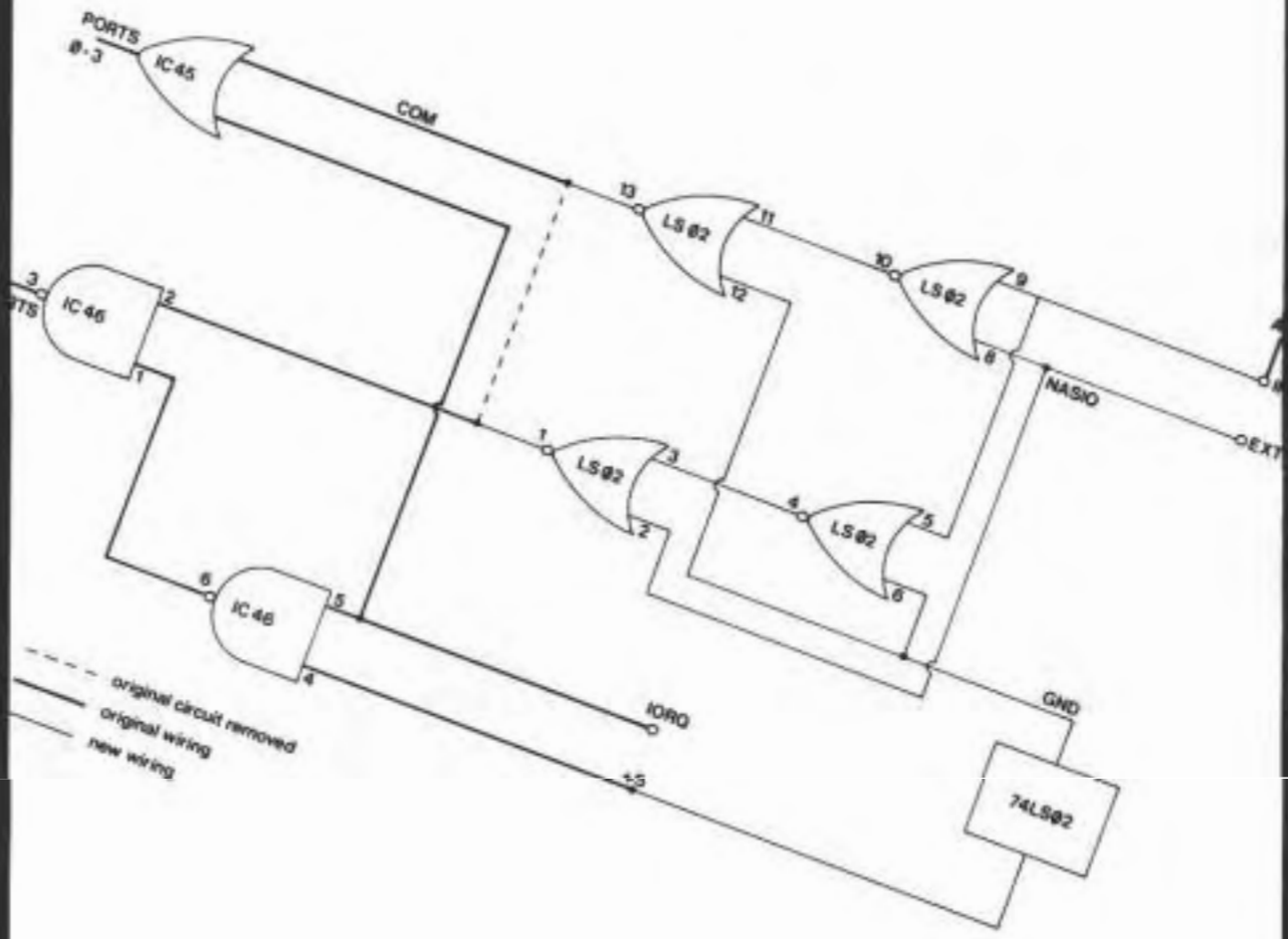
nascom

NEWSLETTER

Volume 3, No. 3
May 1983
£1.25



Lucas Nascom

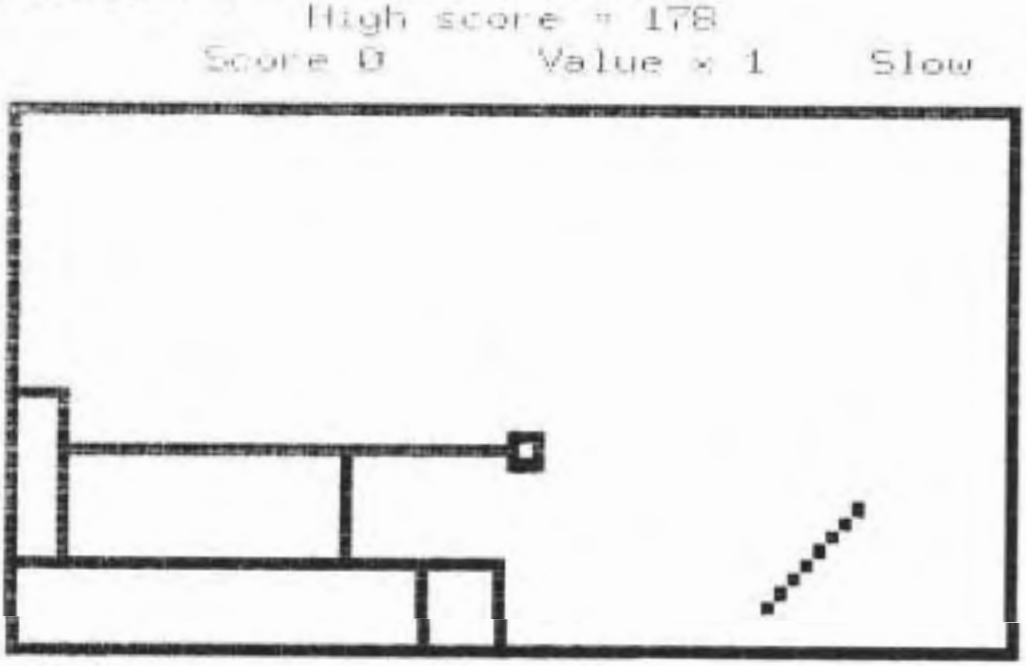
NEW PROGRAMS
from
MICRO POWER

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

Please add 55p for postage & packing and 15% VAT.

NASCOM NEWSLETTER Vol 3, No 3 August, 1983

Contents

Editor - Ian J Clemmett

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

Printed by - Dataform Press

Editorial

I bet you all thought that I'd got lost forever this time. My apologies for the late appearance of this issue but it would have been printed ages ago if a certain company (who shall remain nameless) had sent their advert in on time. Never mind, the magazine is here again.

One mistake from last time, in the COMPASS assembler article there were a couple of errors and they should have been as follows:

```
004B FEE2 100B  JR NEWOPS5A
004C FEE4 3ED9 NEWOPS5 LD A,JR ; pseudo RCAL
004D FEE6 F5   NEWOPS5A PUSH AF ;save character
```

In the next issue I hope to have a review of the HSA-88B Arithmetic Processor board and also an article on DIY lightpens. Until then, ...

IJC

WANTED - FD250 Pertec disc drive. Good condition. Also Nascom 1 I/O board.
Ring 0633-440667

FOR SALE - Nascom 2, cased with 64K RAM A board. No waits. £270
Tel. 01-398-1948

FOR SALE - IMP printer with IMPRINT £140 o.n.o.
12" Green Screen Monitor. £50
Tel. 0209-860480

SYS-EX - A 1K extension program for NAS-SYS monitors - Part 3

by David G. Johnson

This article describes the second half of the commands available under SYS-EX, following on from the previous article.

q

Query tape contents

Read a cassette tape and display on the screen any tape labels as written by the 'l' command, and any file names as written by the 'w' command or by the 8K Basic.

Each file name read is displayed on the next available screen line and, if necessary, the screen lines are scrolled in the normal way. The command is terminated by first stopping the cassette tape recorder and then typing four ESC (SHIFT:ENTER) characters. Four consecutive ESC characters read from the cassette tape input will have the same effect.

This command is one of the four SYS-EX comands that display tape labels (the others are 'r', 's' and 'v'). Tape labels are always displayed within the first 42 character positions of the top line (line 16). Labels that are shorter than the 42 character maximum length are space filled up to the maximum length. Therefore, displayed tape labels overwrite the whole of the top line with the exception of the final six character positions, which are always left unaltered.

r AAAA
--- required file name ---

Read named file

Read a cassette tape and search for a file name which matches with the required file name. When a match is found, read the file into memory using the NAS-SYS Read command.

After entry of the 'r' command, the prompt 'rName:' appears on the next screen line and the required file name is entered on that line. Optionally, 'transparent' characters may be entered as part of the required file name. A transparent character will match with any character read into that position. The transparent character is ASCII 7FH (CONTROL:SHIFT:/).

The tape drive LED is switched on and the cassette tape is searched for tape and file labels. Any tape labels detected are displayed on the top screen line without interruption of the command. As each file label is detected, the file name is displayed on the screen line immediately below the required file name. Any file name or text which was previously displayed on that line is blanked

out.

File names are compared. A required name will match with an identical name read from cassette tape. Also, a required name will match with a longer name read from cassette tape, providing that all the characters match up to and including the last character (i.e. last non-space) of the required name. If a file name read from cassette tape fails to match the required name, the search continues. When a match is found, the NAS-SYS Read command is called to read the file into memory.

The command is terminated after the conclusion of the NAS-SYS Read following a successful file name match. Between entry of the required file name and a successful file name match, the command may be terminated by first stopping the tape and then typing four ESC (SHIFT:ENTER) characters. Four consecutive ESC characters read from the cassette tape input will have the same effect.

AAAA - value to be added to the memory address read from cassette tape. (NAS-SYS 3 only)

This command, along with the 'w' command, was one of the earliest routines to be written. The rest of SYS-EX largely grew from here. It is worth mentioning that any file which can be read by the 'r' command can also be read by the standard NAS SYS 'R' command. Due to the absence of a file name, the reverse of course, is not true.

s

Search for a tape label

Read a cassette tape and search for a tape label as written by the 'l' command. Display the label on the top line of the screen then terminate the command.

The tape drive LED is switched on between entry of the command and command termination. If a tape label is not found, the command may be terminated by first stopping the cassette tape recorder and then typing four ESC (SHIFT:ENTER) characters. Four consecutive ESC characters read from the cassette tape input will have the same effect.

The command may be used for positioning a cassette tape prior to writing a file.

t AAAA
--- text for top line ---

Write to top line of screen

Copies 48 characters to the top line (line 16) of the screen. There are two alternate formats of the command as follows:

1. Copy a line of text from a screen line (1 to 15) to the top line of the screen. The command is entered without any arguments. (i.e. the t should be the only character on the line.) The line which is entered following the command line, is duplicated on the top line of the screen.

2. Copy 48 characters from a specified memory address to the top line of the screen. The command is entered with at least one argument. The first argument entered is the memory address from which 48 characters are copied to the top line of the screen.
AAAA - memory address from which 48 characters are copied.

Using the top line has always been a bit of a problem. With the 't' command the top line can easily be used to hold titles or any other text which is best not scrolled.

u A BBBB

Set user routine address for the 'x', 'y' or 'z' command

Sets up a specified value in one of the NAS-SYS arguments ARG8, ARG9 or ARG10.

The SYS-EX 'x', 'y' and 'z' commands use ARG8, ARG9 and ARG10 respectively, to determine the memory address at which the required user routine is to be called. The 'u' command provides an easy method of setting a value in ARG8, ARG9 or ARG10.

The first argument determines for which command the memory address is being set. '1' sets the memory address for the 'x' command (ARG8). '2' sets the memory address for the 'y' command (ARG9). '3' sets the memory address for the 'z' command (ARG10). The second argument is the value to be set up for the appropriate command.
A - 1, 2 or 3 only. Selects 'x', 'y' or 'z'.
BBBB - memory address at which the user routine will be called.

This is just an easy method of setting up the last three NAS-SYS arguments independently and without having to alter or enter the values for the earlier arguments. The values which are set in the NAS-SYS arguments can be examined at any time by use of the SYS-EX 'a' command.

v
--- required file name ---

Verify named file

This command is identical to the SYS-EX 'r' command with two exceptions. The first exception is that the prompt for entry of the required file name is 'vName:'. The second exception is that when a match between the required file name and a file name read from cassette tape occurs, the NAS-SYS Verify command is called. Therefore, the file is not loaded into memory.

This command is essential to check that a file written by the SYS-EX 'w' command can be successfully read.

w AAAA BBBB
--- file name to be written ---

Write named file

Write a file to cassette tape preceded by a file label containing the specified file name. The file label will be recognised and displayed by the SYS-EX commands 'q', 'r' and 'v'.

As the command uses the NAS-SYS Write command to write the file itself, the format is compatible with files written using the NAS-SYS 'W' command. If required, the NAS-SYS 'R' and 'V' commands may be used to read and verify files written by the SYS-EX 'w' command, although the file label will be ignored by NAS-SYS. The content of memory from address AAAA up to but not including BBBB, is the file which is written to the cassette tape.

After entry of the 'w' command, the prompt 'wName:' appears on the next screen line and the name of the file to be written is entered on that line. The tape drive LED is then switched on and after about two seconds (4MHz), the file label followed by the file itself is sent to the serial output port for writing to tape. Finally, the tape drive LED is switched off.

The format of the file label is as follows: 3 x D3H, 0 to 42 characters of file name, 1 space (00H if name length is 42), 00H.

AAAA - first memory address to be written to tape.

BBBB - one beyond the last memory address to be written to tape.

Many of the comments for the related SYS-EX commands ('l', 'q', 'r', 's', 'v') are of use in understanding the operation and usage of the 'w' command.

A file written using the SYS-EX 'w' command will be recognised in the 8K Basic by both the CLOAD command and the SYS-EX Basic named file facility. However, the file is unlikely to be an acceptable Basic program file and problems could therefore result if it was read into memory during the execution of Basic.

x

Call user routine number 1

Calls a user routine whose execution address has been stored in the NAS-SYS argument ARG8.

The execution address may be set either by entering the appropriate number of arguments (8,9 or 10) on a command line, or by using the SYS-EX 'u' command. If the execution address has a value of 0000H, the command is rejected and an error message is displayed.

On entry to the user routine the NAS-SYS stack is in use. Care should therefore be taken to ensure that: (i) higher addresses on the stack are not corrupted, and (ii) the maximum depth of the stack is not exceeded. A return to SYS-EX may be made by executing a Z80 RET instruction. If the carry flag is set upon return to SYS-EX, the message 'Error' is displayed on the screen prior to the acceptance of further input.

Upon entry to the user routine, various Z80 registers have preset values. These values are: HL = value from ARG1, DE = value from ARG2, BC = value from ARG3, SP = 0C5FH within the NAS-SYS stack. Forty two bytes are available on the stack for use by the user routine. Calls to NAS-SYS routines or to SYS-EX routines, from within a user routine, will require a number of stack levels to be available.

y

Call user routine number 2

Calls a user routine whose execution address has been stored in the NAS-SYS argument ARG9.

The command is identical to the 'x'command except that the argument ARG9 is used in place of ARG8.

z

Call user routine number 3

Calls a user routine whose execution address has been stored in the NAS-SYS argument ARG10.

The command is identical to the 'x'command except that the argument ARG10 is used in place of ARG8.

The three user routine calls, 'x', 'y' and 'z', provide an easy method of attatching individually tailored commands to the monitor. Unlike the NAS-SYS Execute command, these commands allow a return to the monitor (SYS-EX) with a single RET instruction. Also, the 'Error' message can be output under user control by manipulating the carry flag prior to the RET. The commands are ideally suited to calling individual monitor type functions. e.g. - Display the character represented by the ASCII code entered as ARG1 -, or - Print screen contents on attatched printer -. It might even be possible to directly call some of the functions within Basic.

In addition to 'x', 'y' and 'z', there is another method of calling user routines. The commands 'g', 'j', 'm', 'o' and 'p', which are not used directly by SYS-EX, all call a memory address 0400H (1K) beyond the start of SYS-EX (i.e. immediately after SYS-EX). If you put your own code here, these commands can be used in a similar way to 'x', 'y' and 'z'. The register values on entry are the same as for 'x', 'y' and 'z'. A word of warning here - if you haven't put any code beyond SYS-EX, don't use 'g', 'j', 'm', 'o' or 'p' as the results will be unpredictable.

That completes the description of the keyboard commands. In the next issue of the magazine we will look at, amongst other things, how the 8K Basic can make use of forty two character program names.

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.C.s.
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 divison 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

Modifications to Wordease 2

by T. Balls

I have been using Wordease 2 for about two years. As a teacher I find it invaluable for the preparation of worksheets, exam papers and the like. When I bought this program it was a simple choice between it and Naspen and although both had their advantages and disadvantages I settled for the RAM based system so that it could be more easily updated. Nothing is perfect however and there were several deficiencies that I have tried to make good over the time I've been using it. This article, together with the assembler listing, describe these changes. The main changes are as follows:

"Current position" when processing.

The "conditional eject" facility is useful but I found that I very often needed to know how many lines there were remaining on a particular page so that I could decide whether to include an extra paragraph or not. Lines 660 - 1210 encode a routine which outputs the current page being processed and how many lines remain on that page. The right hand side of the top screen line is used during the process phase. Simply put a "halt" at the end of the text to find out how much space is left.

Support for Epson MX 80 printer.

For quite some time I used an ancient Olivetti terminal printer that did not respond to "escape" sequences in the way that more modern printers do. When I started to use an Epson I found several problems - I could switch underline on, but not off again! The justification option in the processor was taking note of the printer control codes so that a line with an underlined word would be justified to a right margin 6 columns in.

The other problem arose when processing printer control codes on the screen via Port1 - the escape sequences tended to mess up the screen display. Lines 1700 - 1930 trap all control-U codes and increment a second line-length counter which is subsequently used for the justification process. There are several patches which are needed to reinstate the correct value at the start of each new line. Lines 3250 - 3420 and 1270 - 1370 trap the output via Port1 and Port2 and modify being output. Output to Port1 (screen) has a table of control codes (<20H) and any of these have Bit 7 set so they are output as graphics characters. In order to send Null codes (00H) to the printer to turn off certain features it is necessary to send 255 instead. The Port2 patch detects these codes and sends 00. The problem is the Wordease uses 00h as the end of line marker in its text buffer. Both these Port patches also detect the "pad" character (see below).

Pad characters.

It is sometimes necessary to be able to put in "hard spaces"

which will not be changed by the justification process. This is especially useful when laying out tables. The "linefeed" key will put a = on the screen but both of the output ports detect this character and change it two a space character.

Additional embedded formatting commands.

Three additional formatting commands have been added to provide centring of lines and automatic paragraph numbering. Lines 1950- 3050 of the source listing code for these additions.

.M when used at the end of a line, in place of .N will cause that line to be centred within the currently set margins. i.e. it will assume that leading spaces, inserted by .T or .I commands, are a part of the text to be included in the centring. .M will be treated like .N by the Adjust command.

.A will insert a right-justified, two-digit number. It will automatically increment each time it is used. The value starts from "1" and it can be reset at any time by using .A0.

.B is similar to .A except that it inserts lower case letters "a" - "z". Again it will automatically increment and it can be reset to "a" with .B0.

The remainder of the source listing deals with the many patches to the original program which are needed in order to implement the changes. The easiest way in which to effect the changes will be to use ROM ZEAP if you have it. Power up ZEAP with its buffer located above 3000H and type in the object code. Load your original Wordease 2 and assemble to memory.

Note that I have a modified NAS-SYS so that my parallel printer output goes via XOUT and that 07H (ctrl-G) produces a "beep" so you might need to alter these locations to NOP.

```
0010 ;   PATCHES  TO  WORDEASE 2
0020 ;
0030 ; SUPPORT FOR EPSON 80 VIA XOUT ROUTINE
0040 ; IN MODIFIED OPERATING SYSTEM [6].
0050 ; CTRL/U HAS BEEN MODIFIED SO THAT EMBEDDED
0060 ; ESCAPE SEQUENCES ARE JUSTIFIED CORRECTL Y
0070 ;
0080 ; PORT 1 VECTOR HAS BEEN INTERCEPTED AND
0090 ; CONTROL CDES HAVE BIT 7 SET.
0100 ;
0110 ; 0AH (LF) IS PROCESSED AS 20H (ie = PAD)
0120 ;
0130 ; CTRL/M USED AT THE END OF A LINE WILL CAUSE
0140 ; THAT LINE TO BE CENTRED.
0150 ;
0160 ; CTRL/A INSERTS 2-DIGIT, AUTO PARAGRAPH
0170 ; NUMBERING. CTRL/A0 RESETS COUNT TO 1.
0180 ;
```

```
0190 ; CTRL/B INSERTS AUTO PARAGRAPH LETTERING
0200 ; USING THE LOWER CASE LETTERS  a - z.
0210 ; CTRL/BO RESETS TO "a".
0220 ;
0230 ;
0240 ; LABELS:
0250         ORG    0
0260 BUFSTR  EQU    1033H
0270 OLDBUF  EQU    2335H
0280 PORT1   EQU    1003H
0290 PORT2   EQU    1007H
0300 PORT20  EQU    100AH
0310 KEYBRD  EQU    100FH
0320 ZCRT    EQU    65H
0330 WORD    EQU    1000H
0340 LNVAL   EQU    106BH
0350 OLDDEF  EQU    1E81H
0360 PTXO    EQU    1F0AH
0370 OUTPUT  EQU    2065H
0380 CMDPTR  EQU    104AH
0390 OBFPTR  EQU    1048H
0400 GETDEC  EQU    21F5H
0410 BLPVCT  EQU    1021H
0420 ERROR   EQU    21B8H
0430 STARTW  EQU    1376H
0440 EXIT    EQU    101BH
0450 VERNUM  EQU    11F2H
0460 LCOUNT  EQU    1055H
0470 PCOUNT  EQU    1054H
0480 ;
0490 ; SET UP X1 AND DOUBLE KEYBOARD REPEAT
0500 ;
0510 STWORD  ORG    WORD
0520         JP     INIT
0530 ;
0540         ORG    OLDBUF
0550 INIT    LD     L,1
0560         SCAL   "X
0570         LD     HL,28H
0580         LD     (0C30H),HL
0590         RST    28H
0600         DEFB   0CH,0          ;CLS
0610         JP     STARTW
0620 ;
0630 ;OUTPUT CURRENT PAGE AND LINES REMAINING
0640 ;DURING PROCESSING OF TEXT.
0650 ;
0660 POSION  LD     A,(LCOUNT)
0670         CALL   GETVAL
0680         LD     (LINENO),HL
0690         LD     A,(PCOUNT)
0700         CALL   GETVAL
0710         LD     (PAGENO),HL
0720         LD     HL,MSSGE
0730         LD     DE,0BE4H      ;TOP OF SCREEN
```

```
0740          LD    BC,15H
0750          LDIR
0760          POP   HL
0770          POP   IY
0780          POP   IX
0790          RET
0800 GETVAL   LD    HL,0          ;CONVERT HEX TO
0810          PUSH  AF            ;ASCII IN RANGE
0820          CALL  HINIBL        ;0 - 99.
0830          CP    0             ;CALL WITH ACCUM
0840          JR    NZ,L1         ;CONTAINING HEX
0850          SET   7,L           ;RETURNS WITH HL
0860 L1       LD    B,A           ;CONTAINING THE
0870          POP   AF            ;TWO ASCII DIGIT S
0880          AND   0FH           ;L HOLDS MSD
0890          ADD   A,0           ;H HOLDS LSD
0900          DAA
0910          BIT   7,L
0920          JR    NZ,L2
0930 LOOP1    INC   H
0940          ADD   A,6
0950          DAA
0960          DJNZ  LOOP1
0970 L2       PUSH  AF
0980          CALL  HINIBL
0990          ADD   A,H
1000          CP    0
1010          JP    NZ,NONZER     ;SUPPRESS
1020          LD    A,20H         ;LEADING ZEROES
1030          JR    LOADL
1040 NONZER   ADD   A,30H
1050 LOADL    LD    L,A
1060          POP   AF
1070          AND   0FH
1080          ADD   A,30H
1090          LD,   H,A
1100          RET
1110 HINIBL   AND   0F0H
1120          SRA   A
1130          SRA   A
1140          SRA   A
1150          SRA   A
1160          AND   0FH
1170          RET
1180 MSSGE    DEFM  "Page:"
1190 PAGENO   DEFS  2
1200          DEFM  " Lines free:"
1210 LINENO   DEFS  2
1220 ;
1230 ; INTERCEPT PRINTER PORT (2) TO DETECT PA D
1240 ; CHARACTER AND 0FFH CODES SENT IN PLACE
1250 ; OF NULL TO EPSON MX 80.
1260 ;
1270 PRINT    PUSH  AF
1280          CP    0AH           ;PAD CHARACTER
```

```
1290          JP     NZ,NOTPAD
1300          LD     A,20H
1310 NOTPAD   CP     0FFH          ;"NULLS"
1320          JP     NZ,SKIP
1330          INC    A
1340 SKIP     SCAL   6EH           ;PRINTER
1350          DEFB   0             ;ROUTINE
1360          POP    AF
1370          RET
1380 ;
1390 ; BLEEP - "NAS-SYS 6" ONLY
1400 ;
1410 BLEEP    RST    28H
1420          DEFB   7,0
1430          RET
1440 ;
1450 ; TEMP STORES
1460 ;
1470 OWNLNL   DEFS   1
1480 CURRNO   DEFS   1
1490 CURLET   DEFS   1
1500 ;
1510 ; SETUP NEW STORES TO DEFAULT VALUES AT
1520 ; THE START OF PROCESSING.
1530 ;
1540 DEFALT   LD     (LNVAL),A
1550          LD     (OWNLNL),A
1560          XOR    A
1570          LD     (CURRNO),A
1580          LD     A,"a
1590          LD     (CURLET),A
1600          JP     OLDDEF
1610 ;
1620 ; CHANGE LINELENGTH COUNTER ON CTRL/L
1630 ;
1640 NEWLNL   LD     (OWNLNL),A
1650          JP     PTXO
1660 ;
1670 ; NEW CTRL EMBEDDED COMMANDS
1680 ; CTRL/U, CTRL/A, CTRL/B, CTRL/AO  AND CTRL/BO
1690 ;
1700 PROSSU   CP     "U
1710          JP     NZ,LETTER
1720          CALL   GETDEC
1730          LD     A,B
1740          PUSH   AF
1750          LD     A,(LNVAL)     ;INCREMENT
1760          INC    A             ;LINELENGTH
1770          LD     (LNVAL),A     ;COUNTER
1780          CP     0FEH          ;CHECK IF LINE
1790          JR     C,CONTIN      ;IS GETTING TOO
1800          LD     DE,126EH      ;AND FLAG
1810          CALL   ERROR         ;LINELENGTH ERRO R
1820 CONTIN   POP    AF
1830          CP     20H
```

```
1840          JP      NZ,OUT
1850          LD      HL,(104AH)
1860          LD      (104CH),HL
1870 OUT      LD      HL,(OBFPTR)
1880          LD      (HL),A
1890          INC     HL
1900          LD      (OBFPTR),HL
1910          LD      HL,(CMDPTR)
1920          LD      A,(HL)
1930          JP      PTXO
1940 ;
1950 LETTER   CP      "B
1960          JP      NZ,CENTRE
1970          CALL    GETDEC
1980          LD      A,B
1990          CP      0
2000          JP      NZ,CONLET
2010          LD      A,"a
2020          LD      (CURLET),A
2030          JP      PTXO
2040 CONLET   LD      A,(CURLET)
2050          PUSH    AF
2060          INC     A
2070          LD      (CURLET),A
2080          CP      7CH             ;CHECK TO SEE IF
2090          JP      C,CONTIN        ;PAST "x"
2100          POP     AF              ;RESET IF SO
2110          LD      A,"a
2120          PUSH    AF
2130          INC     'A
2140          LD      (CURLET),A
2150          JP      CONTIN
2160 ;
2170 CENTRE   CP      "M
2180          JP      NZ,NUMBER
2190          LD      A,(LNVAL)
2200          LD      C,A
2210          DEC     C
2220          LD      B,0
2230          LD      HL,1070H        ;START OF BUFFER
2240          ADD     HL,BC
2250          PUSH    HL
2260          POP     IX
2270          LD      A,20H
2280 FINDSP   CP      (HL)
2290          JR      Z,CENT
2300          LD      (HL),A
2310          DEC     HL
2320          JR      FINDSP
2330 CENT     DEC     HL
2340          CP      (HL)
2350          JR      Z,CENT
2360          PUSH    HL
2370          LD      DE,1070H
2380          OR      A
```

```
2390          SBC   HL,DE
2400          JR    NC,NEXT
2410          LD    DE,1290H     ;CONTROL ERROR
2420          CALL  ERROR
2430 NEXT     POP   HL
2440          PUSH  HL
2450          EX    DE,HL
2460          PUSH  IX
2470          POP   HL
2480          OR    A
2490          SBC   HL,DE
2500          LD    A,L
2510          SRL   A
2520          CP    1
2530          JP    P,KEEPON
2540          LD    DE,129EH     ;JUSTIFICATION
2550          CALL  ERROR        ;ERROR
2560 KEEPON   LD    L,A
2570          ADD   HL,DE
2580          EX    DE,HL
2590          POP   HL
2600          PUSH  DE
2610          PUSH  HL
2620          LD    DE,1070H
2630          OR    A
2640          SBC   HL,DE
2650          PUSH  HL
2660          POP   BC
2670          POP   HL
2680          POP   DE
2690          INC   BC
2700          LDDR
2710          EX    DE,HL
2720          PUSH  HL
2730          LD    DE,1070H
2740          OR    A
2750          SBC   HL,DE
2760          LD    B,L
2770          INC   B
2780          POP   HL
2790          LD    A,20H
2800 LOOP     LD    (HL),A
2810          DEC   HL
2820          DJNZ  LOOP
2830          CALL  2162H        ;PRINT LINE
2840          JP    PTXO
2850 ;
2860 NUMBER   CP    "A
2870          JP    NZ,204EH     ;BACK TO WORDEAS E
2880          CALL  GETDEC
2890          LD    A,B
2900          CP    0            ;CTRL/AO = RESET
2910          JR    NZ,CONT
2920          XOR   A
2930          LD    (CURRNO),A
```

```
2940          JP     PTXO
2950 CONT     LD     A,(CURRNO)   ;RESET TO 0 WHEN
2960          CP     99           ;99 IS REACHED
2970          JR     NZ,OUTNUM
2980          LD     A,0FFH
2990 OUTNUM   INC    A
3000          LD     (CURRNO),A   ;UNITS" PART OF
3010          LD     BC,0AH       ;CTRL/X ROUTINE IN
3020          CALL   2228H        ;WORDEASE
3030          JP     PTXO
3040 ;
3050 ; PATCH END OF LINE PROCESSING AND ADJUST
3060 ; ROUTINES TO ALLOW FOR CHANGES
3070 ;
3080 ENDPRS   PUSH   AF
3090          LD     A,(OWNLNL)
3100          LD     (LNVAL),A
3110          POP    AF
3120          JP     PTXO
3130 ;
3140 ADJUST   CP     "E
3150          JP     Z,1BFBH
3160          CP     "M
3170          JP     Z,1BFBH
3180          JP     1BBAH
3190 ;
3200 ; NEW SCREEN OUTPUT - TRAPS AND MODIFIES
3210 ; CONTROL CODES TO PREVENT VDU CORRUPTION
3220 ;
3230 NEWP1    PUSH   AF
3240          PUSH   HL
3250          PUSH   BC
3260          CP     0AH          ;PAD CHARACTER
3270          JR     NZ,CHKCTL
3280          LD     A,20H
3290 CHKCTL   LD     HL,LOOKUP
3300          LD     BC,TABEND-LOOKUP
3310          CPIR
3320          JP     NZ,VIDEO
3330          OR     80H
3340 VIDEO    SCAL   ZCRT
3350          POP    BC
3360          POP    HL
3370          POP    AF
3380          RET
3390 ;TABLE OF CODES TO BE ALTERED
3400 LOOKUP   DEFB   7,8,9,11,12,14
3410          DEFB   15,17,18,20,27
3420 TABEND   DEFB   0
3430 ;
3440 ; DEFINE NEW BUFFER
3450 ;
3460 NEWBUF   DEFW   BUFEND
3470          DEFB   0FFH
3480 BUFEND   DEFB   0FFH
```

```
3490 ;
3500 ;
3510 ; ***** PATCHES INTO ORIGINAL *****
3520 ;
3530        ORG   PORT1
3540        JP    NEWP1
3550 ;
3560        ORG   PORT2
3570        JP    PRINT
3580        DEFB  0
3590 ;
3600        ORG   BLPVCT
3610        JP    BLEEP
3620 ;
3630        ORG   EXIT
3640        RST   28H
3650        DEFB  0CH,0
3660        JP    0
3670 ;
3680        ORG   KEYBRD
3690        SCAL  7DH
3700 ;
3710        ORG   11E3H
3720        JP    POSION
3730 ;
3740        ORG   1E7EH
3750        JP    DEFALT
3760 ;
3770        ORG   1F8EH
3780        JP    ENDPRS
3790 ;
3800        ORG   1FDDH
3810        JP    ENDPRS
3820 ;
3830        ORG   1FEEH
3840        JP    ENDPRS
3850 ;
3860        ORG   2011H
3870        JP    NEWLNL
3880 ;
3890        ORG   2044H
3900        JP    PROSSU
3910 ;
3920        ORG   20B6H
3930        JP    ENDPRS
3940 ;
3950        ORG   1BB6H
3960        JP    ADJUST
3970 ;
3980        ORG   BUFSTR
3990        DEFW  NEWBUF
4000 ;
4010 ; PATCH TO PREVENT THE SAVING OF CTRL/L VALUE S
4020 ; IN MACRO CALLS.  (CTRL/I UNCHANGED)
4030 ;
```

```
4040            ORG     2128H
4050            DEFB    0,0,0,0
4060            LD      HL,(106CH)
4070            PUSH    HL
4080            CALL    PTXO
4090            POP     HL
4100            LD      (106CH),HL
4110            DEFB    0,0,0,0
4120 ;
4130 ; CHANGE START MESSAGE
4140 ;
4150            ORG     1200H
4160            DEFM    "  Requires NAS-SYS 6 and"
4170            DEFM    " EPSON MX 80"
4180            DEFB    80H
4190 ;
4200            ORG     VERNUM
4210            DEFM    "4T"
```

A REVIEW OF WAIT - GATE

by M.S. Smith

"What is WAIT - GATE" ? I hear you all ask. Well, you should have a good idea if you have studied memory " WAIT - STATES " in general, or in particualar, for Nascoms. For those of you who have not done so, I will now try and explain.

The Nascom 2/3 has facilities for driving RAMs and ROMs of varying access time. This entails providing WAIT - STATES by the use of the Z80 control line WAIT. The effect of WAIT is to give extended memory access cycles, allowing the use of slower memory devices. On the Nascom with WAIT enabled, this is done to every memory access cycle, ROM, or RAM, of the CPU.

Now for the important bit - what the WAIT - GATE can and cannot do for you, and how all this is achieved. With the WAIT - GATE the following are provided; an explanatory booklet / user manual, and benchmark test programs, the results of which will follow later. The first section of the manual explains how the device works, and what it will provide. It then goes on to describe the fitting, testing, and operation, finally concluding with a section on how to use it to best advantage. The clearly written manual and neat, well designed WAIT - GATE are easy to understand , and almost foolproof. Anyway, I didn't find any information lacking or failure in the device.

Basically, the WAIT - GATE works by selecting how long a WAIT is necessary for both RAMs and ROMs. That is, if your memory devices do not require the full amount of WAIT, (ie. less than 525ns access time), which for RAM is usually the case these days. Or, if your RAM and ROM work at different speeds, the gate can be hardware - selected to vary the length of WAIT for each. Not every memory device needs either no WAIT or full WAIT, but for 300 - 400ns devices, a WAIT on M1 cycles will suffice.

Fitting the device requires the removal of two IC's (8 & 18) from the Nascom main board. The WAIT - GATE has two wire - wrap sockets for insertion into the now vacant sockets, in a "piggy back" arrangement. The means that a gap above IC8 and IC18 of a sockets' width should be available. The usual set up for Nascoms that I have met, is for the main board to be uppermost, so this should be no problem at all. Once in place testing can begin.

The WAIT - GATE has a "four DIP" switch and associated selector to choose which WAIT states are implemented. The RAM and ROM sections can be individually selected for either M1 and R/W (full) or M1 only; or for neither WAIT, as required. This gives all the combinations of WAIT.

Testing is quite simple, I found. It's a matter of fitting the WAIT - GATE with all the WAITs (M1 and R/W) in operation, giving a normal and average cursor flash rate. This is followed by setting to no WAITs, which gives a noticeable change of cursor rate. A further test of M1 WAITs only, gives an increase in cursor rate, but less noticeable. (In almost all systems the NAS-SYS ROM and workspace RAM will work without WAITs). Any poor connections of the wire wrap sockets will show up as faults in other sections of the Nascom circuit. These include the RESET switch, single step logic, PIO, and serial interface. The manual explains this, and these functions should be watched during first operations.

TABLE 1. SWITCH DATA

SWITCH No.	BLOCK	CYCLE TYPE
1	A (ROM)	M 1
2	A (ROM)	R/W
3	B (RAM)	M 1
4	B (RAM)	R/W

TABLE 2. MEMORY DEVICE TIMING

WAIT STATE TYPE	MEMORY ACCESS TIME
NONE	<275ns
M1 ONLY	<400ns
FULL (M1 & R/W)	>525ns

Using the above tables, the latter section of the manual explains how the switches in Table 1 are used to provide each ROM and RAM speed in Table 2. Switches 1 and 2 are set for ROM: 3 and 4 for RAM. If an access time of < 400ns and > 275ns is required for ROM and < 275ns for RAM, the switches would be set:

ROM		RAM	
1 UP	2 DOWN	3 DOWN	4 DOWN
M1	NO R/W	NO M1	NO R/W

The main idea behind the WAIT - GATE, says the manual, is:

"In most systems only some memory access cycles actually need WAIT states. The WAIT - GATE circuit provides a flexible means of controlling WAIT states so that the number of unnecessary WAIT states is minimised. This can result in a considerable increase in C.P.U. speed....."

This I must agree with, when you consider that by far the most frequently used operations are the reading and writing of the memory, both ROM and RAM.

The least useful operation of the WAIT - GATE is with both ROM and RAM of either < 275ns or > 525ns access time. The most useful arrangement is with ROM of < 400ns and RAM of < 275ns. To give results, for the test programs given in the manual, over the full range of ROM and RAM set ups would take up a lot of space and reading time. So I will give results for only one arrangement, which I think is the most frequently used today. ie. for 500ns (< 525ns) ROM & EPROM and 250ns (< 275ns) RAM.

PROGRAM	N2 WAIT(S)	WAIT - GATE
ROM BASIC	24.8 s	22.9 s
PASCAL RUNTIME (BLS)	22.5 s	20.7 s
PASCAL COMPILATION (BLS)	5.1 s	4.2 s
MACHINE CODE (IN RAM)	52.7 s	44.0 s
ZEAP TAPE ASSEMBLY	22.1 s	19.9 s

To summerise. I found the WAIT - GATE a very useful time saving device. It had only one noticable side effect which was increased programming speed. (which can't be too bad!) Finally, the most remarkable aspect for me, was the price Phildata are asking for this piece of hardware, just £ 7.95 all inclusive. I should like to thank Mr. D. Johnson of Phildata for providing the WAIT - GATE for review, and assisting with my questions.

Discs and Ports on a Nascom 1

by C. Godfrey

When installing the Nascom F.D.C. card or any assembly using an off-board P.I.O. on a Nascom 1, you will find that the on-board P.I.O. will no longer function. This is because the decoding for selecting ports 0 to 3 (keyboard and UART) from ports 4 to 7 (P.I.O.) relied on the I/O select link being wired to 'INT' which is actually a2.

Once the I/O select is moved to 'EXT', which is actually NasIO, the address line a2 is disconnected and the port decoding is incomplete, causing any reference to ports 4 to 7 to be directed to ports 0 to 3 respectively.

If the P.I.O. was in use prior to the addition of discs then the only advice that Nascom can offer is that you buy a P.I.O. board, not a very economic method of sorting out this simple problem. There follows a method that will solve the port decoding problem for approx. 20p plus labour.

Parts Required.

1 x 74LS02 Quad NOR Gate
3 x lengths of wire 30cm long

Documentation.

Nascom 1 circuit sheet 1

Method.

1. Locate IC 46 (74LS00)
2. Remove, bend out pin 2, replace.
3. Take 74LS02, bend sideways all pins except pins 1,7,14.
4. Solder 74LS02 on top of 74LS00 using pins 7 and 14.
5. Bend pin 1 of 74LS02 to join pin 2 of 74LS00.
6. Solder these two pins.
7. Remove link on I/O select completely.
8. Connections to 74LS02 are as follows:
   pin 10 to pin 11
   pin 3 to pin 4
   pin 2 to pin 8 to I/O select 'EXT'
   pin 5 to pin 9 to I/O select 'INT'
   pin 13 to I/O select 'COM'
   Ground from pin 7 to pin 6 and pin 12

Once the wiring changes have been made, all that needs to be done is to reassemble and test.

Testing.

Using the NAS-SYS commands 'O' and 'Q', type the following:

1. O 6 0F Initialise port A as an output.
2. O 7 0F Initialise port B as an output.
3. O 4 55 Send 55H to port A.

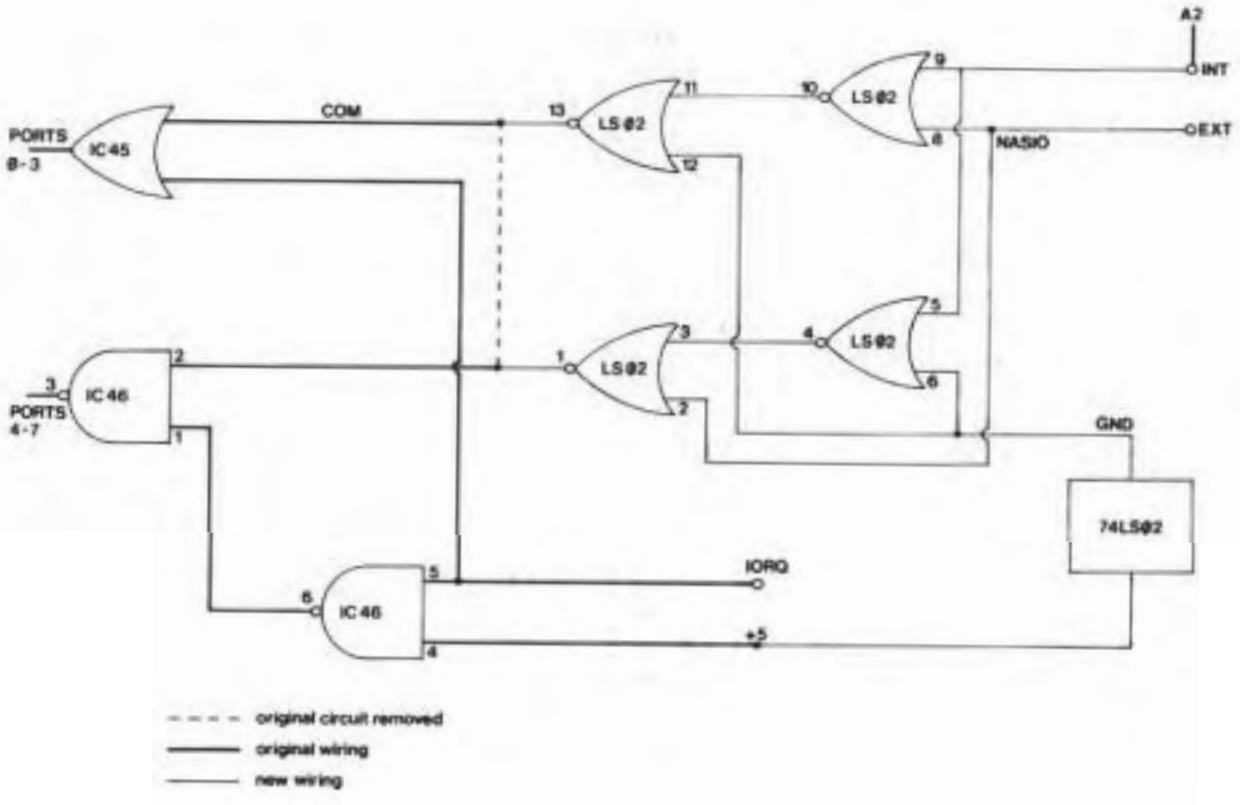
```
4. O 5 AA  Send AAH to port B.
5. Q 4     This should return 55H.
6. Q 5     This should return AAH.
```

If a Graphics expansion board is fitted, care must be taken to ensure that the added IC does not short on the underside of the expansion board as it is mounted directly beneath.

The signal arriving from the F.D.C. card is a decode for ports 0 through to F but as there is no decoding involving a3 it means that there is only true decoding for ports 0 to 7.

Although not tested, it can be assumed that this modification will also work when used with a P.I.O. board, allowing the original P.I.O. to be used as well as the newly added ones.

Should you for any reason no longer require this modification, all that is needed is a new 74LS00 plugging in and the I/O link replacing.



Top specification. High quality. Reliable. Outstanding value.

SPECIAL OFFER!

£225 +VAT

Whilst stocks last!
Cash with order

The new Lucas LX80 printer

Here's a new dot matrix printer that you will want to see and try – if only to confirm that what we claim for it is true.

The LX80 is ideally suited to educational, scientific and personal applications in conjunction with the Lucas Nascom 3 micro, and similar machines.

Phone your nearest Lucas Logic stockist listed below – or if in difficulty phone us on 0926 59411.

We're adding to our dealer list, daily.

FEATURES:

● 80 characters per line, 10 characters per inch ● 80 characters per second bi-directional printing ● Dot matrix print using 7 x 8 dots in an 8 x 9 matrix ● 228 ASCII character set includes normal and italic scripts ● Full graphics facilities – semigraphic characters and 640 bit-addressable dots per line, or compressed 1280 dots per line (7.5 inches) ● Software selection of print – normal, compressed (142 characters per line), double width (40 characters per line) and compressed double (71 characters per line) ● Subscripts, superscripts, underlining, emphasised and double print software selectable ● Vertical and horizontal tab control, including variable line spacing, automatic form feed to preset length and perforation skip ● Backed by Lucas

Lucas Logic

Microcomputers for education, science and business.

Lucas Logic Limited, Welton Road, Wedgnock Industrial Estate, Warwick CV34 5PZ

LUCAS LOGIC STOCKISTS – **Aberdeen**: MicroComms, 0224 633385. **Amersham**: Amersham Computer Centre, 02403 22307. **Bedford**: Kempston News, 0234 857601. **Bristol**: Target Electronics, 0272 421196. **Cardiff**: Llandaff Radio & Television, 0222 563760; Steve Computer Services, 0222 41905. **Crewe**: Mid-Shire's Computer Centre, 0270 211086. **Egham**: Electrovalue, 0784 33603. **Huntingdon**: JPS, 0487 840710. **Ipswich**: M.D.W. Electronics, 0473 78295. **Kenilworth**: Business & Leisure Microcomputers, 0926 512127. **Leeds**: Leeds Computer Centre, 0532 458877. **London**: Henry's Radio, 01-724 3564; Off Records, 01-223 7730. **Manchester**: E.V. Computing, 061 431 4866. **Newcastle-under-Lyme**: Micro-Print, 0782 616481. **N. Ireland**: Newburn Electronics, 09603 78330. **Norwich**: Anglia Computer Centre, 0603 29652. **Nottingham**: Skytronics, 0602 781742. **Plymouth**: S. R. Brewster, 0752 665011. **Poole**: Parkstone Electronics, 0202 746555. **Stroud**: Zeta Computers, 045382 2444. **Torquay**: Crystal Computers & Components, 0803 22699. **Watford**: Computer Centre, 0923 50123; SRS Microsystems, 0923 26602. **Witham**: Selven Systems, 0376 519413.

NEWS FROM NASCOM

This month's news is being written in the middle of the first really good weather of the year - so it's a bit shorter than it would have been otherwise!

On the new product front we have news of the LX80 printer - a low cost 80 character per line, 80 characters per second printer for use with the parallel interface.

The specification of the NAS-CAD computer aided drawing package has grown considerably since our last report, so although a brief outline of the features is given this month you will need to wait for next month's newsletter for full details.

We have more information on using disc files under NAS-DOS who find the manual rather terse (it is!). This will be continued in the next Nascom News when we will also give more information on the operation of the MANOR database manager, which makes extensive use of data files.

The addition of soft function keys at zero cost has been held over to next month - no problems in that we already use it ourselves, but there has not been time to prepare an article for this newsletter.

Mike Hessey

1. APPEAL FOR INFORMATION AND SOFTWARE

Nascom users are exceptionally ingenious, but unfortunately they are also very reticent about what they use their machines for. We are often asked by potential customers 'Has anyone else used a Nascom to.....'. Although we often believe that it has been done we don't have a specific contact. It would help us a lot to be able to compile a directory of users/ applications/ software, and by publishing this we could put you in touch with other users with similar interests, and perhaps save people re-inventing the wheel by writing software which already exists. It could help you make money too - as we may be interested in buying some software ourselves, or contacts you make may be willing to pay, or swap other software.

If you want to take part in this scheme send us a note of your application, hardware configuration and software and how you can be contacted. We would be particularly interested to hear about educational software and NAS-DOS utility programs. The latter could be put into a library, or added to the existing utilities disc.

If you have a particularly interesting application you might

like to consider writing an article on it for inclusion in a future issue to Nascom News.

2 THE LX80 PRINTER

The LX80 is a low-cost 80 character per second printer. It is normally connected to the PIO of any Nascom using suitable cables (extra), and the software described for parallel printers in the last newsletter can be used to control it. A serial version is also available. NAS-SYS/NAS-DOS users should note that we normally supply the printer with the internal switch 2-3 set for no auto-line feed after a carriage return, so you wil need either to set this switch for auto-line feed or add a check in the software drive to output a linefeed (0A ASCII) after each carriage return (0D ASCII). The printer control features associated with the keyboard functions described next month allow for either auto or non-auto linefeed, and a new copy of UTS under NAS-DOS also covers this situation. A modified general purpose parallel printer driver which generates the line feed after a carriage return is listed at the end of this edition of Nascom News.

The printer normally has a maximum print width of 80 characters, but increased and compressed print modes can be initiated by outputting to the printer appropriate escape sequences. In compressed mode up to 142 characters can be printed on each line - very convenient for tabulated output, for example from NAS-CALC.

The printer can also operate in a graphics mode, where each dot is effectively addressable. The AVC software now includes an option for screen dumps to the LX printer, and an example of one of these is attached - the picture was originally generated using NAS-CAD.

Both friction and tractor feed are provided.

A more detailed specification is given in our advertisement. The printer is available from the usual dealers, and may also be found at other outlets.

3. NAS-CAD

NAS-CAD is a computer aided drawing package for use with a Lucas Nascom computer fitted with the Advanced Video Controller (AVC). Development has been carried out on the NAS-DOS version, but a CP/M version will be available soon after the release of the NAS-DOS version. The program is written in Nascom Extended BASIC (XBASIC), but uses machine code routines to increase speed. The program is too large to be entirely in memory, and therefore chaining is used so that only the required parts of the program are in memory at any one time. Loading sections of program from disc can cause brief delays when commands are typed in. However, using the 256K RAM card as a virtual disc under CP/M (this option is available free of charge to registered CP/M users with a 256K RAM card) delays are not noticeable. NAS-CAD

should be available for delivery in September. At the time of writing (mid June) the prototype is operational, but some refinements and further testing are required.

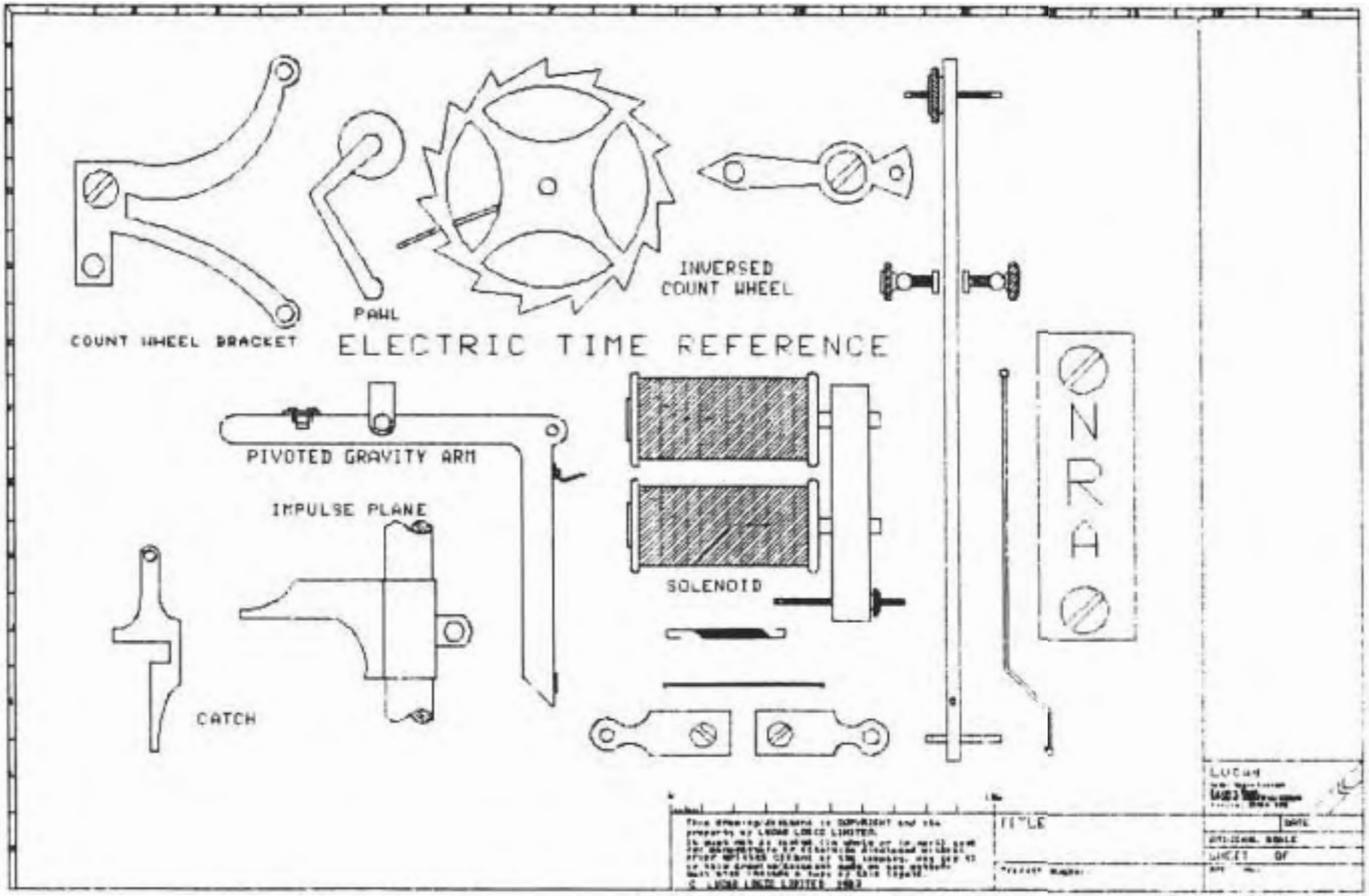
Firstly we must put the facilities offered by NAS-CAD into perspective. Full function commercially available three dimensional computer aided drawing systems generally cost over £100,000, and even low-cost 2 and a half dimensional systems generally cost around £50,000. NAS-CAD as a complete system of hardware and software will cost around £3,000, and does not pretend to offer all the features of systems costing more than ten times as much. Nevertheless it is a completely practical system which can be used in preparing technical drawings, layouts, scheme drawings etc. It can handle up to about 3000 lines per stored shape or object. These objects can be stored on disc and then recalled, thus allowing standard objects to be incorporated in a drawing.

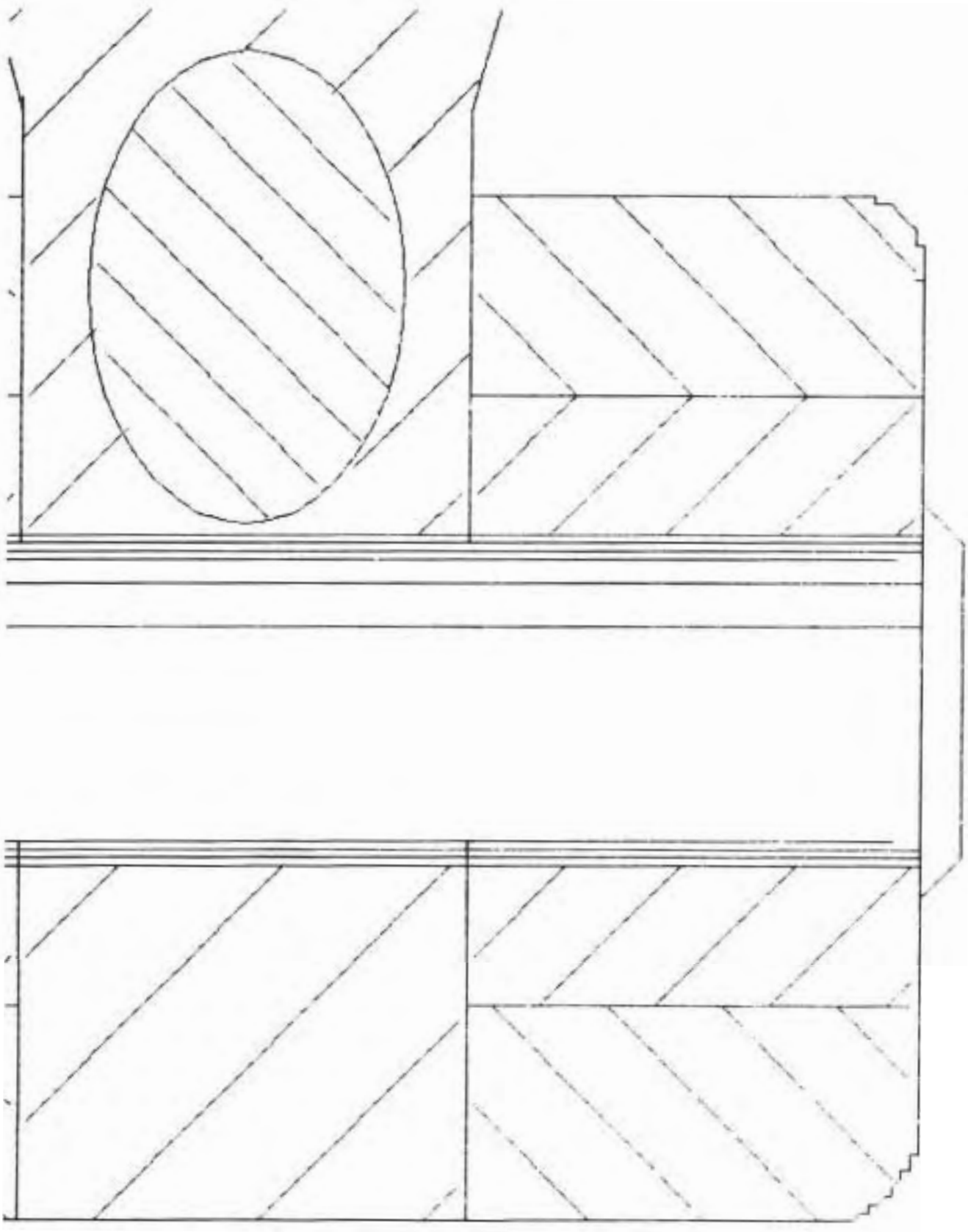
NAS-CAD is used with two monitors (or a monitor and TV). One is used for the picture output from the AVC, while the other displays the 48 x 16 text display. The latter is used to display the menu of commands available. There is a main menu of operations which can be performed, specifying a particular option results in the appropriate sub-menu being displayed. Commands are typed in using the keyboard, which is also used to specify scaling factors, angles of rotation etc.

The graphics screen is used to display the drawing being produced, to a resolution of 380 points horizontally by 256 vertically. A large cross indicates the current cursor position, and is referred to as the graphics cursor. This graphics cursor can be moved around the screen using the normal cursor control keys, although in this case the cursor is of course defining a single pixel at the intersection point. The cursor can easily be positioned to an accuracy of a single pixel, and its current screen co-ordinates will be displayed on command. For fast movement of the cursor the GRAPH key is held down while the cursor control keys are pressed. Points, lines and polygons can be drawn and saved on disc, or recalled from disc and positioned anywhere on an existing drawing. The whole drawing, or a re-loaded object, can be magnified or rotated as required. Colour can be used in the drawing, or colours can be altered.

Any drawing can be modified at any time quite easily. Text, including dimensioning can be added to a drawing.

The full version of NAS-CAD described above is an extremely powerful package and has required substantial development time. This will probably be reflected in the price, but a more modest version at a correspondingly modest price for the enthusiast or educational user will probably also be available retaining the main features of the program. We will give more technical, marketing and availability information in the next issue of Nascom News.


COUNT WHEEL BRACKET
PAWL
INVERSED
COUNT WHEEL
ELECTRIC TIME REFERENCE
PIVOTED GRAVITY ARM
IMPULSE PLANE
SOLENOID
CATCH
ZRO
TITLE



4. CORRECTIONS

One or two corrections need to be made to previous articles.

Firstly the parallel printer driver from BASIC, section 3.2 of the April newsletter. In line 80 the final value is shown as 217, when it should be 201. This error originated in Application Note AN-0006, and was due to an error in converting hex to decimal (I should have used the computer!). The error is also present in the current BASIC manual. In the same article the listing seems to have got slightly corrupted at line number 5 - there should be a line

```
PORTA  EQU  4
```

to declare the A port address - this seems to have become merged with a preceding comment line.

Secondly the keyboard interrupt listing in the last issue. I merged this into the SPEX word processor file without looking properly at the line length. You may therefore find the layout has been right justified on some long lines, producing an odd appearance, depending on how the Editor formatted the pages when producing copies for the newsletter. This should still be understandable, but is a function of how the text was formatted rather than being a fault in NAS-SEMBLER.

5. USEFUL INFORMATION

Some users have asked about the access times of different versions of the Nascom disc drives and the values used in NAS-DOS. The earler Nascom drives had a 30 mS access time, while the latest half height units have 6 mS access time. The different versions can be recognised as follows, listed in chronological order:

1. Earliest drives, all plastic sides (no inset metal plate). Small horizontally mounted pot in the centre of the PCB mounted on the drive. These are 30 mS drives, but many can be made to run at 20 mS, although you may need to tweak the pot.

2. As above, but no pot fitted. These will usually only run as 30 mS drives.

3. As 1.

4. Small metal plates mounted on each side of the drive which carry the mounting screw. These ae 20 mS drives.

5. Half height drives. These have a 6 mS access time.

The access time is controlled by outputting a byte at one point - NAS-DOS. This byte is interpreted as follows:

1F	30 mS
1E	20 mS
1D	12 mS
1C	6 mS

The relevant location will depend on the version of NAS-DOS which you are using:

1.1	D34D
1.2	D346
1.4	D34E

NAS-DOS 1.1 was originally supplied for 30 mS drives, while 1.2 and 1.4 are configured for the 20 mS drive.

Those of you with EPROM blowers can change the relevant location to suit your drives if necessary. Do remember the usual precautions and keep the original chip in case of accidents during the modification process.

6. DISC FILES

As well as allowing programs to be stored on disc NAS-DOS allows data to be storedand loaded. Routines are provided within NAS-DOS itself to assist in data storage and retrieval, and these can be accessed directly from assembly language and Nascom ROM BASIC. ROM BASIC was developed long before discs were available on the Nascom system, and therefore the language itself does not provide any disc commands. Access to the disc is therefore achieved by patching in the various NAS-DOS routines. This is done by means of the USR( ) function in BASIC, the argument being the number of the NAS-DOS routine to be called. It is of course necessary to DOKE 4100,-10234 to point the USR routine to an appropriate translator in NAS-DOS itself. This slightly awkward technique is necessary to allow the commands to be added to a ROM BASIC - it is not a limitation of NAS-DOS. More recently implemented languages (eg Extended BASIC) and applications programs (eg NAS-CALC) include within them meaningful commands for direct access to the discs via NAS-DOS. The DIR (directory) command is an example of this facility.

Before any disc commands are used in a BASIC program under NAS-DOS the command

```
DOKE4100,-10234
```

must be executed by the program to set up a link to the NAS-DOS commands. You must then initialise the NAS-DOS commands and data area with an instruction of the form

```
A=USR(1)
```

When using subsequent disc commands remember that the number of the disc drive is stored in location 3360 (decimal), so if you want to change the disc drive accessed you must include in the

program a line

POKE3360,n

where n represents the disc number - either a constant or a variable.

Before you can load or save data from within a program the name of the file used has to be identified to NAS-DOS. NAS-DOS allows one file name to be used for input and another for output. It does not allow multiple input or output files, so in applications where several input or output files are to be handled the appropriate ones must be opened before access is attempted. There are three forms of file opening command:

A=USR(11),FN$ - opens a file for reading.
A=USR(12),FN$ - opens an existing file for output.
A=USR(13),FN$ - opens a new output file.

Note that in the case of a new output file the length of this file, in sectors of 256 bytes, should have been set by a POKE3367,n instruction, where n is the number of sectors. If there is insufficient space on the disc for this number of sectors the maximum available space will be used.

FN$ is the file name to be used, and must be a string variable (not an array) of length 8 characters. If the name is less than 8 characters long you MUST include extra spaces in declaring the name - eg

FN$="MIKE   "

NAS-DOS returns in the function an indication of whether the file name specified existed - a non-zero result indicates that the file name did not exist. Your program should check this error condition and trap any error before any further file access is attempted.

Note that NAS-DOS organises disc file so that all the contents of a file must be contiguous - ie the file is not divided and spread over the disc. Thus it is possible to delete a lot of small files spread around the disc and have an apparently large amount of disc space available, but as this is not contiguous it may be impossible to create a large disc file, even though its size is less than the total free disc space. The disc utilitiy REORG could be used to copy and compress the free space, although in practice this is an extremely rare problem. When files are deleted in NAS-DOS the space occupied by the file itself is freed for future use automatically, although the entry in the directory is merely marked as a deleted file, rather than being removed from the directory. Any deleted files in the directroy are removed by the JC command of NAS-DOS. This technique can be very useful in sometimes allowing apparently deleted files to be found ane recovered relatively easily from disc if a (user!) program goes berserk and overwrites the directory.

Most disc access commands using the USR function in ROM BASIC,

including the file opening commands described above, require additional variable names to specify the data (or filename) to be saved or loaded. There are a number of significant restrictions associated with these names in ROM BASIC. Again you should realise that these are associated with the simple BASIC Routine Handler incorporated within NAS-DOS for use with the ROM BASIC. These restrictions are not inherent in NAS-DOS itself, and the Extended BASIC, XBASIC, for example, incorporates more sophisticated transfer of data between files and variables which are not subject to these restrictions. The main restrictions are as follows:

1. Only simple character strings may be loaded or saved. Numbers must be converted using the STR$ and VAL functions before/after disc access. The variable cannot be an array element. Thus if you have a 20 element numeric array A( ) which you wish to store on disc you would need to use lines of the form

```
100 FOR I=1 TO 20
110 A$= STR$(A(I))
120 A=USR(32),A$
130 NEXT
```

to save the elements in character string form.

2. While more than one string can be loaded/saved in a single USR( ) statement the strings are stored directly on disc with no separator between them, although after each disc write statement an end-of-record character (0D hex) is output to disc. Therefore in order to read back the data where more than one item has been stored in a record the strings must be of known fixed length otherwise the disc read statement will not be able to determine where one string ends and the next begins. Where the length of the string is variable you will need to fix the length by padding it out to the requied length with additional spaces. Note that this applies where more than one variable is read/written in a single USR call - if only one variable is referenced each time NAS-DOS will use the end-of-record separator which is put in the file after each disc write statement to determine where data which is being read terminates.

These limitations do not affect XBASIC, which allows almost any type of variable to be read or written - see the XBASIC manual for more details on this.

There are essentially two different types of data file which can be saved on disc - at least using NAS-DOS. These are known as sequential files and random files. We will describe sequential files here and introduce random files. next month we will describe random files in more detail.

Sequential files are very comparable with data written onto magnetic tape. Generally to get any item on the file you must start from the beginning and read (or write) each item in turn until you get to the required point. If you are writing the file you must then write any remaining data. Because the end-

of-record marker delimits data items it is not essential that the records are of fixed length. Therefore it is impossible to replace an item in the middle of a file directly - there is no certainty where the record concerned begins,or whether the replacement will exactly fit in the space. Editing of these files can therefore be laborious, as is access to any point in the file. The method which would need to be used to insert an item would be to read each record from an input file and copy it to a new output file. When the point at which the new record is to be inserted is reached the new record would be output, and then the remainder of the file would be read and copied record by record into the new file. The old file could then be deleted and the new one re-named.

Sequential files are quite convenient when you wish to read all the contents of a file into memory and carry out the manipulation of the data in memory. When you have finished processing the data in memory the entire data can be written to a sequential file. Thus in handling data which can be loaded into memory this is a convenient method to use, since operating on the data in memory is inevitably quicker than any method which requires repeated disc access to obtain individual data items. Small data files and index files can conveniently be treated in this way.

A random file can be read/written directly at any point. In NAS-DOS the positions in a file are measured in disc sectors (of 256 bytes) from the start of the file. In its simplest, and most common form, each record will be up to 255 bytes long and will occupy one sector. You can go to any sector directly by specifying its relative sector number in the file prior to the read or write operation. This is done by a DOKE to location 3365 of the relative sector number, starting at 0 for the first sector, eg

```
120 DOKE 3365,10
```

Since you have direct access in this way to any point in a file it is very simple to amend a specific record simply by rewriting the appropriate sector of data on disc. You can if necessary arrange your own 'housekeeping' routines to pack two sets of data into a sector, or to use 2 sectors to store each set of data. For example in the latter case you would access the 10'th item of the data by starting to read at relative sector 18 of the file (the first item is in sectors 0 and 1, the second in 2 and 3 etc).

The problem with a random file is knowing where to store a new item of data, and more importantly where to find a data item. Most records stored on disc will have the most important item in the record at the beginning. The individual items of data in a record are usually known as fields, and the first of these, or the one by which each record is usually referenced (eg in searchesS) is the key field. One method of inserting a new record into a random file would be to store the records in alphabetical order by their key field. This has two major drawbacks - firstly to store an item we would have to move all records up the file to create the space for the new entry. The

second drawback would be in finding a particular entry - we could either read each record in turn starting at the beginning until we get to the right one, or more efficiently we could look at the middle record of the file, determine if this is before or after the required data item, and then look at the middle record of either the top or bottom half of the records, etc. There are time penalties in these approaches since disc access with floppy discs is relatively slow, although the latter, binary seach, method is quite efficient.
Next month we will discuss the alternative methods of controlling access to random files.

7. AVC CONNECTION STANDARDS

We have now defined some standards for connection of colour monitors to the AVC, and these ae used on those production machines which have the AVC fitted as standard.

These connections can also be used for monochrome monitors, although most users have preferred in the past to use the normal BNC connector as the output from the AVC. The advantage of the scheme described here is that the 48 x 16 display can be made permanaently available while using the AVC output for graphics. This can be very convenient if you want to run twin screens - one containing menus or other text. This technique is used in NAS-CAD. An alternative would of course be to use the TV output for the 48 x 16 screen. Although clarity of the picture is not as good on a TV display. A 25 way D-type plug is used for the colour monitor and 48 x 16 displays while the BNC connector carries the usual AVC card output (controlled by the MODE command to give either graphics or 48 x 16 display). Note that a plug is used on the rear panel of the computer (preferably in position CON 4) to reduce the danger of plugging a printer into the AVC and causing damage.

The connections between the 25-way D-type plug and the 16-way Scotchflex connector on the AVC are as follows:

25-way D-type	16-way Scotchflex
1	1
14	2
2	3
15	4
3	5
16	6
4	7
17	8
5	9
18	10
6	11
19	12
7	13
20	14
8	15
21	16

```
25-way          Other
D-type          Connections

 9              AVC VIDEO OUT
22              AVC V GREY
10              AVC GS COM
23              AVC R COM
11              AVC B COM
24              AVC 0 VOLTS
12              AVC 0 VOLTS
25              Nascom 2 VIDEO
13              Nascom 2 0 VOLTS
```

8. DRIVER LISTING FOR NON-AUTO LINE FEED PRINTERS

```
                0000 ;:TCENDRV:
                0001 ;
                0002 ;PARALLEL PRINTER DRIVER
                0003 ;-----------------------
                0004 ;
                0005 ;REV 1.4     28 JUNE 1983
                0006 ;
                0007 ;THIS ROUTINE IS FOR USE WITH PRINTERS
                0008 ;USING A PARALLEL, CENTRONICS, TYPE OF
                0009 ;INTERFACE, CONNECTED VIA THE PIO.
                0010 ;REV 1.4 SUPPORTS PRINTERS WITHOUT AUTO LF
                0011 ;
                0012 ;THE CONNECTIONS SHOULD BE MADE AS
                0013 ;DESCRIBED IN APPLICATIONS NOTES AN-005
                0014 ;AND AN-006.
                0015 ;
                0016 ;THE MAIN DRIVER ROUTINE IS LOCATED AT
                0017 ;THE BEGINNING, AND THE CONFIGURATOR
                0018 ;ROUTINE IS LOCATED AT INIT.  THE
                0019 ;CONFIGURATOR MUST BE EXECUTED ONCE AT
                0020 ;THE START OF ANY SESSION TO ENABLE
                0021 ;THE PIO PORTS IN THE CORRECT MODE.
                0022 ;
                0023 ;*****************************************
                0024 ;
                0025 ;MAIN DRIVER
                0026 ;
                0027        ORG     0
                0028 PORTA  EQU     4 ;PIO PORT ADDRESSES
                0029 PORTB  EQU     5
                0030 CONPTA EQU     6
                0031 CONPTB EQU     7
                0032 ;
                0033        ORG     0C80
0C80 FE 0A      0034 CENTP  CP      0A      ;STRIP OFF LF'S
0C82 C8         0035        RET     Z
0C83 CD 91 0C   0036        CALL    CENTQ
0C86 FE 0D      0037        CP      0D      ;OMIT IF AUTO LF
0C88 C0         0038        RET     NZ      ;THESE LINES UP TO
0C89 3E 0A      0039        LD      A,0A    ;THE NEXT RET
0C8B CD 91 0C   0040        CALL    CENTQ
```

```
0C8E 3E 0D          0041          LD      A,0D
0C90 C9             0042          RET
                    0043 ;
0C91 F5             0044 CENTQ    PUSH    AF
0C92 DB 04          0045 CP1      IN      A,(PORTA) ;CHECK IF BUSY
0C94 CB 47          0046          BIT     0,A
0C96 20 FA          0047          JR      NZ,CP1 ;WAIT TILL FREE
                    0048 ;
0C98 F1             0049          POP     AF
0C99 F5             0050          PUSH    AF
0C9A D3 05          0051          OUT     (PORTB),A
0C9C CB CF          0052          SET     1,A ;STROBE THE DATA OUT
0C9E D3 04          0053          OUT     (PORTA),A
0CA0 00             0054          NOP     ;DELAY FOR SETTLING
0CA1 CB 8F          0055          RES     1,A
0CA3 D3 04          0056          OUT     (PORTA),A
0CA5 00             0057          NOP     ;DELAY AGAIN
0CA6 CB CF          0058          SET     1,A
0CA8 D3 04          0059          OUT     (PORTA),A
0CAA F1             0060          POP     AF
0CAB C9             0061          RET
                    0062 ;
                    0063 ;INITIALISER - CONFIGURES THE PIO PORTS
                    0064 ;AND SETS THE 'U' VECTOR
                    0065 ;
0CAC 3E CF          0066 INIT     LD      A,0CF ;PORT CONFIGURATION
0CAE D3 06          0067          OUT     (CONPTA),A
0CB0 3E FD          0068          LD      A,0FD ;MAKE ALL PORT A  BITS
INPUT
0CB2 D3 06          0069          OUT     (CONPTA),A ;EXCEPT  1  (DATA
STROBE)
0CB4 3E 02          0070          LD      A,2 ;SET STOBE LINE HIGH
0CB6 D3 04          0071          OUT     (PORTA),A
0CB8 3E 0F          0072          LD      A,00F
0CBA D3 07          0073          OUT     (CONPTB),A
0CBC 21 B0 0C       0074          LD      HL,CENTP ;SET THE 'U' VECTOR
0CBF 22 78 0C       0075          LD      (00C78),HL
0CC2 DF 5B          0076          SCAL    05B ;RETURN TO NAS-SYS
```

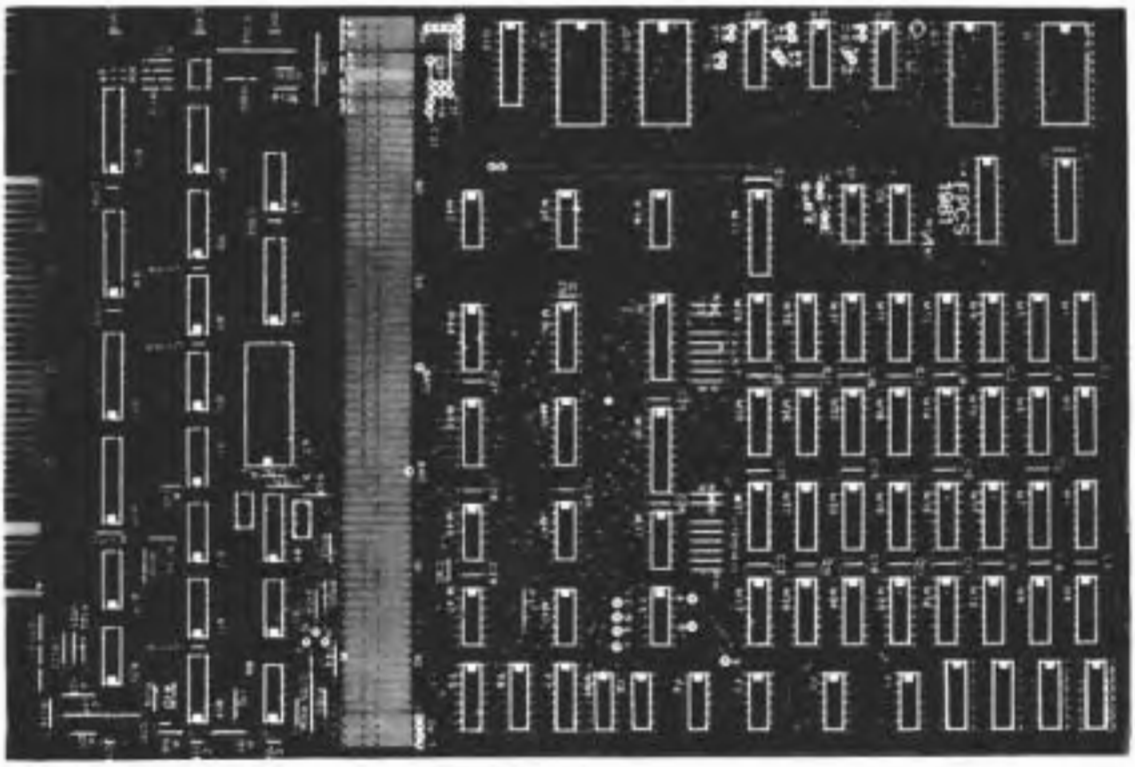
==========

Private Ads.

FOR SALE - Nascom 2 with 48K RAM B board in Verorack. Sound board, manuals, Tandy monitor, games tapes
£375 o.n.o. 08675-3750

WANTED - Gemini GM809 disc controller card and Pertec FD250 disc drive. Will part exchange a brand new unused TEAC 55E double-sided, half-height disc drive against your working FD250.
Wood, 'Limes', Druidstone Road, St. Mellons, Cardiff CF3 9XD
Tel. 0222-791425

FOR SALE - Nascom 2 cased with 48K, Naspen, ZEAP, toolkit and graphics. Also much software, magazines. £260 o.n.o.
Hobbit microcassette drive, cased with spare tapes. £70
IMP printer. Little used with spare paper. £125
Tel. Colchester 841293



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%.