 is ASCII for D, but with bit 7 set. What will

```
81 41 42 43 C4 nn
```

do? Well it will add ABCD to STRDIS, but will also let CONTROL know that this is the last routine to be called and so CONTROL will be exited leaving PRDIS to be called - and then the next instruction.

Routine 2 - This adds the Gth item, in the following list, to STRDIS leaving HL' pointing to one after the last character in the list. You can see that this breaks down into 3 parts

i) Move HL' to the beginning of the required element of the list.

ii) Add the element to STRDIS. It may be more than one character.

iii) Move HL' to one after the end of the list.

If you use, as I did, the 'Bit 7' technique for your list, then part (ii) may just be a call to Routine 1. See how much work you can save by spending a little time on these routines.

Routine 3 - This is exactly the same as Routine 2, only not for G, but for H - not the in HL. In fact I will nearly do this one for you!!. Store G somewhere. Put H in G's place, call Routine 2, replace G. Easy isn't it?

When all these routines are done, you will find that most of the higher ones will call one lower, and so nested subroutines build up. In fact to single step through the disassembleris fascinating as you can watch the stack pointer (SP) drop like a stone as all the return addresses get pushed on. Then, as it finishes with each routine the stack pointer slowly grovels it's way back to where it began. The first few times you watch this your heart will be in your mouth as you wait with bated breath for the return addresses to be removed, or will it crash? It still fascinates me.

Routine 4 - This routine is really like accessing a two dimensional string array, and adding the string to STRDIS. There are six lists of eight elements. Don't worry!! Not more data, no, these lists were discussed last time. The actual list will be specified in CONTROL's command, ie as the data bits (3, 4 & 5 - go back if you have forgotten!). The string to be added to STRDIS is, in fact the Gth item in the specified list. Easy!! Now the lists each have a code (for the data bits for CONTROL)

Bits 345	Name
000	r(G)
001	ss(G)
010	dd(G)
011	n(G)
100	c(G)

```
        110     a(G)
```

For example if the byte given to CONTROL was £24 (hex), ie 00/010/100 and G was 2 then the routine should add D to STRDIS.

The first least 3 significant bits are the routine code, next 3 up represent the list - as above.

Routine 5 - Easy, do Routine 4 but use H (not of HL fame) instead of G. Go back and see what I did for Routine 3. Get the idea?

Well thats all for now, I don't want you all to end up by going mental with frustration!! Remember that the disassembler as a whole will hang around a) The main subroutines, b) Data and c) Pointers. All must function perfectly. If they do then all will go sweetly, if not then....

Letters

Snowdinger 2 revisited.
This letter arrived from Mr. Kemplen, 2 Vicarage Close, Menheniot, Liskeard, Cornwall about the Snowdinger 2 circuit.

> "...After an evenings work I produced a circuit board and added it to the machine. Great. It worked on tabulate and the odd machine code games that I have, so I loaded Basic and tried again. Dismal failure I'm afraid. After about 30 seconds the Basic crashes back to monitor. ..."

Any ideas. I've sat and thought about it but I have not been able to come up with any reason why it should work from Nas-sys and not Basic. Mr. Kemplen is also offering a PCB for Snowdinger 2 so if you want one you should get in touch with him.

For Sale - Nascom 2, 48K + graphics, Hobbit, Imp, Nasdis, Zeap, Naspen, Books + Manuals, other software including Sargon Chess £570
Tel. Colchester 841293

For Sale - Memory cards, both with Page mode fitted. OK at 4MHz (no waits)
48K RAM B £70 (save app. £60 on new kits)
64K G802 £80 (save app. £50 on new kits)
C. Bowden 0209-860480

For Sale - Nascom RAM A board - 4MHz, no RAM £20, Hisoft Pascal (8x2716) £40 (full documentation), Bits & PC's Toolkit (2x2708) £13, Naspen (2x2708) £12 (full documentation for both), EPROM Programmer 2708/16 £15, Nascom 3A power supply £15, I/O board with PIO and UART (all documentation). Phone 0249-814361

For Sale - HSIN Digital cassette system, save & load files at 6000bps (equivalent 7,500 baud CUTS). Software control in 2x2708 + lots of software on digital cassette. Cost £350 inc 9 certified digital (mini) cassettes, will accept £175 ono
Gemini EPROM board with 8x2716 + 8x2708, total cost £100, only £50
Bits & PC's PCG board. Programmable character generator + software. Cost £70, only £40. Castle cassette enhancing unit. Cost £20, only £10.
Phone 0249-814361

For Sale - Nascom 2, 48K RAM B, in vero rack, sound, manuals, Tandy monitor, lots of Micro Power games. £375 ono.
Phone 08675-3750

For Sale - I/O Systems High Resolution Board with software and documentation. £45 or exchange for a RAM A.
Nas-Sys ROM £9
Ring 05433-2855 evenings

Software Standards

by Rory O'Farrell

For some years now, Nascom users have been fortunate in having available to them one of the best machine code monitors for microprocessors. The limits of memory within which Richard Beale worked to write nas-sys 1 & 3 are now not as tight, in view of the drop in memory prices over the last two years and the ready availability of 64K machines at a reasonable price.

The journal of the Nascom users is the proper place to start a discussion on how such a monitor ought now to be extended. We are no longer tied to tight memory constraints. Chris Blackmore, with his MONITOR.COM has shown how Nas-sys can be adapted to run on CP/M based systems. Nas-sys 3 uses only 63 of the 256 possible 'Sys calls. Extensions to 'Sys were suggested as possible for the Nascom AVC card. Real Time Clocks can be purchased or built to run off a pair of PIO ports. A number of 80-column display cards are on the market. Three (more now) colour display cards are currently available.

Each of these items needs driving software. Such software is dependant on the particular hardware configuration used on the object being driven. For example, my RTC circuit, homebrewed, uses an inverted chip select to that written about by David Hunt in 80-Bus News No. 4. In consequence, I have to use (slightly) different driving software which has to be embedded in a program or accessible in a standard way. I suggest that the standard way for Nascom users should be through the calling conventions of Nas-sys.

It will be simple enough to define certain SCAL numbers for specific tasks, and to define the way they pass their parameters/results. For example, the RTC might be called using whatever SCAL is defined for it with HL pointing to the area where the time/date is to be stored. The colour boards such as the Pluto or Climax might also avail to the 'Sys facilities in that predefined graphics routines (Draw line, Circle,etc) might be assigned 'Sys numbers.

Implementation of each of these elementary functions (primitives?) will be system depentant of course. It will be necessary for each Nascom owner to do a little bit of work building up the library of routines to suit his own system, but when that is done, programs can be interchanged between owners in the knowledge that they will run on one machine or another provided that the host machine has all the facilities required. Obviously, one can not expect a program to read Day, Date and Time on a machine that has not got a clock fitted, but one could expect an error message and a safe return. Consider a Pluto owner who sends a program to a Climax owner. Under the old machine code system, the Pluto program hasn't a ghost of a chance of running on the Climax. Using the suggested Extended 'Sys, the program will be able to rely on certain standard primitives being available. It will not have to concern itself with the precise byte by byte structure of those primitives. Instead, it will know (and I use the word usefully!) that circles are drawn by a particular SCAL number and lines by another.

I hope that this letter will serve to provoke other Nascom Users into considering this proposal.

This is something that has been in the back of my mind for quite a while now and as Mr. O'Farrell has put something in writing, I would like to take the proposal further. Being the editor of this magazine, I get to see a vast amount of ingenious software and hardware ideas. Some of it is from professional sources but the vast majority of the ideas come from private individuals who have applied some ingenuity and imagination to a problem. I would like to propose that we set about creating a specification for a new standard of operating system for the Nascom computer. This would take into account most, if not all, of the current hardware available but also leave the doors open for further expansion (speech and speech recognition, picture recognition, etc). It would need to support page mode (Gemini boards, MAP-80 256K boards, etc) and it would also be nice to see the facility to communicate with other machines and possibly additional processors.

Well, these are my thoughts on the subject, what are yours? Get your pens out and start writing.

IJC

Nascom & Gemini USERS

NEW 32K C.M.O.S. BATTERY BACKED RAM BOARD

from

MICROCODE (CONTROL) LIMITED

FEATURES:

EFFECTIVELY REPLACES EPROMS.
Does away with the inconvenience of EPROM programming and the compromise of assigning valuable address space to ROM.

ON BOARD RE-CHARGEABLE Ni-Cad BATTERY RETAINS MEMORY FOR OVER 1000 Hrs.
Battery is automatically charged during power-up periods.

HIGH SPEED OPERATION up to 6 MHz WITHOUT WAIT-STATES.

FULLY NASBUS[1] and GEMINI-80 BUS[2] COMPATIBLE.

PAGE MODE SCHEME SUPPORTED.
The board can be configured to provide one 32k byte page or two completely independent 16k byte pages.
Complete pages of 64k bytes are simply implemented by adding more boards on to the bus.

SOFTWARE and/or HARDWARE READ/WRITE PROTECTION.
4K blocks in either page are link selectable to be aligned on any 4K boundary.

FULLY BUFFERED ADDRESS, DATA AND CONTROL SIGNALS.

MEMORY I.C. SOCKETS ARE LINK SELECTABLE TO SUPPORT ANY 24 PIN 2k byte MEMORY I.Cs.
Thus the board can support up to 32k bytes of any mixture of cmos, nmos rams or 2716/2516 eproms.
All options are link selectable using wire links plugged into gold-plated socket pins, avoiding the risk of damage and the inconvenience caused by soldering links directly to the board.
The printed circuit board is manufactured to the high quality demanded by industrial users and conforms to B.S.9000.
The board comes assembled and tested and is supplied with a minimum of 2k bytes of low-power cmos ram. Fully documented.

AVAILABLE NOW!

PRICES:

Board with 32k bytes	£184.95
Board with 16k bytes	£149.95
Board with 2k bytes	£99.95
TC5517AP Very low power 2k cmos memory I.C.	£6.50

Please add £1.50 (P&P) & VAT @ 15%

[1]nasbus is a trademark of nascom microcomputers a division of LUCAS LOGIC
[2]trademark of GEMINI MICROCOMPUTERS LIMITED

Cheques/PO's made payable to:-

MICROCODE (CONTROL) LTD
40 WELL TERR., CLITHEROE,
LANCASHIRE, BB7 2AD
ENGLAND. 0200 27890

For all Microprocessor Applications

Consultancy, Design and Manufacturing Services to Industry

Call
MICROCODE (CONTROL) LTD
40 Well Terr., Clitheroe,
Lancashire, BB7 2AD
England. Tel. 0200 27890

NEW PRODUCT

MICROCODE (CONTROL) LTD.
40 WELL TERR., CLITHEROE,
LANCS, U.K. 0200 27890

"Aspirins are not the only solution to your backplane problems"

BP14C BACKPLANE

14 SLOT 80-BUS & NASBUS TERMINATED BACKPLANE

BP14C has been designed, using mainframe techniques, to solve microcomputer problems. As the speed of microcomputers increases, the demands on the bussing system become more and more critical. Without the use of a properly terminated backplane, system performance and reliability will inevitably suffer.

BP14C has been designed to overcome the three major backplane problems:

1. Noise generated by reflected signals.
2. Noise generated by crosstalk.
3. Noise generated and transmitted through backplane power rails.

These problems have been attacked on the BP14C by:

1. Terminating ALL active bus signals into a potential balanced R.C. filter.
2. Interlacing all active bus signals with *ground 'shield' tracks.*
3. Forming a complete ground plane on one side of the backplane and using very wide tracks for the other power rails.

The Backplane measures 14" by 8" and supports up to 14 slots. Smaller lengths can be obtained by a simple 'score and break' operation.

The backplane will fit neatly into a 19" rack with spare space available for a Power Supply Unit.

The backplane features proper implementation of both the interupt and the bus request daisy chains.

BP14C is another product designed in line with Microcode (Control) Limited's commitment to industrial quality 80-bus and NASBUS performance.

Price £47.00 plus £1.50 (Post and packing), plus VAT

Cheques/PO's made payable to:-

MICROCODE (CONTROL) LTD.
40 Well Terrace, Clitheroe,
Lancs., U.K. Tel. 0200 27890

SYS-EX A 1K extension program for NAS-SYS monitors.
Part I

by David G. Johnson

Over the next few issues of the Nascom Newsletter, a description of the development and the contents of a 1K extension to NAS-SYS will be given. The monitor extension is called SYS-EX and is fully compatible with both NAS-SYS 1 and NAS-SYS 3. A full hex dump of SYS-EX will be printed in a future issue of the magazine.

So what is SYS-EX and what does it do? In addition to the normal NAS-SYS keyboard commands, SYS-EX uses twenty one of the lower case letters of the alphabet to extend the range of commands available. These commands plus an additional range of routines may be called from assembler programs in a very similar way to NAS-SYS routines. An added bonus for the users of Nascom 8K Basic, is the set of routines which allow you to use program names up to forty two characters in length. The forty two character names are completely compatible with the standard one character names.

Wherever possible within SYS-EX, the door has been left open to allow further additions to the available facilities. Indeed, without making any changes to the SYS-EX code or tables, it is quite possible to define your own commands which can be called by typing a single letter in the same way as normal NAS-SYS commands. SYS-EX uses its own internal table of routine addresses which could, if required, be amended to 'customize' the monitor extension further. SYS-EX is written entirely in relocatable code and the address table entries are all relative addresses. SYS-EX does not require any of its own workspace, although certain routines do make use of some of the fields within the standard NAS-SYS workspace.

The monitor extension can be loaded from tape and run in RAM whenever required. However, the author has found it extremely convenient to put SYS-EX into EPROM (a single 2708 is ideal) and set the hardware reset jump (Nascom 2) to cold start SYS-EX at switch on.

Here is a full list of the SYS-EX additional keyboard commands. A brief description of each command is included and a full description of how to use each command will follow at a later date.

a Display the 10 NAS-SYS arguments. (ARG1 to ARG10)

b Tabulate memory in double byte format. (i.e. 'HiLo' instead of 'Lo space Hi' with the NAS-SYS Tabulate command.)

c Swap the contents of two specified areas of memory.

d Convert a decimal number in the range -32768 to +65535 to hexadecimal.

e Erase a byte at a specified memory location and move the contents of memory down to fill the gap.

f Find up to nine occurrences of a specified string. (Hexadecimal or character string input.)

h Convert a hexadecimal number in the range 0H to FFFFH to decimal.

i Insert a byte at a specified memory location and move the contents of memory up to make space.

k Keyboard input only. Suppress those annoying spurious characters when your cassette recorder switches off.

l Label a cassette tape. Tape labels are displayed on the top line of the screen when tapes are read. Label names may be between zero and forty two characters long.

n Return to NAS-SYS. (Rarely needed as all NAS-SYS commands can be used normally from within SYS-EX.)

q Query tape contents. Read a cassette, scanning for and displaying all tape, program and file labels.

r Read a named file. Works as for NAS-SYS 'R', but only loads the file with the name you specify.

s Search for and display the next tape label as written by the SYS-EX 'l' command.

t Print on the top line of the screen. The required text can be typed in, or may come from any specified memory location.

u Set up user routine addresses for the 'x', 'y' and 'z' commands.

v Verify a named file. Works as for NAS-SYS 'V', but only verifies the file with the name you specify.

w Writes a named file. Works as for NAS-SYS 'W', but first writes a file label with the name you specify. Names may be between zero and forty two characters in length.

x Calls the user routine whose address is stored in ARG8. (ARG8 may be set up using the 'u' command.)

y Calls the user routine whose address is stored in ARG9. (ARG9 may be set up using the 'u' command.)

z Calls the user routine whose address is stored in ARG10. (ARG10 may be set up using the 'u' command.)

SYS-EX commands are entered in the same way as NAS-SYS commands except that the lower case leters of the alphabet are

used in place of the upper case letters. During the operation of SYS-EX, all of the NAS-SYS keyboard commands remain available for use in the normal way. With the exception of the 'd', 'h' and 'u' commands, all of the SYS-EX commands use the arguments ARG1 through ARG10 in the same way as does NAS-SYS. The 'd' command is a special case in that the NAS-SYS arguments ARG1 through ARG10 are neither updated nor used by the command. Also, the single required argument for the 'd' command may be in a format which is normally unsuitable as a command argument. (i.e. a minus sign and up to five digits are permitted.) The 'h' command updates ARG1 through ARG10 in the normal way, however, the command itself will only recognise an argument which is entered on the same line as the command letter. The 'u' command places a value in ARG8, ARG9 or ARG10 without the necessity of entering 8, 9 or 10 values on a command line.

Certain SYS-EX commands require further input parameters to be supplied after entry of the command line. The find ('f') command requires the entry of a find mask, while the remainder of commands in this category require the entry of a file or a tape label name.

Single step and non-maskable interrupt processing is in no way altered by SYS-EX. However, after each of these facilities has been used, control of keyboard input returns exclusively to NAS-SYS. i.e. the SYS-EX keyboard commands become temporarily unavailable until a SYS-EX warm or cold start is executed. The same situation applies when a program is ended by a breakpoint or by a NAS-SYS MRET instruction. (.. or by the action of the RESET button unless the hardware reset jump is set to the beginning of SYS-EX.)

A number of the features within SYS-EX have been designed specifically with the Basic programmer in mind. Perhaps the most notable amongst these features are the routines which provide enhanced alternatives to Basic's CLOAD, CLOAD? and CSAVE commands. The SYS-EX routines not only permit program names up to forty two characters in length, but also allow access to tape labels written using the SYS-EX 'l' command. Program files which are written using the SYS-EX routine will be successfully recognised and read by the standard Basic commands. Similarly, program files which are written using the standard Basic command will be successfully recognised and read by the SYS-EX routines.

The routines which convert between decimal and hexadecimal numbers are particularly useful when using the PEEK, POKE, DEEK and DOKE commands in Basic. Both for memory addresses and for the values contained in particular memory locations, Basic uses the decimal notation. However, much of the useful documentation quotes values in hexadecimal. The SYS-EX 'd' and 'h' commands provide easy methods of converting in either direction.

Other features will also prove useful to the Basic programmer. For example, the SYS-EX 'k' command will in many circumstances be invaluable in suppressing the spurious input

characters which are often received from a cassette recorder (and which make a mess of the screen) after a program has been read from tape.

Many of the features within SYS-EX will be of interest to the assembler programmer. All of the SYS-EX commands may be called by an assembler program. The available commands include all of the SYS-EX keyboard commands plus several additional useful routines. All calls are made to the same entry point within SYS-EX. This entry point is three bytes on from the beginning of SYS-EX. The required routine number is specified by the value in the byte which immediately follows the call instruction. Those of you who are familiar with using the NAS-SYS routines from within an assembler program, will notice the similarity in the use of SYS-EX routines. By way of an example, if SYS-EX is installed between B000H and B3FFH, a straightforward call to a SYS-EX routine is coded as follows:

```
CD  03  B0  nn
```

where nn is the required routine number. The routine number for a keyboard command is the ASCII code of the command letter. (e.g. 'a' and 'z' are represented by 61 and 7A respectively.) In addition to the keyboard commands, a further nine routines are provided in routine numbers 78 to 83. Complete details on these routines will be printed in a future issue.

Having described some of the features of SYS-EX, it might now be of interest to explain something of the structure of this monitor extension. SYS-EX begins with a control section which takes over from NAS-SYS in recognising and routing both upper and lower case keyboard commands. The mechanism for calling SYS-EX commands from an assembler program is embedded within this control section. Part of the control section processing works out the absolute address at which SYS-EX is actually installed. This absolute address is used in conjunction with an internal table of relative addresses, to calculate the absolute address of each required routine. The internal table of relative addresses is located between offset 0139H and 017EH inclusive. Each table entry is two bytes long with the first table entry representing the 'a' command, the second table entry representing the 'b' command and so on. Entries in this table represent the start of the routine relative to the beginning of SYS-EX.

In common with NAS-SYS, SYS-EX does not use the alternate set of registers nor does it use the index registers. SYS-EX does not require any of its own stack or work space, although it does make use of the NAS-SYS stack. Certain fields within the NAS-SYS workspace are used by SYS-EX, although the places where such fields are used will always be explicitly mentioned within the text.

And now for a bit of history. SYS-EX began it's life about a year and a half ago when the author decided that it would be useful to have the facility to name machine code files written

to tape. This was to be achieved in a similar way to the naming of program files from the 8K Basic. The required routines were written and provided many advantages over the facilities within Basic. (e.g. forty two character names and tape label processing.) However, a big problem remained. How could the new routines be easily used? The aim was to find a method of calling the new routines by using a single letter followed by the necessary arguments - that is, in the same way as you might use NAS-SYS 'W' or 'R'. After much head scratching and may reems of assembler coated paper, the problem was solved. It was decided to use the lower case letters of the alphabet as the command letters ('w' for named file write, 'r' for named file read, 'v' for named file verify, etc.) Part of the head scratching had resulted in a method of 'hooking into' the main input loop of NAS-SYS, thereby providing a means of branching to SYS-EX processing whenever a lower case command letter was keyed. This method did not require any changes to NAS-SYS, so there was no non-standard EPROM blowing to do. During the development of SYS-EX, the author upgraded his Nascom 2 from NAS-SYS 1 to NAS-SYS 3. As a result of this, the 'hooking into' routine was amended to make the whole thing compatible with all present versions (and maybe future versions too?) of NAS-SYS. As the author did not posess an assembler of any description (pencil, paper, Z80 manual and NAS-SYS 'M', being the available tools), and as it was not known where in memory the code would end up, it was decided to make the routines fully relocatable. (i.e. no absolute jumps or calls and no table data containing or referenced by absolute memory addresses.) This policy of creating entirely relocatable code was continued throughout the development - sometimes at great personal risk to the author's mental health. The result is that the final article can be run at any available memory location without need for re-assembly.

Having created the mechanism to call new commands as if they were normal NAS-SYS commands, it seemed a shame to stop development with only the named file commands present. After all, there were many useful features that the author had always hankered after, which could now be easily tagged on to NAS-SYS. Besides which, there were still some twenty unused lower case letters just begging to be used. All that was now needed was to write the routines! After a good deal more head scratching, many more reams of assembler coated paper and a partially seized brain, (counting backwards in hex for those relative jumps is detrimental to health) out popped the final version.

"What shall we call this thing?" the author asked himself. (Talking to himself was now a regular occurrence. - Total madness was not far away.)

"SYS-EX", he replied to himself. The monitor extension (and insanity) were complete.

THE M.A.P. VFC FOR NASBUS & 80 BUS

80 Column x 25 line screen, and Floppy disk controller on a single 8" x 8" professionally built plug in card. Fully compatible with the current range of NASCOM and GEMINI products. FEATURES: 80 x 25 paged memory mapped screen, Flicker free display, Standard ASCII character set, 128 graphic chars or inverse video, Onboard software, Video switch & keyboard port options, 5 1/4" floppy disk controller. Available separately as kits or built boards.

VIDEO CARD : £89 KIT £110 BUILT
F.D.C. : £95 KIT £115 BUILT
VIDEO & FLOPPY: £165 KIT £199 BUILT

Also available, floppy disk drive system using 96 TPI, SS/DD Teac half height drives, in a slim case complete with P.S.U. and all cables.

SINGLE DRIVE SYSTEM 500K £299 BUILT
DOUBLE DRIVE SYSTEM 1 Mb £499 BUILT
6K RAM CARD £105 KIT (64k Vers)
256K RAM CARD £150 BUILT (64k Vers)

We can support most systems with CP/M software - Ring for details.

PRICES EXCLUDE P&P (£1.50) & V.A.T.

COME AND SEE US AT OUR NEW SHOP.

MAP 80 SYSTEMS LTD.,
Chertsey Computer Centre
1 Windsor St., CHERTSEY,
Surrey. Tel 09328 64663

PARKSTONE ELECTRICS

NASCOM AUTHORISED DEALER

GEMINI PRODUCTS ALSO STOCKED

Computer Systems Built for Special Applications

Personal PEARL now available on NASCOM

Create all your business programs for less than £200

PARKSTONE ELECTRICS
18, Station Road,
Lower Parkstone,
Poole, Dorset, BH14 8UB.
Tel: Parkstone (0202) 746555

PLUTO

A BREAKTHROUGH IN COLOUR GRAPHICS

PLUTO is an intelligent, high performance colour display controller that is both compact and affordable:

Technical Profile:-

- **8" x 8" PCB**
- **640 (H) x 288 (V) x 8 colours**
- **16 bit Intel 8088 microprocessor**
- **192 Kbytes of dual ported memory**
- **Fast parallel I/O interface for connection to most computers**

Use PLUTO with any machine

PLUTO'S in built software gives fast access to a comprehensive command set:-

- **High level software protocol.**
- **High speed vector drawing. (over 150,000 pixels per sec.).**
- **Built-in character display freely mixable with graphics.**
- **On-board storage of user-defined symbols.**
- **Complex polygon flood fill**

An extended command set ROM is available providing the following functions:-

- **Raster operations with rotation in multiples of 90°.**
- **Complex polygon colour and pattern fills.**
- **Circles and arcs, etc.**

Prices:-

PLUTO ..... £399 + VAT
BABY PLUTO (320 x 288) .... £299 + VAT
DOUBLE RES. OPTION ..... £50 + VAT
8 MHz PROCESSOR OPT ... £50 + VAT
EXTENDED COMMAND ROM... £60+VAT
MANUAL ..... £6 ONLY

(OEM Discounts Available)

io research ltd.

6 Laleham Avenue · Mill Hill London NW7 3HL
Telephone: 01-959 0106

Micro Power 64K RAM Board Review

by S. Weir

I had deliberately waited several months after it was first announced so that, firstly, any bugs would be ironed out before I bought one, and secondly, so that I could read the review of it that was promised in Micropower. Unfortunately, not all the bugs had been found (or they had been left in deliberately to make life more interesting?) and the promised review did not appear. Perhaps it was thought that there was not enough interest in the board - there probably aren't too many unexpanded Nascom around now I suppose. However, this board makes an inexpensive way to add 64K of RAM and a PCG to a Nascom 1 - total cost about £100. (A Nascom 2 does not require the buffer board to be populated and the cost is about £85.)

The PCB itself is of the highest quality and is worth every penny of the £43 - except for one small point - there is a small piece of track missing, about half an inch long, on the underside of the buffer board section near to the Nasbus connectors. It should connect from the track coming from B17 pin 5 to the Nasbus. Not only does this mess up the addressing of the memory but it also causes havoc with the I/O (ie. keyboard!) since the missing track means that A3 is not connected to B4, the port decoder. You can imagine what fun I had trying to sort that out. This is the only PCB error I found, however, I noticed during my investigations into the track layout that the DMA daisy chain does not appear to have been implemented (not that it really matters to most people). Instead of going to Nasbus pin 17, /BA0 is connected to /BA1 near to B13.

On the whole, the manual supplied with the board is very good with clear explanations of the hardware. For those of us with a nervous disposition, do not put the tantalum capacitors in the board the way the instructions say. This results in your lovely new 64K board exuding several puffs of smoke and an obnoxious smell of fried tantalum capacitors! This brings back memories of the Nascom 3A power supply - funny how these designer chaps don't know their +'s from their -'s.

The instructions for fitting 6116's to the PCG seem wrong to me - when fitting one 6116 it should be fitted at G1, not G3, otherwise it is sitting on top of the 6576. Also, the instructions say that when fitting two 6116's to get a full 4K of programmable graphics, the links L1 and L2 should be set up as for 4118's. I think this is wrong and they should be as described for the 6116. Details are given for switching in the extra 6116 via G7. This disables the 6576. An alternative method is to keep the 6576 enabled and select between 6116's by taking G6 pin 13 high or low.

The explanation of the memory mapper is, on the whole, clear but I would have liked to know how to map the PCG at 0000 which at the moment I can not see how to do. Also the method given for

the port decoding seems incomplete. It is necessary to connect /IOEXT as well as /NASIO to the lowest decode of B4 (ie. pin 9) otherwise the Nascom 1 on-board port decoder will not be encoded. (At the moment I am not clear as to why /IOEXT and /NASIO are seperated on the buffer board.) It is possible to use the on-board PIO by using a circuit by Mr. D. Ritchie, published in INMC.

The values of the electrolytic power supply decoupling capacitors are not given anywhere, nor is there any information on getting the extra data and address signals out of the Nascom 1 video circuitry. However, this information is available in Micropower, Vol. 1, No. 1 in the article by Mr. S. Hope, but note that the connectors to pins 10 and 14 of the character generator socket must be swapped over.

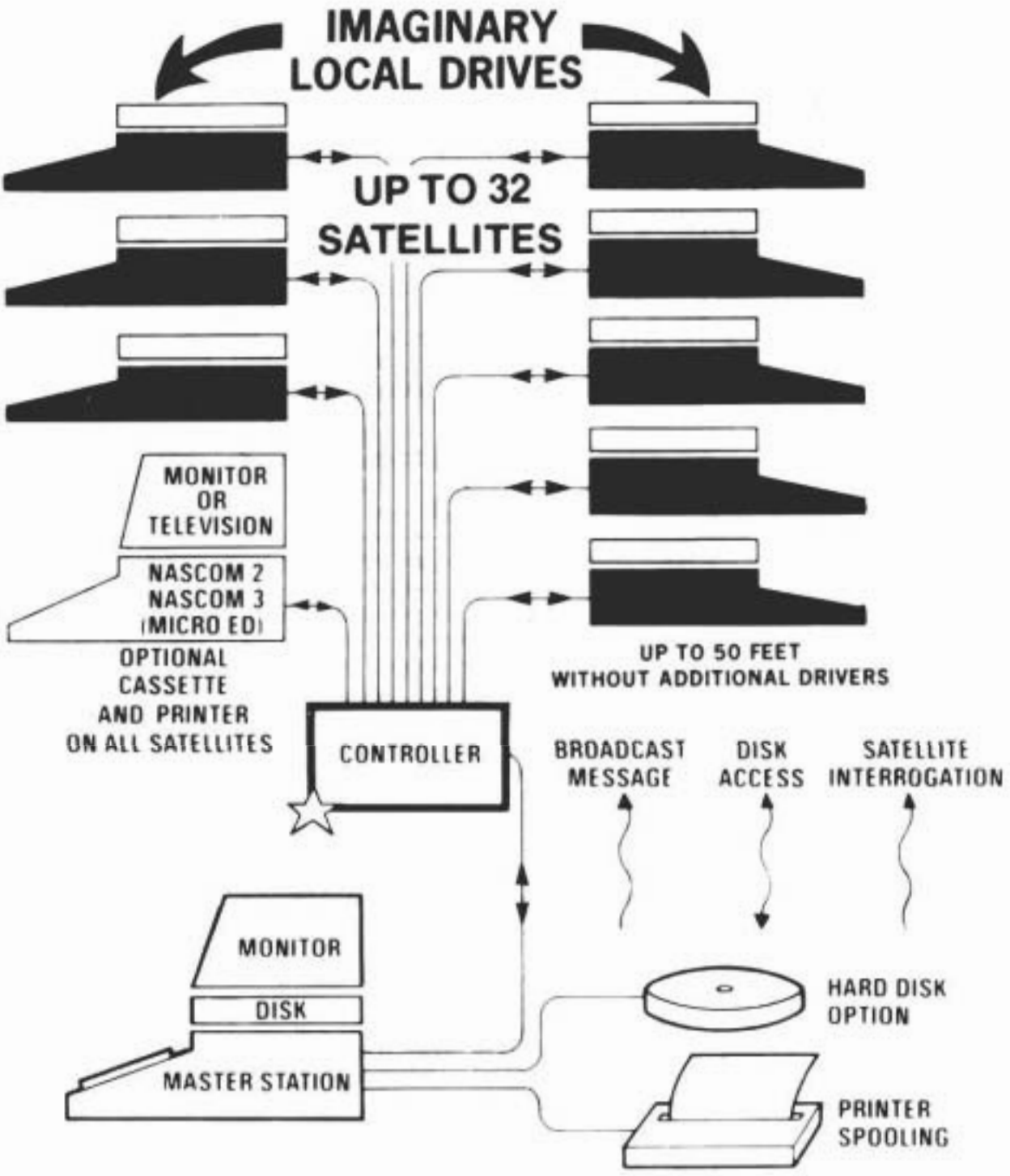
Actual construction is quite straight-forward. I used sockets throughout despite the warning in the manual. The silk-screened legends for the R's and C's are a little ambiguous in places. There is a small amount of handwiring to do - mainly between the address lines and the PCG, and a few to add from the memory mapper. I used verowire with the combs glued to the outside edge of the board to guide the wires.

Overall then, a good quality product allowing one to expand a Nascom to 64K at a very reasonable price. Unfortunately, it took me a while to get mine going due solely to errors in the documentation and on the PCB itself. At least I know more about my Nascom now than I did before I started building the PCB.

Lucas Nascom Systems

NAS-NET 1

A Networking System for Nascom Computers



NAS-NET 1
A Networking System for Nascom Computers

NAS-NET 1 is a low-cost system which allows the sharing of resources between a network of linked Nascom microcomputers.

A system consists of a central 'master' computer linked to up to 32 satellite computers. Satlelites can be located at distances of up to at least 50 feet from the master. Data is transmitted between the master and satellites at a speed of around 64,000 baud, giving fast response in most circumstances. The master computer needs to be a Nascom 1, 2 or 3 computer equipped with disc drives and the NAS-DOS disc operating system. The satellites can be any Nascom 1, 2 or 3 computer fitted with the NAS-SYS 1 or NAS-SYS 3 operating system. The satellites are individually wired to the location of the master computer using 3 core cables connected to their standard RS 232 connections. These are connected through a multiplexer unit which is connected to the master via the standard RS 232 connection of the master station. The satellite machines require no modification other than fitting of 1K of EPROM and 3K of RAM (for a modified version of the NAS-DOS operating system), and setting of the required data transmission clock. The master station requires the fitting of a special NAS-SYS monitor EPROM, one connection from a board test point to the PL4 connector and setting of the data transmission clock. The PL4 connector can still be used for driving a parallel printer.

The satellite machines when connected to the master appear to have all the facilities normally found on a dedicated NAS-DOS machine, except for potentially dangerous commands such as Format. They are able to use the discs for both program and data storage and retrieval. The satellites will not be affected by the network until they request access to the disc. The satellites can also be disconnected from the network and used in a purely local mode with a cassette recorder. They perform identically to a standard Nascom (although the disc commands will of course be ineffective). Note that a cassette recorder cannot be used while machines are connected to the network.

The master computer retains full access to the disc operating system and all its normal functions, except that the cassette recorder cannot be used. When not connected to the network the master computer behaves exactly like an unmodified machine and retains all its disc commands.

The system allows all the machines in the network to share access to the disc drive(s) and there will also be support of spooling of data for printing by the master on a parallel printer. A serial printer can be supported by the master station by fitting the Nascom I/O card to the master. The high data rate normally gives reasonably quick response to disc access by any terminal. Note however that the satellites are handled in turn by the master, and in the extreme case of all satellites attempting to store or retrieve data simultaneously there will, inevitably, be a delay while they are all processed. As a low-cost, simple system NAS-NET 1 does nor provide any file security facilities, other than write-protection for individual files to prevent unauthorised access to files or to prevent one user writing to files while another is reading them.

This specification is provisional, and further details of the operation of the system will be released when finalised.

Lucas Logic Limited
Welton Road Wedgnock Industrial Estate
Warwick CV34 5PZ
Tel: Warwick (0926) 497733 Telex: 312333

NEWS FROM NASCOM

In future we intend to divide the news from Nascom into three main sections - new products, news of existing products and technical tips.

This month's newsletter includes a brief description of the Lucas Nascom LX and the SPEX word processing package. A few notes on the use of some existing software packages follow, and we finish with an article on using various printers with the Nascom - a subject on which we regularly receive requests for information.

We did promise in the last issue an article on the AVC, but we are now holding this over for the next issue, due to shortage of space.

We have recently changed our arrangements for dealing with customer enquiries. As before your first contact should be with your dealer, but if you do need to contact us direct ask for the Sales Department for price/availability/literature, or the Customer Service Desk for technical or maintenance information. Please make sure that you know the serial number of any purchase on which you have a query. I must emphasise the importance of returning registration forms - an amazing number of people do not bother, and they are of course the ones who don't receive automatic product information updates or, in the case of some software products, revised releases when these are issued. Since software is easily pirated (not of course by the people who read this magazine) we insist on serial numbers being quoted in the case of software queries.

In the next issue we will have the article on the AVC and its support software. We will also give a description of how to provide a buffered keyboard, giving the ability to type in while the computer is calculating, at a cost of about 2p!

1. NEW PRODUCTS

THE LUCAS NASCOM LX MODEL

Those of you who went to the 'Which Computer?' exhibition will have seen the new Lucas Nascom LX model. This is a new style of packaging intended to make the appearance of the machine more attractive to the business user. It is an additional model rather than a replacement for any existing product. The internal electronics are similar to that used in Nascom 2/3. We will be giving more information on this in the next issue.

THE SPEX WORD PROCESSOR

SPEX is a word processing program developed specifically for

use on the Nascom microcomputer. It can be used on any Nascom 1, 2 or 3 (or the LX) with NAS-SYS and tape or NAS-DOS and discs. A minimum of 32K of user memory is required.

SPEX operates in two modes - command and edit. In command mode a range of instructions is available allowing text data to be loaded, saved, and printed. In the case of discs the directory can also be listed and files may be deleted. In edit mode text can be typed in and corrected. For initial use no specialist knowledge is required - the machine will perform like a normal typewriter. However, in addition to the normal Backspace key there are keys to move left, right, up and down the screen, and by positioning the flashing cursor symbol at an appropriate position corrections can simply be typed in. These screen editing facilites are consistent with the normal Nascom screen editor. Further commands are available to allow fast scanning through the text, location of page boundaries, locating and optionally changing each occurrence of a word or phrase, etc.

Instructions can be included in a text file to fix paper and margin widths, and these may be changed at will during entry of the text. Text can be right justified or spread to fill the exact width of paper specified. Facilities are included to allow automatic centring of headings on a page and for making use of bold face printing on printers equipped with this facility.

Printing on paper can be initiated simply by typing PRINT in command mode. Text will be formatted as specified by the width, margin and other commands. Page boundaries are detected automatically, and there is an option for stopping at the end of each page (for sheet paper) or continuous printing (for continuous stationery). A new page can be specified if required by a command in the text file. In addition to printing the entire text the print operation can be started from any required point within the text file. Most types of printer can be used with the SPEX program via the interfaces fitted as standard to the Nascom computer.

Many optional additional commands are provided to allow page numbering, titling of pages, deletion of blocks of text, moving or duplicating passages within the text, etc. A HELP command is provided to assist the user, and to give access to more specialised controls, such as setting double line spacing for draft documents.

SPEX is an extrememly powerful and versatile word processing package, but one which it is also very easy to use. Although a tape version is available SPEX will probably be primarily of interest to the disc user, who can load it very quickly. NAS-PEN will continue to be available in EPROM and on disc, and is of course supported by NAS-DOS.

2. NEWS OF EXISTING PRODUCTS

NAS-SYS 3A

We are now using NAS-SYS 3A instead of NAS-SYS 3 in the Nascom 3. The only difference from NAS-SYS 3 is that the OB2 80 is performed automatically to switch in the normal display rather than the AVC in machines fitted with the AVC. If you have NAS-SYS 3, an AVC and the means to blow your own EPROM you can change your existing NAS-SYS 3 to 3A by making the following changes:

Location 0407:

```
3E 80 D3 B2 EF 2D 4E 41
53 2D 53 59 53 20 33
```

CP/M AND NAS-DOS

A number of users have asked for information about using both NAS-DOS and CP/M on the same machine. The difficulty arises of course from the need to change the memory map - normally achieved by re-wiring the header plug LKS1. We described in an Application Note (AN-0011) how this can be achieved using a multi (8 !!) way switch. A neater solution is now provided by Business and Leisure, who are selling a small PCB which mounts onto the board in place of LKS1 and the MD PROM. This incorporates a number of relays, and it only needs a two-pole switch to activate this board to switch from NAS-DOS to CP/M. We have used this ourselves, and it is an Approved Product. You can contact B & L on 0926 512127.

MANUALS

After I wrote the section on manuals in the last newsletter describing the new documentation some (person?) decided to *change the volume numbers! The corrected list now reads as* follows:

Volume	Contents
1	User manual - Intro' to BASIC, Hardware manual, NAS-SYS 3
2	Graphics manual - AVC software and hardware
3	Personal data file
4	Operating systems - Polysys, NAS-DOS, CP/M
5	System software - ZEAP, NAS-SEMBLER, Pascal
6	Hardware options - RAM B, PSU, I/O board
7	NAS-DOS/NAS-SYS applications software
8	CP/M applications software

All these volumes, and the individual manuals within them, are available separately, as are the binders. Your usual supplier can give you further information and prices.

NAS-SEMBLER NEWS

1. A fault occurred in the automatic serialisation of some copies of NAS-SEMBLER which can result in errors when using the /Z80 command in edit and the EDIT command in Z80. Fortunately this can be corrected quite easily. Simply execute EDIT in the normal way and then use the COPY command to re-save the program. Do the same for Z80. This ensures that the programs are saved with the correct length, so as not to conflict with the data area.

2. The manual (Rev 1.2) refers to the DELETE command on page 10 and in the summary on page 14 as being used to delete lines. The DELETE command in fact is used to delete files (as described on page 8 of the manual). The command used for deleteing lines in ERASE x y where x is the line number of the first line and y is the line number of the last line to be deleted. This error crept in when the meaning of the two commands was reversed at a late stage of development.

NAS-DOS NEWS

Very surprisingly a fault has come to light within the last 3 months. This can occur when there is an error on the disc and the head is not returned to the initial point. Unfortunately the error procedure does not issue a command to return the head to the first track, but rather to step back and out again. In an extremely small number of cases (about 6 to date) this can result in a situation where the drive apparently becomes inoperative. A few drives have been returned as faulty for this reason, when in fact no fault existed. There is just one command which does initialise the heads on the disc and this is the JF - format - command. In the very unlikely event that you should get an apparent failure of one of your drives, which will not read a disc, it is worth trying issuing a JF command to reset the heads. Don't do this with a good disc in place though - either put a blank disc in or issue the command with no disc in place.

This problem is being corrected in Rev 1.4 of NAS-DOS. If you wonder what Rev 1.3 was, this was used in the Networking systems and offered, in that application, a slight speed advantage over Rev 1.2. This change is also incorporated in Rev 1.4.

MANOR NEWS

When the last newsletter was written the release of Rev 4 of MANOR was predicted. Since then a number of additional improvements have been implemented, and the next release after Rev 4 will therefore be Rev 6. In addition to the speed increases this offers records of up to 510 bytes, 42 fields, upper or lower case commands, access to two disc drives doubling the maximum number of records to 4000 and a CHange command

within HINTON to allow existing files to be changed without needing to create new ones.

I would again remind users that copying a file structure is better accomplished using FILCPY than the CReate command of HINTON, particularly since some users have experienced problems with the latter!

Rev 6 should be issued by mid-March to all existing users WHO HAVE REGISTERED - copies will not be issued to anyone who has not returned the registration form. This will, however, have to be the last free update issued to users of copies of MANOR sold before July 1982 - we have already extended the free update service beyond that originally offered, and updates are largely improvements rather than corrections to errors.

EXTENDED BASIC

All the versions (tape, NAS-DOS and CP/M) of this are now available. Versions including the graphics commands for use with the AVC are provided on the same tape/disc as the normal version as standard.

It should be noted that the existing version of XBASIC 3.1T has an error in the printer routine (device 1) - the AF register is not saved before checking the status of the handshake line, and a SCAL 6F is used to drive the printer instead of SCAL 6E. As you cannot easily patch this in the available space we suggest that you use your normal printer driver (or see the listing in AN-0006 in this issue) and insert the address of this routine at 3B32. Remember that you should not put the driver at 0C80 for XBASIC, as this is the keyboard input buffer, and that if you put it at the top of memory you should use a CLEAR statement to protect it from BASIC.

NAS-NET - NETWORKING FOR NASCOM COMPUTERS

Networking allows several computers to share the disc and printer resources of one of their number. Where several computers are used together - such as in education - this can result in significant savings. NAS-NET is a system for Nascom computers which achieves this objective. NAS-NET has been in use for over 6 months in several sites, and so can be regarded as a well proven system. More information is given in the advertisment elsewhere in the Newsletter.

3. TECHNICAL TIPS

USE OF PRINTERS WITH NASCOM 2 AND 3

We probably receive more enquiries relating to the use of printers with the Nascom than on any other subject. The hardware and software implications of using vaious types of printer with the Nascom are described in Application Note AN-0006, which is reproduced below.

1. Introduction

Nascom microcomputers are designed to allow the use of a wide range of different printers. The procedures used to connect these different printers depend on the individual printer, but can generally be divided into three groups.

1. Serial printers without handshake. Most slower printers fitted with a serial interface to RS232, V24 or 20mA current loop standards can run continuously at their data transfer speed - usually 300 baud (equivalent to 30 characters per second). Both Nascom 2 and Nascom 3 have a suitable serial interface fitted as standard.

2. Faster serial printers running at a data transfer rate of 1200 baud (equivalent to 120 characters per second) can accept coded characters at this speed, but cannot continuously print this fast. Instead they have an internal memory in which the data is stored prior to printing. If the computer continues to output characters at full speed the printer memory fills, and if the printer could not tell the computer to wait some data would be lost. Such printers are therefore normally fitted with an additional connection, called the 'handshake'line, which is connected to the computer to notify the computer when the printer memory is full. A connection is available on Nascom 2 and 3 for this purpose, although an additional software routine is required to support it.

3. Parallel interface. In this case instead of the code for each character being sent serially along one wire a separate wire is used for each code bit of the character. The interface is frequently referred to as a 'Centronics' interface, snce it has been widely used by one of the largest American specialist printer manufacturers, Centronics. The uncommitted parallel ports from the PIO fitted as standard to the Nascom 2 and 3 can be used to drive such printers.

This application note describes in detail how to use any of these types of printer with a Nascom 2 or 3 computer. Normally your dealer will be able to supply the correct connecting cable, but for those wishing to make up their own full details of the connections required are given. If you have a suitable cable then you should concentrate on the software details. These software procedures are very easy to implement, although the extensive description of all possible methods of use given in this note may make them appear rather more difficult than is actually the case.

2. Printer connections

2.1 Serial printers without handshake

Your dealer will normally supply an appropriate cable and

connectors. Those making their own cable should use the standard 25-way D-type connectors and make the following connections:

Computer D-type	Function	Printer D-type	From PL2 Pin
1	Protective ground	1	
2	Data transmitted by computer	3	6
3	Data received by computer	2	3
7	Signal ground	7	11

The connection for the computer to receive data is only required if the printer also has a keyboard for data input.

2.2 Serial printers with handshake

These printers require the additional 'handshake' line to be connected in order that the printer can notify the computer when printing is to be suspended pending emptying of the printer memory. Normally your dealer will supply suitable connectors and cables, but if you wish to make up your own you should follow the instructions below.

The printer handshake may be either at a 5 volt TTL level, as in the case of the Nascom Imp printer, or may be at 12 volt RS232 levels. The connections to be made should be exactly as described for the serial printer without handshake but with the following additional connection.

Computer D-type	Function	Printer D-type	From PL2 pin
20	Handshake at 12 volts	20	8

Pin 8 of PL2 must be connected to TP3 on the main board. (Standard on all recent production Nascom 2's and 3's).

The handshake signal input must be limited to 5 volt TTL levels, which is done as follows: (this connection is standard all on current Nascom 3's)

Connect pin 20 of the D-type socket in the computer to a 1K ohm resistor, the other end of the resistor being connected to the junction of 2 diodes (eg 1N4001). One of these diodes is connected to the ground pin (7) and the other to the grey lead from pin 8 of PL2; in both cases the cathode of the diode (the end marked with a ring in most cases) is connected to the common junction.

2.3 Parallel printer connections

Once again your dealer should be able to supply suitable cables and connectors, which will be made up according to the data given below for those assembling their own connections.

The printer can conveniently be connected to the uncommitted PIO chip supplied with Nascom 2 and 3. We suggest using a 25-way D-type connector for this purpose. The wiring between this and the Nascom 2 board should be carried out as described in application note AN-005 (this connection is already made on Nascom 3).

The connections between this D-type connector and the printer should be made as follows:

D-type at comp.	Function	From PL4 pin	Centronics Connector
1	PIO B5/Printer data 5	1	7
2	PIO B6/Printer data 6	3	8
3	PIO B7/Printer data 7	5	9
7	PIO A0/Printer busy	13	11
8	PIO A1/Printer strobe	15	1
14	PIO B4/Printer data 4	2	6
15	PIO B3/Printer data 3	4	5
16	PIO B2/Printer data 2	6	4
17	PIO B1/Printer data 1	8	3
18	PIO B0/Printer data 0	10	2
21	Ground	16	14

3. Software

NAS-DOS users who have returned their registration form will have received a Utilities Disc containing some utility routines to simplify use of printers of any of these types from BASIC, the monitor NASPEN and ZEAP.

3.1 Serial printers without handshake

When using printers which do not require the 'handshake' line the printer is simply enabled by the XO command in NAS-SYS, or by the appropriate POKE's in BASIC.ie

NAS-SYS 1	NAS-SYS 3
1 DOKE 3189,1925	1 DOKE 3189,1919
2 DOKE 3187,1917	2 DOKE 3187,1911
3 POKE 3112,0	3 POKE 3112,0

(See the NAS-SYS and BASIC manuals for further details)

The routine may be disabled using the NAS-SYS N command, a reset or by a DOKE 3187,1919 (1913 in NAS-SYS 3) from BASIC.

ZEAP and NAS-PEN both assume printers of this type for printed output. Instructions for use of a printer with Nascom Pascal are included in the manual.

3.2 Serial printers with handshake

NAS-DOS owners are catered for by the utilities included on the utilities disc.

To enable the computer to recognise the handshake signal when using printers requiring this, it is necessary to enter the following routine in to a suitable memory location 0C80 using the NAS-SYS monitor, or the equivalent POKE's from BASIC:

```
0C80 21 88 0C 22 78 0C DF 5B
0C88 F5 DB 00 E6 80 28 FA F1
0C90 DF 6E C9
```

To access the printer from NAS-SYS, first enable this routine by typing EC80 and then use the U command.

In BASIC the routine can be treated as follows (located at X, where X=3200 for location 0C80 hex)

```
5  X=3200
10 FOR I =0 TO 10
20 READ A
30 POKE X + I,A
40 NEXT
50 DOKE 3192,X
60 DOKE 3187,1918:REM 1912 IF NAS-SYS 3
70 DATA 145,219,0,230,128,40,250, 241
80 DATA 223,110,201
90 END
```

In NAS-PEN the routine should be entered and enabled using NAS-SYS; NASPEN should be cold started and then exited, and location 101D modified using the command M101E DF 75. If the NAS-PEN file is saved this modified printer vector is also saved, although it is still necessary to enter the code and initialise it.

NAS-DOS users are provided with the utility program UTS which is loaded at the top of memory and allows printer output to a self-contained routine of the type described above.

3.3 Parallel printers with Centronics interface

A suitable driver is included in the UTS file on the NAS-DOS utilities disc. The coding for a suitable routine is attached. The routine is used as the User output routine under NAS-SYS in exactly the same way as described for the serial printer with handshake.

In NAS-SYS the routine should be entered using the M command, and enabled by the EC9B command.

From BASIC the procedure is as described above, except that the range of the FOR loop should be 0 TO 50 and the DATA statements should be as follows:

```
      70 DATA 245,219,4,203,71,32,250,241,245,211
      75 DATA 5,203,207,211,4,0,203,143,211,4,0
      80 DATA 203,207,211,4,241,217
      85 DATA 62,207,211,6,62,253,211,6,62,2,211,4
      87 DATA 62,15,211,7,33,128,12,34,120,12,223,91
```

Note that the initialiser must be executed once, and this must be done with the EC9B command from NAS-SYS. The DOKE commands will enable the driver subsequently.

The routine is disabled by the NAS-SYS N command, or a reset, as described in section 3.1.

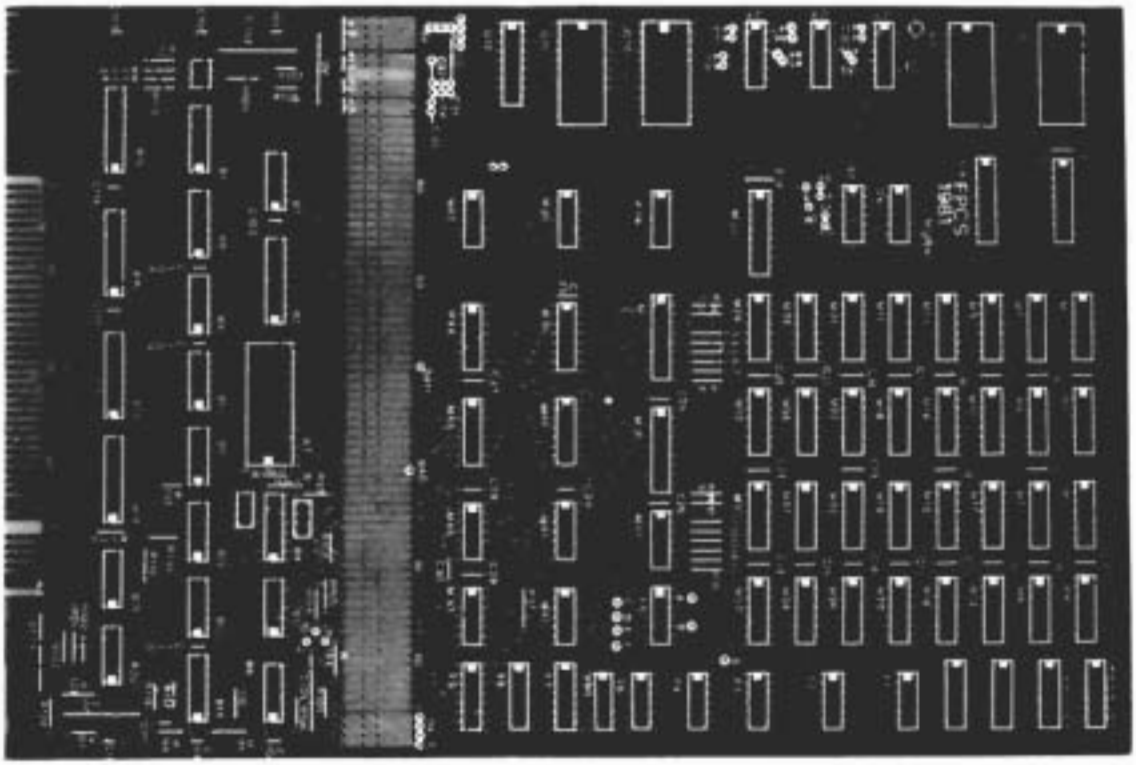
```
      NAS-SEM - Nascom Z80 assembler Page 0000

                0000 ;:PRINTDRV:
                0001 ;
                0002 ;SERIAL PRINTER WITH HANDSHAKE
                0003 ;-----------------------------
                0004 ;
                0005 ;REV 2.1    4 DECEMBER 1981
                0006 ;
                0007 ;THIS ROUTINE IS FOR USE WITH SERIAL
                0008 ;PRINTERS REQUIRING A 'HANDSHAKE'
                0009 ;CONNECTION.
                0010 ;
                0011 ;THE HANDSHAKE LINE SHOULD BE CONNECTED
                0012 ;AS DESCRIBED IN APPLICATIONS NOTES
                0013 ;AN-004 AND AN-006,AND THESE ROUTINES
                0014 ;SHOULD BE USED AS DESCRIBED IN THOSE
                0015 ;APPLICATIONS NOTES.
                0016 ;
                0017 ;**************************************
                0018 ;
                0019 ;INITIALISER - EXECUTE THIS IMMEDIATELY
                0020 ;
                0021          ORG  0C80
0C80 21 88 0C   0022 HANDS    LD   HL,HANDR ;GET OUR USER ROUTINE AD
0C83 22 78 0C   0023          LD   (0C78H),HL ;PUT IN THE USER VECTOR
0C86 DF 5B      0024          SCAL 05B ;RETURN TO NAS-SYS

                0025 ;
                0026 ;PRINTER HANDSHAKE PROCEDURE
                0027 ;
0C88 F5         0028 HANDR    PUSH AF ; SAVE THE CHARACTER
0C89 DB 00      0029 HERE     IN   A,(0) ; WAIT UNTIL PRINTER IS FREE
0C8B E6 80      0030          AND  080H
0C8D 28 FA      0031          JR   Z,HERE
0C8F F1         0032          POP  AF ; NOW RETRIEVE THE CHARACTER
0C90 DF 6E      0033 CON      SCAL 6EH ; AND PRINT IT
0C92 C9         0034          RET
```

```
       NAS-SEM - Nascom Z80 assembler  Page 0000

                    0000 ;:CENDRV:
                    0001 ;
                    0002 ;PARALLEL PRINTER DRIVER
                    0003 ;-----------------------
                    0004 ;
                    0005 ;REV 1.3      4ADDRESSES
                    0028 PORTB  EQU    5
                    0029 CONPTA EQU    6
                    0030 CONPTB EQU    7
                    0031 ;
                    0032        ORG    0C80
0C80 F5             0033 CENTP  PUSH   AF
0C81 DB 04          0034 CP1    IN     A,(PORTA) ;CHECK IF BUSY
0C83 CB 47          0035        BIT    0,A
0C85 20 FA          0036        JR     NZ,CP1 ;WAIT TILL FREE
                    0037 ;
0C87 F1             0038        POP    AF
0C88 F5             0039        PUSH   AF
0C89 D3 05          0040        OUT    (PORTB),A
0C8B CB CF          0041        SET    1,A ;STROBE THE DATA OUT
0C8D D3 04          0042        OUT    (PORTA),A
0C8F 00             0043        NOP    ;DELAY FOR SETTLING
0C90 CB 8F          0044        RES    1,A
0C92 D3 04          0045        OUT    (PORTA),A
0C94 00             0046        NOP    ;DELAY AGAIN
0C95 CB CF          0047        SET    1,A
0C97 D3 04          0048        OUT    (PORTA),A
0C99 F1             0049        POP    AF
0C9A C9             0050        RET
                    0051 ;
                    0052 ;INITIALISER - CONFIGURES THE PIO PORTS
                    0053 ;AND SETS THE 'U' VECTOR
                    0054 ;
0C9B 3E CF          0055 INIT   LD     A,0CF ;PORT CONFIGURATION
0C9D D3 06          0056        OUT    (CONPTA),A
0C9F 3E FD          0057        LD     A,0FD ;MAKE ALL PORT A BITS INPL.
0CA1 D3 06          0058        OUT    (CONPTA),A ;EXCEPT 1 (DATA STROBE)
0CA3 3E 02          0059        LD     A,2 ;SET STOBE LINE HIGH
0CA5 D3 04          0060        OUT    (PORTA),A
0CA7 3E 0F          0061        LD     A,00F
0CA9 D3 07          0062        OUT    (CONPTB),A
0CAB 21 80 0C       0063        LD     HL,CENTP ;SET THE 'U' VECTOR
0CAE 22 78 0C       0064        LD     (00C78),HL
0CB1 DF 5B          0065        SCAL   05B ;RETURN TO NAS-SYS
```



64 KILOBYTE RAM
and
BUFFER CARD
with PROGRAMMABLE GRAPHICS

This 64K RAM card is suitable for the Nascom 1 or 2. The double sided glass-fibre P.C.B., 302 mm (12 ins.) by 203 mm (8 ins.), holds up to 4 blocks of 16 Kb dynamic RAM (4116). When all four blocks are fitted the whole of the Z80 address field is occupied by RAM. The on board mapper allows parts of this address field to be selectively inhibited in either read or write mode, or both. The mapper divides the address field into 4K blocks, and any two selected blocks can be further subdivided into 2 x 2K blocks.

The graphics section is entirely separate from the dynamic RAM, but it can be mapped in at any chosen 2K boundary. It can use an EPROM (2716) to give a pre-programmed character set, or static RAM (2 x 4118, or 6116) to provide user-programmable characters.

For the Nascom 2 the memory and graphics section can be separated from the "buffer" section; the resulting 8 x 8 card can be plugged into a standard Nasbus (80-bus) edge connector For the Nascom 1 the bottom 8 x 4 ins. section of the card provides full buffering between the Nascom 1 43-way connector and Nasbus. In addition the following extra facilities are also provided:-

1. Power-on jump; this allows the processor to execute a program at any preset 4K boundary on power-on or reset.
2. Synchronised Reset; the reset pulse is synchronised with the processor M1 cycles, to prevent corruption of data in dynamic RAM
3. Wait state generator; one wait state can be added to memory or input/output access
4. ROM socket; a 28 pin or 24 pin socket can be placed at position B3, and via a series of links this can accommodate a 2716, 2732, 2764 or the standard Nascom Basic ROM
5. Input/output; a partial decode is provided which allows for 64 input/output addresses.

The 64K RAM card is available now, price £39.50, from

MICRO POWER Ltd.,
8/8A, Regent Street, Leeds LS7 4PE
Tel. (0532) 683186
Please add 55p p/p and V.A.T. at 15%.