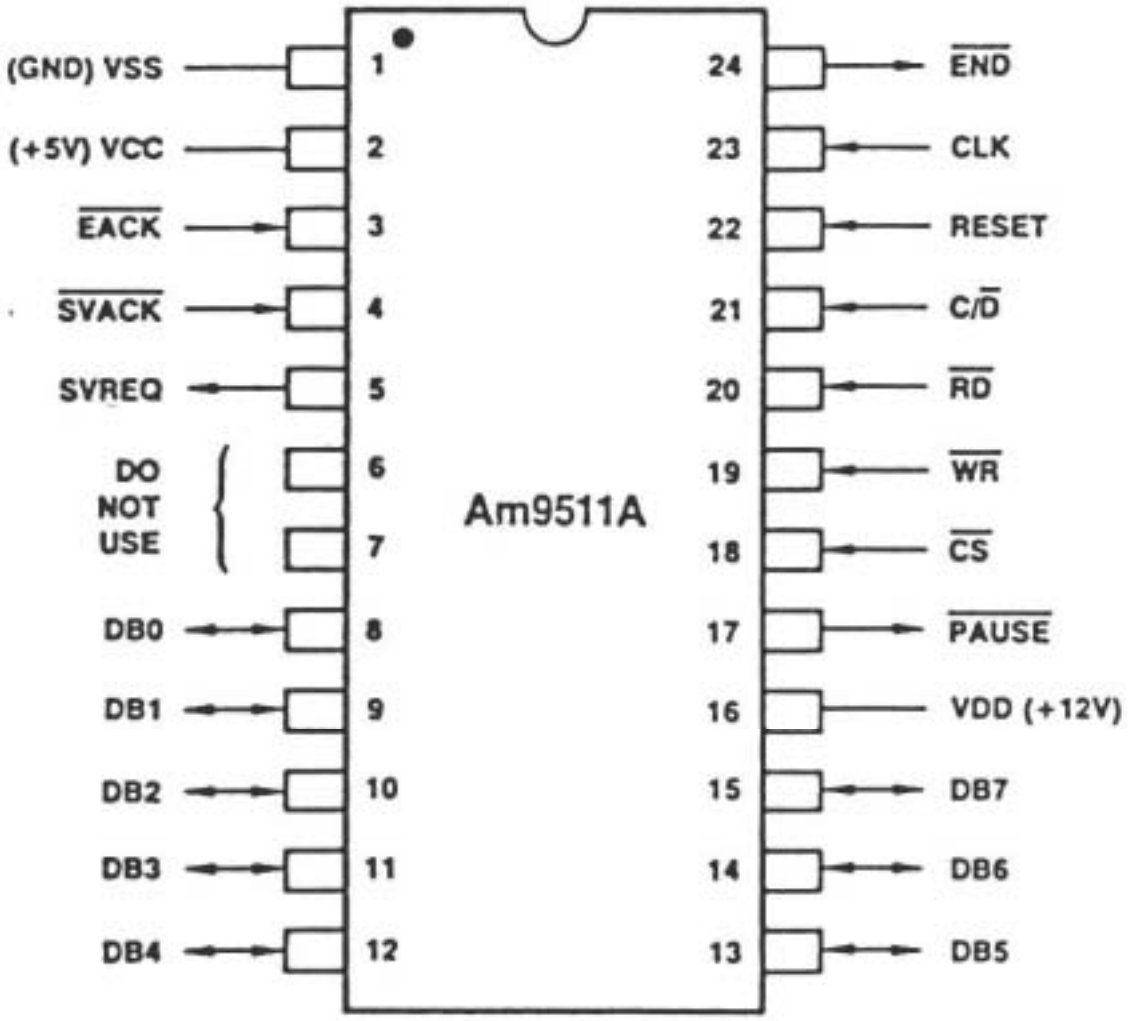
# nascom

# NEWSLETTER

Volume 3 Number 4
December 1983
£1.25




Lucas Nascom

# Back Issues

Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues

We currently have copies of all the back issues of the Nascom Newsletter and Micropower magazines.

We are now in a position to sell these off at rock bottom prices to clear our shelves.

Volume 1 (Issues 1 to 4)

1 magazine 50p
3 magazines for £1

Volume 2 (Issues 1 to 6)

1 magazine 50p
3 magazines for £1

Our stocks are limited so order now for those missing issues.

Please add 55p P&P to your orders

Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues
Back Issues

NASCOM NEWSLETTER Vol 3, No 4 December, 1983

Contents



Editor - Ian J Clemmett

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

Printed by - Dataform Press

## For Sale

For Sale - Zeap 2.1 (4 x 2708) £22, NasDis + NasDebug (4 x 2708) £20, Bits and PC'c Toolkit (2 x 2708) £10, Nas-Sys 1 ROM £6, I/O High Resolution board £35. All fully documented.
Ring (Burscough) 894948

For Sale - Nascom 2 in a verorack with 48K RAM B and sound board. Monitor, manuals and programs.
£300 ono.
Also AVT Monitor £60 ono.
Ring Kidlington 3750

For Sale - 32K RAM board
G805 FDC board + DOS
Pertec 200 drive
Reasonable offers accepted
Ring Newport (Gwent) 612616

# SYS-EX A 1K extension program for NAS-SYS monitors - Part 4.

by David G. Johnson

Erratta to Part 2, (Vol.3, No.2)
1) Page 6
```
    ADD L  )     ADD 30H turn into Ascii code
```
should read
```
    ADD L
    ADD 30H  ) turn into Ascii code
```
2) Page 6
```
    .ROUT    print character from A  CP  20H  Z  flag  if  final
space
```
should read
```
    .ROUT    ) print character from A
    CP 20H   ) Z flag if final space
```

In the last issue we completed the detailed description of the SYS-EX keyboard commands and promised that a description of the SYS-EX BASIC routines would follow in the next issue. Well, here is the next issue and we begin with the SYS-EX BASIC routines.

To recap briefly, SYS-EX provides an alternate set of commands to the 8K BASICs CSAVE, CLOAD and CLOAD?. These commands allow program names to be up to 42 characters long. The commands allow access to tape labels written using the SYS-EX "1" command. The commands are fully compatible with CSAVE, CLOAD and CLOAD? and programs saved with one set of commands can be recognised and read using the other set of commands.

The SYS-EX BASIC named file and label commands are called by using the USR function. Prior to using USR, the start of the SYS-EX BASIC routines should be DOKE'd into memory. The BASIC entry point in SYS-EX is located 01E0H beyond the start of SYS-EX. When the monitor extension is installed at the recommended memory address (B000H to B3FFH) the required BASIC statement is:

```
DOKE 4100,-20000          (ie. B1E0H = -20000 decimal)
```

As SYS-EX is fully relocatable, you may have decided to put it at a different memory address. If so, just add 01E0H to the hex start of SYS-EX then convert this to decimal using the "h" command. You won't get the nice easy to remember number (-20000), but it will work just the same.

The routines may be called by a statement such as PRINT USR(n) or ?USR(n) or I=USR(n). Even though a PRINT statement or an assignment statement is used, nothing will be printed or assigned to a variable on completion of the SYS-EX commands. This is because SYS-EX warm starts BASIC when it finishes a command.(ie. it does not just return to BASIC)

READ NAMED PROGRAM FILE - USR(0) (CLOAD equivalent)

Read a cassette tape and search for a file name which matches with the required program name. When a match is found, read the program into memory.

After entry of the USR(0) statement, the prompt

rName:

appears on the next screen line and the required program name is entered on that line. If the first program is required (irrespective of the name) just press ENTER at this point. Optionally, 'transparent' characters may be entered as part of the required program 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 line of the screen without interruption of the command. As each file or program label is detected, the name is displayed on the screen line immediately below the required program name. Any name or text which was previously displayed on that line is blanked out. The program name and the required name are compared. A required program 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 (ie. 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 program is read into memory.

Between entry of the required program name and a successful name match, 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 cassette tape will have the same effect.

The command will also recognise as program names, any file names written using the SYS-EX "w" command. If a file which is not a BASIC program file matches with the required name and is read into memory, it may be necessary to RESET the computer and to cold start BASIC once again.(NAS-SYS "J")

VERIFY NAMED PROGRAM FILE - USR(1) (CLOAD? equivalent)

Read a cassette tape and search for a file name which matches with the required program name. When a match is found, read the program but do not load it into memory.

After entry of the USR(1) statement, the prompt

vName:

appears on the next screen line and the required program name is entered. The action of the command is identical to the READ NAMED PROGRAM FILE command, with the exception that the required program is NOT loaded into memory. The command is used to check that a program file may be successfully read from cassette tape.

SEARCH FOR TAPE LABEL - USR(2)

Read a cassette tape and search for a tape label as written by the SYS-EX "l" command. Display the first label found on the top line of the screen then terminate the command.

The command may be used for positioning a cassette tape prior to using the WRITE NAMED PROGRAM FILE 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 recorder will have the same effect.

WRITE NAMED PROGRAM FILE - USR(3) (CSAVE equivalent)

Write the BASIC program currently in memory to cassette tape. The program is preceded by a file label containing the specified program name. The file label will be recognised and displayed in full by the SYS-EX commands "q", "r" and "v" in addition to the WRITE and VERIFY NAMED PROGRAM FILE commands. The BASICs CLOAD and CLOAD? commands will also recognise the file label, but will use and display only the first character of the program name.

After entry of the USR(3) statement, the prompt

wName:

appears on the next screen line and the name of the program to be written is entered on that line. The tape drive LED is then switched on and, after a short delay, 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.

NB.
It is possible to escape from the above commands while the file name prompt is displayed on the screen.(ie. rName: , vName: , wName:) To do this, backspace over the prompt or move the cursor to another line before pressing ENTER. An error message will be produced, but no harm will be done.

RETURNING TO SYS-EX FROM BASIC

It is useful to have an equivalent of the MONITOR command which, instead of returning solely to NAS-SYS, returns to SYS-EX so that the full range of NAS-SYS and SYS-EX keyboard commands

may be used. The SYS-EX equivalent of MONITOR is to perform a warm start to SYS-EX. This may be achieved from BASIC in the following way.

DOKE the SYS-EX warm start entry point into location 4100. The warm start entry point is 000BH beyond the start of SYS-EX. If SYS-EX is installed at B000H, the required BASIC command is:

```
DOKE 4100,-20469          (ie. B00BH = -20469)
```

BASIC can now be left by calling the USR function. (eg. PRINT USR(0) )

One example of the usefulness of this feature is when a conversion between decimal and hexadecimal is required. In this case, the procedure would be:

1. Leave BASIC using:

```
DOKE 4100,-20469 : PRINT USR(0)
```

2. Perform the conversion using "d" or "h"

3. Return to BASIC using NAS-SYS "Z".

If you put SYS-EX into EPROM and set the hardware jump to start SYS-EX, then a press of the RESET button will take you straight from BASIC to SYS-EX. As this performs a cold start, the screen will also be cleared.

Now for something for Z80 assembler programmers. An earlier part mentioned that all of the SYS-EX routines may be called from an assembler program. The available routines are all of the keyboard commands plus 9 further commands. All routines are called by calling the same entry point within SYS-EX (3 bytes on from the start of the monitor extension). The byte following the call instruction should contain the required routine number. In the case of the keyboard commands, the routine number is the ASCII code of the lower case letter. Routines start at 61H ("a"). The 9 additional routines are numbered 7BH through 83H.

Parameters required for a called command are set up in the NAS-SYS arguments before the command is called. Commands do not preserve the contents of registers AF, BC, DE, HL, so the user must save any values required. If, on return from calling a SYS-EX routine, the Carry flag is set ON, an error condition has been detected and the command may not have been completed.

The 9 routines which are available only by calling from an assembler program are now described.

Routine no. 7B Name: FCEP1

Find in memory a single occurrence of the mask which immediately follows on from the routine number. The mask may be any length and is terminated with 00H. (eg. to find AA BB CC , code CD 03 B0 7B AA BB CC 00 ) The search starts at the memory address contained in ARG1 in the NAS-SYS workspace and continues to FFFFH or to the first occurrence of the required mask, whichever occurs first. If the mask is found, the Zero flag is set OFF at completion and the address of the match is in the HL register pair. If the search stops at FFFFH, the Zero flag is set ON.

Routine no. 7C Name: FCEP2

Find in memory a single occurrence of the mask whose start address is contained in ARG2 in the NAS-SYS workspace. The mask may be any length and is terminated with 00H. The search starts at the memory address contained in ARG1 in the NAS-SYS workspace and continues to FFFFH or to the first occurrence of the required mask, whichever occurs first. If the mask is found, the Zero flag is set OFF at completion and the address of the match is in the HL register pair. If the search stops at FFFFH, the Zero flag is set ON.

Routine no. 7D Name: SGNNO

Output in ASCII the decimal equivalent of the signed hexadecimal value in NUMV in the NAS-SYS workspace. The number of digits in NUMN in the NAS-SYS workspace is used when determining the sign. (eg. If NUMV = 00FFH and NUMN = 2 or less, then the result will be -1. If NUMV = 00FFH and NUMN = 3 or more, then the result will be 255.) If the result of the conversion is negative, a leading minus sign is output. Decimal values in the range minus 32768 to plus 32767 may be output.

Routine no. 7E Name: USGNO

Output in ASCII the decimal equivalent of the unsigned format hexadecimal value in NUMV in the NAS-SYS workspace. Decimal values in the range 0 to 65535 may be output.

Routine no. 7F Name: BKRM1

Blank out the current screen line from the cursor position onwards. Do a carriage return unless the cursor is at the start of a line.

Routine no. 80 Name: BKRM2

Blank out the current screen line from the cursor position onwards. Leave the cursor in its original position.

Routine no. 81 Name: DTOHX

Convert an ASCII format decimal number (leading minus sign permissible), to a four digit hexadecimal value. Put the result into both NUMV in the NAS-SYS workspace and the HL register pair. Prior to calling the routine, the memory address of the first character of the decimal number (if negative, the minus sign) must be placed in ARG1 in the NAS-SYS workspace. Leading spaces are not permitted. If the decimal number is positive, the A and the C registers return with the value 00H. If the decimal number is negative, the A and the C registers return with the value FFH. NUMN in the NAS-SYS workspace is set to the number of numeric digits in the input decimal number. If the input value contains characters other than 0 to 9 or if it is outside the range -32768 to 65535, conversion does not take place and the Carry flag is set ON. On a valid return, the Carry flag is set OFF.

Routine no. 82 Name: USCR1

Call a routine which starts 0400H (1K) beyond the start of SYS-EX. This routine is provided in order to allow user expansion of monitor facilities. The routine should only be called if the user has placed suitable code immediately following SYS-EX. HL , DE and BC are set to ARG1, ARG2 and ARG3 in the NAS-SYS workspace, respectively. The A register is set to 82H.

Routine no. 83 Name: USCR2

Identical to USCR1 except that the A register is set to 83H.

Well, that's it for this issue. In the next magazine, the finishing details of the monitor extension will appear. Also, the full hex dump of the monitor extension will be printed so that you can finally try out all of the SYS-EX facilities.

# A LIGHT PEN FOR THE NASCOM2

J.AUCKLAND

Some time ago I constructed a light-pen to interface with my Nascom 2, and I can declare that it is the cheapest on the market! The pen has been in use for quite some time and seems to be reasonably accurate, taking in to account the minimum resolution available with the standard Nascom Graphics chip.

The design for the light pen evolved because of my need for a device that would allow me to select routines from menus of routines. Some of these routines were to be used to input data by using the screen. I wanted a routine that would enable me to define the envelope shape of a particular sound that was to be generated in a micro-processor controlled synthesiser. Thus a peripheral was needed that would permit the definition of envelope shapes by drawing the envelope shapes directly on to the screen. It seemed that a light pen would be the most obvious device to use and so I looked to various manufacturers for a suitable light pen. Unfortunately, it transpired that the majority of pens were either out of my price range, or not quite the design that I required, and so I decided to build a pen specifically for the Nascom 2. The design works with my system, and the pen and its associated hardware is far cheaper than any other pen that I know of.

## PEN DESIGN

Having looked at the various photo-electric devices that could be employed in the design of the pen I decided to use a Light Activated Switch, which is an RS component type 305-434 (5V), as the photo-sensitive device. This device includes a variable-threshold switch that not only enables the sensitivity of the pen to be altered, but also reduces the component count by having the threshold switch as an integral part of the sensor.

Refering to figure 1 it can be seen that the output from pin 4 of the sensor is usually buffered by a TTL gate to bring the switched output to a clean TTL compatable level. The time-constant components, R1 and C1, set the sensitivity of the device, large values of RC give a high degree of sensitivity. It would be quite possible to use the circuit in figure 1 to directly interface with an input port. On the prototype, a pulse of 20us duration was output from the photo-sensor every time the raster struck the active area of the pen. This could be used to identify an individual pixel, or sub-pixel, of the screen area. By loading each successive VIDEO RAM location with the data FFH and by scanning the input ports for a HIGH from the pen, it is possible to identify the pen location on the video display unit. For a higher degree of accuracy, each VIDEO RAM memory location can be sequentially loaded with the data C1H, C2H, C4H, C8H, D0H, E0H. This data will then enable the pixel to be subdivided into 6 sub-pixels, and will cut a

small block of white at the top left-hand corner of the pixel, then at the middle left-hand side, and then at the bottom left-hand corner of the pixel. The last three bytes of data put the small block at the top right, mid right, and bottom left-hand corners of the pixel. Therefore, by successively loading each contiguous pixel with the six bytes of data, it becomes possible to identify the pen location on a screen with a resolution of

(48 x 2) x (16 x 3) = 4608 locations

If the output from the pen were connected to bit 7 of port 4, a program for identifying the pen position might look something like figure 3.

This method, crude though it may be, does also need synchronising with the BLANKING signal so that the "pen search" can commence as the screen scan starts. Otherwise, pixels may be tested after the raster scan has passed, and prevent the pixel under the pen from being tested at the same time as the raster is being displayed under the pen. This method of scanning without waiting for the blanking signal, gives rise to a screen full of random dots which, interesting though it may be for the first 10 secs., does not really solve the problem in hand. Also, if the scan program has to wait for the beginning of a new picture scan before it can load the screen with one of the six sub-pixels, it can be seen that each pixel will take at least 6 x 20ms to scan. So to scan all of the 768 locations will take

6 x 768 x 20ms = 92.16 seconds

With no stretch of the imagination can I say that this is satisfactory. A faster method would be to fill the screen with white and to scan the input ports for a HIGH from the pen. When the raster strikes the light-activated switch the other port could latch in the memory address of the VIDEO RAM that was being accessed at the time of the raster strike. As there are 1024 VIDEO RAM locations, it would be necessary to load in 8 bits to one port, and another 2 bits into the other port. This method would give the pen location on any of the 768 screen pixels. If the screen were successively filled with the six sub-pixel bytes, the resolution could be increased to the full 4608 locations.

As the VIDEO RAM is scanned under hardware control it is possible to tap off the required address lines and to feed them directly to the input ports. The program that I was using at the time of the pen's design, required the identification of the pen location as it traversed the screen in a left-right direction. This would enable me to draw a wave shape directly on to the screen and then store it for later use. The routines had to be identified from a selection of menus, and so the pen location was required for selection identification.

Refer to figure 4 it can be seen that the VIDEO RAM is addressed by A0 to A9, the address lines A0 to A5 sequentially address 64 screen memory locations, 48 of which are displayed, in a left-right direction. A6 to A9 select the 16 rows of 64 locations in a top-bottom manner. Though it must be noted that the screen top row is in fact the last row, row to be addressed, ie. 0BCAH to 0BF9H. Even though the VIDEO RAM scanning hardware is scanning from 0000H to 03FFH on the address lines, IC47, which is the NZMD PROM, selects the VIDEO RAM

write-enable when memory locations 0800H to 0BFFH are addressed. Thus there is an offset of 0800H to consider when evaluating the pen location from the VIDEO RAM hardware address lines A0 to A9.

## VIDEO ADDRESS LINES

To obtain the address lines, a small amount of soldering is required. It is possible to tap off the address lines directly from the board, though I prefer to keep all soldering and additions to the component side of the board, and it was for this reason that I decided to adopt a "piggy-back" approach to the address line location.

There are a number of ways to find the pen location by using the address lines. the approach that I took was to identify 1 sub-pixel in a column of 48 sub-pixels. As long as I knew which column I was addressing I could find the pen location on that column. It would be possible to find the pen position on the left-right axis as well as the up-down axis, as long as I looked at all the address lines, but this would mean looking at more than 1 input port, and I was already using that for something else.

Address lines A6 to A9 can be used to identify which of the 16 character lines is being addressed. This brings the pen location resolution to 1 in 16. To increase the resolution we can look at either all 14 raster lines, or, as the minimum sub-pixel is only 4 raster lines, we can look at the address lines that relate to the selection of each group of 4 raster lines. The RS lines, RS0 to RS3, are used to select the appropriate raster lines for each pixel, and so by looking at RS2 and RS3 it becomes possible to identify on which of the sub-pixels the pen is resting.

A6 to A9 can, therefore, be treated as the Most Significant Bits, and RS2 and RS3 as the Least Significant Bits, of a six-bit address that can identify a location in a column of 48 locations.

To tap off these address lines, I soldered 16-pin header-plugs directly on to IC68 and IC53. This method allowed me to solder wires to the plug with ease, and gave test-points to see if I had blown the chip by soldering too close to it! The six address lines were taken along the board and put on to the bus at pins 59 to 64. On the Gemini 80-BUS, these pins are now allocated as interrupt request lines, powerfail warning, and backup power, so it might be advisable not to put the video address lines on to the bus if you can help it. The Video Blanking signal, VBLANK, was also taken to the bus so that the screen data would be updated only in the screen blanking period. this prevents screen flicker when drawing images on to the screen.

## OTHER HARDWARE

For certain routines, I wanted to know if the pen was in the same place, or whether it had moved from an area of white to an area of black. Unfortunately, you can not merely test the pen status. This is because the sub-pixel consists of 4 raster lines, each being 64us in

duration, and each raster-strike on the active area of the pen produces a pulse of 20us duration. As the picture frame is re-displayed every 20ms, the sub-pixel will produce a train of four 20us pulses every 20ms. It is only through the persistance of vision that the image appears to be constantly displayed.

It now becomes necessary to build a circuit that will output two pulses. One pulse will be of 20us duration, to indicate that the pen has been struck by a raster, the other pulse will go high when the pen has been struck, and remain high for period of time that is greater than 20ms and will also be kept high every time that the pen is struck. By doing this, a high state is output as long as the pen remains over the displayed area, even though the area is being refreshed every 20ms and being displayed for only 80us.

Figure 5 is the circuit diagram for the pen interface. IC1 is an NE555 timer configured as an astable multivibrator, the frequency of which is adjusted by RV1. IC2 is a 7493 Binary counter and IC3 and 4 are 7400 NAND gates and 74LS126 TRI-STATE buffers.

On power-up, the 7493 Q outputs are in a low state and so IC3(a), in conjunction with the inverted output of IC2 and the output from the clock generator, begins to clock the counter through 16 counts until Q3 goes high. At this point, one of the inputs to IC3(a) becomes low and so the clock pulses are inhibited and the 7493 ceases to count. The HIGH at Q3 is inverted by IC3(b) and buffered by the TRI-STATE buffer IC4(a), the output of which will be enabled only when SW1 is closed.

It can be seen that the output to the ports is now held at a LOW state. If the pen is now activated a 20us pulse will be sent to both the OUTPUT1 and the RESET input of the counter. Q3 now goes LOW, and thus IC3(b) goes HIGH, enabling the clock input and allowing the count to begin again. The output2 has now gone HIGH indicating a hit. As long as the frequency from the clock generator is lower than 20ms x 16 ie about 800Hz, then IC2 can not count up to 16 before the receipt of another reset pulse. Therefore, if the pen is struck by the raster every 20ms, IC2 will be reset and Q3 will remain LOW and the output2 HIGH.

Even though the pen has moved from a white area to a black area, it will not register the change until 16 clock pulses have been received, and RV1 can be used to slow down the clock and increase the time taken to register a change in state. This can be used to effectively slow down the speed of the pen. Similarly, the frequency can be increased to a point where the reset pulse has no effect. The two LEDs indicate the pen's status. LED1 indicates whether the pen has registered a hit, and LED2 shows if the data has been enabled into the ports.

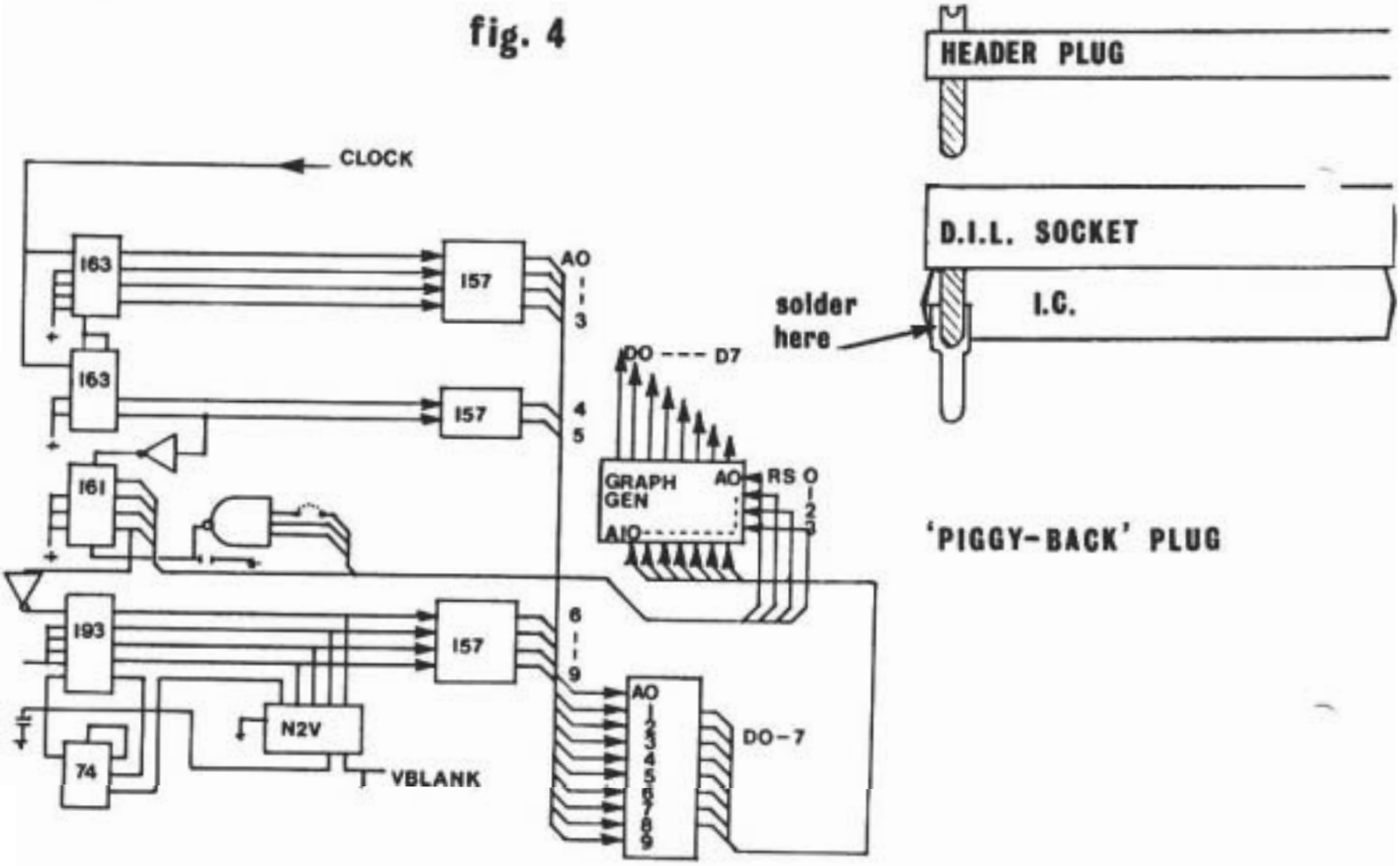
The prototype was built on a small piece of veroboard and housed in a small diecast aluminium box, and seems to have tolerated an immense amount of knocking. The pen was constructed using a small length of plastic tube with a 5-pin plug and socket to allow it to be disconnected from the system. I found that it was easier to assemble the pen in three pieces than to try and poke everything down the tube. If the sensor is glued into a smaller tube, (I used a drilled, solid piece of plastic for this) and then that inserted into a larger tube, the pen can be easily separated for modificaction. Similarly, the

switch can be set at the junction of two tube halves to facilitate re-wiring if necessary.

It should be possible for most people to construct a light pen using the above design, though I am quite sure that most readers will be aware that this is only one way of obtaining information as to the whereabouts of the raster.

Another method could be to count the blanking pulses, or indeed build a circuit on the lines of the NASCOM VIDEO RAM scanning circuitry, or buy a video board and have done with the interface problems!

**fig. 4**



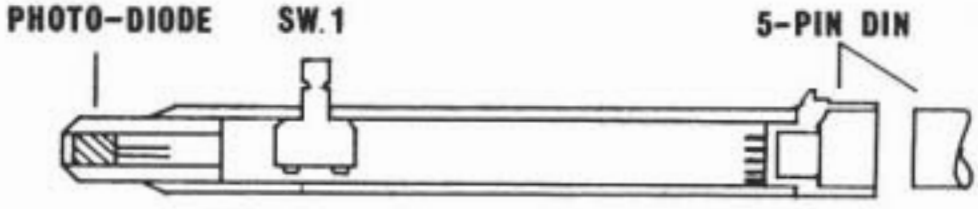

**VIDEO RAM**

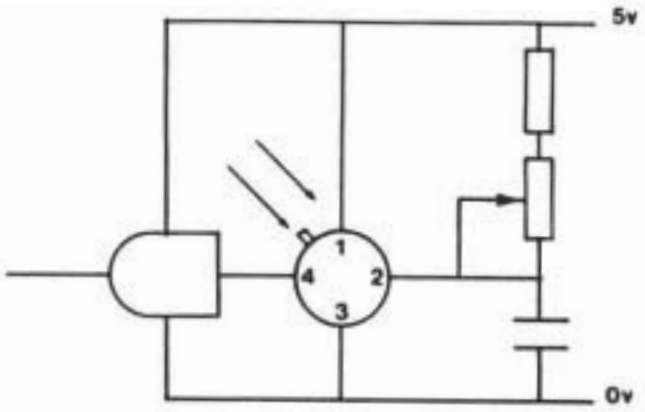



**'PIGGY-BACK' PLUG**

**NASCOM VIDEO-SCAN HARDWARE**

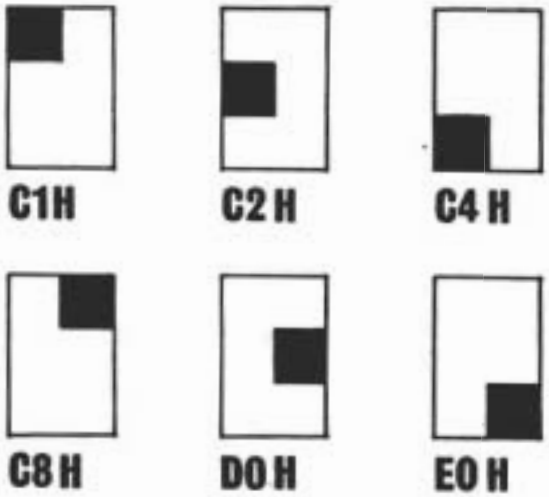
**LIGHT PEN**






LIGHT-ACTIVATED SWITCH

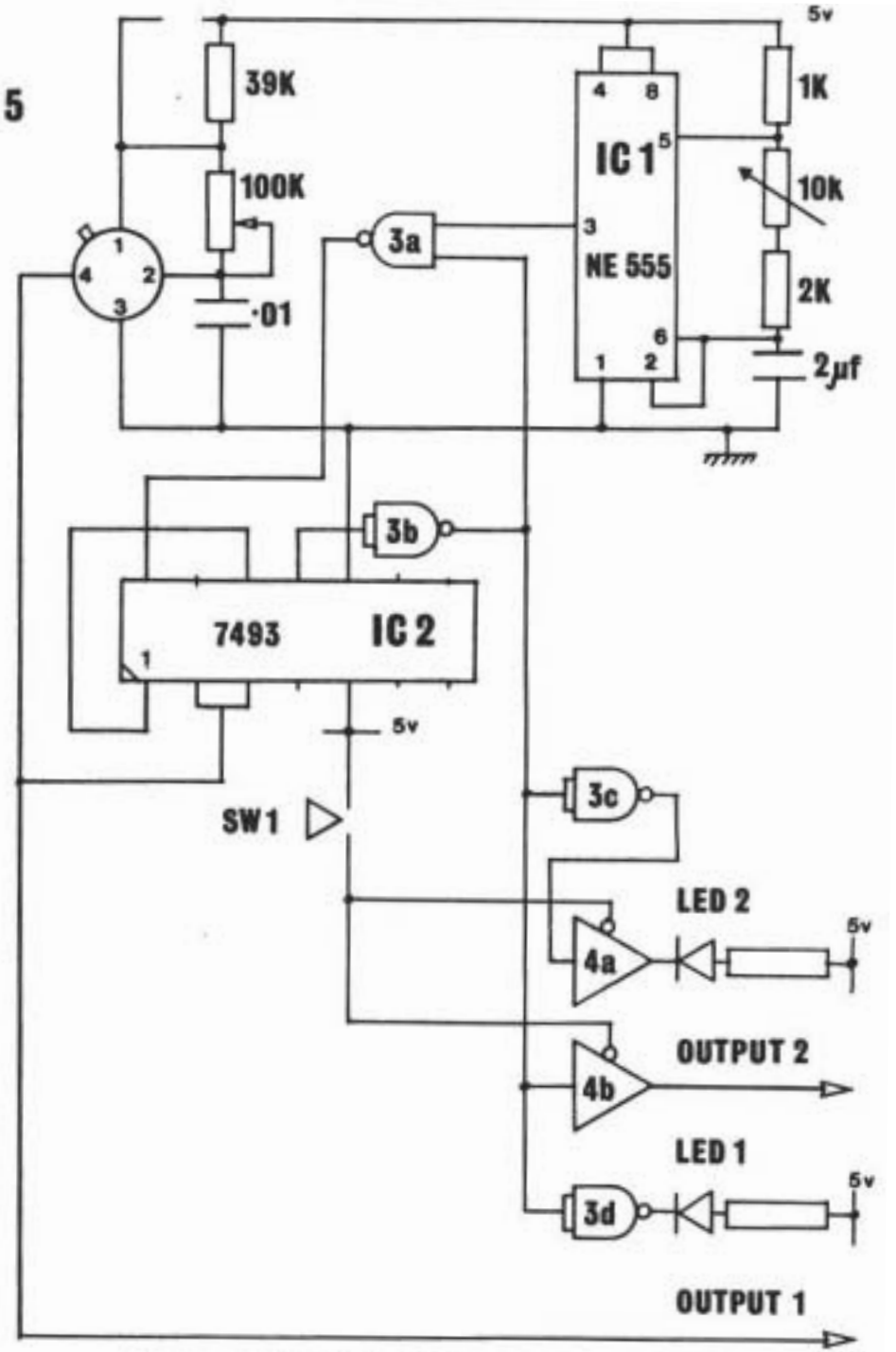
fig.1




NAS-GRAPH IMAGES

fig.2

fig. 5




LIGHT-PEN INTERFACE

```
ZEAP Z80 Assembler - Source Listing

                0040 ;*****************************************
                0050 ;THIS PROGRAM FINDS THE PEN LOCATION BY
                0060 ;SEQUENTIALLY LOADING THE SCREEN WITH
                0070 ;IMAGES AND THEN LOOKING AT THE PEN STATUS
                0080 ;TO SEE IF THE RASTER IS ILLUMINATED UNDER
                0090 ;THE PEN.VBLANKING IS TAKEN TO BIT 6 OF
                0100 ;THE INPUT PORT,THE PEN OUTPUT TO BIT 7.
                0110 ;*****************************************

                0140 ;EQUATES FOR THE PROGRAM
1A8A 080A       0150 ORIGIN EQU  080AH       ;SCREEN TOP LEFT
1A8A 0010       0160 MARGIN EQU  16          ;DON'T TEST MARGINS
1A8A 00C1       0170 IMAGE1 EQU  0C1H         ;PIXEL TOP LEFT
1A8A 00C2       0180 IMAGE2 EQU  0C2H         ;PIXEL MID LEFT
1A8A 00C4       0190 IMAGE3 EQU  0C4H         ;PIXEL BOTTOM LEFT
1A8A 00C8       0200 IMAGE4 EQU  0C8H         ;PIXEL TOP RIGHT
1A8A 00D0       0210 IMAGE5 EQU  0D0H         ;PIXEL MID RIGHT
1A8A 00E0       0220 IMAGE6 EQU  0E0H         ;PIXEL BOTTOM RIGHT
1A8A 0020       0230 SPACE1 EQU  20H         ;SPACE
1A8A 0010       0240 LINES  EQU  16          ;NO.OF LINES TO TEST
1A8A 0030       0250 COLUMS EQU  48          ;NO.OF COLS TO TEST

0C80            0280         ORG  0C80H            ;WORKSPACE RAM
0C80            0290         ENT

0C80 3E0C       0310         LD   A,0CH             ;CLEAR THE SCREEN
0C82 F7         0320         RST  30H

0C83 3E4F       0340         LD   A,4FH             ;SET PORT A TO BE
0C85 D306       0350         OUT  (6),A            ;IN INPUT MODE

0C87 210A08     0370 START   LD   HL,ORIGIN    ;POINT TO TOP LEFT
0C8A 0610       0380         LD   B,LINES          ;16 CHARACTER LINES
0C8C C5         0390 NEWLNE  PUSH BC              ;SAVE THIS

0C8D 0630       0410         LD   B,COLUMS         ;48 COLUMS

0C8F 36C1       0430 SCANST  LD   (HL),IMAGE1 ;LOAD FIRST IMAGE
0C91 CDB70C     0440         CALL PENTST          ;IS THERE A HIT?

0C94 36C2       0460         LD   (HL),IMAGE2
0C96 CDB70C     0470         CALL PENTST

0C99 36C4       0490         LD   (HL),IMAGE3
0C9B CDB70C     0500         CALL PENTST

0C9E 36C8       0520         LD   (HL),IMAGE4
0CA0 CDB70C     0530         CALL PENTST
```

```
0CA3 36D0      0550          LD    (HL),IMAGE5
0CA5 CDB70C    0560          CALL  PENTST

0CA8 3620      0580          LD    (HL),SPACE1      ;ERASE IMAGE5
0CAA 23        0590          INC   HL               ;NO HIT,THEN NEXT PIXEL
0CAB 10E2      0600          DJNZ  SCANST           ;DO FOR 48 COLUMS

0CAD C1        0620          POP   BC               ;RETRIEVE LINES TO DO
0CAE 111000    0630          LD    DE,MARGIN        ;MISS OUT THE MARGIN
0CB1 19        0640          ADD   HL,DE
0CB2 10D8      0650          DJNZ  NEWLNE

0CB4 C3870C    0670          JP    START            ;TRY AGAIN


               0700 ;*****************************************
               0710 ;ROUTINE TO TEST FOR A BLANKING PERIOD
               0720 ;AND A RASTER SCAN HIT.
               0730 ;*****************************************

0CB7 DB04      0750 PENTST   IN    A,(4)        ;TEST THE PORT!
0CB9 CB77      0760          BIT   6,A              ;IS THE BLANKING LOW?
0CBB 20FA      0770          JR    NZ,PENTST        ;NO,THEN LOOK AGAIN

0CBD DB04      0790 BLANK    IN    A,(4)         ;TEST UNTIL THE SCREEN
0CBF CB77      0800          BIT   6,A              ;SCAN COMMENCES
0CC1 28FA      0810          JR    Z,BLANK

0CC3 DB04      0830 PENHIT   IN    A,(4)        ;NOW LOOK FOR A HIT
0CC5 CB77      0840          BIT   6,A
0CC7 280B      0850          JR    Z,RTN            ;NEW BLANK PERIOD?

0CC9 CB7F      0860          BIT   7,A              ;HIT? TEST UNTIL A NEW
0CCB 28F6      0870          JR    Z,PENHIT         ;BLANKING PERIOD

0CCD D1        0900          POP   DE               ;REMOVE RETURN ADDRESS
0CCE DF        0910          RST   18H              ;PRINT LOCATION ON
0CCF 66        0920          DEFB  66H              ;SCREEN

0CD0 DF        0940 AGAIN    RST   18H              ;PRESS ANY KEY TO
0CD1 62        0950          DEFB  62H              ;START AGAIN
0CD2 30FC      0960          JR    NC,AGAIN

0CD4 C9        0980 RTN      RET
```

**fig. 3**

# 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 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

## Doctor Knowall's Glossed-over Glossary

| | |
|---|---|
| ARRAY | Exclamation of joy when a row of Bytes are found (see below) |
| ASCII | Alcoholic drink from Japan, thought to be the main cause of dyslexia among programmers |
| BINARY | A place where Ascii is consumed |
| BYTE | A Binary that serves hot food |
| BIT | Tart, usually found in Bytes |
| BUG | Sorry, that should be MUG - something that Ascii is served in |
| CURSOR | Someone who, having bought a *VC board, has found out that it would have cost less to interface a Spectrum |
| DYNAMIC RAM | Cudgel for beating absent-minded programmers over the head with |
| HARDCOPY | A painful condition suffered by those without peripherals |
| HEX | Demoniacal spell. Also a Binary with no Ascii |
| I/O | A tune from Snow White |
| MEMORY | NASBUG !! |
| PEEK | Shy machine code programmer |
| PIXEL | Garden ornament, usually found in the gardens of Peekers |
| PERIPHERALS | Pity the man without them (the author has only one) |
| SYNTAX | Levy plced on those who frequently nip down to the Byte for a Bit |
| XMAS | A period of intensive use of computing terms and computers |

Confused ? .... GREAT !!

# BruteSort

by J. McAfee

There are probably better sorts around for the Nascom, but either they are being kept a dark secret, or I have missed their publication (not a difficult trick in Saudi Arabia), so I decided to write my own. Being an ardent exponent of the art of coarse programming, my code owes nothing to elegance and even does a couple of naughties, hence the name, BruteSort.

I am not sure of the sorting jargon but I think it's a modified Shell-Shuffle-Shuttle sort (or have I just invented a test of soberiety? Try saying that when you have had a few). The logic of the thing was taken from a BASIC listing in a magazine and there are probably more efficient algorithms around but I've found it adequate for my needs. No! Lets be honest, it impresses the hell out of me.

I'll leave it to others to obtain accurate timings, but going by the kitchen clock, BruteSort takes about 3 seconds to sort a 1000 element array of random strings. This sorted array can then be sorted into descending order in about 1.6 seconds while it takes a dazzling 0.9 seconds to resort the array with one element out of sequence. All these timings were on a machine running at 4MHz.

The code sits at A000H and is 12BH bytes long, so remember when initialising BASIC that 'Memory Size ?' must not exceed 40960.

There are several ways in which BruteSort can be used. Lets take the following string as a typical example in the array dimensioned DA$(500).

```
"2540208BLOGS J        LONDON        SORTER          "
```

This string is a concatenation of four fields:-

| | | |
|---|---|---|
| 1. | ID No. | 8 characters |
| 2. | Name | 15 characters |
| 3. | Home | 13 characters |
| 4. | Occupation | 15 characters |

To sort the entire array by name (omitting the zeroeth element, but BruteSort can handle it if you wish)...

```
A=USR(0)DA$(1),500,9,23,A
```

...will do the trick provided that A does not evaluate to -1. If it does then a descending sort will be returned.

Let's say, for some unfathomable reason, we wish to sort

from element number 125 to element number 327 inclusive by ID No., then we can use something like...

```
100 F=125:L=327-125+1:D=1
110 A=USR(0)DA$(F),L,1,8,D
```

Double dimensioned arrays are not supported but...

```
A=USR(0),DD$(1,1),50,1,10,S
```

will sort the array from DD$(1,1) to DD$(50,1) without reference to any other dimensions. ie. the elements DD$(1,2) to DD$(50,2) will remain in their previous sequence. Perhaps BruteSort II will properly support multi-dimensioned arrays but I can visualise the 1000 element sort dragging on for quite a bit longer with that enhancement.

The penalty paid for having such brief code is that there is no validation of the parameters and it's also fussy about syntax (extra blanks, etc.). Its up to the BASIC calling program to check that all parameters are within valid range. Another limitation is that all the elements in the array should have characters filling the entire sort field. Sorting field 1 to 10 on a 6 character string may have unpredictable results and an array with a null string in the sort range will most likely get corrupted.

With some more understanding of how BASIC works the routine could be polished up a bit to remove these restrictions. Who is going to be first with SuaveSort?

I would be glad to hear from any other Nascomaniacs in Saudi Arabia (or anywhere else for that matter). I can be contacted via

J.R. McAfee
c/o Saif Establishment
P.O. Box 927
Al-Khobar
Saudi Arabia

```
100 REM DEMOSORT  DEMO PROGRAM FOR BRUTESORT
110 REM
140 A=DEEK(-24576)+DEEK(-24500)+DEEK(-24400)
150 IF A=-36794 THEN 230
160 Y=32768:GOSUB 1120:REM LOAD BRUTESORT
170 REM
190 REM TOP OF MEMORY MUST HAVE BEEN SET TO
200 REM A000H ON BASIC INITIALISATION
210 REM
230 CLEAR 50000:DOKE 4100,-24576:REM A000H
240 TT$="UNSORTED <--> SORTED"
250 D=0:REM - 1 WILL RETURN DESCENDING SORT
260 REM
270 CLS:SCREEN 15,7:INPUT"How many words ";X$
280 NUM=VAL(X$):IF NUM<1 OR NUM>1000 THEN 270
```

```
290 CLS:SCREEN 9,8
300 PRINT"Generating"NUM"random words"
310 DIM A$(NUM),B$(NUM)
320 FOR I=1 TO NUM:A$="":FOR J=1 TO 10
330 A$=A$+CHR$(INT(RND(1)*26+65))
340 NEXT:SCREEN 19,6:PRINT I
350 A$(I)=A$:B$(I)=A$:NEXT
360 REM
380 REM
390 REM SORT ROUTINE IS CALLED USING USR JUMP
400 REM IN THE FORM X=USR(Y)X$(F),NO,SF,EF,DIR
410 REM WHERE:-
420 REM
430 REM X AND Y ARE DUMMIES, THOWN AWAY BY
440 REM THE ROUTINE
450 REM
460 REM X$() - SUBSCRIPTED VARIABLE MAY HAVE ANY
470 REM VALID NAME. NOTE, NO SEPARATOR BETWEEN
480 REM USR CALL AND VARIABLE NAME
490 REM
500 REM F - NUMERIC EXPRESSION REPRESENTING THE
510 REM FIRST ARRAY ELEMENT TO BE SORTED
520 REM
530 REM NO - NUMERIC EXPRESSION REPRESENTING THE
540 REM NUMBER OF ELEMENTS TO BE SORTED - MUST
550 REM BE GREATER THAN OR EQUAL TO F
560 REM
570 REM SF - NUMERIC EXPRESSION REPRESENTING THE
580 REM START OF THE FIELD WITHIN THE STRING
590 REM
600 REM EF - NUMERIC EXPRESSION REPRESENTING THE
610 REM END OF THE FIELD WITHIN THE STRING,
620 REM MUST BE GREATER THAN OR EQUAL TO SF
630 REM
640 REM DIR - NUMERIC EXPRESSION REPRESENTING THE
650 REM DIRECTION OF SORT. IF DIR EVALUATES
660 REM TO -1, A DESCENDING SORT IS RETURNED.
670 REM ANY OTHER VALUE RETURNS ASCENDING SORT
680 REM
710 REM
720 SCREEN 19,10:INPUT"READY":O$
730 A=USR(0)A$(1),NUM,1,10,D
740 CLS:SCREEN 15,7:PRINT"SORT COMPLETED"
750 REM
780 REM
790 SCREEN 15,11:INPUT"PRESS ENTER":O$
800 CLS:FOR I=1 TO LEN(TT$)
810 POKE 3025+I,ASC(MID$(TT$,I,1)):NEXT
820 FOR I=1 TO NUM
830 PRINT"        "B$(I)"    "A$(I)"  ";
840 NEXT
850 END
860 REM  BRUTESORT
870 REM
880 REM
```

```
890 DATA -7711,11725,5103,-4845,1107,-12895
900 DATA -5761,-4837,1619,-12895,-5761,-10981
910 DATA 21485,-24312,32717,31721,-27695,2610
920 DATA -12895,-6090,16845,-12819,-5749,5093
930 DATA -19845,552,318,2866,417,2,17389,-24302
940 DATA 5153,2465,9086,28518,23533,-24314
950 DATA -4689,850,14339,-4631,4683,2977,30731
960 DATA -13647,-4579,17389,-24302,5153,2465
970 DATA 9086,28518,4130,8865,-24308,3114,-4703
980 DATA 4187,-20575,21229,3618,-4703,4187,6561
990 DATA 23533,-24316,10537,-6887,-7715,10981
1000 DATA -24306,10537,-6887,-7683,19437,-24312
1010 DATA 9086,28518,-5367,32481,26147,2415,2618
1020 DATA 18337,-27110,-15654,-15712,-24394,4899
1030 DATA -3056,318,2849,-20831,-4158,-15456
1040 DATA -24374,8623,-24309,-15698,-24337,28381
1050 DATA -8960,358,32509,-8960,119,32509,-8959
1060 DATA 375,30205,-768,372,10927,-24306,23533
1070 DATA -24304,21229,31442,-4704,3163,5025
1080 DATA 21485,-24308,1578,-20575,21229,28882
1090 DATA -15456,-24493,4865,99,0,264,100,96,1,2
1100 DATA 0,1,4,13,40,121,364,1093,3280,9841
1110 DATA 29524,-32768,-28673
1120 FOR I=40960 TO 41261 STEP 2:READ X
1130 DOKE I+2*Y*(I>=Y),X:NEXT:RETURN
```

# Corn Corner

by Cresby

Naspen Extension from B&L (£15 disk or tape)

I only have the tape version of this but find it solves some of the gripes of Naspen. Generally, it is claimed to be faster in most commands but more spefically on the time consuming block inserts, etc. A nice touch is character A0H which displays as a triangle but prints as a space to keep blocks on one line. Eg. equations. New commands include:-

'B' block delete

and additions to the find command like:-

'r' replace
'g' global (replace all) - designed to help bad spoolers lyke mee

'T' top kill - deletes all before cursor

tHERE IS A CAPS LOCK WHICH IS HANDY

'U' can unwrap the display which I prefer to use on difficult formatting problems

'J' command can now fail safe, and

'Q' will overide the stop print character

The cursor stays on screen in nearly all situations now, which is an improvement, but the cursor also goes past the end of the line in 'U'nwrap which makes it hard to see where the insertion of spaces is.

The extension allows you to insert almost the whole character set so that control characters can be dumped to the printer, but, in desperation at Naspen itself (which recodes some keys) I have had to modify buffer locations in Nas-Sys starting around 1021H to get the right characters quickly. Even then, codes like 00H and FFH must be avoided which precludes some data bytes to my printer set-up facility. The real plus was the re-discovery that 'i' allows insertion of 'ESC' which, with my printer, invokes all the fancy fonts and highlights.

The disc version is by and large the same but sorts out all the disc write commands transparently except that it asks for a filename after the 'W' command.

On Heat.

Rewarming a cold start is not for the faint-hearted, maybe my hardware problems are confusing it all, but I have not succeeded (is my meory RAMshackled?). In fact, I have several times killed text (with and without extension) totally but refound that text in Nas-Sys by filling suspect locations with 20H, removing the stop FFH character at 1021H and then putting the correct end-of-file address in 101BH and 101AH (high byte in high address). The location 101CH can be ignored it seems. Tabbing upwards from 1021H (in 0EH blocks) it is obvious when the end-of-file appears. You will see about 8 FFHs or at least the end of your schpiel. Take it from me, you can reheat Naspen too often.

RAMbling on

What is the best way to psych a chip? I find that the hottest ones are generally candidates for replacement when searching for faults. Using one of the more sensitive parts of the body as a thermometer gives rise to in-jokes about kissing RAMs, not to mention BUS drivers. Jest not, it works. Experimentors are advised to switch off first - 4MHz tastes vile!

PROMulgating

I don't recall seeing any review of the I/O PROM programmer but having used one for six months, I am full of praise. In fact, the only real hassle was having to solder one cable end and the PROM personality plug. The software is the operating manual almost, which is how programs should be - at man's service. It covers 2708, 2716 and 2732 PROMs but with a little guile 2764s and 27128s can be accomodated. Points to note are the need for 21 volts not 25. If you need the faster 2732A, beware of the same lower programming voltage requirement.

Chipping in

CMOS versions of everything are pouring out from all manufacturers. National have a Z80. They, Motorola and RCA have many 74HC series parts fast enough and beefy enough to do most TTL tasks. Projects planned for your intrepid reporter include using CMOS 2716s, first to reduce power to 10mW per PROM and also to keep things cooler. After that, the hottest parts are the Z80 and bipolar PROMs. Another project is to try using X2816s, they are 5 volt only EEPROMs made by Xicor. If you have £20 plus VAT (Vear and Tear), professional users can get them from Micro Call, Thame, Oxon (tame oxen indeed). The nice touch is that they sport on-board latches which means that providing you only write to them once in 10mS they behave like 2K, non-volatile RAMs. I have experimented with them but the first try failed. We are close though, so stay tuned to this channel folks. Unless anyone has tried using 2816 or 27c16s, in which case they can write to Cresby (in a plain, brown envelope).

About software standards

1) I am writing in response to your plea for comments on Mr. O'Farrell's article re:standardised extensions to Nas-Sys. I agree entirely with his motives in providing standard calls for add-ons but I would like to suggest a few alterations to his scheme

a) Graphics boards

Rather than separate SCAL calls for line, circle, etc., would it not be better to use one SCAL GRAF and load up the arguments and/or registers?

b) Floating Point Handling

Whilst floating point routines have been published in many places, they are still not widely available to everyone.

I believe extended calls to Nas-Sys should include floating point arithmetic and functions as well as perhaps extended precision integers. In doing this it will turn Sys into a 'software bus' onto which can be tacked all manner of language and arithmetic applications.

c) Nas-Sys (COM) Naughties

Many moons ago, in the days when Z**1's weren't around, there was an article in INMC (issue 5 or 6 I think) entitled 'Nas-Sys Naughties'. The purpose of it was to ward off people making direct calls to Nas-Sys as they had done with the Nas-Bugs. May I now suggest another naughty - Direct Keyboard Scans!!!

Many games programs and certain languages, ie BASIC and BLS Pascal, scan the keyboard directly. (BLS Pascal even makes a direct call to Nas-Sys). This practice is all very well and good on a Nascom but it is perfectly possible to run Sys on a Gemini by fudging the video RAM (see Richard Beal's program in 80-Bus News).

So, in conclusion, I would repeat my wish that Sys should act as a 'software bus' running the 'hardware bus' we all know.

R.D.E. Brown,
London SW7

2) The proposal by Mr. O'Farrell in the Nascom Newsletter Vol.3, No.1 was a very good idea. A lot of people have already added new routines to their Nas-Sys. I know that Polydos uses its own set of new routine numbers and I think that Nasdos does the same. It would be nice if we could get some standard out of it. But I think that there is another problem. Where in memory should these extra routines be placed? Somewhere in the top of memory most people would say, but I think that this is only half a solution. Why not place the whole monitor on another page? This is what I have done. Nas-Sys is placed on a G803 EPROM board (but I think that it should be a simple matter to modify the CPU-board such that it could be paged in and out). The monitor pages itself in and out when Nas-Sys routines are

called. In this way you can have more than 60K of RAM space for user programs (in fact the monitor occupies less than 300 bytes of RAM space) regardless of how many new routines you have added to Nas-Sys.

S.D. Pedersen,
Copenhagen,
Denmark

The simplest clock of all ????
On many Nascoms, the PIOs are not used. If you have one of those unused PIOs, here is a description of how to make a timer out of it.
You only have to do the following:-
- connect the ASTROBE of the PIO (PL4 pin 11) to ground
- connect TP1 to PIO A7 (PL4 pin 24)
- type in the following program and execute it at 0D10H

```
PIOA    EQU 4
PIOAC   EQU 6
MRET    EQU 5BH
INTVEC  EQU 0C80H  ;interrupt vector
TIME    EQU 0C82H
        ORG 0D00H
;interrupt routine - increment time
INTROU  PUSH HL
        LD HL,(TIME)
        INC HL
        LD (TIME),HL
        POP HL
        EI
        RETI
        ORG 0D10H
;initialising routine
INIT    DI
        IM 2 ;set mode 2
        LD A,INTVEC/256 ;most significant byte
        LD I,A          ; of INTVEC to I-reg
        LD C,PIOAC      ;PIOA's control port
        RCAL PUT        ;Fill it with control bytes
        DB 06           ;number of bytes to PIOAC
        DB 4FH          ;if it is waiting on a byte
        DB 11001111B    ;mode - control
        DB 10000000B    ;bit 7 in, others out
        DB 10110111B    ;enable PIO interrupt
        ;                'or' interrupt bits (doesn't matter
        here)
        ;                interrupt when high (doesn't matter
        here)
        ;                mask follows
        DB 01111111B    ;only interrupt on bit 7
        DB INTVEC&00FFH ;least significant byte of INTVEC
        LD HL,INTROU    ;address of interrupt routine
        LD (INTVEC),HL  ;into (INTVEC)
        EI
```

```
        SCAL MRET
PUT     EX (SP),HL
        LD B,(HL)
        INC HL
        OTIR
        EX (SP),HL
        RET
        END
```

Now you can read the time in the addresses 0C82H-0C83H and time differences can be measured by subtracting two time values. As TP1 gives 300Hz, you can measure time differences of up to 3.5 minutes with an accuracy of 1/300 of a second. You can reset the timer at any time, either from Nas-Sys (using the M command) or from BASIC (using DOKE). The timer cannot be used under Nas-Sys 1 as this is not interruptable.

S.D.Pedersen,
Copenhagen,
Denmark

XBASIC tip

A tip to all bone idle users of XBASIC who, having written their ultimate, high-speed data-handling routine which has lots of internal GOTO/GOSUBs, find that when they install it in their main program it runs very slowly because it is located way down the program, miles from the start, here is an easy solution:-

```
 10 program
.
.
30 GOSUB routine
.
.
9999 REM routine
10000 TEXTPTR=DEEK(&1006):HOLD 10000
10009 REM routine proper
.
.
10999 REM REM end of routine
11000 DOKE &1006,TEXTPTR:RETURN
```

You may say 'why not use MGE before the RETURN', well, it doesn't work, it terminates the program and returns to command level which is understandable.
I have used this method in a program which has a 'SORT' routine and two 'SEARCH' routines and all three run much faster. I timed the sort routine and it runs 28% faster using the above method.
Now, can anyone tell me how to set up a type-ahead buffer for XBASIC as the authors seem to have used their own keyboard routine and I haven't, as yet, managed to find it?

I.P. Coole,
London SW7

# Review of the Belectra Arithmetic Processor Board.

by IJC

A common problem with micros is their lack of speed and accuracy when it comes to calculations. For the average user, the fact that a figure is only accurate to the 6th decimal place may be rather trivial but when it comes to complex graphics calculations or financial formulae the effects can be compounded to make the figures unacceptable. Also, when calculating trigonometric functions in a program, (commonly used in graphics plots again) speed is very important. The time taken to calculate the LOG of a number is horrendous when compared to the rest of the program. One way round this is to use look up tables but this uses up a lot of memory and isn't very accurate.

The answer, as far as the Nascom is concerned, has come in the form of the Belectra HSA-88B Arithmetic Processor board.

The board (or at least the one I was loaned for review) arrived well packaged and swaddled in silver paper. I also received a rather thin manual and a data booklet on the AM9511A processor (the heart of the board). The board isn't a full size 8x8 board but it is blue. (This seems to be coming the Nascom industry standard colour.) The board doesn't look as polished as some of the other Nascom boards available but it was adequate and who cares if it doesn't look pretty anyway.

Even for its small size, the board is only sparsely populated and at £199 you may wonder what you are paying for. The answer is, the AM9511 processor. It does account for a rather large chunk of the total cost.

The board is supplied ready wired to a default port and with a default clock speed which most users should not need to change. However, by means of wire links, you can select which two ports you want to address the board on (80/81H, 90/91H, A0/A1H, B0/B1H, C0/C1H, D0/D1H, E0/E1H or F0/F1H) and you may have to change the link on the clock speed if you are running a 2MHz Nascom. Clean, simple and straight-forward. The board does not support DBDR so it is incompatible with the Nascom 1 which *seems a shame with so many still being in use.*

The arithmetic processor works totally independently of the Nascom Z80 and is accessed via 2 I/O ports. To use the processor couldn't be simpler. The 9511 contains a stack, a command register and a status register. The data to be operated on is firstly pushed onto the stack (always least significant byte first), a command byte is sent to the command register (eg. 16-bit multiply - 6EH), the Nascom then waits for the arithmeic to finish (by testing the status port) and then it reads the result from the stack port again (most significant byte first this time). One example that they give in the manual is as follows:-

```
        LD C,STACK      ;register C points to 9511 stack port

        OUT (C),E       ;push operands onto 9511 stack
        OUT (C),D       ; (least significant byte first)
        OUT (C),L       ;
        OUT (C),H       ;

        LD  A,6EH       ;issue '16 bit multiply' command
        OUT (COMMAND),A

  WAIT  IN A,(STATUS)
        BIT 7,A
        JR Z,WAIT       ;wait until operation is finished

        IN H,(C)        ;pop result (DE times HL) back into HL
        IN L,(C)        ;(most significant byte first)
```

All very straight forward and simple to use.

The manual supplied is rather sparse. It tells you how to set up the board and the principles behind how you get it to work but it leaves you to the 9511 data book to give you a description of the different commands available and how to use them. It doesn't seem quite the right way about doing things but it does give you all the facts that you need to make the board work and start to earn some of it's worth.

The machine code programmer should find this board very easy to get along with and easy to use. For those of you who would want to use it in an applications type environment, Hisoft HP5 in CP/M format is supplied (but not with my review board because a) the Pascal was not availbale and b) I haven't got access to CP/M).

Using the processor from a high-level language will either require patching in the required routines or tansferring everything to machine code calls (USR). If the board is successful, I would imagine that patches for most commonly used high-level languages will be published. If, however, you need to use USRs, the speed gain by the processor will probably be lost in setting up the entry point and parameters.

In conclusion then, the HSA-88B board is a nice complement to the Nascom computer. A system comprising of a graphics card (AVC or other) plus the processor board should be capable of some quite impressive feats. It is a rather expensive board but if you want the arithmetic capability of the 9511, you are unlikely to get it much cheaper. Programming the processor is very straight forward and most people should find it easy to get to grips with. It does allow parallel processing to an extent (the HSA-88B can be calculating while the Nascom Z80 is drawing) but the addition of an interrupt facility would be very welcome. I think that on the whole, this is the only point missing from a very nice product.

# 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 LUCAS NASCOM

This months excuse for a short contribution is pressure of work! Once again I have had to postpone the article on function keys - anyone who is really impatient can obtain a listing of part of some software which uses this technique by sending me an SAE. Due to our planned expansion of te business we will shortly be appointing extra staff. We need experienced hardware engineers, system and applications software engineers, technical support engineers and a documentation engineer. If there is anyone out there who feels that he or she might meet the bill, please write to our Personnel Department.

This month sees the release of information on the Winchester disc system, described below. In fact a number of these have already been delivered to customers at the time of writing. Other developments are mainly by way of enhancements to existing software products. We are continuing to try to improve the standard of our documentation, both on new products and existing ones. In particular the release of NAS-DOS 2.1 also sees a new manual, which we hope will be found to be more 'user friendly' than its predecessor. All manual updates are now generally be accompanied by a pocket-sized summary card.

In mentioning new releases of existing software can I yet again make a plea for you to register your software purchases using the cards supplied with the products. This is in your own interests - special software upgrade offers and technical support will ONLY be provided to owners of registered products. From our knowledge of sales and registrations it is obvious that not everyone registers. There are no conditions attached or loss of any rights if you do register.

Returning to the question of documentation and general product presentation (particularly software) it is interesting to see how the microcomputer industry in general is now placing much more emphasis in this area. As more new computer users appear, particularly in business, software needs to become more polished and easy to use, with corresponding improvements in documentation. However, while not wishing to discourage a long needed move in that direction, I do have some reservations. There seems to be a danger that presentation of a product and documentation are becoming excessive - enormous manuals and ever more fancy binders supporting increasingly complex programs. Are there still at least some new users who really only require straightforward basic software which is easy to understand and can be explained fully in a manual of less than 100 pages? Any comments would be received with interest.

At the time of writing there is a special offer on the Lucas LX80 printer - 80 characters per second, 80 columns and while the offer lasts only £225. There are many special printing features, including expanded and compressed text, italics, overprinting, undrlining and graphics. This offer will be for a limited period.

Mike Hessey
Technical Manager

## THE WINCHESTER DISC

Winchester discs differ from the more familiar floppy disc drives in that their mode of operation is quite different, and the storage medium is sealed in the drive unit rather than being removable. Because the disc operates in a sealed environment, with no variation in the storage medium, the speed of access can be far faster than with a floppy disc, and the capacity is also much greater. At the present time we are offering three versions of the Winchester disc - 5M byte, 10M byte and 20M byte versions. These contrast with the 375K bytes of a single sided floppy disc and 750K of a dual sided disc. The disc is controlled by the computer in a different way to the floppy disc, and therefore a different controller card is fitted in the computer. This replaces the normal floppy disc controller card, but also provides a floppy disc controller on the same board. There is also a high speed communications facility on this board for use in networking or other applications.

The Winchester disc with its large storage capacity and fast access will be of particular interest to the business user with large data files, and to users of networks. Support for the Winchester disc is available through both the NAS-DOS and CP/M disc operating systems. In the case of CP/M the Winchester is incorporated in the system by running a special program (automatically if you wish), while in the case of NAS-DOS the Winchester version of the operating system should be specified when ordering.

## THE NAS-DOS DISC OPERATING SYSTEM

NAS-DOS is the disc operating system used by Nascom computers which provides compatibility with the simple tape operating system. A range of applications software and system software is availble, generally best described as cheap and cheerfull. None costs more than £70, and while lacking the sophistication of some of the more expensive programs for the CP/M operating system they have the considerable advantage of being easy to use and fast in operation. NAS-DOS was the first disc operating system released by us for the Nascom computers, and has always been the most popular. A new release of this operating system, Rev 2.1, is under test at the time of writing, and will be available by the time this article appears in print.

As in previous updates to NAS-DOS the latest release is fully compatible with previous versions, and represents refinements to the existing product rather than a major new development. While a special offer will be made to existing registered owners of NAS-DOS to allow them to upgrade to the latest release, we would suggest that for most people there will be little advantage in doing so. Details of the special upgrade offer will be released in the next issue of the magazine.

The changes incorporated in Rev 2.1 of NAS-DOS are as follows:

1. Standard version supports 6mS track to track disc drives - the half-height units now being used by Nascom. Special versions of the operating system to suit older, lower speed, drives can be supplied to order.

2. Improved error messages

3. File locking

4. File identification by terminal number when using the NAS-NET networking system.

5. Formatting removed from the operating system to make way for other improvements, and also to eliminate the possibility of faulty assembler programs managing to re-format the disc!

6. New and improved utility programs on disc, now including a disc formatting program. This disc will be supplied with NAS-DOS in future, rather than at registration time, since alhtough the existing utilities are not essential to use of the system, disc formatting is!

7. New and expanded manual.

A special version of NAS-DOS for use with the Winchester disc is also available. This supports the Winchester as the main disc drive, and uses utility programs to read files from floppy disc where this facility is required (eg for program loading and transfer).

## NAS-NET NETWORKING SYSTEM

NAS-NET is a networking system which allows several Nascom computers to be linked together to share the resources of a central disc and printer. All the satellite computer (1 to 31 are allowed) appears to have a normal NAS-DOS disc operating system, but automatically accesses the disc of the master computer when disc access is required. The system has been proved in customer sites for over 12 months, and has proved extremely popular, technically and from the point of view of its low price and genuine availability. Schools and other educational establishments in particular have been major customers.

Rev 2.1 of NAS-NET has now been released. Again it is fully compatible with the previous releases, and upgrading to the latest issue is not necessary for existing users. There are a number of internal improvements, but the most immediately obvious to the user is that files stored on disc are now given a 'terminal number', and are accessible only by that terminal number and the master computer. The terminal number is specified by the user on joining the network, and is not a function of a particular piece of computer hardware. Some changes have also been made to the way that access is gained to the central printer via 'spool' files on the disc. These too are now automatically identified by terminal number, and printing of these can be initiated from the satellite computer.

Most NAS-DOS applications software is now available for use with NAS-NET. In practice the difference from the standard versions of the software is limited to the printer access - in the network version this will automatically make use of the printer on the central computer. Software now available in this form includes the NAS-PEN and SPEX word processors, the NAS-CALC spread sheet program and the MANOR database manager.

The new Winchester disc mentioned above can now be used with NAS-DOS, although a special version of NAS-DOS must be specified at time of purchase. In a networking system the effects of the increased storage capacity and fast storage and retrieval of data are particularly dramatic.

## CP/M NEWS

The CP/M disc operating system has generally proved most popular with those who wish to run standard business applications software. Both CP/M and the programs to run with it are generally significantly more expensive than the corresponding NAS-DOS versions. A very wide range of software is available to use with CP/M from many sources. If you are buying software from some other source do be careful to ensure that the disc is supplied in the right format - most suppliers should be able to provide Nascom format. A range of the more popular software is available from oursleves through the usual dealers, and this is of course in the correct format and will normally also have been configured for the correct terminal facilities. There are differences in the single and double sided disc drive formats, and although you can access discs easily via a utility program (SINGLE or DOUBLE) it will generally be more convenient in you specify the appropriate version at the time of ordering.

The most popular languages used when programming with CP/M are Nascom Extended BASIC and Microsoft's BASIC 80 (also known as MBASIC). Both are available through Nascom dealers. Easily the most popular applications programs are Sorcim's Supercalc and MicroPro's Wordstar, again available through your dealer. We can also supply, amongst others, DBASE II , FMS and Datastar, which are database programs, Mailmerge, Spellstar, Calcstar and Datastar from the same stable as Wordstar, the various financial programs which make up the FINAS/ACCENSE suite and the NEWSPAC Newsagents package. The range is steadily increasing, so consult your dealer if your requirements are not on this list. Soon to be released is a very powerful low-cost computer aided design and drawing package, known as LOTTI, which is mentioned elsewhere in this newsletter.

Sorcim have recently released a new and improved version of Supercalc - Supercalc 2. This contains a number of improvements over the previous version, and a new manual is provided. This is a much larger manual than previously, and more thought seems to have gone into its structure. Overall a worthwhile improvement to what was already an extremely fine product.

MicroPro have also re-vamped their products, with Wordstar Release 3.30 and corresponding MailMerge and Spellstar programs. New documentation here too, properly printed at last and very methodically and well laid out - although I don't feel the 'cartoons' really add anything of value. The total volume of the documentation is rather alarming, particularly if you get Reportstar and Datastar at the same time! All the manuals are in the same, A5 (half A4) size, and are in cloth bound ring binders in cloth bound card sleeves - very impressive in appearance. A new product to me is StarIndex which was included in the documentation. This looked a very good way of preparing indexes, particularly when modifying manuals moves the page numbers. Unfortunately although the manual was present in both sets which we

ordered, there was no disc! Further comments on this product next time, I hope. Overall again we have an 'industry standard product being improved further, with particular attention to presentation.

A number of improvements have been made in our implementation of CP/M 2.2 - referred to as our implementation Rev 2.3 (sorry about the confusing terminology). In particular a compressed version of the text generation used by the display card (AVC) is available when graphics is not being used. This allows a maximum CP/M user area of 58K, rather than the 55K of in the previous version. When using graphics, however, the larger version of the support software is still required. A new manual accompanies the latest version, describing the changes.

The availability of the large capacity and fast access Winchester discs described above will be of particular interest to many CP/M users. Not only can larger volumes of data be handled (almost esential for some of the financial packages) but also the speed of operation of many of the programs is dramatically increased.

Another feature which can be used to advantage by CP/M users is 'virtual disc . This allows additional memory fitted to the 256K memory card now supplied with most CP/M machines to be treated as a disc, which of course can be accessed very rapidly. Support for this option is now included with CP/M as we supply it. Those who require this facility but were not supplied with it at the time of purchase, and have registered their CP/M ownership, should contact their dealer or our Warwick office.

## THE BEST OF BOTH WORLDS - NAS-DOS AND CP/M

You do not have to make a decision to use only one of the two operating systems NAS-DOS and CP/M. In an Application Note (AN-0011) we described how to fit a switch to permit selection between the two operating systems. Since it is necessary to alter the memory arrangement in the computer between the two systems the switching arrangement is not as simple as you might suppose - the original note required the use of a 7 (seven) pole switch and a considerable amount of wiring. A better solution was worked out by B & L Microcomputers who designed a small PCB to fit in the computer which used relays for the switching, reducing the external wiring to an on/off switch. We now have an even simpler solution using a PCB fitted with only one chip, a resistor pack and some diodes. A simple on/off switch again controls which operating system is selected. This board is available, with full fitting instructions, from your dealer, or your dealer will fit the unit for you for a small charge.

Both NAS-DOS and CP/M can be organised to use only half of the Winchester disc, so that in a switched system you can store both types of file. Files are not of course compatible between the two systems.

## ANOTHER OPERATING SYSTEM - UCSD !

While CP/M has become the most widely used operating system on microcomputers there are a number of other systems which are not limited to implementation on a single make of computer. One of the

best known of these is the UCSD system, a particular feature of which is to maximise the portability of software between different computers. UCSD can be run on a number of much larger computer systems as well as on 16 and 8 bit microcomputers, which illustrates its flexibility.

UCSD is most frequently referred to with its implemetation of the Pascal programming language, but in fact other languages such as BASIC and Fortran are also available. A certain amount of applications software is also available.

UCSD has been implemented on the Nascom by Mike York, and we have recently taken delivery of a copy. UCSD is completely new to me, and the previously mentioned shortage of time has prevented any serious usage yet. However, it looks very interesting and I look forward to using it and commenting again in the future. Initial reactions are that it is very different from the other operating systems on the Nascom, and requires rather a lot of disc grinding to get started! It is quite easy to install - replace NAS-SYS (or the CP/M boot EPROM) with Mike's ingenious ALLBOOT EPROM, fit a CP/M MD PROM and header (if not already fitted) and away you go. You can also run it in a switchable system with NAS-DOS using the switch unit described above. A drawback for some users will be the price - typically several hundred pounds for the operating system and a language.

For further details contact Mike York on 01-874-6244.

## THE MANOR DATABASE MANAGEMENT PROGRAM

MANOR is a versatile, low cost data management program designed to run with computers fitted with the NAS-DOS operating system. In the 18 months or so of its use it has proved one of the most popular of the NAS-DOS applications programs - perhaps because there are so many uses for a program of this type. The current release is Rev 6, and all registered owners should have a copy of this version, since a free update service is included covering a 12 month period. If you have not received the update, and you have registered the product, please write in giving the serial number, date and place of purchase. Although our system should catch those who register the earlier versions after the release of the new one, it does occasionally fail to do so.

One bug has been reported in the HINTON initialisation program of the current Rev 6 of MANOR. This only affects the CHange option when this is used to swap the fields round so that the first (or key) field is changed. The fix for this bug has the rather nice feature that not only does it now work (I hope!) but it is shorter and faster than the original. To make the corrections type the following:

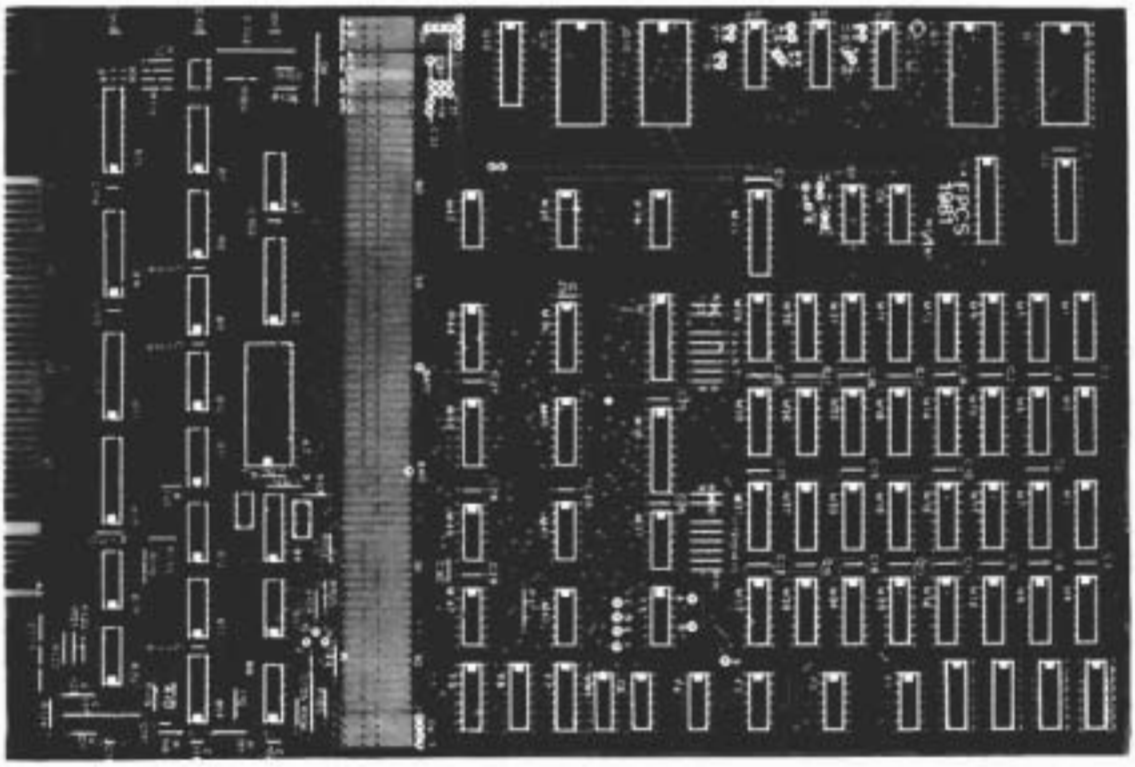
```
JL:HINTON
Z
6260GOSUB30:A$(I)=A2$(0)
6276FORI=1TONC-1
6274
6275
6277
JB
```

# LOTTI - COMPUTER AIDED DRAWING AND DESIGN

A Nascom computer fitted with the Advanced Video Controller (AVC) offers tremendous potential for use of graphics, as the many thousands of owners will now be aware. Use of graphics is made particluarly simple by the provision of a wide range of instructions for use in ROM BASIC or Nascom Extended BASIC (XBASIC), and now in Nascom Pascal. The AVC is fitted as standard to all Nascom and Lucas LX computers supplied with CP/M as original operating system, and this also provides in these machines a 25 line by 80 column text display. Up till now, however, to use the graphics it has been necessary to write your own programs.

There are many uses of computer graphics in the fields of drawing and design - technical illustration, layouts, general arrangements, detailed drawings, manuals, teaching aids etc. Typical minicomputer based computer aided draughting and design (CAD) systems can cost from around £30000 to £250000. With LOTTI Nascom Microcomputers has developed a full-feature, practical low cost design system. A typical system with all the hardware (including a plotter) and software would cost under £3000. The AVC is used to produce graphic output on one (colour or monochrome) monitor, while the second displays command and menu text generated from the normal Nascom video display. You can move a cross-hair cursor around the screen using the cursor control keys, and can specify points, lines, circles, arcs and polygons. Any shape may be stored on disc and then recalled, either for use in its own right or to become a part of a larger drawing. Magnification, rotation and slanting can be performed, and output can be sent to a matrix graphics printer and plotter.

LOTTI is designed for ease of use and versatility at a very low price. While the system cannot claim to rival £250000 sytems it does represent a useful tool for many applications. We are getting a terrific response to the information released on the product so far, and a few evaluation copies are out on trial. The first release is planned for October/November, with further developments to follow. The package will be on display at Compec, and it is hoped to include a detailed article in the next issue of Nascom News.



# 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%.**