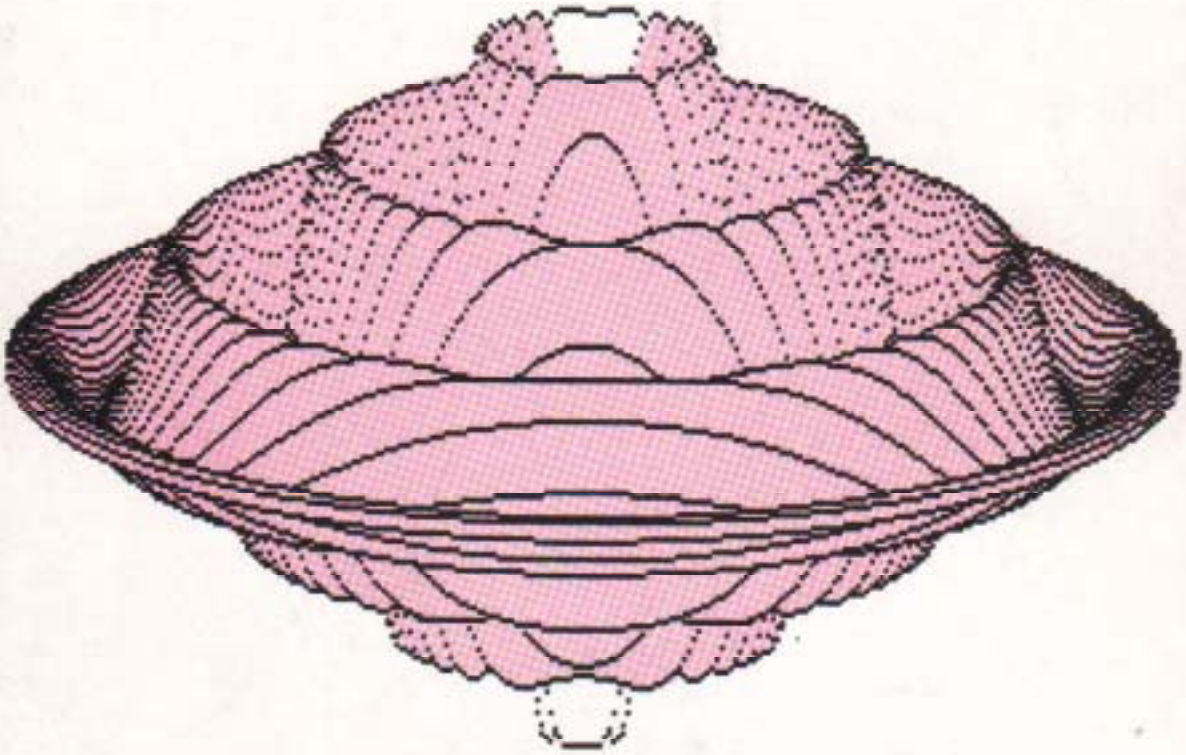
80-BUS NEWS

MARCH – APRIL 1983 VOL.2 ISSUE 2

- MINI-CASSETTE REVIEW
- UTILISING CCPZ
- HIGH SPEED CASSETTE INTERFACE
- FREE GRAPHICS SOFTWARE



£1.50

The Magazine for NASCOM & GEMINI USERS

March – April 1983

80-BUS NEWS

Volume 2. Issue 2.

CONTENTS
========

 Published by Interface Data Ltd and printed by Excelsior Photoprinting Ltd., High Wycombe.

SUBSCRIPTIONS

Annual Rates (6 issues)	UK	£9	Rest of World Surface	£12
	Europe	£12	Rest of World Air Mail	£20

Subscriptions to 'Subscriptions' at the address below.

EDITORIAL

Editor : Paul Greenhalgh Associate Editor : David Hunt

Material for consideration to 'The Editor' at the address below.

ADVERTISING

Rates on application to 'The Advertising Manager' at the address below.

ADDRESS: 80-BUS News,
Interface Data Ltd.,
Woodside Road,
Amersham, Bucks, HP6 5EW.

EDITORIAL

The Bad News

It had to happen, didn't it? I open my big mouth (in the last issue) and confidently state that as we had actually managed to produce several consecutive issues reasonably on time, the trend would continue. And what happens? A conspiracy, that's what, the net result of which is that this will probably now reach you approx. 6 weeks late (if not later).

The Good News

One consolation (as a result of this conspiracy) is that I have already managed to get a considerable amount of material together for the next issue, and all things being equal (which they rarely are) it should follow this in a matter of days.

More Bad News

In the Editorial of the last issue I stated that I was trying to get together information on the I/O ports used by 80-BUS/Nasbus compatible boards. The purpose of this is to draw up an I/O map of all boards past/present and (individual manufacturers allowing) future so that everyone can see what product clashes with what (which some unfortunately do), and what space is available to locate that 256 channel, polyphonic, quadrophonic noise-maker that you are currently building. I gave all of the manufacturers that I am aware of several weeks and at least **TWO** reminders to give me the necessary information, and still I have not received all of the information I requested. So to **Climax, EV Computing, Gemini Microcomputers, MAP 80 Systems** and **Microcode Processes** many thanks for your time and effort in returning the requested information to me so promptly and so completely. And as for the others, that is **Lucas/Nascom** and **IO Research - RASPBERRIES !!!** As a result of this apathy I am going to hold the I/O map over to the next issue ... so this is your last chance Nascom and I/O. If you don't send me the details that are, after all, in your own interest (of selling product) to have published correctly, I shall 'interpret' the ports used myself, and suggest that EVERY 80-BUS reader sends you a personal letter asking to verify that we got it right. OK?

More Good News

I am pleased to report that CCsoft have very gratiously given us their permission to publish the code for their Nas-Graphpac package. This links with the standard Nascom ROM BASIC to provide twenty additional commands for controlling the Nascom's 96x48 block graphics. In addition they have allowed us to reprint the System manual for the package, and Gemini (who have bought the rights to CCsoft's other Graphpacs) have also given us permission to reprint the Command manual. This means that we are able to bring you totally free, gratis and for nothing software which normally retails (on cassette) for £20 + VAT. All you have to do is type it all in!

Even More Good News

Last issue (I think) we included an advert for a book on 'Nascom BASIC disassembled' that said something to the effect of 'coming soon'. Well the publication of that has slipped too, and so we have decided, in our extremely infinite wisdom, to 'serialize' it starting in the next issue. This is a series that Gemini and Nascom owners alike should find of interest, as the listing is extremely thoroughly commented and should give the 80-BUS reader interested in machine code programming (all one of him) a great insight into many programming tricks that keep code as short and fast as possible.

LETTERS TO THE EDITOR

NASCOM SCREEN MOD.

Further to Dave Hunt's screed in the last issue of 80-BUS News where he answered someone's query regarding the graphics conversion for Nascom 2 to display full characters. The more complicated mod. is detailed below which also has the added benefit of stabilising the whole display screen (no more screen weave). Readers not familiar with the mod. may find this interesting.

Bend the pins of the indicated integrated circuits at right angles so they do not make contact with the socket and can have wires soldered to them:

IC53 pins 1 and 11
IC56 pins 5 and 6
IC68 pins 1 and 10

Now link the following points with thin wire:

IC68 pin 1 to 8
IC68 pin 10 to 16
IC53 pin 11 to IC68 pin 5 to IC56 pin 5
IC56 pin 6 to IC44 pin 11
IC53 pin 1 to IC68 pin 11

R. Dowling, Welling, Kent.

DISMAY

I am yet again dismayed when I receive your excellent magazine. I say dismayed because page upon page contains references to disks and things such as modifications to your BIOS etc.,etc.

My poor old Nascom badly needs financial help to enable it to emerge into the wondrous creature that it really is. The problem to be quite frank is that I have a wife and a money-gobbling house to support as well. Can you imagine the complaints when I finally gave in and purchased a printer - No, not a Qume Sprint 5 as you may have noticed. [Ed. - this looks like a Qume Sprint 5 to me! (Oops!)] One good thing in my favour was the purchase of an official NAS-CASE, which did at least give us back our dining room table, let the card frame go for a rest in the loft, and made the Nascom look attractive.

All this brings me to the point. There are many of us out here in Nascom/Gemini land who will probably never own a 5.4 MByte Winchester with an 800KByte floppy, so why bother rubbing salt in our wounds and raging on about them. It's bad enough going into H****S Computer Store and seeing D*** H*** leering at you from behind his green screen when you are cheerfully told that they don't have a Licon keytop or switch.

It is high time that some dealers and your magazine support the people who have spent sums of money in the past and still require support on their somewhat down market equipment until funds are available to buy the good stuff.

R.D. Madge, West Wickham, Kent.

[Ed. - It's very difficult to know just where to pitch the content of the magazine, and so the policy to date has been to publish material on different subject matter in proportion to that received.

One aspect of this, as I have said before (at the Brighton conference), is that the contents of this rag are written entirely by readers, and as such it is the keenest readers who contribute, and they fairly naturally seem to be the ones who have added varying quantities of expansion. I have to date endeavoured to keep the general (e.g. non-disk) content as high as possible, and shall continue to do so, as long as the relevant material keeps arriving.

Another viewpoint is to draw a parallel with other hobbies/professions. Many people buy car magazines, for example, NOT for the information on driving techniques, achieving economy etc. (i.e. how to use what they have), but for details of all the accessories that they may one day add, and for the reviews of the Ferraris, Lambourghinis etc. that they recognise they have little chance of ever owning or even driving.

With the Nascom, Gemini or Nascom/Gemini hybrid that everyone(?) reading this owns, or has access to, there is the in-built flexibility to expand it from the very basic (sic) model upto an extremely powerful and comprehensive system. This magazine will continue to inform the reader, as well as we can, what is available to achieve this expansion, and we will try to present objective evaluations of how well this is executed. At the same time we will continue to present information on how to make the most of the system that you currently own, regardless of its state of expansion.

That is, provided the material is sent to us.]

PEN MODIFICATION BUG

The journal of the CP/M Users Group (UK) Vol. 1 No. 6, May 1982 describes a number of bug fixes to some of the standard Digital Research CP/M utilities (e.g. MOVCPM, ASM, PIP, SUBMIT) and useful changes to BIOS and BDOS. (N.B. The patches to MOVCPM.COM need to be applied to addresses 100H higher up.)

While trying out some of these ideas I decided to check the change I wrote up in the 80-BUS News Vol. 1 Iss. 2, April - June 1982 headed PEN - 1 on pages 5 and 6. Unfortunately I found a mistake. The byte at address 296A should be patched from 18 to E2. It just goes to show there is many a slip between proving a mod. and writing it up for 80-BUS News.

Has anyone thought of a mod. to the Nascom 2 board which would enable the EPROM bootstrap loader to be ghosted out and thereby allowing the whole 64K RAM to be used for CP/M 2.2 with an IVC?

S. Willmott, West Drayton.

CLUB IN HANTS?

My lonely Nascom 2 has waited for three years for a club to spring up within the ALTON/CAMBERLEY area but thus far I haven't even seen any clubs formed for other (Gulp) machines.

I would like to hear from anybody in this neck of the woods who would like to exchange this or learn that! My current interests would be to link up Nascomers over the Welybone using the LetserP protocol or something like it; well this is the age of ET isn't it?

And now a puzzle:
Following power up, my RAM board memory contains random data which is correctly refreshed. I can write zeros into the memory but not one's. Anybody who has

analysed the circuit diagrams and thinks they understand them should (?) be able to pin-point the problem without any more info. than this - please!

Kevin Weatherhead, Sheen, Old Odiham Rd., Alton, Hants. GU34 4BW.

HISOFT PASCAL

In case Dr Dark hasn't got round to the review of Hisoft Pascal 4 promised in 80-BUS News, here are the results I obtained running the PCW Benchmark tests on this version of the compiler. Incidentally, because of the speed of the code, most of the loops were set to 30,000 and the resulting time divided by three, in the case of the Magnifier the routine was enclosed in an outer 100-times loop.

	Sec.		Sec.
Magnifier	0.22	Forloop	2.7
Whileloop	4.3	Repeatloop	3.7
Literalassign	3.3	Memoryaccess	3.3
Realarithmetic	14.0	Realalgebra	15.0
Vector	8.0	Equalif	6.0
Unequalif	5.7	Noparameters	3.3
Value	3.7	Reference	3.7
Maths	6.0		

No great accuracy is claimed for these figures, but they show some slight improvement on the NASPAS 3 timings, save for the REAL tests and the MATHS test where the old timings seemed to have been halved. All the tests were run on a Nascom 2 running at 4MHz with no Wait-states. The options to remove overflow checks etc. were used, and the most significant speed improvement came from invoking option C-, which removes the keyboard check at the start of each loop.

Having only had this compiler for about a month, I haven't yet been able to try out all the features - well actually the time has been spent on converting one or two very lethargic BASIC programs into PASCAL and enjoying the substantial increase in speed. During the course of this exercise one or two peculiarities came up which appeared not to be covered in the manual. These were:-

1) CASE Statements. The semi-colon after the last statement in the case-list must be ommitted.

2) RESET & REWRITE. The disk file name can be a variable,permitting filename entry from the screen, but it must be formatted as stated in the manual.

3) READ. The check on a digit or '+' or '-' during an integer read doesn't appear to work.

4) FILES. This one had me wondering a bit. During the course of a program a string included a Line Feed character (0AH) that shouldn't have been there. It seems that during a disk read, if the end of the buffer coincides with the middle 0D0AH character-pair, then LF is not skipped, and the next character read picks it up.

I.P. Forbes, Brighton.

*** Printing by GAS *** R. Cutler

The time had come when hard copy was an absolute necessity, but how did one obtain (other than beg, borrow or steal), a mechanical marvel of reasonable quality? A CREED was ruled out since I could not afford any structural alteration's like joist support's or double glazing.
(I almost bought a QUME daisywheel but they didn't have the right colour).
Think's.....try the WATER board, GAS board, ELECTRICITY board, GPO even.
WATER board:- Well er...I don't er... we've got 'one' but when it goes wrong we buy another.
GAS board:-YES, IBM magnetic tape typewriter's, (GOLFBALL'S to you).
Will you be making any redundant in the near future? YES, come over and have a look at one.
I was there like a shot and came face to face with an IBM MT-typewriter.
I'm afraid we can't just sell the printer you would have to take the whole system. This dampened my enthusiasm because the 'whole' system looked completely un-affordable. (have you ever seen an IBM MT-typewriter?).
It consist's of a 'metal' three draw desk with the printer on top, and alongside that is a rather large tape control unit. The latter, (by the way), just fit's into the boot of a MARINA.
Anyway, to cut a long story short, I managed to acquire this highly sought after piece of equipment for a princely £50.

*** Now to drive it. ***

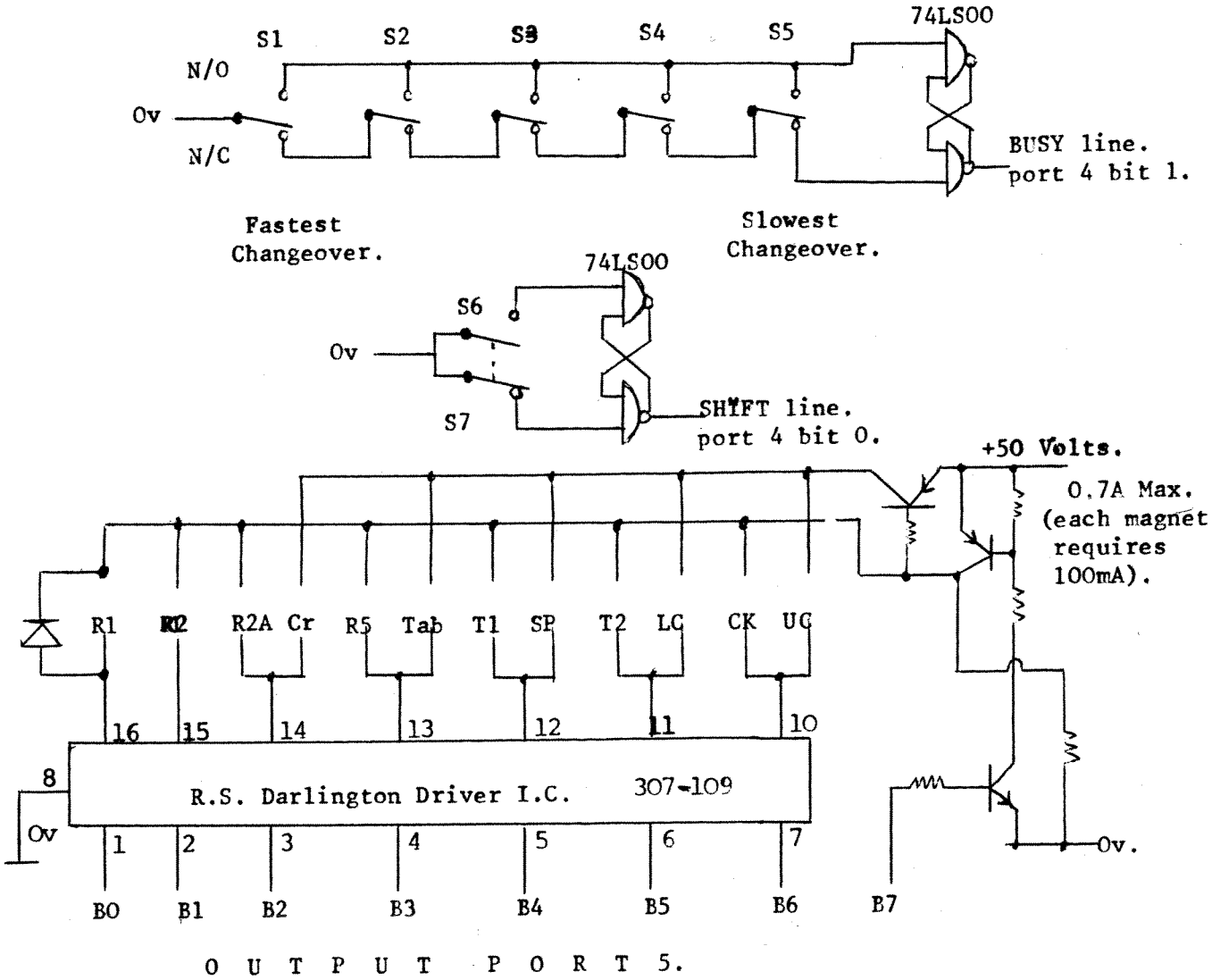
I eventually intend to drive the printer from an RS232 serial port using an 8035 micro. This will be programmed to recieve RS232 output at whatever speed is selected by link's on the RS232 input socket. (If you are not familiar with the 8035/48/51 series micro, perhaps an article would be of interest, it's an ideal control micro?).

For the time being, however, I wanted hard copy, so decided to use a PIO.
There are 7, 48 volt print magnet's on an IBM called R1 R2 R2A R5 T1 T2 CK.
I've wired these via, drivers to port 5 bit's 0 to 6 respectively.
There are also a few control magnet's, I have used CRLF TAB SP LC UC connected to bit's 2 to 6 respectively of the same port.
A glance at the circuit will show that bit 7 is used to select either print magnet's or control magnet's. This was done so that a 12 core cable could be used between the computer and printer.

Two other signal's are required, a BUSY line and a SHIFT line. These are produced as shown in the circuit. When the printer is ready, BUSY=1.
The SHIFT line is used to identify which case the printer is in, and also serves to 'time' the case change. Port 4 bit 0 is used for SHIFT, and bit 1 for BUSY.

The mechanical timing of an IBM is fairly complicated, so to get a correct BUSY line signal, the feedback contact's must be connected in the correct sequence as shown.

The contact's work on the following principle:-
When the IBM is ready, the N/C contact's are used to set BUSY high.
A character is sent.
When the N/O contact's have closed, the magnet's may be de-energised,
And when the N/C contact's have re-closed the IBM is ready for another character.
Refer to JAN 79 Practical Computing for more detailed information.



S1:-Is the main timing contact and is found on the left hand side of the unit. There are two set's by the timing wheel and it's the inner set.
S2:-Is the space and back-space contact,it's situated at the rear of the unit, second set from the right.
S3:-Carriage return and line feed contact, rear of the unit, extreme right.
S4:-CR interlock, right hand side toward's the rear. If CR is operated, S4 can be seen to move.
S5:-TAB interlock, rear of unit extreme left.
S6:-Right hand side, closes when in upper case.
S7:-Near S6, closes when in lower case.
Note that all magnet's must have a diode across them for two reason's.
1/. To protect the driver transistor's.
2/. To make the magnet's release as quickly as possible.
The 5 volt supply for the 74LS00 was provided by the PIO socket on the NASCOM, so that if a 12 core screened cable is used (i.e, screen = Ov), there will be ONE spare way.

The print magnet's, R1 through CK, are underneath on the left, and from left to right are:-
T2 CK T1 R2A R1 R2 R5
As for the control magnet's, it's simply a matter of applying power to each one in turn and noting the effect. They are situated on the right hand side, underneath.

```
                0010 ;* * * * * * * * * * * * * * *
                0020 ;* Golf-ball Print Handler,*
                0030 ;* Load $UOUT with start,  *
                0040 ;* OC78/79 for NAS-SYS, or,*
                0050 ;*    OF05/06 for ZEAP.    *
                0060 ;* Ray, Cutler,    25/3/82,*
                0070 ;* * * * * * * * * * * * * * *

                0090 ;Note that no NAS-SYS re-start's are used
                0100 ;so that the routine will run with other
                0110 ;software.

                0130 ;The routine is relocatable except for
                0140 ;line 590

B000            0160        ORG  £B000

B000 0004       0180 PORTW  EQU  4
B000 0005       0190 PORTX  EQU  5
B000 0006       0200 PORTY  EQU  6
B000 0007       0210 PORTZ  EQU  7

B000 E5         0230 START  PUSH HL;Save registers,
B001 D5         0240        PUSH DE
B002 C5         0250        PUSH BC
B003 F5         0260        PUSH AF
B004 3E0F       0270        LD   A, £OF
B006 D307       0280        OUT  (PORTZ),A;PIO/B MODE 0,
                0290 ;Control Port for IBM magnets,

                0310 ;PORTX = data port for magnet's,
                0320 ;CK  T2  T1  R5  R2A  R2  R1 when B7=1,
                0330 ;UC  LC  SP  TAB CRLF        when B7=0,
                0340 ; 6   5   4   3   2   1   0 respectively.

B008 3E4F       0360        LD   A, £4F
B00A D306       0370        OUT  (PORTY), A;PIO/A MODE 1
                0380 ;Control port for SHIFT and BUSY status,

                0400 ;PORTW = Data port,
                0410 ;Bit 0 = SHIFT (1=LC, 0=UC),
                0420 ;Bit 1 = BUSY  (1=Ready, 0=wait),

B00C F1         0440        POP  AF;A=Valid ASCII code,
B00D F5         0450        PUSH AF
B00E FE09       0460        CP   9;Tabulate? (Cntrl I)
B010 2835       0470        JR   Z, TABULE
B012 FE0D       0480        CP   £OD;CRLF?
B014 282D       0490        JR   Z, CRLF
B016 FE5F       0500        CP   £5F;Left arrow?
B018 2831       0510        JR   Z, SPACE
B01A FE21       0520        CP   " +1
B01C 382D       0530        JR   C, SPACE;It's less than 21H,
B01E FE7B       0540        CP   "z+1
B020 3029       0550        JR   NC, SPACE;It's greater than 7AH,

                0570 ;A=£21 to £7A except £5F
                0580 ;i,e, All printable codes,
B022 2189B0     0590        LD   HL, TAB1
B025 D621       0600        SUB  £21;A=0 to £59
B027 0600       0610        LD   B, 0
B029 4F         0620        LD   C, A
B02A 09         0630        ADD  HL,BC
B02B 7E         0640        LD   A, (HL)

                0660 ;A=Valid IBM code,
                0670 ;Bit 7 = Upper-case,
B02C 57         0680 SHIFT  LD   D, A;Save it,
B02D CB7F       0690        BIT  7,A
B02F DB04       0700        IN   A,(PORTW)
B031 2008       0710        JR   NZ,UPPER

B033 CB47       0730 LOWER  BIT  0,A;1=LC 0=UC
B035 2031       0740        JR   NZ,OUT6   ;It's another lower case,
B037 3E20       0750        LD   A, £20;Do lower case,
B039 1816       0760        JR   OUT2

B03B CB47       0780 UPPER  BIT  0,A;1=LC 0=UC
B03D 2829       0790        JR   Z,OUT6    ;It's another upper case,
B03F 3E40       0800        LD   A, £40;Do upper case,
B041 180E       0810        JR   OUT2

B043 3E04       0830 CRLF   LD   A, 4;CRLF relay,
B045 1806       0840        JR   OUT1

B047 3E08       0860 TABULE LD   A, 8
B049 1802       0870        JR   OUT1

B04B 3E10       0890 SPACE  LD   A, 10H;Space relay,

B04D D305       0910 OUT1   OUT  (5), A;Send non-print code,
B04F 181C       0920        JR   WAIT

B051 D305       0940 OUT2   OUT  (5), A;Send shift code,
B053 FE20       0950        CP   20H
B055 2808       0960        JR   Z, OUT4

                0980 ;An upper case code has been sent so
                0990 ;wait for a '0' from port 4 bit 0,
B057 DB04       1000 OUT3   IN   A,(PORTW)
B059 CB47       1010        BIT  0,A
B05B 20FA       1020        JR   NZ,OUT3
B05D 1806       1030        JR   OUT5

                1050 ;A lower case code has been sent so
                1060 ;wait for a '1' from port 4 bit 0,
B05F DB04       1070 OUT4   IN   A,(PORTW)
B061 CB47       1080        BIT  0,A
B063 28FA       1090        JR   Z,OUT4

B065 AF         1110 OUT5   XOR  A ;Power down the SHIFT magnets,
B066 D305       1120        OUT  (PORTX),A

B068 7A         1140 OUT6   LD   A, D;Recover character to print,
B069 F680       1150        OR   80H;Select print magnet drivers,
```

```
B06B D305      1160        OUT  (PORTX),A

B06D DB04      1180 WAIT   IN   A,(PORTW);Have interlock's changed
               1190 ;over yet?
B06F CB4F      1200        BIT  1, A
B071 20FA      1210        JR   NZ, WAIT;No
               1220 ;Make sure it's not contact bounce
               1230 ;otherwise the magnets may be de-energised
               1240 ;too early causing missing characters.
B073 219001    1250 DELAY  LD   HL,400
B076 2B        1260 DEL1   DEC  HL
B077 7C        1270        LD   A, H
B078 B5        1280        OR   L
B079 20FB      1290        JR   NZ, DEL1
B07B AF        1300        XOR  A
B07C D305      1310        OUT  (PORTX),A;Power down all magnets.

B07E DB04      1330 WAIT1  IN   A,(PORTW);Have interlock's
               1340 ;assumed their original position yet,
               1350 ;ready for another character?
B080 CB4F      1360        BIT  1, A
B082 28FA      1370        JR   Z, WAIT1;No
               1380 ;IBM ready for next code.

B084 F1        1400 END    POP  AF
B085 C1        1410        POP  BC
B086 D1        1420        POP  DE
B087 E1        1430        POP  HL
B088 C9        1440        RET

               1460 ;BIT 7 High denotes UPPER case.

               1480 ;          .    "    £    $    %    &    '
B089 1693B3B2  1490 TAB1   DEFB £16 £93 £B3 £B2 £89 £BB £13
     89BB13
               1500 ;      (    )    *    +    ,    -    .    /
B090 B0B1B6BF  1510        DEFB £B0 £B1 £B6 £BF £0A £40 £16 £BE
     0A4016BE
               1520 ;      0    1    2    3    4    5    6    7
B098 9929363E  1530        DEFB £99 £29 £36 £3E £39 £33 £32 £3B
     3933323B
               1540 ;      8    9    :    ;    (    =    )    ?
B0A0 3A308B0B  1550        DEFB £3A £30 £8B £0B £B0 £3F £B1 £8A
     B03FB18A
               1560 ;      AT   A    B    C    D    E    F    G
B0A8 B99AA0AA  1570        DEFB £B9 £9A £A0 £AA £AB £A3 £8E £8F
     ABA38E8F
               1580 ;      H    I    J    K    L    M    N    O
B0B0 A19287A2  1590        DEFB £A1 £92 £87 £A2 £A9 £9F £A6 £99
     A99FA699
               1600 ;      P    Q    R    S    T    U    V    W
B0B8 83829B91  1610        DEFB £83 £82 £9B £91 £A7 £AE £9E £90
     A7AE9E90
               1620 ;      X    Y    Z    (    /    )    '    SP
B0C0 AF81B7B0  1630        DEFB £AF £81 £B7 £B0 £BE £B1 £13 £00
     BEB11300
               1640 ;      '    a    b    c    d    e    f    g
B0C8 131A202A  1650        DEFB £13 £1A £20 £2A £2B £23 £0E £0F
     2B230E0F
               1660 ;      h    i    j    k    l    m    n    o
B0D0 21120722  1670        DEFB £21 £12 £07 £22 £29 £1F £26 £19
     291F2619
               1680 ;      p    q    r    s    t    u    v    w
B0D8 03021B11  1690        DEFB £03 £02 £1B £11 £27 £2E £1E £10
     272E1E10
               1700 ;      x    y    z
B0E0 2F0137    1710        DEFB £2F £01 £37
```

```
TB000 B100 FF 8 FF
 B000 E5 D5 C5 F5 3E 0F D3 07 3E 4F D3 06 F1 F5 FE 09
 B010 28 35 FE 0D 28 2D FE 5F 28 31 FE 21 38 2D FE 7B
 B020 30 29 21 89 B0 D6 21 06 00 4F 09 7E 57 CB 7F DB
 B030 04 20 08 CB 47 20 31 3E 20 18 16 CB 47 28 29 3E
 B040 40 18 0E 3E 04 18 06 3E 08 18 02 3E 10 D3 05 18
 B050 1C D3 05 FE 20 28 08 DB 04 CB 47 20 FA 18 06 DB
 B060 04 CB 47 28 FA AF D3 05 7A F6 80 D3 05 DB 04 CB
 B070 4F 20 FA 21 90 01 2B 7C B5 20 FB AF D3 05 DB 04
 B080 CB 4F 28 FA F1 C1 D1 E1 C9 16 93 B3 E2 89 BB 13
 B090 B0 B1 B6 BF 0A 40 16 BE 99 29 36 3E 39 33 32 3B
 B0A0 3A 30 8B 0B B0 3F B1 8A B9 9A A0 AA AB A3 8E 8F
 B0B0 A1 92 87 A2 A9 9F A6 99 83 82 9B 91 A7 AE 9E 90
 B0C0 AF 81 B7 B0 BE B1 13 00 13 1A 20 2A 2B 23 0E 0F
 B0D0 21 12 07 22 29 1F 26 19 03 02 1B 11 27 2E 1E 10
 B0E0 2F 01 37 00 00 00 00 00 00 00 00 00 00 00 00 00
 B0F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

A Change of Tactics

D. R. Hunt

PROLOGUE

In the past we have played down the numerous letters we have received describing 'How I made my Nascom'. We have published a couple describing systems in use, but in general we have confined ourselves to reviews of component parts (boards) and descriptions of uses of component parts, rather than describing a complete system made out of the parts. Spurred on by the appearance of the Gemini GM835 Winnie just after Christmas and the Gemini implementation of a piece of software called CCPZ, I recently took the plunge and added this to my system. I intend to describe the change in operating procedure brought about by using CCPZ, but first, as an aid to understanding what it's all about, I intend to break with the previous tradition and describe the evolution of my computer and the uses to which it has been put.

EVOLUTION

In the beginning there was a Nascom 1, the subject of much of my earlier series, the Dodo's Guide to Assembler Programming. Most of what I know about home computing is directly attributable to that machine. I could never lay claim to being an expert on the Nascom 1, as perhaps I was more fortunate than some in that it worked first time, and most of it remained working until I finally disposed of it in its final rebuilt form a year back. I believe it is still working well with no problems, somewhere. My Nascom 1 soon sprouted 2K of additional RAM, addressed at F800H - FFFFH, which was originally designed as a programmable character generator, and this was shown on the Nascom stand at the PCW exhibition in 1978. The additional 2K of RAM was more than just a PCG, it could of course be used as 2K of expansion RAM. Bearing in mind that Nascom had not yet got round to making any of their own, Richard Beal borrowed the machine shortly after the exhibition and implemented the public domain 2K Tiny BASIC written by the American with the Chinese name (which unfortunately I have forgotten). This explains why Nascom Tiny BASIC resided between F800H and FFFFH. Then came the 32K RAM (A) card and buffer card with their additional problems. Overall, the Nascom 1 was a most useful machine and taught me a lot.

Nascom 2 came along, and I naturally acquired one with 16K of memory and indulged in a massive teach yourself BASIC programming exercise. Most of my knowledge of BASIC programming is down to the Nascom 2. As new goodies appeared, so these were either bought or acquired, again in the cause of self education.

Nascom got themselves into money trouble and there seemed as if there would be no disk systems for it, a great pity. So the Henelec/Gemini disk system was born, designed by Peter and Richard (not Beal) with a little help from myself. The original disk system required field testing, and I gave it as good a bashing as most. In the process, learning a lot about disk systems. All this was done on the basis of "if you make enough mistakes, you're bound to learn to get it right in the end".

So my system grew both in power and in size. My wife started to object to the pile of cards and bits on the living room table, so the system ended up 'cased', perhaps housed is a better word, in an MFI office desk. With suitable modifications, the desk is still in use today, supplemented by a matching wood finished filing cabinet. The desk is about 5'6" long and 2'6" deep, and with a large cupboard and three drawers it is ideal. It even has a modesty board towards the back fitted with shelves 8" deep. Absolutely the right size for Nascom boards. Perhaps the clincher was the price, at £49.50 (inc VAT) it was about the cheapest computer case on the market, offering all the advantages 'her at home' could ever require. From her point of view it is a piece of furniture, it is neat and tidy, all my junk remains in one place not cluttering up the table, and the only bits that show are the monitor, the keyboard and the printer, all of which are removeable when guests arrive. From my point of view, it means I have a permanent place to work at, without causing friction with the missus, all my

tapes and disks live in the drawers and all the manuals in the cupboard (and these days in the filing cabinet as well). The computer is safe and away from prying fingers, being buried on the shelves behind the desk. The only snag as far as I can see is the weight of the thing, like all MFI stuff, it might be cheap, but it makes up for it in weight. With the cupboard full of books, the keyboard, monitor and printer in place, and the drawers full, it must weigh nearly a couple of hundredweight, a distinct disincentive to making changes to the computer as it means pulling the whole lot away from the wall to get at anything.

With the advent of the disk system, so CP/M came into my life. This proved to be quite useable but a lot less clever than the nice Nascom operating systems Richard is wont to write. Richard converted to CP/M about the same time and started to write the SYS series of overlay BIOSes for CP/M which brought some of the niceties of NAS-SYS to CP/M. Of course, with the advent of disks, everthing I had previously developed as a discipline in using the machine had to change, and with it the sort of things I was doing. Less learning 'how to use' the machine and more actually 'using' it, if the difference is discernible. Exercises in overlaid BASIC programs for business use, and of course a lot more writing both for this magazine and others using newer and ever improving versions of NAS-PENs.

Despite the danger of my having to wear a truss through moving the desk, the system has grown over the past couple of years. Various goodies from the Gemini range have been added, the GM809 disk controller, doubling the density of disk system, and the Gemini GM812 IVC card, bringing the video display up to the standard I required. I would have liked to have played around with colour, but the Nascom AVC was delayed so long, and when it finally arrived was found to be too slow for my purposes, my enthusiasm waned and I asked myself if the investment in colour was really worth it for the sort of things the computer is used for. Pluto would be nice, but prohibitively expensive when the cost of a high resolution monitor to do it justice is considered. So colour was dropped.

With my re-awakened interest in radio (hardly a new found interest as I have been dabbling with radio for the past 20 years, long before the home computer was ever thought of), the need to quieten the system RF-wise became a pressing consideration. For those not aware of the problem, try putting a portable radio next to your Nascom and see what happens (to the radio that is). The radio becomes unuseable except as an uncontrollable squawk box, emitting squeaks and whistles all over the place. Now ALL computers chuck out RF interference and Gemini are as bad as Nascom in that respect. Anyway, the system had to be rebuilt into a totally shielded RF proof box. Up until then it had been what you might call a 'distributed system', with bits of it on various shelves. Unfortunately, when designing the box, and bringing it all together, it was soon realised that using the Nascom with its 12" x 8" board would mean that the box would be too big to fit on the shelves of the desk. So somewhat reluctantly the Nascom was pensioned off and a Gemini GM813 put in its place. This meant a spare RAM card which could be used as virtual disk as this feature had recently been added to SYS.

So the system slowly evolved and was used under CP/M, using two drives. The A: drive always containing what I called the 'system disk', of which I had three. The first 'system disk' had all the utilities I used for assembler programming, the assembler and linker, the debugger, PEN, and so on. The second disk had all the BASIC bits on it, the BASIC itself, the compiler and so on. The third disk was my INMC/80BUS (word crunching) disk, containing PEN and the proof reading program SPELLGUARD with its 150K dictionary [Ed. - you trying to tell me you actually 'Spellguard' this lot?]. Work was always carried out on the B: drive, and various 'system disks' were used in drive A: in conjunction with the relevant 'work disk' in drive B:. I have found this an ideal way of working and saw little reason to change it.

But change it must. With about 80 disks in use, the system was becoming unmanageable, and the investment in disks alone was worth a tidy sum. Three things happened to force me into making the system into what it is today. Firstly I hadn't used the virtual disk much as the capacity with one RAM card was just too little, however, it had impressed me with its speed. MAP changed that with the introduction of their 256K RAM card. Next, the Gemini Winnie was put into use at work, and the change in operating procedure forced by its sheer size (5.4 Mbyte might not sound a lot, but you fill it and then try finding something in the directory) introduced me to using the USER areas in CP/M for the first time. I had never previously used the USER areas under CP/M as their operation under CP/M is distinctly unfriendly and result in the duplication of lots of common files thus using up more disk space. The Winnie showed me the delights of using CCPZ, a modified form of which is provided with the Winnie. Lastly two double sided quad density 80 track drives became available, offering 788K per drive, and the thought of reducing the number of disks in use really struck home.

The transformation took place one weekend shortly after Christmas, the two drives and the MAP card were fitted and apart from some aggro with one of the drives, which will cause me to have some very rude things to say if I ever meet a rep. from the Micropolis disk drive company, about their ex-factory quality control, all fired up correctly. Richard's latest SYS, SYSB15, was reassembled to cope with the double sided drives and the MAP card; and CCPZ, which I had had for some time, but not implemented as it looked very complicated, was patched as necessary and away it all went.

The most time consuming part of the whole job was transferring my 80 odd disks from 35 to 80 track. There are three approaches one can take. The disks can be transferred from a 35 track machine to an 80 track machine using either the serial or parallel interfaces. Or you can borrow a machine which is fitted with both types of drive and PIP them across using the serial interfaces. Or, as the track pitch of the 35 track drives is 48 tracks per inch and the track pitch of the 80 track drives is 96 t.p.i, you can make one of the 80 track drives 'double step' reading alternate tracks and PIP the disk to the other drive (in a Gemini system you can only resort to the latter if double sided drives are fitted).

I used a combination of methods one and two. We have two machines at work, one 35 track and one 80 track, hooked together by a length of bell flex and a simple (but expensive) piece of software called BSTAM which allows the two machines to talk to each other serially at 9600 BAUD. About half my disk library was copied using BSTAM, taking the best part of a week. The remainder was copied using a machine called the Octopus at Gemini in Amersham. It's a sort of Galaxy fitted with standard 8" and 35 and 80 track 5.25" drives, and has a Winnie lurking on it somewhere. It can be programmed to copy almost anything to anything and is used by software houses for just that purpose. This copy service is one Gemini normally charge for, but they wouldn't charge li'l' ol' me .... would they? An afternoon spend on the Octopus finished the job. So within the week I was back to normal with a fully operating system and my library of disks reduced to just 25. I didn't try using the third method, as I don't like mucking about with disk primitives in case I get it wrong. Disk systems are a bit 'Catch 22ish', in that a non working disk system can't be diagnosed because it doesn't work, at least not without resorting to a ROM based operating system. Anyway it would have taken me a week to make the software work satisfactorily, so what odds, it wouldn't have been any quicker, and I would have ended up with brain ache as well.

So the system in its present form consists of a Gemini GM813 CPU/RAM card, a GM812 IVC card with GM821 keyboard, a GM809 FDC card, a GM822 RTC card, a MAP 256K RAM card, a Nascom I/O card and two double sided Micropolis 1015-F6 quad disk drives which give a shade under 1.6 Mbyte of disk storage. The whole lot is powered by a Tangerine 6 amp switch mode power supply made by Astec (similar to

the Gemini GM817 PSU but without the RF shielded case). An ICL Keyedit monitor and a NEC PC-8023B-C printer all mounted in, on, or about the MFI desk complete the ensemble.

The system is now mostly used for word processing (that is writing this stuff) using the just available DISKPEN (GEMPEN) 3, running the radio logbook program (the subject of my discussion on data bases elsewhere in this issue), the development of stock control and financial programs using DBASE II for work, and the odd bit of assembler work as and when I have a bright idea and, of course, find the time.

CCPZ

Now on to CCPZ. It was designed by a whole group of bods from the US CP/M Users' group and for a committee effort it's quite good. This is another overlay job and to understand how it is introduced into the system we must consider the three parts of CP/M 2.2. Looking at CP/M it may very conveniently be split into three parts, the first part, the Central Command Processor (the CCP), the Basic Disk Operating System (the BDOS), and the Customized Basic Input Output System (the CBIOS). The CBIOS handles all the necessary interfaces between CP/M and the computer, handling the keyboard, the screen, the printer and the disk primitives. It is CBIOS which Richard replaces with his SYS programs. The BDOS is the heart of CP/M, and is where CP/M makes the decisions as to where it has to store or retrieve things from disk of differing sizes and formats. The BDOS has so far been left strictly alone. The CCP is where all commands from the various inputs are interpreted and the way in which CP/M acts on its own internal instructions, DIR, ERA, etc. The CCP is what is of interest when investigating CCPZ.

The approach adopted by the CCP committee has been to write an entirely new CCP program in Z80 code. As the CCP is located at the beginning of CP/M, and its size and locations are fixed by the way in which CP/M works, and the naughty way in which some programs use it direct, its size is strictly limited to 2K. The difficulty of mucking the MOVCPM.COM relocator about has also meant that a new CCP is not easily patched into MOVCPM.COM, so a different approach towards implementation has been adopted than that used by SYS. As Z80 code is more compact than 8080 code, coupled with the fact that the general opinion is that CP/M is compiled from some efficient high level language and is therefore to some degree wasteful of space, the CCP produced by the committee was originally much shorter than the original. This allowed room for expansion. CCPZ has grown some extra commands, all very useful to the machine code programmer.

CCPZ FEATURES

So what has CCPZ got to offer over the normal CP/M CCP, well the major feature is not a command at all. It's primary use is when the CP/M USER areas are in use, and it is the way in which it looks for files. Imagine the situation where I am using drive B: as the default, and I want to edit this piece of text, I forget that drive B: is the default and type:

B>PEN B:80BUS6-2.PEN

to get this file in. Well of course PEN is on A: isn't it, so the result will be the error message:

PEN?

won't it? Not so with CCPZ, the process is as follows:

Look on the default for the command file, if found, execute it

Nothing found so select the specified default drive (I specified A: when I assembled CCPZ).

Command file found, so load it.

Reselect the original defaults.

Now execute it.

See how it works, it looks for the command file in an hierarchical fashion, first on the specified or default drive, if not found, then on to the final default drive you specified when assembling CCPZ (you can also specify the

default USER area on the final default drive using the DFU command). Further, this hierarchical search extends through the CP/M USER areas. Suppose I was on drive B:, USER 2, and PEN was stored on drive A:, USER 0; then the search sequence would start with the default (or specified) drive and USER, if the search failed, then on to the default (or specified) drive, USER 0, if that failed then on to the final CCPZ default, USER 0. Only then, if the search failed would any error message be displayed. Note also that the hierarchical search does not take place on the command extension, in this instance the 80BUS6-2.PEN. Having found PEN, the whole lot resets to the original conditions. So PEN searches for 80BUS6-2.PEN on the original default. If PEN can't find what it's looking for then PEN says so. So the important thing to note is that CCPZ only does the hierarchical bit when searching for the command file, what follows is not the business of CCPZ, it's up to the command file to decide where it gets the command extension from.

Another nice little tweek is the prompt itself, when you change USER area, the new area number is displayed in the prompt. So that you can see at a glance that:
B12>
you are on default drive B: USER area 12. Nice isn't it. Yet another comes to mind, the ERA command, when you erase a file(s) the ERA command displays what it has erased. (Even better if it stopped after displaying the list and asked you if you meant it, but it doesn't.) Well you at least know what you erased, and if you find one you didn't mean, then a little utility I have called UNERA does, guess what?

As to the commands, well all the original CP/M commands are there of course, but some work differently:

Command:	DIR	
Function:	To display a listing of the names of the files on the disk.	
Forms:	DIR <afn>	Displays $DIR files
	DIR <afn> S	Displays $SYS files
	DIR <afn> A	Displays both $DIR and $SYS files
Note:	If the directory is scanned for a name and no such name exists on the disk the message 'No Files' will be displayed. However, if the search was for $DIR files and the file existed as a $SYS file or vice versa, then no message will be displayed indicating that the file does exist but the file attribute is not as specified.	

Command:	ERA	
Function:	To erase the specified R/W file from the directory.	
Forms:	ERA <afn>	Erase both $DIR and $SYS files from disk.
Notes:	The name(s) of the file(s) erased is displayed as a cross check.	

Command:	LIST	
Function:	To print the specified file to the CP/M LST: device.	
Forms:	LIST <ufn>	Print the file (no paging).
Notes:	The file will be printed regardless of the $DIR or $SYS attribute.	

Command:	TYPE	
Function:	To print the specified file to the CP/M CON: device.	
Forms:	TYPE <ufn>	Type the file with the paging default.
	TYPE <ufn> P	Type the file with the paging default negated.
Notes:	Any character will continue typing at the end of a page. ^C will terminate typing, ^P and ^S will work as usual.	

```
Command:   SAVE
Function:  To save an image of the TPA to disk starting at 100H.
Forms:     SAVE <n> <ufn>        Save n decimal pages to disk.
           SAVE <n>H <ufn>       Save n hexadecimal pages to disk.
           SAVE <n> <ufn> S      Save n decimal sectors to disk.
           SAVE <n>H <ufn> S     Save n hexadecimal sectors to disk.

Command:   REN
Function:  To change the name of a disk file.
Forms:     <ufn.new>=<ufn.old>  Change the .old name to the .new name.

Command:   USER
Function:  To change the current USER number.
Forms:     USER <user number>   Change to new decimal USER number.
           USER <user number>H  Change to new hexadecimal USER number.

Command:   DFU
Function:  To temporarily change the default USER number for the command
           hierarchy search.
Forms:     DFU <n>              Change default to n decimal.
           DFU <n>H             Change default to n hexadecimal.

Command:   JUMP
Function:  To 'CALL' the routine at the specified address.
Forms:     JUMP <aaaa> <cmd. parms.> or
           JUMP <aaaa>H <cmd. parms.>   Call the routine at <aaaa>H.
           JUMP                         Call the routine at 0000H (warm boot).

Command:   GO
Function:  To 'CALL' the routine at address 100H.
Forms:     GO <cmd. parms.>     Call the routine at 100H.

Command:   GET
Function:  To load the specified file to the specified address.
Forms:     GET <aaaa> <ufn>     Load the file to address aaaaH.
```

So as you can see CCPZ has a lot to offer in the command mode, the GET and SAVE commands are very useful to the machine code programmer.

So how do you fit it, well I acquired my copy from Richard (I think) and it in turn originated from the UK CP/M User's Group. Four files were supplied, a source file, installation documentation, instructions for use and a complete history/philosophy commentary. From the documentation it was soon apparent that at least one other file was missing, but fortunately a file called STATUS from 'you know who's' utilities disk proved to be a similar thing. To assemble it I also needed the DRI MAC macro assembler as the macros in the source file precluded the use of my particular favourite the Microsoft Macro 80 assembler. The installation words needed reading twice to understand what it was on about, but once the basic idea was grasped the well documented 'demonstration run' was easily understood (except the assembly parameters for DRI MAC which as I didn't read the manual, I didn't understand).

The principle is easy, and works on the philosophy of;

"Notice you rarely need to change the size of the CP/M system, hence, you rarely use MOVCPM. Well as you rarely change the system size, it doesn't really matter if the system size becomes fixed, now does it? So decide on your optimum system size, find out the start and end addresses of of the CCP within your optimum system. Give the CCPZ ORG statement this data and assemble your copy of CCPZ to this size. Having got the CCPZ, use MOVCPM to make a system to your chosen optimum size, but instead of using SYSGEN to put it on the system tracks,

save it to disk instead. You've now got two files on the disk, the CCPZ and a CP/M system of matching size. Now use DDT or ZSID, or some such to merge the two files together putting CCPZ over the top of the original CCP and resave the the file. Finally use SYSGEN to put that file onto the system tracks. Bash RESET, and providing all is Ok, it should all fire up."

This philosophy is fine provided you don't change system size often. If you do, then all you need do is to keep a copy of the merged CCPZ and the CP/M system assembled for your favourite size on a disk, and when you change size just generate a CP/M to the system tracks using MOVCPM in the normal way. Ok, so you lose the features of CCPZ. When you've done, simply SYSGEN the CCPZ system image back onto the system tracks and you're all set up again. (The version of CCPZ supplied with the Gemini Winnie has a relocator incorporated and whole thing is merged into MOVCPM. You can then use MOVCPM properly. Pity they don't release this as a program, but I suspect that permutation problems would win in the end.)

Well this was all detailed in the installation documentation, and with the help of the dummy run, fairly easily understood. Finding the start of the CCP in an active CP/M of the chosen size proved a little difficult because the little program that should have been supplied to give you the address wasn't there. So I thought about this for a while. My first thought was to use GEMDEBUG (I like 'BUG' over the others because I find it fairly easy to use even though it's nothing like as powerful as ZSID) to find out where the CCP normally resided, but that was no use, as 'BUG' overlays the CCP and alters its size, and hence its jump location. At least I suppose this to be case with 'BUG', its's certainly true of ZSID and DDT, I couldn't be bothered to phone David to ask him. So using a debugger was out, I could calculate it, but as I only had a hazy idea of what I was about, this didn't sound too clever, as it would introduce one more variable into the system to cause it to crash. Then I remembered STATUS, a little file I have rarely used because I thought I knew where everything was on my computer, apart from that, STATUS tests all the ports to see if something happens, as soon as it hits OB3H, of course the video card resets, so you have to be very quick with a ^S if you hope to see the information prior to the video card reset. Anyway, I ran STATUS, and on the second try I caught it before it got to port OB3H and, LO, there was my CCP start address.

Armed with the start address of the CCP, I had to modify the ORG statement of the source file. Easy? Not so. The source is 52K long and my version of PEN will only hold about 50K. I could have used WORDSTAR I suppose, but I don't like it, it always wants to fight back at me. I always find it so much trouble to find out what the commands are, I find its layout of five sub-menus illogical and the commands I want to know about always seem to be scattered about in the five sub-menus and never in one. Further, there is no index to the sub-menus, so I have to work through them in order to find things. Enough slagging WORDSTAR, I know there are people out there who like it, and it is capable of jobs that PEN could never do (like writing books). Personally I find it all a bit of a pain, so enough personal bias from me. Anyway, horror of horrors, I resorted to ED!!! Now if there was something that should have been strangled at birth ..... With the aid of the CCPZ documentation, I patched the ORG statement and a couple of other things in the source, like "Do I want the prompt on USER 0 to be displayed as A> or A0>", and things like that, and wrote the lot out to disk.

DRI MAC to the fore, I typed the assembly parameters exactly as in the demo (I have no idea what they meant) and let it get on with it. I ended up with a CCPZ.HEX file and a CCPZ.REL file. Ok, take the .REL file and LINK it, to make it appear as a true code file, but the linker wasn't having any. Apart from that it was departure from the CCPZ instructions, but I thought I'd be clever and do it my way. Alright, don't give up, get LOAD out of it's dark drawer, blow the moths out of it, and see if that does the trick on the .HEX file. After all it's supposed to take a .HEX file and convert it into a .COM file. Of course, LOAD

wasn't interested in the idea, I kept getting funny totally uninformative error messages. Anyway as the linker and LOAD both didn't want to know, and as I didn't have the manual for the linker anyway, it was back to the instructions. These woffled on about using DDT and reading in the .HEX file with an offset calculated from the start address and the destination. Frankly it didn't make a lot of sense, as it seemed to mean that DDT could take a HEX file, convert it to code and stuff it in the right place. This sounded cleverer than the DDT I know and hate, but as I hadn't read all the words about DDT ... perhaps ... you never know. So I did it and it appeared to work. I examined the code at the start of the CCP as now overlayed by CCPZ and it was certainly different, but did CCPZ work? So save the file and SYSGEN onto the system tracks and find out!!

Now this is the natural cliff-hanger, where I should say this is the end of part one and watch this space for future developments. Aren't you lucky this isn't a serial, because of course it worked. I still don't know why, and if it hadn't I'd have actually had to read ALL the documentation to find out why. (Have you noticed I'm a great believer in "If it doesn't work try reading the instructions".) Anyway it worked, and I proceeded to check all the functions. There was one nice little tweek that Gemini have in their modded version on the Winnie, and that was to do with the USER command. You could either type:

```
A>USER 13
```

or

```
A>.13
```

Now as . is three letters less than USER, this seemed a good idea. So back to the source. It was easy enough to find the command jump table, so I added the extra command '.' and reassembled it. Tuf, it now looked for the user number, the '13', in the wrong place. I could get away with:

```
A>.  13
```

or even

```
A>  .13
```

But of course the routine that actually got the user number wasn't bright enough to look for it someplace else if you actually moved it. Oh well, back to the source and find the routine, which, surprise surprise, was labelled GETNUM. At that point I looked at the clock and noted that it was well passed the witching hour and I was in imminent danger of turning into a pumpkin so I left it ..... and left it ..... and left it. One day I'll get round to it but honestly, is it worth the bother I ask myself, and of course this is the way it will stay until the day when I have to change the user number 99999 times in one evening, THEN I'LL FIX THE B....R!!!

A CHANGE OF TACTICS

At long last, and about time too, the bit about the change of operating methods that this all brought about. Now I have lots of software, some purchased, odd bits stolen, some borrowed, a little 'liberated' and enough of my own to cause me considerable grief when trying to find what I'm looking for. Have you noticed that a lot of proprietary software often has a file called INSTALL, so if you are trying to keep things together, it's a bit daft to put two suites of programs on one disk without renaming the INSTALL program of at least one of them. Now consider the case where 788K is available on one disk, this problem is not going to apply to just a couple of suites of programs, more like half a dozen. How to keep them separate? The simplistic answer is to use different user areas for each. Fine, but how to find them.

A long time ago I came across the cataloguing suite by Ward Christennsen in the CP/M users library and to date this has proved ideal, although I took it apart a long time ago because I didn't like some of its 'features'. Richard also had a go at it soon after. Well of course, this couldn't keep track of things on different user areas, but one Sunday evening soon fixed that. Now CATUD can be called from any user area, CCPZ will of course find it where I left it, having loaded CATUD, CCPZ will switch back to the original user area and then execute the program. CATUD then constructs the names list for possible insertion into the

master catalogue in RAM. At that point I added a routine to save the original drive and user numbers and then do a hierarchical search similar to CCPZ for the master catalogue. The same as CCPZ with the exception that it always ends it search for the file at drive A: USER 0 as it was too much like hard work to make the thing search CCPZ for the final DFU number (and what about if CCPZ wasn't there anyway, it could do all sorts of wierd and wonderful things). Having found the master catalogue, this is updated from the names list. CATUD then restores the original drive and user and then goes home. I didn't need to update the catalogue search program CAT as I reckon that it will naturally be on the same drive/user as the master catalogue.

Of late GEMPEN/DISKPEN has being undergoing a face lift with the addition of lots of new features, two of which involve looking for overlay files, the master HELP overlay and the master MENU overlay, a quick word with Peter and PEN grew the hierarchical search as well.

So having found a way of cataloguing the software on various user areas, how to get it there? Even when using CCPZ, PIP is only capable of PIPing software from one drive to another with the same user number, so that's not a lot of use. [Ed. - wrong, Wrong, WRONG. If you actually READ the doc. that came with the GM835 Winnie that you have at work you will find it refers you to the 'G' option of PIP. Now go and read up on PIP at once!!] The answer came with SWEEP, another find from the CP/M User Group library. This is an all laughing dancing utility which not only allows movement from one user area to another (even on the same disk) but allows you to tag a load of files (and of course not tag others) for bulk movement or bulk erase. It also has the ability to rename files and a sneeky command for allowing you to 'TYPE' ASCII files to the screen to find out if it is the one you wanted. All clever stuff. I think it was written in some high level compiling language because of its size, 26K, but even so it earns its keep.

Lastly, having gained the disk capacity, implemented a means of ready access, fixed up a means of cataloging the files and got a means of putting the files where I wanted them, comes the way of organising them. The cataloging suite requires you to give the disk a name, and with my mod, the user areas on the disk can all be given different names. So let's say I want to sort out a disk full of source files into different groups. Now my disks live in books of 20 plastic wallets made for the job, a bit pricey at about £13.00 each, but ideal. Anyway, for sake of argument, the disk we are to play with is disk number 6. So I decide the CCPZ source suite with its .DOC files will live on USER 0, so I SWEEP them across and give that user area the name -CCPZASM.060. The next lot is the suite of .MAC files for taking the cataloging suite apart and putting it back together again. These are 'SWEPT' across to USER 1 and given the name -CATMAC.061. Next comes a suite of .MAC files about the RTC, a couple of mine, and Richards lot, these go on USER 2 and are called -RTCMAC.062. --- And so on --- Notice that the file type starts with .06, meaning disk 6, and the last number is the user area used for the particular suite. So when I ask CAT where the devil did I put the file called TIME.MAC the reply:

```
   File            Disk
TIME.MAC       RTCMAC.062
```

immediately tells me that the file is on disk 6, USER 2, get the idea?

It's all very simple, and of course entirely unoriginal, but it's surprising those I've mentioned it to have all seemed very impressed, hence the preceding 34K of file describing the system and how I use it.

Now for the commercial, the utilities I've mentioned are advertised someplace in this magazine and CCPZ is available from the same source for a tenner + VAT (state disk format when ordering). Of course the suppliers aren't going to accept responsibility if you muck up your CP/M master disk, that's your problem and serve you right for doing it on the master disk. Also, if you intend to have a go at fitting CCPZ you'll have to find your own copy of DRI MAC.

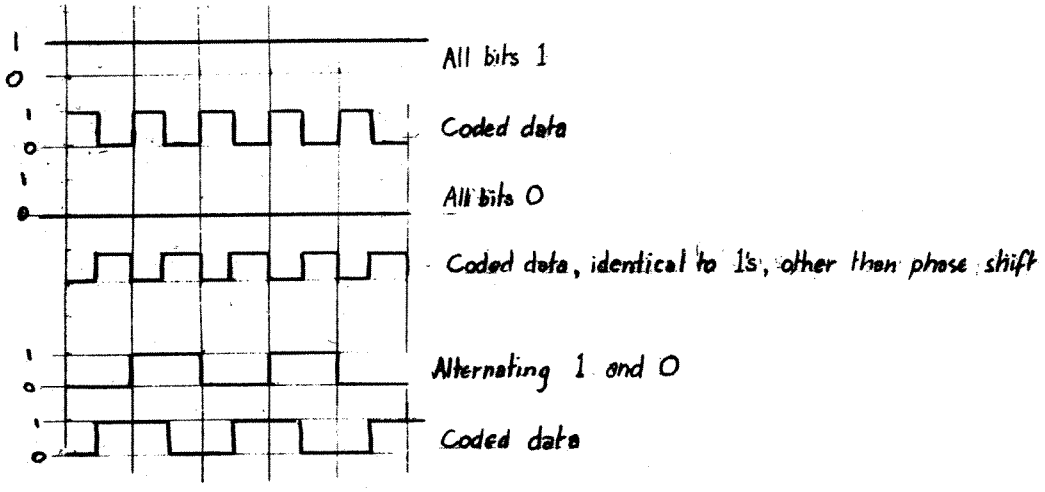
Overall CCPZ is well worth it and a well designed piece of software.

HIGH SPEED CASSETTE INTERFACE

John R. Hunt

I was interested in W Van Malderen's article in Vol.1 Issue.3: it confirmed what I had thought for some time, that with appropriate circuitry data rates significantly higher than Nascom's standard 1200 and 2400 Baud are possible.

I would like to point out a possible problem with the Biphase (Manchester II) coding used, namely that the recorded data for a stream of ones and a stream of zeros are ambiguous, as the following sketch shows.

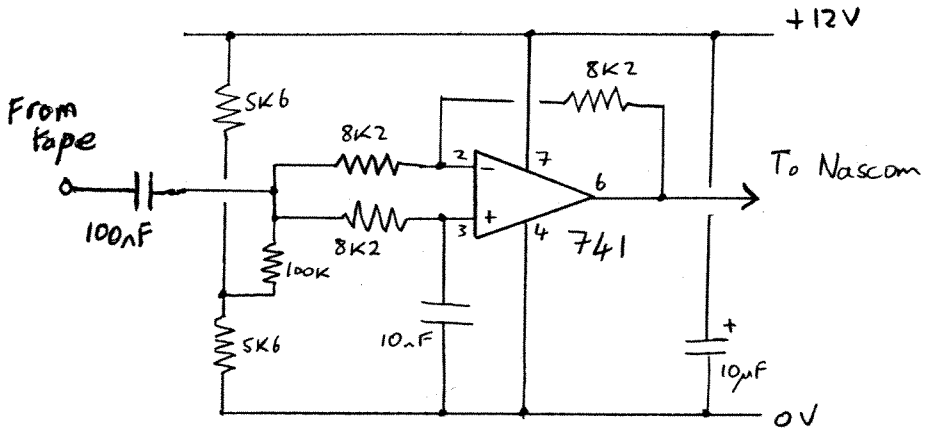


Thus, when the tape is replayed, the 'blank' section before the data starts can be decoded randomly as mark or space, and in the latter case the first data character will be missed. This does not matter for tapes recorded in Nas-Sys format as a block of nulls is output before the data, and these will guarantee synchronisation of the decoder. For other file structures (eg Hisoft Pascal 4?) reliable data recovery will not be achieved.

There is a related Biphase coding method, Biphase-M, which does not suffer from this deficiency, and is also unaffected by possible signal inversion in the recorder. I refer you to an article in the Feb 1982 issue of Wireless World for a discussion of this coding method along with circuits for coder and decoder. Apart from the addition of a phase-locked loop for clock recovery, these are only a little more complex than W Van Malderen's circuit. I intend to experiment with this when time permits.

The article in Wireless World also points out the desirability of phase equilisation of the signal on playback for reliable operation. Based on the Wireless World circiut I have built an equiliser which I have installed permanently between the ouput of my portable recorder and Nascom. The result has been a great improvement in the reliability of reading data tapes, particularly for tapes recorded on other machines. Most significantly, error free reading can now be achieved over a very wide range of volume control settings. The circuit is powered from the 12V supply available on the serial

connector, and is as follows:

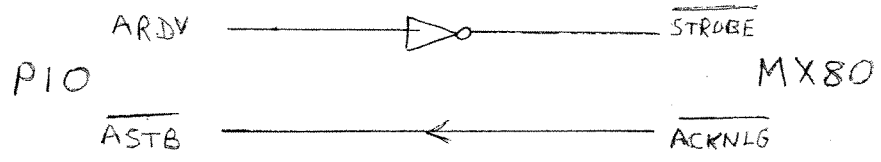


I would also like to correct the impression that a Biphase signal contains only frequencies between half the Baud rate and the Baud rate itself. By chance I had been experimenting with FFT's (Fast Fourier Transforms) and had calculated spectra for NRZ, Biphase and Biphase-M encoded random data streams shortly before the arrival of issue 3. These show that in fact the spectra of both Biphase signals reach a peak at the Baud rate, and have minima at 0 frequency (ie DC) and twice the Baud rate. NRZ on the other hand has a minimum at the Baud rate but is not DC free and so cannot be used for taping data. This is not to say however that Biphase data cannot be recovered after passing it through a bandpass filter such as a tape recorder. (My FFT program is written in Hisoft Tape Pascal 4, and takes about 30 seconds to calculate a 512 point spectrum.)

INTERFACING THE EPSON MX80

John R Hunt

Can it be that I am the only reader of 80-bus News to realise that the PIO's RDY and /STB lines are ideally suited to handshaking with the printer's /STROBE and /ACKNLG lines? All that is required is the addition of an LS TTL inverter:



The port must be in mode 0 (output). Busy should still be taken back to an input line on the other port to allow the driver software to monitor the printer status, unless that is you drive the printer on interrupt. My printer is at the end of 3m of ribbon cable, but I have had no problem with spurious signals or the lack of a program determined settling time.

I get around the inconsistency of some programs using CR and others using CR and LF at line ends as follows. The printer procedure ignores LF's completely, and converts CR to LF before sending it to the printer, which has AUTO FEED off. CH is converted to CR in order to preserve compatibility with Nas-Sys cursor control. The MX80's block graphic character set has different codes to Nas-Graphics. This short procedure converts from Nascom to Epson graphcs:

```
;Nas graph char in A
      SUB #C0
      LD D,A
      LD A,#A0
      LD B,6  ;6 bits
      LD HL,CTAB
C1    SRL D
      JR NC,C2
      OR (HL)
C2    INC HL
      DJNZ C1
;exit with Epsom graph char in A
      RET
CTAB  DEFB 1,4,16,2,8,32
```

Finally, I disagree with David Hunt's comments about XON/XOFF protocol. At work I am a user of small computers (mainly PDP 11), which use only this protocol. Both in terms of cabling costs and interface simplicity it is greatly to be preffered to handshaking with DTR etc. Indeed, few DEC interfaces support handshake lines, and XON/XOFF is the only available method for 20mA current loop communication.

Classified Ads.

==========

These ads. are printed free-of-charge (when space permits) provided that they are for the sale of private property and are not for commercial gain. If you want to advertise games, or that Greenhouse Controller Board then you'll have to pay. Write for details of our super rates.

Nascom RAM card A.
32K fitted + 4K firmware; works at 4MHz without wait states £50 ONO.
Tel: Clive Graham, Crawley 547400.

Nascom 2.
Cased with 48K, graphics, Nas-Sys 3 & 1, Imp printer, Hobbit microcassette system, toolkit, Zeap, Nas-Dis, Naspen, games, spare cassettes, magazines and books. Complete working system, £450 for clearance.
Colchester (0206) 841293.

Cheap printers:
1) Epson MX-70 (80-col + dot gaphics).
2) Golfball Terminal/Typewriter with Serial interface (135 baud - 15cps)
(software and baud rate generator for Nascom also available)
£190 ono each. Mike, 01 874 6244.

NAS-GRAPHPAC

Written by CCsoft, NAS-GRAPHPAC links with the standard Nascom ROM BASIC to provide twenty additional commands for controlling the Nascom 2's low-resolution graphics. We are grateful to CCsoft for giving us permission to publish the object code and System Manual for this package, and to Gemini Microcomputers for giving us permission to publish the Command Manual. The documentation, therefore, is in two parts.

The System Manual.

This gives specific details for implementing Graphpac on the Nascom and a summary of relevant commands.

The Command Manual.

This gives the general description of the commands and features provided by all current versions of Graphpac.

Copyright and right to make backup copies

On receipt of this computer program and associated documentation, you are granted a non-exclusive license to execute the enclosed software and to make backup copies of the computer program for single user use only. Mbaslink and Graphpac are subject to copyright and all rights are reserved.

Disclaimer

No representations or warranties with respect to the contents hereof are made and any implied warranties of merchantability or fitness for any particular purpose are specifically disclaimed.

NAS-GRAPHPAC SYSTEM MANUAL

INTRODUCTION

This package has been introduced for the Nascom microcomputer to give greater flexibility while controlling the low resolution graphics. It is recommended that the user becomes familiar with the use of the Nascom system and ROM BASIC before any attempts are made to run Nas-Graphpac. Nas-Graphpac in fact consists of two modules, Graphpac, the graphics subroutines and also Mbaslink which handles all the interfacing with the Nascom ROM BASIC.

Existing graphics commands (SET, RESET and POINT) are still available while Nas-Graphpac is active, existing BASIC programs will therefore run without modification.

The MONITOR command will re-initialise Nas-sys by jumping to location zero instead of doing an SCAL MRET.

HARDWARE REQUIREMENTS

This package requires a Nascom 1 or 2 fitted with the Nascom graphics ROM and at least 16k of RAM.

MEMORY USAGE

The program is supplied in relocatable format and once configured for the host system, it will be 2K in length and can reside anywhere in the memory map above 4800 hex. (including ROM if required). Workspace RAM can optionally be set at 0D00, 0E00, 0F00 hex or just below the relocated package.

COMPATIBILITY WITH OTHER SOFTWARE

Nas-Graphpac is fully software compatible with the Nas-sys series of Monitors, however, because provision was not made for extending the command table in the Nascom ROM BASIC it has been necessary to use the flexibility of the Nas-sys monitor to achieve the interface. The method used is to copy out the SCAL table from the Nas-sys ROM into workspace RAM. Various modifications are then made to the entries allowing BASIC to link with Nas-Graphpac. Once Nas-Graphpac is active on the system it fools the Nascom ROM BASIC into thinking a non-maskable interrupt has occurred. Control then passes to Nas-Graphpac via the Nas-sys SCAL table. As the SCAL table is copied into RAM and modified, it is unlikely that other Nascom software that also does this, will be compatible.

Because of the number of different system software and hardware options available on the Nascom it is impossible to support them all, Nas-Graphpac has therefore only been tested on a standard cassette based system running an unmodified Nas-sys monitor. Having said this, we would like to hear of successful interfacing to Polydos 1, Polydos 2, Nas-dos etc.

If Nas-Graphpac does not operate correctly on your system because of software incompatibility with your uprated operating system, all is not lost. A USR mode has been incorporated inside Nas-Graphpac and graphic commands can be used via the USR function. See appendix 3 at the back of this document for detailed descriptions.

CONFIGURING NAS-GRAPHPAC

Load Nas-Graphpac into memory from the listing given at 1000 hex and using the Nas-sys Write command make a tape copy. Decide where you want Nas-Graphpac to execute on your system. Normally this will be in the top 2k of available RAM which will depend on the size of your memory board and whether you have system firmware in ROM (Nas-Dis/Debug or Zeap). Execute Nas-Graphpac at 1000 hex followed by the address where the program is to eventually run:

E 1000 7000 (enter)

will generate a Nas-Graphpac that will run at 7000 to 77FF hex.

The Graphpac workspace will be located just below the relocated package at 6F00 hex.

If it is intended to run Nas-Graphpac in ROM it will be necessary to locate the Graphpac workspace in the low memory area 0D00 - 1000 hex. This can be achieved by entering the following command line.

E 1000 aaaa L (enter)

where aaaa is the address at which the program will run and where L is the letter D,E or F to indicate which 100 hex bytes in the low memory area is to be used. D equates to 0D00 hex etc. This decision will depend on what other software is running on the system.

The relocation process in both cases is straight forward and when complete the user will be prompted to make a tape copy of the generated version at its execution address (a copy is placed at 4000 hex. if RAM is not available at the target address). Once this tape is made the original Nas-Graphpac tape will not be needed unless a new version is required at a different address, and should be kept in a safe place as a master backup tape.

WARNING....

It is essential that you do not ask the relocator to configure Nas-Graphpac so that it crosses the 32K address boundary. If you have a 16k Nascom system you can only configure Nas-Graphpac to start at 4800 hex. If you have a Nascom equipped with 32K or more, never ask the relocator to address Graphpac between 7801 hex and 7FFF hex. as it is probable that the package will not work correctly.

EXECUTING NAS-GRAPHPAC (COLD START)

It is important that if CCsoft's Nas-Debug is active on the system, it should be disabled by pressing RESET or executing Nas-sys at address zero or by using the Nas-sys N command before Nas-Graphpac is used.

Your generated Nas-Graphpac must first be loaded using the Nas-sys READ command. Execute Nas-Graphpac at the first address that was displayed during the READ process. i.e.

E 7000 (enter)

where Graphpac was relocated originally at 7000 hex.

Nas-Graphpac will be initialised together with the BASIC and an attempt to set the MEMORY SIZE at 101 hex bytes below Nas-Graphpac will be made irrespective of the position of the Nas-Graphpac workspace. If Nas-Graphpac resides in ROM, the MEMORY SIZE message will be printed and the normal BASIC cold start procedure is used.

If it is wished to set the MEMORY SIZE manually this can be achieved by entering a second argument on execution.

E 7000 5FFF (enter)

Graphpac will be executed at address 7000 hex and the MEMORY SIZE will be set at 5FFF hex. This will be confirmed on entry to BASIC as the free memory size will be printed in decimal. If the second argument entered is invalid, the MEMORY SIZE message will be displayed and the user will be prompted to enter the correct value (in decimal). DO NOT just press the enter key without entering a decimal value if this point is reached, unless Nas-Graphpac resides in ROM.

EXECUTING NAS-GRAPHPAC (warm start)

If the user has returned to Nas-sys, Nas-Graphpac and BASIC may be re-entered by executing Nas-Graphpac at it's normal execution address. (The same as cold start.)

E 7000 (enter)

A second argument should not be entered unless the user is aware of the consequences as explained in appendix 2.

ENTERING NAS-GRAPHPAC COMMANDS

Nas-Graphpac commands can be entered in the DIRECT MODE or as part of a statement line. If the command is to be entered while in the DIRECT MODE, it must ALWAYS be preceded by a colon (:) otherwise a syntax error will occur. Nas-Graphpac commands can be used as part of a statement line using normal BASIC conventions except a colon must precede a Nas-Graphpac command that follows the word THEN in an IF THEN statement.

BASIC variable and array names should not match Nas-Graphpac command names. (although this is usually acceptable if a LET is not implied and the word LET is used in a statement line.)

BASIC PROGRAM EXECUTION TIME

Every care has been taken to ensure the graphic routines execute as fast as possible and speed problems should not be encountered while actually producing graphic displays, however, there is a time overhead caused by searching the Graphpac command table, before deciding whether Graphpac or BASIC is to take control. In long program loops that do not include Nas-Graphpac commands (i.e. setting up a large array) it may be worth while disabling Graphpac temporarily. This is achieved by the following line which POKE's the N.M.I. flag in BASIC's workspace.

POKE 4173,0

Before a Nas-Graphpac command is executed, the N.M.I. flag must again be modified with the following line

POKE 4173,1

This facility should only be used in a program where the execution speed is critical. An INPUT statement, keyboard interrupt or a return to the direct mode will cause Graphpac to be enabled.

Another problem with program speed is associated with screen blanking. It is possible to plot and erase short lines on the screen extremely quickly, so much so, that if this process is repeated in a FOR NEXT loop the line may appear to be plotted in slow motion due to the blanking effect. The following program illustrates the problem (on a 4 Mhz Nascom 2).

```
10 CLS : STARTAT 48,23 : PENFLIP
20 FOR A=1 TO 400
30 DRAW 8,8 : PR : NEXT
```

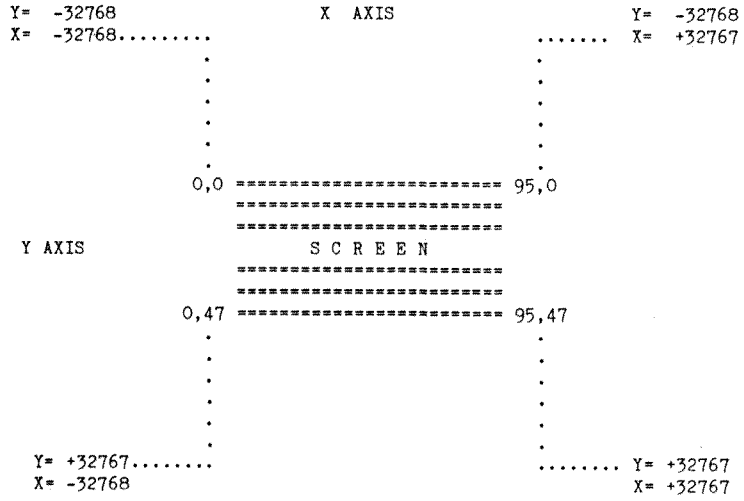
Insert extra colons before the PENRET command to vary the timing and observe the results.

PLOTTING ON THE SCREEN

Graphic co-ordinates have been changed slightly while using Nas-Graphpac. "Y" co-ordinates 45, 46 and 47 no longer appear on the top unscrolled line of the screen but are found in their correct positions near the bottom of the screen. This should not affect existing software as SET, RESET and POINT are unchanged and use the old graphics co-ordinates. X,Y co-ordinates are still numbered in the usual way. co-ordinate 0,0 is found in the top left hand corner and 95,47 is found in the bottom right corner. An option is available whereby attempting to plot at X,Y co-ordinates off of the screen will not cause an FC error. With this facility it is quite possible to place the CAP at X,Y co-ordinates 200,23 and draw a circle using the CIRCLE command with a radius of 175. Part of the circle will appear on the screen. Arguments given to Nas-Graphpac commands must not exceed 32767 or BASIC will cause an FC error.

The location which has to be PEEKed after a PTEST command is executed can be found the the ROM BASIC workspace at 4177 decimal (1051 hex).

The following chart may help you to understand the physical relationship between graphics co-ordinates.



APPENDIX 1

The following commands are included in NAS-GRAPHPAC. Detailed descriptions on each command are given in the Command Manual. The standard BASIC "SCREEN" statement remains operational and therefore the documentation in the Command Manual relating to the Graphpac "SCREEN" command should be ignored.

GRAPHICS COMMAND SUMMARY

GCLR	Clear screen of graphic characters
FCON	Cause FCerr when plotting off the screen
FCOFF	Plotting off screen is not trapped
PSET X,Y	Set bright the specified pixel
PRESET X,Y	Reset dark the specified pixel
STARTAT X,Y	Position CAP at specified co-ordinate
SA X,Y	
PENUP	This pen mode causes no action while moving CAP
PU	
PENDOWN	Prepare to set pixels following the CAP's movement
PD	
PENFLIP	Prepare to invert pixels
PF	
PENERA	Prepare to erase pixels
PE	
PENRET	Restore previous CAP co-ordinates
PR	
DRAWTO X,Y	Move CAP to new X,Y
DRAW X,Y	Move the CAP relative to the current X,Y
PLOT A,D	Move the CAP to new X,Y based on Angle,Distance
DOCAP	The pixel at the CAP is set, reset or inverted
RATIO n	Correction factor for CIRCLE and PLOT
CIRCLE R,A1,A2	Draw a circle or arc
CAP	Print CAP on the screen
PTEST X,Y:PEEK(4177)	Test specified pixel using location 1051 hex.

APPENDIX 2

USING THE PARKINSON TOOLKIT WITH NAS-GRAPHPAC

There are several different BASIC toolkits available for the Nascom from various sources, none of them official Nascom software products. One of the more popular toolkits written by David Parkinson and available through your local Microvalue Dealer, may be used with Graphpac.

The COLD start procedure is more complicated than normal if the Toolkit is to be used and the method need not only be used with this software but can be used with any other routines which needs to be initialised at the same time as BASIC and Nas-Graphpac.

First Load Nas-Graphpac at it's execution address (if not in ROM).

Load the toolkit at 1200 hex and execute. The second argument must be used to limit the amount of memory used by BASIC to ensure that Nas-Graphpac is not over-written. If Nas-Graphpac was loaded at 9000 hex use the following line.

E 1200 8EFF (enter)

This leaves Nas-Graphpac and the workspace situated in the memory below it untouched. BASIC will now be initialised and entered. Use the MONITOR command to restore control to Nas-sys.

Execute Nas-Graphpac at it's normal execution address + 3. In the example above this would be 9003 hex. The second argument must specify the address to jump to after Nas-Graphpac has been initialised, in this case, the Toolkit. The line would therefore read as follows, the Toolkit warm start being 2 below the maximum address of the toolkit program.

E 9003 8EFD (enter)

All Nas-Graphpac and Toolkit commands can now be used.

Graphpac commands are not reduced in size to one byte as BASIC reserved words are. For this reason Parkinson's Toolkit (or indeed any other) cannot recognise Nas-Graphpac command words, these will appear as two letter variable names while using the cross reference facility.

APPENDIX 3

USING THE USR FUNCTION TO OPERATE NAS-GRAPHPAC.

If for any reason Nas-Graphpac does not run correctly on your system due to, for example, software associated with your discs or high speed tape interface, use the BASIC USR function. Programs will have to be written slightly differently and commands will not be readily available in the direct mode, but all facilities are still available.

Load the relocated Nas-Graphpac into memory at it's execution address. COLD start BASIC, being careful to set the MEMORY SIZE correctly so that Nas-Graphpac and it's workspace are not over-written. The next stage can best be described by the use of a short BASIC program.

Lets assume that Nas-Graphpac has been loaded at 9000 hex. and that BASIC has already been entered.

```
10 DOKE 4100, -28666
20 CLS
30 C=USR(0): STARTAT 48,23: PENFLIP
40 FOR A= 0 TO 359
50 C=USR (0): PLOT A,18: PENRET
60 NEXT A: GOTO 40
70 END
```

Line 10 sets up the address that the USR will jump to. In our case as Nas-Graphpac was relocated for 9000 hex the address to use was 9000 + 6, which equates to -28666 decimal.

Now every time that a USR function is now reached Nas-Graphpac is entered. C is a dummy variable but any variable name may be used.

The whole of the remaining line MUST contain only valid Nas-Graphpac commands or a syntax error will occur. Normal BASIC statements cannot be mixed with Nas-Graphpac commands. Any number of Nas-Graphpac commands may be included in a line following a USR provided they are separated by a colon. Nas-Graphpac commands can be used in the direct mode but C=USR(0) must still be typed, which makes the facility almost useless.

```
1000  C3 0A 10 32 00 00 00 00 - 00 FF 3E FF 32 09 10 3A  C..2.... ..>.2..:
1010  0B 0C FE 01 20 32 EF 20 - 53 65 63 6F 6E 64 20 61  ..~. 2o  Second a
1020  72 67 20 67 69 76 69 6E - 67 20 4F 52 47 20 61 64  rg givin g ORG ad
1030  64 72 65 73 73 20 6D 75 - 73 74 20 62 65 20 67 69  dress mu st be gi
1040  76 65 6E 2E 0D 00 DF 5B - 2A 0E 0C 7C FE 48 30 2E  ven...[ *..|~HO.
1050  EF 73 70 65 63 69 66 79 - 20 61 6E 20 61 64 64 72  ospecify  an addr
1060  65 73 73 20 67 72 65 61 - 74 65 72 20 74 68 61 6E  ess grea ter than
1070  20 34 37 46 46 20 48 65 - 78 2E 0D 00 DF 5B FE 80   47FF He x..._[~.
1080  30 05 3E 7F 32 C2 14 22 - 05 10 11 00 08 19 22 03  0.>.2B." ......".
1090  10 2B 7E 2F 77 BE 2F 77 - 28 43 EF 47 65 6E 65 72  .+~/w>/w (CoGener
10A0  61 74 65 64 20 76 65 72 - 73 69 6F 6E 20 77 69 6C  ated ver sion wil
10B0  6C 20 6E 6F 74 20 72 75 - 6E 20 69 6E 20 52 41 4D  l not ru n in RAM
10C0  20 6F 6E 0D 74 68 69 73 - 20 73 79 73 74 65 6D 0D   on.this  system.
10D0  00 AF 32 09 10 3A 0B 0C - FE 03 C2 72 11 3A 0B 0C  ./2..:.. ~.Br.:..
10E0  2A 05 10 11 00 01 B7 ED - 52 EB FE 03 DA 10 11 2A  *.....7m Rk~.Z..*
10F0  10 0C 7C B7 C2 72 11 3E - 0D BD 11 00 0D 28 11 3E  ..|7Br.> .=...(.>
1100  0E BD 11 00 0E 28 09 3E - 0F BD 11 00 0F C2 72 11  .=...(.> .=...Br.
1110  EB 22 07 10 01 E5 12 21 - D3 1A 23 5E 23 56 7A B3  k"...e.! S.£^£Vz3
1120  28 13 EB 09 D5 5E 23 56 - E5 2A 05 10 19 EB E1 72  (.k.U^£V e*...kar
1130  2B 73 E1 18 E5 21 D1 1B - 11 36 00 B7 ED 52 23 5E  +sa.e!Q. .6.7mR£^
1140  23 56 7A B3 28 0E EB 09 - D5 5E 23 56 73 2B CB FA  £Vz3(.k. U^£Vs+Kz
1150  72 E1 18 EA 21 F3 1B 23 - 5E 23 56 7A B3 28 37 EB  ra.j!s.£ ^£Vz3(7k
1160  09 D5 5E 23 56 E5 2A 07 - 10 19 EB E1 72 2B 73 E1  .U^£Ve*. ..kar+sa
1170  18 E5 EF 53 70 65 63 69 - 66 69 65 64 20 77 6F 72  .eoSpeci fied wor
1180  6B 73 70 61 63 65 20 69 - 73 20 69 6E 76 61 6C 69  kspace i s invali
1190  64 2E 0D 00 DF 5B 3A 09 - 10 3C 20 0C 21 E5 12 ED  d..._[:. .< .!e.m
11A0  5B 05 10 01 EF 07 ED B0 - 21 E5 12 11 00 40 01 EF  [...o.mO !e...%.o
11B0  07 ED B0 3A 09 10 3C 28 - 3C 21 00 40 22 05 10 21  .mO:..<( <!.%"..!
11C0  00 48 22 03 10 EF 47 72 - 61 70 68 70 61 63 20 73  .H"..oGr aphpac s
11D0  75 69 74 61 62 6C 65 20 - 66 6F 72 20 52 4F 4D 20  uitable  for ROM
11E0  61 74 20 34 30 30 30 20 - 2D 20 34 38 30 30 20 68  at 4000  - 4800 h
11F0  65 78 2E 0D 00 EF 52 65 - 6C 6F 63 61 74 69 6F 6E  ex...oRe location
1200  20 63 6F 6D 70 6C 65 74 - 65 0D 0D 14 14 57 20 00   complet e....W .
1210  2A 05 10 DF 66 2A 03 10 - DF 66 EF 0D 13 13 00 DF  *.._f*.. _fo.....
1220  5B 00 00 00 00 00 00 00 - 00 30 00 17 00 00 00 00  [........ .0......
1230  00 00 00 30 00 17 00 00 - 00 00 00 00 00 00 00 FF  ...0.... ........
1240  C9 00 30 00 17 00 00 00 - 00 00 00 00 00 00 00 00  I.0..... ........
1250  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
1260  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
1270  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
1280  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
1290  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
12A0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
12B0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
12C0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
12D0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
12E0  00 00 00 00 00 C3 09 00 - C3 11 00 C3 E7 01 21 2A  .....C.. C..Cg.!*
12F0  00 3E AA BE 20 77 2A 0B - 10 7D 84 FE EB 20 13 CD  .>*> w*. .|.~k .M
1300  D8 00 21 0B 0C 46 AF 77 - 3E 01 B8 CA FD FF 2A 0E  X.!..F/w >.8J}.*.
1310  0C E9 3A 2A 00 FE AA 28 - 1C EF 20 42 41 53 49 43  .i:*.~*( .o BASIC
1320  20 4E 4F 54 20 69 6E 69 - 74 69 61 6C 69 73 65 64   NOT ini tialised
1330  2E 0D 00 DF 5B EF 20 42 - 41 53 49 43 20 77 6F 72  ..._[o B ASIC wor
1340  6B 73 70 61 63 65 20 63 - 6F 72 72 75 70 74 69 6F  kspace c orruptio
1350  6E 0D 20 46 6F 72 63 65 - 64 20 43 4F 4C 44 20 73  n. Force d COLD s
1360  74 61 72 74 2E 0D 00 32 - 2A 00 C3 00 00 77 21 DE  tart...2 *.C..w!^
1370  FE 22 7E 0C 21 DF E2 01 - 63 00 11 00 10 ED B0 EB  ~"~.!_b. c....mOk
1380  F9 CD DF E4 CD 81 EB 32 - AA 10 32 F9 10 CD D8 00  yM_dM.k2 *.2y.MX.
1390  21 00 00 77 BE 3E C3 77 - 3A 0B 0C 47 3E 00 32 0B  !..w>>Cw :..G>.2.
13A0  0C C2 36 E0 78 FE 02 21 - FF FE 20 0E 2A 0E 0C 7E  .B6`x~.! .~ .*..~
13B0  2F 77 BE 2F 77 3E 00 C2 - 36 E0 C3 6D E0 EF 20 4D  /w>/w>.B 6`Cm`o M
13C0  62 61 73 6C 69 6E 6B 20 - 31 2E 30 20 2F 20 4E 61  baslink  1.0 / Na
13D0  73 2D 47 72 61 70 68 70 - 61 63 20 31 2E 30 0D 20  s-Graphp ac 1.0.
13E0  43 43 73 6F 66 74 2E 20 - 28 43 29 20 31 39 38 32  CCsoft.  (C) 1982
13F0  0D 00 2A 71 0C 11 82 00 - 19 01 90 00 11 34 00 ED  ..*q.... .....4.m
1400  B0 11 67 01 2A A8 00 22 - 06 00 ED 53 A8 00 21 B2  0.g.*(." ..mS(.!2
1410  FF 22 71 0C 21 00 00 22 - 68 00 3E C3 32 02 00 32  ."q.!.." h.>C2..2
1420  05 00 32 4D 10 21 30 00 - 22 08 00 22 12 00 21 17  ..2M.!0. "..".."..!.
1430  00 22 0A 00 22 14 00 21 - 00 00 22 1F 00 3E FF 32  ."..".."..! ..">.2
1440  29 00 32 1E 00 21 00 B6 - 22 2F 00 C9 E5 3E F9 32  ).2..!.6 "/.Ie>y2
1450  4D 10 21 0E 00 39 BE 20 - 11 23 7E FE E5 38 0B FE  M.!..9>  .£~~e8.~
1460  E7 28 0C CD 70 FD 28 02 - E1 C9 CD 26 02 E1 C9 CD  g(.Mp}(. aIM&.aIM
1470  20 02 E1 FE 1B C8 7E FE - 3A 28 13 B7 C2 AD E3 23   .a~.H~~ :(.7B-c£
1480  7E 23 B6 CA 7A E8 23 5E - 23 56 ED 53 5C 10 EB 13  ~£6Jzh£^ £VmS\.k.
1490  CD 29 02 CB 7F 20 5B 21 - 19 02 22 2D 00 21 0C 02  M).K. [! .."-.!..
14A0  22 03 00 21 55 07 CD 29 - 02 D5 1A 13 23 E6 7F BE  "..!U.M) .U..£f.>
14B0  28 F8 1B 3E 7F BE 38 08 - 23 BE 30 FC 23 D1 18 E6  (x.>.>8. £>0}£Q.f
14C0  7E E6 FF 23 6E 67 E3 2A - 2D 00 E3 E9 C1 E1 E5 7E  ~f.£ngc* -.ciAae~
14D0  23 A7 20 FB 2B E3 EB C5 - 21 FF 01 22 2D 00 21 AD  £' {+ckE !.."-.!-
14E0  E3 22 03 00 CD 29 02 13 - FE 3A 28 B7 B7 C2 AD E3  c"..M).. ~:(77B-c
14F0  C9 F1 EB D1 D1 C1 F1 F1 - F1 F1 2B C3 16 E8 CD 29  IqkQQAqq qq+C.hM)
1500  02 EB C3 91 01 CD 70 FD - C0 DF 62 C3 05 00 1A FE  .kC..Mp} %_bC...~
1510  20 C0 13 18 F9 CD 44 02 - EB CD 90 E6 2C CD 51 02   %..yMD. kM.f,MQ.
1520  EB 2A 0C 00 ED 4B 0E 00 - C9 FD 21 0C 00 CD 29 02  k*..mK.. I}!..M).
1530  EB CD 51 02 EB C9 CD 41 - ED CD 8B E9 FD 73 00 FD  kMQ.kIMA mM.i}s.}
1540  23 FD 72 00 FD 23 C9 CD - BB 02 CD 30 02 D5 CD 74  £}r.}£IM ;.MO.UMt
1550  05 CD 5E 03 D1 CD 29 02 - FE 3B C0 13 C3 62 02 CD  .M^.QM). ~;%.Cb.M
1560  BB 02 CD 30 02 D5 ED 5B - 08 00 19 22 21 00 2A 0A  ;.MO.Um[ ..."!.*.
1570  00 09 22 23 00 CD 5E 03 - D1 CD 29 02 FE 3B C0 13  .."£.M^. QM).~;%.
1580  C3 7A 02 CD BB 02 CD 30 - 02 D5 22 21 00 60 69 22  Cz.M;.MO .U"!.`i"
1590  23 00 CD 5E 03 D1 CD 29 - 02 FE 3B C0 13 C3 9E 02  £.M^.QM) .~;%.C..
15A0  2A 08 00 22 12 00 2A 0A - 00 22 14 00 C9 CD 30 02  *.."..*. ."..IMO.
15B0  AF 32 1E 00 CD 33 02 D5 - 22 0E 00 ED 43 0C 00 2A  /2..M3.U "..mC..*
15C0  08 00 22 16 00 2A 0A 00 - 22 18 00 2A 0C 00 CD 74  .."..*.. "..*..Mt
15D0  05 CD 2A 03 2A 0C 00 ED - 5B 10 00 B7 ED 52 19 30  .M*.*..m [..7mR.0
15E0  27 23 23 22 0C 00 CD 74 - 05 2A 1A 00 22 08 00 2A  '££"..Mt .*.."..*
15F0  1C 00 22 0A 00 CD E0 03 - CD 2A 03 ED 4B 18 00 2A  .."..M`. M*.mK..*
1600  16 00 CD 58 03 C3 EF 02 - D1 3E FF 32 1E 00 C9 2A  ..MX.Co. Q>.2..I*
1610  21 00 22 1A 00 2A 23 00 - 22 1C 00 C9 CD 44 02 2A  !."..*£. "..IMD.*
1620  0C 00 7C E6 01 67 22 1F - 00 C9 ED 4B 14 00 2A 12  ..|f.g". .ImK..*.
1630  00 18 0A CD 30 02 ED 43 - 14 00 22 12 00 ED 43 0A  ...MO.mC ."..mC.
1640  00 18 0D 3A 30 00 B7 C2 - E0 03 22 0A 00 2A 21 00  ...:0.7B `."..*!.
1650  22 08 00 C9 21 2F A6 22 - 2F 00 C9 21 00 AE 18 F7  "..I!/&" /.I!...w
1660  21 00 B6 18 F2 21 00 00 - 18 ED 01 00 B6 CD A4 03  !.6.r!.. .m..6M$.
1670  2A 31 00 22 2F 00 C9 01 - 2F A6 18 F1 01 00 A6 3E  *1."/.I. /&.q..&>
1680  02 32 51 10 CD A4 03 18 - E7 2A 2F 00 22 31 00 ED  .2Q.M$.. g*/."1.m
1690  43 2F 00 CD 30 02 D5 EB - 60 69 CD CC 04 CD BD 03  C/.MO.Uk `iML.M=.
16A0  D1 C9 3A 29 00 B7 C8 3A - 33 00 B7 C8 3A 1E 00 3C  QI:).7H: 3.7H:..<
16B0  CA A0 E9 ED 4B 18 00 2A - 16 00 CD 58 03 3E FF 32  J imK..* ..MX.>.2
16C0  1E 00 C3 A0 E9 2A 08 00 - ED 5B 21 00 01 00 04 CD  ..C i*.. m[!....M
16D0  6B 04 E5 2A 0A 00 ED 5B - 23 00 06 02 CD 6B 04 D1  k.e*..m[ £...Mk.Q
16E0  B7 ED 52 19 38 02 EB 0C - 22 27 00 ED 53 25 00 13  7mR.8.k. "'.mS%..
16F0  D5 CB 3A CB 1B 21 9C 04 - 79 87 87 4F 06 00 09 22  UK:K.!.. y..O..."
1700  2B 00 0E 00 C5 D5 2A 0A - 00 ED 5B 08 00 CD CC 04  +...EU*. .m[..ML.
1710  CD BD 03 C1 D1 E1 2B 7C - B5 C8 E5 2A 27 00 19 E5  M=.AQa+| 5He*'..e
1720  AF ED 42 FA 49 04 2A 25 - 00 09 44 4D 3E 02 C5 2A  /mBzI.*% ..DM>.E*
1730  2B 00 85 6F 7E 23 5E 4F - 17 9F 47 2A 08 00 09 22  +..o~£^O ..G*..."
1740  08 00 7B 17 9F 57 2A 0A - 00 19 22 0A 00 C3 24 04  ..{..W*. .."..C$.
1750  7A AC F2 80 04 CB 7A 28 - 04 EB C3 7B 04 78 81 4F  z,r..Kz( .kC{.x.O
1760  CD 0B 06 19 C9 CB 7C CA - 8C 04 CD 0B 06 EB CD 0B  M...IK|J ..M..kM.
1770  06 B7 ED 52 D0 19 EB CB - 7C 20 03 78 81 4F ED 52  .7mRP.kK | .x.OmR
```

```
1780  C9 FF 00 FF FF 00 FF FF - FF FF 00 FF 01 00 01 FF  I....... ........
1790  01 01 00 01 FF 00 FF 01 - FF 01 00 01 01 00 01 01  ........ ........
17A0  01 D5 2A 0A 00 ED 5B 08 - 00 CD CC 04 CD BD 03 D1  .U*..m[. .ML.M=.Q
17B0  C9 7B D6 60 3E FF 32 33 - 00 D0 7A B7 C0 7D D6 30  I{V`>.23 .Pz7%}VO
17C0  D0 7C B7 C0 AF 32 33 00 - 3A 30 00 B7 C8 FE B6 28  P|7%/23. :0.7H~6(
17D0  1B FE AE 28 08 3A 2F 00 - B7 28 23 18 18 D5 E5 7B  .~.(.:/. 7(£..Ue{
17E0  45 CD 19 05 E1 D1 FE 02 - C8 B7 20 09 7B 45 CD 2D  EM..aQ~. H7 .{EM-
17F0  05 D8 B1 77 C9 7B 45 CD - 2D 05 D8 A0 77 C9 7B 45  .X1wI{EM -.X wI{E
1800  CD 2D 05 38 07 A1 28 06 - 3E 01 18 02 3E 02 32 51  M-.8.!(. >...>.2Q
1810  10 C9 B7 1F 0E 07 30 02 - 0E 38 6F 78 06 03 26 11  .I7...0. .8ox..&.
1820  25 90 30 FC C6 04 47 3E - 84 07 10 FD A1 4F 7C B7  %.0|F.G> ...}!O|7
1830  37 F8 7D FE 30 3F D8 44 - 11 40 00 21 CA 0B ED 52  7x}~0?XD .%.!J.mR
1840  10 FC F5 3E 07 BC 20 03 - 21 CA 0B F1 5F 19 79 2F  .|u>.< . !J.q_.y/
1850  47 7E FE C0 D0 3E C0 B7 - C9 01 68 01 B7 CB 7C 28  G~~%P>%7 I.h.7K|(
1860  07 ED 4A FA 7C 05 18 06 - ED 42 F2 83 05 09 B7 11  .mJz|... mBr...7.
1870  5A 00 06 FF 04 ED 52 F2 - 8F 05 19 E5 C5 CD C6 05  Z....mRr ...eEMF.
1880  08 DC 0B 06 ED 5B 08 00 - 19 22 21 00 C1 04 E1 CD  .\..m[.. ."!.A.aM
1890  C6 05 ED 5B 1F 00 7A B3 - C4 E8 05 08 DC 0B 06 ED  F.m[..z3 Dh..\..m
18A0  5B 0A 00 EB B7 ED 52 22 - 23 00 C9 CB 40 CA D3 05  [..k7mR" £.IK%JS.
18B0  11 C7 06 CD 0B 06 18 03 - 11 13 06 EB 19 19 7E 23  .G.M.... ...k..~£
18C0  66 6F B7 CB 48 28 01 37 - 08 ED 5B 0E 00 44 4D 21  fo7KH(.7 .m[..DM!
18D0  00 00 3E 09 CB 42 28 04 - 09 30 01 1C 3D 28 0A EB  ..>.KB(. .0..=(.k
18E0  29 EB 29 30 EF 1C C3 EF - 05 6C 63 CB 3C CB 1D C9  )k)0o.Co .lcK<K.I
18F0  7D 2F 6F 7C 2F 67 23 C9 - 00 00 09 00 12 00 1B 00  }/o|/g£I ........
1900  24 00 2D 00 35 00 3E 00 - 47 00 50 00 59 00 62 00  $.-.5.>. G.P.Y.b.
1910  6A 00 73 00 7C 00 84 00 - 8D 00 96 00 9E 00 A7 00  j.s.|... ......'.
1920  AF 00 B7 00 C0 00 C8 00 - D0 00 D8 00 E0 00 E8 00  /.7.%.H. P.X.`.h.
1930  F0 00 F8 00 00 01 08 01 - 0F 01 17 01 1E 01 26 01  p.x..... ......&.
1940  2D 01 33 01 3B 01 42 01 - 49 01 50 01 57 01 5D 01  -.3.;.B. I.P.W.].
1950  64 01 6A 01 70 01 76 01 - 7C 01 82 01 88 01 8E 01  d.j.p.v. |.......
1960  93 01 99 01 9E 01 A3 01 - A8 01 AD 01 B2 01 B7 01  ......£. (.-.2.7.
1970  BB 01 C0 01 C4 01 C8 01 - CC 01 D0 01 D1 01 D7 01  ;.%.D.H. L.P.Q.W.
1980  DB 01 DE 01 E1 01 E4 01 - E7 01 EA 01 EC 01 EE 01  [.^.a.d. g.j.l.n.
1990  F1 01 F3 01 F5 01 F7 01 - F8 01 FA 01 FB 01 FC 01  q.s.u.w. x.z.{.|.
19A0  FD 01 FE 01 FF 01 FF 01 - 00 02 00 02 00 02 D5 21  }.~..... ......U!
19B0  0A 08 01 F0 03 CB 7E 28 - 02 36 20 ED A1 EA D0 06  ...p.K~( .6 m!jP.
19C0  AF D1 C9 3E FF 18 01 AF - 32 29 00 C9 D5 2A 08 00  /QI>.../ 2).IU*..
19D0  CD FB 06 2A 0A 00 CD FB - 06 3E 0D CD 4D 07 D1 C9  M{.*..M{ .>.MM.QI
19E0  CB 7C 28 0C 3E 2D CD 4D - 07 11 00 00 EB B7 ED 52  K|(.>-MM ....k7mR
19F0  01 05 00 11 10 27 CD 30 - 07 11 E8 03 CD 30 07 11  .....'MO ..h.MO..
1A00  64 00 CD 30 07 11 0A 00 - CD 30 07 11 01 00 CD 30  d.MO.... MO....MO
1A10  07 3E 20 18 1D 3E FF 0D - B7 3C ED 52 30 FB 19 F6  .> ..>.. 7<mRO{.v
1A20  30 FE 30 20 04 CB 40 28 - 04 CB C0 18 05 79 B7 C0  O~O .K%( .K%..y7%
1A30  3E 30 C5 E5 F7 E1 C1 C9 - C3 02 00 44 52 41 57 26  >OEewaAI C..DRAW&
1A40  9E 02 44 52 41 57 7A 02 - 50 4C 4F 54 62 02 53 54  ..DRAWz. PLOTb.ST
1A50  41 52 54 41 54 4E 03 44 - 4F 43 41 50 BC 04 50 44  ARTATN.D OCAP<.PD
1A60  7B 03 50 55 80 03 50 46 - 76 03 50 00 4F 57 4E 7B  {.PU..PF v.P.OWN{
1A70  03 50 45 4E 55 50 80 03 - 50 45 4E 46 4C 49 50 76  .PENUP.. PENFLIPv
1A80  03 50 45 4E 52 45 54 45 - 03 50 45 4E 45 52 41 6F  .PENRETE .PENERAo
1A90  03 50 45 6F 03 50 1C 85 - 03 50 1D 92 03 47 43 4C  .PEo.P.. .P...GCL
1AA0  52 C9 06 53 41 4E 03 50 - 54 45 53 54 97 03 43 49  RI.SAN.P TEST..CI
1AB0  52 43 4C 45 C8 02 50 52 - 45 03 43 41 50 E7 06 46  RCLEH.PR E.CAPg.F
1AC0  43 11 DE 06 46 43 4F 46 - 46 E2 06 52 41 54 49 4F  C.^.FCOF Fb.RATIO
1AD0  37 03 53 07 01 00 04 00 - 07 00 1B 00 86 00 A9 00  7.S..... .......).
1AE0  AC 00 C3 00 1D 01 86 01 - 8B 01 AC 01 B3 01 B9 01  ,.C..... ..,.3.9.
1AF0  BF 01 C2 01 F4 01 00 02 - 1A 02 1E 02 31 02 39 02  ?.B.t... ....1.9.
1B00  49 02 4D 02 63 02 66 02 - 6A 02 6D 02 71 02 78 02  I.M.c.f. j.m.q.x.
1B10  7B 02 7E 02 91 02 95 02 - 9C 02 9F 02 A2 02 AE 02  {.~..... ....".".
1B20  B2 02 B9 02 C9 02 D0 02 - EA 02 ED 02 02 03 11 03  2.9.I.P. j.m.....
1B30  14 03 1E 03 21 03 38 03 - 4F 03 63 03 89 03 A0 03  ....!.8. O.c... .
```

```
1B40  AF 03 B6 03 B9 03 D6 03 - EB 03 F8 03 11 04 29 04  /.6.9.V. k.x...).
1B50  2C 04 3F 04 69 04 6E 04 - 76 04 7C 04 83 04 86 04  ,.?.i.n. v.|.....
1B60  8A 04 C5 04 C8 04 FD 04 - 0A 05 13 05 1C 05 7F 05  ..E.H.|. ........
1B70  86 05 93 05 99 05 9D 05 - AB 05 B4 05 B8 05 C9 05  ........ +.4.8.I.
1B80  CC 05 CF 05 D4 05 02 06 - D9 06 EC 06 F2 06 F7 06  L.O.T... Y.l.r.w.
1B90  02 07 12 07 18 07 1E 07 - 24 07 2A 07 5B 07 61 07  ........ $.*.[.a.
1BA0  67 07 70 07 77 07 7B 07 - 7F 07 83 07 8A 07 91 07  g.p.w.{. ........
1BB0  9A 07 A2 07 AA 07 AE 07 - B2 07 B6 07 BC 07 C0 07  ."."*... 2.6.<.%.
1BC0  C7 07 CF 07 D3 07 D8 07 - DD 07 E4 07 EB 07 ED 07  G.O.S.X. ].d.k.m.
1BD0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
1BE0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
1BF0  00 00 00 00 0A 00 2E 00 - 83 00 18 01 20 01 23 01  ........ .... .£.
1C00  27 01 2A 01 33 01 38 01 - 3B 01 44 01 47 01 4D 01  '.*.3.8. ;.D.G.M.
1C10  50 01 56 01 5B 01 5E 01 - 64 01 B6 01 BC 01 E3 01  P.V.[.^. d.6.<.c.
1C20  F7 01 FD 01 27 02 3D 02 - 41 02 46 02 83 02 87 02  w.}.'.=. A.F.....
1C30  8A 02 8E 02 A6 02 AB 02 - BC 02 BF 02 C2 02 C5 02  ....&.+. <.?.B.E.
1C40  CD 02 D4 02 D8 02 DB 02 - DE 02 E1 02 E4 02 E7 02  M.T.X.[. ^.a.d.g.
1C50  F0 02 F4 02 FF 02 05 03 - 08 03 0B 03 0E 03 18 03  p.t..... ........
1C60  1B 03 27 03 2B 03 2E 03 - 31 03 34 03 3B 03 42 03  ..'.+... 1.4.;.B.
1C70  47 03 4A 03 53 03 56 03 - 5A 03 5F 03 66 03 69 03  G.J.S.V. Z._.f.i.
1C80  6C 03 73 03 8C 03 8F 03 - A5 03 A8 03 AC 03 BE 03  l.s..... %.(.,.>.
1C90  C3 03 C8 03 D0 03 D3 03 - DB 03 E1 03 E5 03 EF 03  C.H.P.S. [.a.e.o.
1CA0  F3 03 04 04 08 04 1B 04 - 22 04 26 04 37 04 42 04  s..... ".&.7.B.
1CB0  4B 04 57 04 5B 04 62 04 - 66 04 BE 04 C2 04 D2 04  K.W.[.b. f.>.B.R.
1CC0  E1 04 E4 04 F1 04 A1 05 - A5 05 AF 05 BC 05 C3 05  a.d.q.!. %./.<.C.
1CD0  E6 05 E4 06 E9 06 EF 06 - 54 07 00 00 00 00 00 00  f.d.i.o. T.......
1CE0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
1CF0  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00  ........ ........
```

GRAPHPAC

The documentation for Graphpac is in two parts.

The Command Manual.
This gives the general description of the commands and features provided by all current versions of Graphpac.

The System Manual.
This gives specific details for implementing Graphpac on your computer and a summary of relevant commands.

Disclaimer
No representations or warranties with respect to the contents hereof are made and any implied warranties of merchantability or fitness for any particular purpose are specifically disclaimed.

INDEX

GRAPHPAC COMMAND MANUAL

Introduction.

Graphpac allows the user to SET, RESET, INVERT or TEST any pixel on the screen. Routines are also supplied to draw lines using rectangular (based on X,Y values) or polar co-ordinates (described by an angle and distance). This manual gives a general description of Graphpac features and commands in a similar format and notation to Microsoft Mbasic documentation. An entry is given for each command showing the syntax used.

Plotting on the screen.

The resolution of the graphics is determined by the hardware used and can range from 96 pixels across the screen by 46 pixels down, to 512 X 256 or more.

Graphpac commands control an invisible graphics cursor, the position of this cursor is described by X,Y co-ordinates. Values supplied to Graphpac are treated as signed 16-bit values between -32768 and +32767. A screen X co-ordinate value of +32767 is the maximum distance to the right of the screen that the graphics cursor may be placed. Negative values are to the left of the screen, -32768 being the maximum. It is not necessary to use the INT function before passing a value to Graphpac.

Note that if Graphpac is used with Microsoft Mbasic or BASIC-80, numbers between +32768 and +65535 will be accepted. Graphpac will interpret these values and all values in between as negative numbers between -32768 and -1.

As a general rule numbers greater then 32767 should not be passed to Graphpac.

GRAPHPAC COMMANDS

Not all the commands described in this document will be available in the software supplied. Some would be unsuitable or impractical on your particular hardware, in other cases they may just be unnecessary. The documentation specific to your hardware (Graphpac System Manual) will inform you which commands are available.

Graphpac commands are stored in memory usually as normal ASCII characters and not as one byte tokens which represent normal BASIC reserved words.

The PLOT DRAW and DRAWTO commands may have more than one set of values passed at a time by separating each set with a semi colon.

```
i.e.    10 DRAWTO 0,0 ; 20,0 ; 20,20 ; 0,20
```

PSET (pixel set)

Graphpac format:

```
PSET X,Y
```

Purpose:
To SET a pixel.

Remarks:
X and Y are graphic co-ordinates. If a pixel is already bright there will be no change.

Example:

```
10 CLS
20 FOR A=0 TO 30
30 PSET A,A
40 NEXT A
```

This short routine will draw a diagonal line on the screen.

PRESET (pixel reset)

Graphpac format:

PRESET X,Y

Purpose:

To RESET a pixel.

Remarks:

X and Y are graphic co-ordinates.

Example:

```
10 CLS
20 FOR A=0 TO 30
30 PSET A,A
40 NEXT A
50 FOR A=0 TO 30
60 PRESET A,A
70 NEXT A
80 GOTO 20
```

This routine will continuously draw and erase a line on the screen.

PTEST (pixel test)

Graphpac format:

PTEST X,Y: A=PEEK (PVALUE)

Purpose:

To test if a specified pixel on the screen is dark or bright.

Remarks:

Normally PTEST is implemented as a BASIC function, so that the line A=PTEST X,Y would be valid. It has been impossible in the space allocated for Mbaslink to include functions in its command table, therefore this function has to be completed in two stages. PTEST X,Y tests the specified pixel and stores the result in a memory location. It is then the responsibility of the BASIC program to retrieve this value by PEEKing this location. In the statement line above, the numeric variable would hold the result of the pixel test. If the test confirmed that the pixel was bright, the result in A would be 1. If the pixel was dark the result would be 0. If the specified X,Y coordinate does not lie within the screen display area the result would be 2.

The address of (PVALUE), the location to PEEK is given in the Graphpac System Manual.

PEN COMMANDS

A collection of PEN commands have been made available in Graphpac. The PEN commands control an invisible graphics cursor. The position of this cursor we have chosen to call the CAP (Current Active Point). The CAP can be moved around on or off the display area by various graphic statements and depending on the PEN mode active at the time, a line may be drawn, erased, or inverted. The commands that control the PEN mode may be abbreviated if required.

In the interests of speed Graphpac works in 16-bit arithmetic, this limits the distance the CAP may be moved in the X or the Y plane, in one operation to 32767. It is therefore probable that moving the CAP from the centre of the screen to X,Y co-ordinates -32767,-32767 in one operation may introduce a small error which can be confirmed by using the CAP command.

STARTAT (Start at)

Graphpac format:

STARTAT X,Y or SA X,Y

Purpose:

This command allows the CAP to be moved unconditionally to the specified X,Y graphics co-ordinate as if the PENUP mode is effective.

Remarks:

STARTAT would normally be used as part of the first graphic statement of a program, to position the CAP ready for subsequent plotting. The current PEN mode is not altered. PENRET will now be inactive until a subsequent DRAWTO, DRAW or PLOT command is encountered.

PENUP

Graphpac format:

PENUP or PU

Purpose:

To allow the CAP to be moved around the display area using DRAW, DRAWTO or PLOT commands without producing a trace or changing the existing display.

PENDOWN

Graphpac format:

PENDOWN or PD

Purpose:

All following DRAW, DRAWTO, PLOT or DOCAP commands will cause any pixels the CAP passes through to be set bright.

PENFLIP

Graphpac format:

PENFLIP or PF

Purpose:

All following DRAW, DRAWTO, DOCAP and PLOT commands will cause any pixels the CAP passes through to be inverted, that is, dark pixels will be SET bright and bright pixels will be RESET dark.

PENERA

Graphpac format:
PENERA or PE

Purpose:
All following DRAW, DRAWTO, DOCAP and PLOT commands will cause any pixels the CAP passes through to be erased.

PENRET

Graphpac format:
PENRET or PR

Purpose:
This allows the CAP to return to the co-ordinates that existed before the last DRAW, DRAWTO or PLOT commands.

DRAWTO

Graphpac format:
DRAWTO X,Y

Purpose:
To draw, erase or invert a line based on co-ordinates specified in rectangular notation.

Remarks:
The CAP co-ordinates are changed to those specified by X,Y and a line may be drawn, erased or inverted (depending on the current PEN mode) to these co-ordinates from the previous CAP. The previous CAP co-ordinates are saved in case of a following PENRET command.

DRAW

Graphpac format:
DRAW X,Y

Purpose:
To draw, erase or invert a line based on relative co-ordinates.

Remarks:
The DRAW command has a similar syntax to the DRAWTO command however the X,Y values supplied are added to the current CAP co-ordinates to produce a new CAP. Draw is useful if an object or shape has to be produced several times but at different screen co-ordinates. The shape is specified by a subroutine using the DRAW command and whenever the routine is called, the shape will be produced relative to the current CAP.

RATIO

Graphpac format:
RATIO N

Purpose:
To adjust for the height to width ratio of the screen when using PLOT or CIRCLE. The argument passed with the RATIO command will be used to modify the Y axis co-ordinates.

Remarks:
Normally the display on the screen is taller than it is wide and a circle plotted on the screen would appear oval with the sharp ends in the vertical plane. RATIO allows the user to introduce a correction factor whenever PLOT or CIRCLE are used so that a circle does in fact appear as a circle. RATIO is followed by a single argument. This argument should be in the range 0 - 511. If 511 is exceeded the value will be MOD 512 (513 will be treated as 1). A value of 512 (or zero) will cause no correction to take place and a true (distorted) display will be produced. A value of one will cause maximum correction to take place and in fact a circle will appear as a single horizontal line on the screen. The optimum value is somewhere between the two and should be decided by the user for his particular hardware configuration.

PLOT

Graphpac format:
PLOT A,D

Purpose:
To draw, erase or invert a line based on co-ordinates given in polar notation.

Remarks:
The new CAP co-ordinates are calculated, based on a specified angle "A" and distance "D". "A" is an angle between 0 and 359 degrees, which is calculated clockwise, zero degrees being towards the top of the screen. This value can be exceeded as a wrap around effect takes place, 360 degrees being the same as 0 degrees, 450 degrees being the same as 90 degrees etc. "D" is a distance specifying the number of pixels. This distance is modified automatically by a value dependant on the argument given in a previous RATIO command and also the value of the angle A. As the argument given in the RATIO command only modifies calculations in the Y axis, there will be no modification to the length of lines plotted at angles 90 and 270 degrees, however as the angle nears the vertical (0 or 180 degrees) the modification will be at it's greatest. The old CAP co-ordinates are saved in case of a following PENRET command.

CIRCLE

Graphpac format:
CIRCLE R,A1,A2

Purpose:
To draw a circle or arc on the screen.

Remarks:
A circle or arc is drawn the centre of which is the CAP and radius is R. A1 and A2 must be positive values with A2 greater than A1. An arc will be drawn from angle 1 (A1) to angle 2 (A2) in a clockwise direction. If angle 1 is zero and angle 2 is 360 a circle will be drawn. The maximum radius is 511 and if this is exceeded the radius will be MOD 512 as for the distance in the PLOT command.

It is possible to place the CAP on the circumference of a circle after it has been drawn by using the PLOT command, with the circle radius given as the plot distance. Obviously the same RATIO value must be used for the PLOT and CIRCLE commands for the result to be correct.

DOCAP

Graphpac format:
DOCAP

Purpose:
To alter a single pixel at the CAP co-ordinates dependent on the current PEN mode.

Example:
The following is a BASIC program that follows the actions taken by the CIRCLE command. Execution is slower as the CIRCLE command is a machine code subroutine and does not involve any of the overheads associated with a BASIC program.

```
10 STARTAT X,Y : REM THE MIDDLE OF THE SCREEN
20 FOR A=0 TO 360
30 PENUP
40 PLOT A,R : REM R IS THE RADIUS OF THE CIRCLE
50 PENDOWN : DOCAP : PENRET
60 NEXT A
```

CAP

Graphpac format:
CAP

Purpose:
The CAP command has been added to assist the programmer. It reports the CAP co-ordinates, allowing a programs action to be investigated.

Remarks:
CAP reports the co-ordinates on the screen at the current cursor location in much the same way as a normal print statement.

CAP%

Graphpac format:
CAP% CC,RR

Purpose:
CAP% CC,RR reports the co-ordinates on the screen at the specified CC,RR screen co-ordinates. This action does not effect the cursor address so that subsequent print statements will act correctly.

LCAP

Graphpac format:
LCAP

Purpose:
This reports the CAP co-ordinates on a printer allowing a program's action to be investigated.

Remarks:
LCAP will report the CAP co-ordinates to a printer (or the current LST: device if running under CP/M).

CLS

Graphpac format:
CLS

Purpose:
To clear the screen of all graphics and alpha-numeric characters.

GCLR (Graphics Clear)

Graphpac format:
GCLR

Purpose:
To clear the screen of graphic characters.

Remarks:
A search is made in the screen memory for any graphic characters. If any are found they are replaced with a space character. (20 hex.)

CLEOL (CLear To End Of Line)

Graphpac format:
CLEOL

Purpose:
To clear the display from the current cursor position to the end of the current line.

SCROLL

Graphpac format:
SCROLL n

Purpose:
To cause scrolling to affect a number of lines (n) on the video display.

Remarks:
The number of lines specified in the scroll command will be situated at the bottom of the screen, allowing a display area above to be fixed for graphics plotting. Alpha-numerics can still be used on the display area with the use of the SCREEN command. If it is wished to return scrolling to normal, use the command SCROLL 25. Scrolling will only take place on receipt of a line feed while the cursor is on the bottom line of the display.

SCREEN

Graphpac format:
SCREEN CC,RR

Purpose:
To place the cursor at a specified screen co-ordinate ready for subsequent PRINT statements.

Remarks:
On a 25 line by 80 character display, CC is a column number 1-80, RR is a row number 1-25. SCREEN 1,1 specifies the top left character on the screen and SCREEN 80,25 specifies the bottom right character.

BAR GRAPH STATEMENTS

Three commands have been included to allow the construction of vertical bar graphs.

VBAR (Vertical bar)

Graphpac format:
VBAR CC,RR,N

Purpose:
To display a solid vertical bar on the screen.

Remarks:
The bar which is one character wide, is constructed from the bottom of the character box specified by the screen position CC,RR for a height of N pixels. CC and RR are screen co-ordinates as used by the SCREEN command.

VBARH (half tone vertical bar)

Graphpac format:
VBARH CC,RR,N

Purpose:
To display a shaded vertical bar on the screen.

Remarks:
This command acts in a similar manner to VBAR above.

DOWN (vertical print command)

Graphpac format:
DOWN CC,RR " any ascii characters"

Purpose:
To easily label graph Y axis or vertical bar charts.

Remarks:
CC and RR are screen co-ordinates as used by SCREEN, VBAR and VBARH.

PSI (Put Screen Image)

Mbaslink format:
RESET:PSI "DRIVE:FILENAME":RESET

Purpose:
To save the screen image on disk.

Remarks:
If Graphpac is running with Microsoft's Mbasic under CP/M care must be taken to ensure that the correct directory information is sent to the disk, PSI should always be preceded and followed with a RESET statement. This will ensure that the latest directory information is written to disk before the screen image file is created and that Mbasic is aware of the file afterwards. The user should also be aware of the effect RESET has on any disk files in use by a BASIC program, consult the documentation which accompanied your BASIC interpreter.
If drive is not specified the currently logged-in drive will be used. If the filename is greater than eight characters only the first eight will be used. The filetype extension automatically supplied is .IMG (for image).

GSI

Graphpac format:
RESET:GSI "DRIVE:FILENAME"

Purpose:
To display on the screen an image previously saved, using the PSI command.

Remarks:
Due to the way Mbasic handles the disk I/O it is necessary to precede PSI with a RESET statement, otherwise rubbish may be displayed on the screen. It will be noticed, if the user experiments, that RESET is not always necessary, however, it is advised that the user conforms with this procedure. The closing quotation mark should always be included even if GSI is the last command in a BASIC statement line.

NS (Normal Screen)

Graphpac format:
NS

Purpose:
To re-configure the screen format for normal text. i.e. 25 lines of 80 characters.

Remarks:
This statement has been included to allow convenient switching to a normal screen format so that fast scrolling and a normal width display are available if required. On execution the screen will also be cleared.

GS (Graphics Screen)

Graphpac format:
GS

Purpose:
To re-activate the graphics format screen.

LOWRES

Graphpac format:
LOWRES

Purpose:
To select the low resolution graphics mode for all subsequent graphics plotting.

HIRES

Graphpac format:
HIRES

Purpose:
To select the high resolution graphics mode for all subsequent graphics plotting.

Line clipping and error trapping.
On systems where it has been possible to implement a wrap around affect, the commands WRAP and CLIP have been included (to enable the wrap around affect or clip lines at the edge of the screen respectively). As an alternative where this has not been possible the commands, FCON and FCOFF have been implemented so that the user has the choice of error trapping if required.

WRAP

Graphpac format:
WRAP

Purpose:
To cause subsequent line plotting that leaves the edge of the screen to wrap around and re-enter the screen on the opposite side.

Remarks:
This command together with CLIP are only implemented on systems where the screen resolution allows a wrap around effect to take place.

CLIP

Graphpac format:
CLIP

Purpose:
To disable the wrap around feature of the graphics.

Remarks:
Lines that leave the screen in any direction are clipped and do not re-appear on the opposite side of the screen. Any graphic statement that attempts to act on an X,Y co-ordinate outside the specified screen range will have no effect, however the CAP will still be updated. This command disables the wrap around effect caused by WRAP. It should be noted that CLIP only acts on PLOT,DRAW, DRAWTO and CIRCLE commands, normal pixel control using PSET,PRESET and PTEST will always act on the screen the wrap around feature always being operational.

FCON

Graphpac format:
FCON

Purpose:
To enable error trapping when a Graphpac command is asked to act on an X,Y co-ordinate that is not on the screen.

Remarks:
After FCON has been executed any attempt to use X,Y co-ordinates that are not on the screen will cause an illegal function call error. FCON is active when the package is first executed.

FCOFF

Graphpac format:
FCOFF

Purpose:
To disable error trapping when X,Y co-ordinates that are off the screen are used.

Remarks:
Graphpac commands which are called to act on X,Y co-ordinates that are off the screen may be executed. No physical change to the screen contents will occur however, the CAP will be updated where applicable.

SPOKE (Screen POKE)

Graphpac format:
SPOKE A,V

Purpose:
To POKE a location inside the video screen memory.

Remarks:
Where the design of the hardware causes the video memory to be outside the normal Z80 64K memory map, SPOKE has been implemented. The normal BASIC POKE statement cannot reach the video RAM. SPOKE has been included to give this facility. The syntax is the same as the POKE statement. "A" represents a location in the video memory, the value represents an offset (in characters) from the start of the display. "v" is the byte to be POKE'd.

DOCTOR DARK'S DIARY – EPISODE 15

MATTERS ARISING

Mike York's letter in the last issue commented on my remarks about a couple of aspects of Pascal, so I thought it would be a good idea to see if I could justify what I said. First, string handling. The definition of Pascal does not contain any reference to the string handling functions Mike York mentions. I assume that these are a non-standard extension provided in UCSD Pascal. They will make programming of string handling very easy for him, but will also render his programs less portable. As I understood it, one of the aims of Pascal was that if a program worked on one machine, it would work on another. For this very reason, I try to use only standard Pascal when I give you my free routines. It should then be possible to compile the code using any decent compiler, which is surely better than if I told a Com-Pas user that they had to get another compiler to use my stuff? Second, big numbers. I have sometimes had cause to write programs that handle numbers with more than the 36 digits UCSD copes with. In these circumstances, it would seem that I was not "doing it the hard way", so much as doing it the only way. The whole point of my comments on the subject was to show how Pascal enables you to write programs that work on numbers of a very large size, let us say for the sake of example a hundred digits. I am sorry if I did not make this clear. To avoid any further misunderstanding, I should like to point out that the foregoing remarks are not an attack on either Mike York or the UCSD operating system. I have read a lot about UCSD in the glossy magazines, who rave about it. So when I saw it advertised in the said glossies, I thought, "at last, our prayers are answered." Then I saw the price.....

If Mr R A C Treen of Burgess Hill (see 80-BUS News Vol.2 Issue 1 page 14) had written to me telling me how to get past the Plover room without falling down a pit, I would have told him how to open the clam. But! I now know how to deal with the Plover room, and am able to score 250 out of 350 on the Syrtis Adventure. The next hundred points seem to be much more difficult to get, however. Mr Treen, you need something like a crowbar to open the clam (which actually isn't a clam at all!), and one of the items of treasure which you may not have found yet is a bit like a crowbar.....

I have found out, with the help of my brother, quite a few useful actions that help with Level 9's Dungeon Adventure... For instance, I know how to deal with the giant rat, and what the octopus is for. Even the armed skeletons are no longer a threat, but if anyone can explain why the transport system is affected by which collar you are wearing and what the rule governing which collar to wear is, I should be glad.

BOOK REVIEW 1 - MASTERING CP/M

This one is by Alan R Miller, who is a professor at the New Mexico Institute of Technology, and has been using CP/M since it first emerged. It is published by Sybex, and costs £14.50, if you get it from Maplin, which I did. And a most unusual thing happened. Maplin were out of stock, and I had to wait a whole fortnight! Not like them at all, so the book must be selling pretty well. It has about 400 pages, and contains quite a lot of listings of short programs to illustrate the points under discussion.

Any new book on CP/M's mysteries has to be compared with the ones already out. This is better than Rodney Zaks' "The CP/M Handbook", which is really for people who are not doing any programming of their own in machine code. And it is better than Fernandez and Ashley's "Using CP/M", which is for similar people, who in addition believe that you learn more if you fill in blank lines all through the book.

There are chapters on such useful subjects as "Adding features to BIOS". The sort of thing that the other two books needed, and didn't have. The author

has a good approach to programming, in that he recommends that you build up a library of macros, rather than re-invent the wheel each time you need a particular routine, and he supplies the code for the actual macros. He shows how to read and write disk files from machine code programs, and gives a good explanation of how the directory works, which is something most authors avoid like the plague. There are several appendices, mostly useful, and a good index too. So, what is the catch? Well, remember that the professor has been using CP/M for a long time? He started out with the 8080 processor, and all the example programs and macros are in those funny mnemonics that we modern Z80 fans can't stand. You could use the indices to translate from one to the other, if you wanted to: if you have the assembler he recommends (the grossly, absurdly over-priced Macro 80), it will accept either sort of mnemonics. Even with this disadvantage, this is still the best book on CP/M that I have seen so far. I was preparing a series of articles about using CP/M routines in your own machine code programs, but I won't bother.....

BOOK REVIEW 2 - THE SOUL OF A NEW MACHINE

This one is on sale at W H Smiths, if you can get past the hordes of children fingering all the computer magazines they stock (about 500, from the look of things), reading the blurbs on all those garish cassette cases under the S*ncl**r Sp*ctr*m (which is busy demonstrating itself to an audience of baffled shoppers, most of whom seem to think it is Channel 4 television they are watching). It is actually in the shelf with the top ten paperback books of the moment, and costs a mere £1.95, which is a nice change from computer book prices in general. It is a description of how a firm called Data General rushed out their 32 bit machine in an impossibly short time. The reviews I had read about this book made it sound as if they burned out the brains and bodies of hundreds of their staff in the process, but having read it, I think they had it pretty cushy. Only one bloke actually left, and he stayed in the business, in spite of the note he left, saying he refused to deal in any unit of time shorter than a season. The engineers in particular seem to have lived a life of comparative luxury. Did you ever hear of a logic analyser being used on a Nascom RAM A board? They don't know they are born, as the saying goes. Mind you, I found it a good read. There are rude words in it, but only one, in various forms, since American swearing seems pretty monotonous and is rather unimaginative. Nobody says swut, turlingdrome, or (heaven forbid) Belgium! Almost forgot, it's by Tracy Kidder, and has a p-picture of a P-Penguin on it.

VIRTUAL DISK TESTED

I said I would do some sort of benchmark program to find out what sort of difference using virtual disks makes to the speed things run at. True to form, I have not yet done things, but I have an estimate. Loading the 32K of an Adventure game from disk takes about 8 seconds on my system, although I believe some of the more recent drives than my Pertec will do it faster. Loading the same 32K from virtual disk takes under half a second. It should be pretty obvious from this that any program that does updates of files on disk will speed up in a dramatic way, so I tried out a program called Marvin. Hey Presto! The main bottleneck, or one of them any way, disappeared, as the file update went through in about two seconds. Even the accompanying Zap program goes acceptably fast, when run from the virtual drive.

PLUTO UPGRADE ROM

As if the facilities provided by the Pluto were not sufficiently incredible, there is an upgrade ROM available, which adds some extra commands to those already available. It costs £60.00, however, which is pretty steep for an EPROM and a revised manual. The manual is a distinct improvement on the earlier one, thank goodness. However, for people like me, it would have been a good idea if they had said where the new ROM goes. Do not put it in the nice empty socket, because that is not where it goes. Take out the old 2732 ROM, and put the new 2764 in that socket, with the new chip's pin 1 in the hole

where the old chip had it's pin 1. You now have a spare 2732, and no idea what to use it for. It's a good job the new chip doesn't do any harm when it is in the wrong socket. Apparently, one of the things the spare socket can do is this: you put a set of demo routines in it, connect the board to five volts and a monitor (no host computer, no 80-BUS, just a p.s.u. and a monitor!) and it sits there demonstrating itself! I was told this one by one of the designers, when I rang up to ask why the ROM didn't work in the wrong socket. He told me lots more, most of which I have sadly forgotten, especially the bit about how to upgrade Pluto to double resolution without paying £50.00. I suspect you can do the double speed processor mod. for less, as well....

Anyway, back to the new ROM. It gives you the ability to draw dotted line, using an eight bit mask, as well as the solid lines already provided for. Circles and arcs of circles can be drawn, if you can work out what parameters to send, and it wants eight bytes of them! The circle routine is faster than any of the others I have seen, which is as it should be. The most interesting new facility is the pattern fill routine. Any shape can be "wallpapered" with any predefined pattern, and this makes pretty pictures of hosts of golden daffodils a snap to program!

Apparently, quite a few people are working on world shattering software for the Pluto, but as yet none of it seems to be appearing in the adverts. Can't wait to buy some, in case you are reading this, world shatterers! Of course, if they want to send me a free copy of anything they want to be reviewed, they are welcome to do so, and will get the usual impartial.....

APATHY RULES OK!

I expect you remember reading my idea for a circulating disk full of juicy software, to replace the program library. I expect that quite a few of you thought "That's not a bad idea, with a few changes we could....." Well it is a shame that only three people out of however many disk users read this could be bothered to write in. This is apparently a large response, by standards of this apathetic group. I have written to the people who did summon up the energy to write, apologising on behalf of the rest of you. I am surprised that a machine so obviously intended for people with enthusiasm should have been bought by so many idle bodies. Apart from the five or six people who write this magazine, and the three who wrote to me, is there anyone out there alive? Because I can see no point in me doing all this if you are just going to sit there and fiddle with your appendages. This is supposed to be a user group, not a one way process where a few of us write, and you all take and never give. Where is your "Share and Enjoy"? Belgium, man, Belgium.....

MORE APATHY, AND SERVE US ALL RIGHT

It must be a great comfort to the people who keep all their work to themselves to know that you are right never to send in an article, because they don't pay you for months on end. And you are right not to write to the people running this magazine, because they never, but never, answer your letters of complaint. Assuming it gets printed, and it seems doubtful that the Mafia [Ed. - Watch it or we'll senda de boys around!] would allow open criticism of their methods, they now owe me the £10.00 a page for this article. So where has all the money gone? Must be the nice daisy wheel printer and Wordstar, I suppose.

FINALLY

I have found out how to make Adventure programs save to disk instead of tape, but as there are only three disk users who can write, there is no point printing it. Turlingdromes.....

THE HOBBIT DIGITAL MINI-CASSETTE SYSTEM REVIEWED. BY ALF WANT

MANUFACTURER ... IKON COMPUTER PRODUCTS, KILN LAKE, DYFED.
PRICE ..... £99 + VAT
DIMENSIONS ..... Height 9cm Height below mounting board 7cm
Width 10cm Depth 9cm

Three Nascom 2 computers were in use in our primary school. The stage was reached where they were ready to go into general school use as opposed to usage in the computer clubs. With a dozen Maths and English games written we needed an operating system that would make the entry of these programs truly child's play. A five year old (or even his teacher!) must be able to load the game that is needed. Disks would be ideal but we could not afford three sets of drives. Our computers will be too widespread and too mobile even for mains power and a networking system would be hopelessly impractical. The Hobbit unit reviewed here meets all our requirements so completely, and its use has already proved so succesful that it was decided to write this review. I have not used the rival decks produced in Scotland so can offer no comparisons with this Welsh unit. Being English I can claim neutrality.

If you are thinking of instant retrieval of programs then forget the Hobbit. A new cassette has to be formatted with certain timing information (F command : 240 seconds). It is best to format both sides while you are at it. It is then simplicity itself to write programs to tape, whether they be BASIC, FORTH, PASCAL, or ZEAP. I did feel the instructions were a little incomplete here, but I managed so they can't be too bad. I do not know if Naspen can be filed but if one of the Hobbit routines is not suitable then it would be easy to write one that is. I do feel this to be a major omission from the Hobbit manual, especially as Naspen will be our next purchase! The important thing is the ease of finding the program again. The tape with the program on it has to be mounted (M command : 80 seconds). The names of the programs on the tape can then be displayed (N command : instant). I do not like the standard display format of five names, followed by the next five on ENTER,.....up to 69 files. Nor do I like only six character names, but given this it should have been possible to display all file names on one screen. I am hoping to alter this on our systems. The chosen program is loaded into memory and runs or executes automatically. (R command : 6 seconds to 68 seconds).

There are 56 KBytes of storage on each side and we are able to record about six of our school programs in that. If a cassette has been mounted for a class to use it averages about 40 seconds to run one of its files. If a new cassette is needed it takes about 230 seconds. Reliability is 100 per cent all the time and appears to me to better the Commodore and Research Machines disks, both of which have given me problems. I doubt if the simple Phillips mechanism will ever need servicing.

The unit arrives ready built, well built, without plugs and uncased. It is delightfully compact and is the first piece of computer hardware that has brought forth, "Oh isn't it sweet!" from wife and daughter instead of the usual, "How much did that cost?" It uses mini-cassettes, much smaller than audio cassettes, and two were supplied with our two drives. The whole cube shaped unit would fit easily into a Nascom 3 case, or a Kenilworth. Ours mount just above the keyboard. They may be mounted horizontally, vertically or any angle inbetween. Ours even worked upside-down. Connections are 12 volts (120 mA), 5 volts (10 mA) and Earth. Eight Data/Control lines go to port A on the Nascom, the big 26 pin plug sometimes called PL4, the one with control port 6 and data port 4.

Any fool could follow the Hobbit directions thus far. Not so easy is setting up the two 2708 EPROMS. As usual all the information is in the Nascom manual but it does take some reading. These EPROMS are normally supplied to work at D000 in the memory map. So connect pins 4, 6 and 7 to pins 10 and 11 on LKS1 and switches 7 and 8 on LSW1 up or closed. This will make the eight memory sockets on the Nascom board decode as one 8 KByte block from C000 to DFFF. The

D000 EPROM will go in socket B1 and D400 EPROM in socket B2. Note that ROM ZEAP usually lives at D000 so if you keep ZEAP there you will have to tell Ikon where you wish to put their software. Remember that one of the advantages of a reliable storage system like the Hobbit is that utility programs such as ZEAP, Nasdis, Debug, Pascal and Forth may be used efficiently without using up all your valuable 64 KByte memory page.

The commands and error messages are listed with this article but are much better explained in the Hobbit Booklet. A good description is given of how to use the Hobbit routines in programs. This is so important for educational work where there is a constant need to record results and use large quantities of data. The example program is good but I would have liked a couple more, particularly one inputting data from the tape into the program. It says much for the documentation of the Hobbit and of NasSys 3 that I have been able to adapt the system to suit our special requirements. Believe me, if I can do it anyone can. I will willingly supply any other school with a copy of my EPROM at cost if they write to me. You will probably prefer to use the information to taylor your own system. I believe that although no source code is provided the Hobbit manual provides all the addresses needed and is a good compromise between perfect documentation and cost.

Finally I must add how using the unit with Blue Label Pascal (now Nascom Pascal) has totally changed my preparation of educational programs. All my Procedures and Functions are recorded onto the Hobbit cassette that contains the Pascal itself. I can get 69 on one side and 58 on the side with the Pascal on. I can find the ones I need for a program quickly, and load them automatically while I drink my coffee. I never have to type the same routine twice, or write one out to keep. I have vowed never to use BASIC again.

But of course the ten and eleven year olds in the school do use BASIC, and the automatic operation of the Hobbit has made things much easier for them too.

A super value product, but I hope I have made clear its limitations without playing down its potential. The manufacturers are most courteous and helpful on the telephone and delivery is very fast and efficient. Our school units were posted two days BEFORE our cheque could possibly have arrived.

COMMANDS :

B XXXX YYYY	Records a BASIC program, or similar.
C	Change - Used to file a name.
D	Delete - Erase a file.
E	End - rewinds cassette after use.
F	Format a new tape.
K	Kill - deletes all files on a tape.
L XXXX	Loads a file into location XXXX.
M	Mount a new cassette ready to read or load it.
N	Names - List all files on a tape.
R	Read - reads and starts a program.
Sx	Selects one of two drives.
T	Transfers a file from one drive to a second.
X	Return to Nas-Sys.
W XXXX YYYY ZZZZ	Writes a file from location XXXX to location YYYY and if it is a program file will execute it at ZZZZ. Note that it works with BASIC only if BASIC has been cold started.
Z XXXX YYYY	Writes a ZEAP file to tape.

ERRORS :

A	That file name already exists.
B	Bad file structure.
C	A block won't read correctly after sixteen tries.
D	Cassette full up.
E	No such file exists on this tape.
G	Cassette has been write-protected.

```
0001                    ; UNERA - CP/M 2.2 utility program
0002                    ;
0003                    ; By Anders Hejlsberg, July 1982
0004                    ;
0005                    ; UNERA is used to un-erase a CP/M file which was
0006                    ; deleted by accident. UNERA is invoked by the command
0007                    ; line 'UNERA d:nnnnnnnn.ttt', where d:nnnnnnnn.ttt is
0008                    ; the name of the file to un-erase. The drive number
0009                    ; may be omitted, in which case the currently logged
0010                    ; drive is selected. Note that it is only possible to
0011                    ; un-erase a file if no new files have been created
0012                    ; since the file was deleted. Following a succesfull
0013                    ; un-erasure, UNERA will display 'FILE FOUND'. If
0014                    ; the file specified cannot be found among the deleted
0015                    ; files, a 'NO FILE' message is displayed.
0016
0017
0018 0100                       ORG     100H            ;Define origin
0019
0020                    ; CP/M equates
0021
0022       =0000        JWB     EQU     0000H           ;warm-boot jump vector
0023       =0004        CURDRV  EQU     0004H           ;Current drive store
0024       =0005        BDOS    EQU     0005H           ;BDOS jump vector
0025       =005C        DFCB    EQU     005CH           ;Default FCB
0026
0027                    ; BIOS jump vectors (offset from WBOOT)
0028
0029       =0018        SELDSK  EQU     18H             ;Select disk
0030       =001B        SETTRK  EQU     1BH             ;Set track
0031       =001E        SETSEC  EQU     1EH             ;Set sector
0032       =0021        SETDMA  EQU     21H             ;Set DMA address
0033       =0024        READ    EQU     24H             ;Read current record
0034       =0027        WRITE   EQU     27H             ;Write current record
0035       =002D        SECTRN  EQU     2DH             ;Translate sector add
0036
0037                    ; Entry point
0038
0039 0100 01E101                LD      BC,SECBUF       ;Point to SECBUF
0040 0103 3E21                  LD      A,SETDMA        ;Set DMA address
0041 0105 CDBE01                CALL    BIOSC
0042 0108 3A5C00                LD      A,(DFCB)        ;Get drive code
0043 010B 3D                    DEC     A               ;Adjust
0044 010C F21201                JP      P,UNDO          ;Not default => skip
0045 010F 3A0400                LD      A,(CURDRV)      ;Get default drive nbr
0046 0112 4F            UNDO:   LD      C,A             ;Put drive number in C
0047 0113 3E18                  LD      A,SELDSK        ;Select disk
0048 0115 CDBE01                CALL    BIOSC
0049 0118 7C                    LD      A,H             ;Is drive nbr invalid?
0050 0119 B5                    OR      L               ;(DPH address zero)?
0051 011A CAB301                JP      Z,NOFILE        ;Yes => quit
0052 011D 5E                    LD      E,(HL)          ;Get sector translation
0053 011E 23                    INC     HL              ;vector address
0054 011F 56                    LD      D,(HL)
0055 0120 ED53DB01              LD      (XLT),DE        ;Store in XLT
0056 0124 110900                LD      DE,9            ;Point to DPB vector
0057 0127 19                    ADD     HL,DE
0058 0128 5E                    LD      E,(HL)          ;Get DPB address
0059 0129 23                    INC     HL
0060 012A 56                    LD      D,(HL)
0061 012B EB                    EX      DE,HL           ;Put in HL
0062 012C 5E                    LD      E,(HL)          ;Get sectors/track count
0063 012D 23                    INC     HL
0064 012E 56                    LD      D,(HL)
```



```
0065 012F ED53DD01              LD      (SPT),DE        ;Store in SPT
0066 0133 110600                LD      DE,6            ;Point to DRM field
0067 0136 19                    ADD     HL,DE
0068 0137 5E                    LD      E,(HL)          ;Get number of directory
0069 0138 23                    INC     HL              ;entries
0070 0139 56                    LD      D,(HL)
0071 013A ED53DF01              LD      (DRM),DE        ;Store in DRM
0072 013E 110500                LD      DE,5            ;Point to OFF field
0073 0141 19                    ADD     HL,DE
0074 0142 4E                    LD      C,(HL)          ;Get first track number
0075 0143 23                    INC     HL
0076 0144 46                    LD      B,(HL)
0077 0145 C5            UND1:   PUSH    BC              ;Save track number
0078 0146 3E1B                  LD      A,SETTRK        ;Set track
0079 0148 CDBE01                CALL    BIOSC
0080 014B 010000                LD      BC,0            ;Clear sector number
0081 014E C5            UND2:   PUSH    BC              ;Save sector number
0082 014F ED5BDB01              LD      DE,(XLT)        ;Get XLT vector address
0083 0153 3E2D                  LD      A,SECTRN        ;Translate sector number
0084 0155 CDBE01                CALL    BIOSC
0085 0158 44                    LD      B,H             ;Get physical sector nbr
0086 0159 4D                    LD      C,L             ;into BC
0087 015A 3E1E                  LD      A,SETSEC        ;Set sector
0088 015C CDBE01                CALL    BIOSC
0089 015F 3E24                  LD      A,READ          ;Read sector
0090 0161 CDBE01                CALL    BIOSC
0091 0164 21E101                LD      HL,SECBUF       ;Point to first entry
0092 0167 0604                  LD      B,4             ;Set entry count
0093 0169 C5            UND3:   PUSH    BC              ;Save entry count
0094 016A E5                    PUSH    HL              ;Save entry address
0095 016B 7E                    LD      A,(HL)          ;Get delete flag
0096 016C FEE5                  CP      0E5H            ;Deleted entry?
0097 016E 201E                  JR      NZ,UND5         ;No => skip it
0098 0170 115C00                LD      DE,DFCB         ;Point to user name
0099 0173 060B                  LD      B,8+3           ;Compare name and type
0100 0175 23            UND4:   INC     HL              ;Point to next character
0101 0176 13                    INC     DE
0102 0177 1A                    LD      A,(DE)          ;Are characters equal?
0103 0178 96                    SUB     (HL)
0104 0179 87                    ADD     A,A             ;(Ignore bit 7)
0105 017A 2012                  JR      NZ,UND5         ;No => skip
0106 017C 10F7                  DJNZ    UND4            ;Loop 11 times
0107 017E E1                    POP     HL              ;Restore entry address
0108 017F C1                    POP     BC              ;Restore entry count
0109 0180 3600                  LD      (HL),0          ;Undelete!!!
0110 0182 0E01                  LD      C,1             ;Directory sector
0111 0184 3E27                  LD      A,WRITE         ;Go write sector
0112 0186 CDBE01                CALL    BIOSC
0113 0189 11C801                LD      DE,FFMSG        ;Go say 'FILE FOUND'
0114 018C 1828                  JR      EXIT
0115 018E E1            UND5:   POP     HL              ;Restore entry address
0116 018F C1                    POP     BC              ;Restore entry count
0117 0190 112000                LD      DE,32           ;Point to next entry
0118 0193 19                    ADD     HL,DE
0119 0194 10D3                  DJNZ    UND3            ;Do 4 entries
0120 0196 2ADF01                LD      HL,(DRM)        ;Get entry count
0121 0199 11FCFF                LD      DE,-4           ;Decrement by 4
0122 019C 19                    ADD     HL,DE
0123 019D 3014                  JR      NC,NOFILE       ;All done => skip
0124 019F 22DF01                LD      (DRM),HL        ;Save entry count
0125 01A2 C1                    POP     BC              ;Restore sector number
0126 01A3 03                    INC     BC              ;Increment
0127 01A4 60                    LD      H,B             ;Put into HL
0128 01A5 69                    LD      L,C
```

```
0129 01A6 ED5BDD01          LD      DE,(SPT)        ;Time for next track?
0130 01AA B7                OR      A
0131 01AB ED52              SBC     HL,DE
0132 01AD 209F              JR      NZ,UND2         ;No => skip
0133 01AF C1                POP     BC              ;Restore track number
0134 01B0 03                INC     BC              ;Increment
0135 01B1 1892              JR      UND1            ;Loop
0136
0137                    ; Here if file not found
0138
0139 01B3 11D301    NOFILE: LD      DE,NFMSG        ;Say 'NO FILE'
0140
0141                    ; Print message and exit via warm-boot. Warm-boot must
0142                    ; be used, or otherwise disk space is not reclaimed
0143                    ; properly due to an incorrect allocation vector
0144
0145 01B6 0E09      EXIT:   LD      C,9             ;Print message
0146 01B8 CD0500            CALL    BDOS
0147 01BB C30000            JP      JWB             ;Go warm-boot
0148
0149                    ; Call BIOS routine
0150
0151 01BE D5        BIOSC:  PUSH    DE              ;Save DE
0152 01BF 2A0100            LD      HL,(1)          ;Get addr of WBOOT jump
0153 01C2 5F                LD      E,A             ;Put offset into DE
0154 01C3 1600              LD      D,0
0155 01C5 19                ADD     HL,DE           ;Get jump vector address
0156 01C6 D1                POP     DE              ;Restore DE
0157 01C7 E9                JP      (HL)            ;Go to BIOS routine
0158
0159                    ; Messages
0160
0161 01C8 46494C45 FFMSG:  DB      'FILE FOUND$'
0162 01D3 4E4F2046 NFMSG:  DB      'NO FILE$'
0163
0164                    ; workspace
0165
0166 01DB +0002     XLT:    DS      2               ;XLT vector address
0167 01DD +0002     SPT:    DS      2               ;Sectors/track store
0168 01DF +0002     DRM:    DS      2               ;Directory entry count
0169 01E1 +0080     SECBUF: DS      128             ;Sector buffer
0170
0171 0261                   END
```



```
BDOS    0005    BIOSC   01BE    CURDRV  0004    DFCB    005C    DRM     01DF
EXIT    01B6    FFMSG   01C8    JWB     0000    NFMSG   01D3    NOFILE  01B3
READ    0024    SECBUF  01E1    SECTRN  002D    SELDSK  0018    SETDMA  0021
SETSEC  001E    SETTRK  001B    SPT     01DD    UND0    0112    UND1    0145
UND2    014E    UND3    0169    UND4    0175    UND5    018E    WRITE   0027
XLT     01DB
```

```
10 REM ***************************************
20 REM *          FOUR IN A LINE             *
30 REM ***************************************
40 REM *   CONVERTED FOR NASCOM 2 UNDER      *
50 REM *   NAS-SYS 1 MONITOR AND NAS-GRA     *
60 REM *          BY ROGER DOWLING           *
70 REM ***************************************
72 CLS
80 DOKE3189,1922:DOKE3187,1919:REM N MODE
110 DOKE3340,33:DOKE3342,32268:DOKE3344,-10293
120 DOKE3346,-20533:DOKE3348,-8329
130 DOKE3350,-13986:DOKE4100,3340:X=USR(0)
150 GOTO1340
160 PRINT"The object of this game is to get FOUR"
170 PRINT"IN A LINE."
180 PRINT"The board is a 6 * 7 matrix."
190 PRINT"Columns are filled from the base upward."
200 PRINT"A winning line may be horizontal,vertical,"
210 PRINT"or diagonal."
220 PRINT"In play,only three keys are used :"
230 PRINT"SHIFT LEFT moves the pointer left."
240 PRINT"SHIFT RIGHT moves the pointer right."
250 PRINT"SPACE BAR drops your marker to the"
260 PRINT"foot of selected column."
270 DIMS(44),T(11):PRINT:GOTO290
280 FORI=1TO8000:NEXT
290 M$="HIT ANY KEY TO CONTINUE":GOSUB1260
300 GOSUB1380
310 E=0:F=0:G=0:FORA=1TO42:S(A)=0:NEXT
320 H$="FOUR IN A LINE":FORI=1TOLEN(H$):POKE3030+I,ASC(MID$(H$,I,1)):NEXT
325 DOKE4100,3328:X=USR(0):DOKE4100,3200
330 PRINT:PRINTTAB(7)"SCORES":PRINT:PRINT
340 PRINT"YOU";TAB(9)S1
350 PRINT:PRINT"ME";TAB(9)S2:PRINT
360 PRINT"DRAWN";TAB(9)S3:PRINT:PRINT:PRINT
370 RESTORE
380 IFOO<>1.5THEN400
390 GOTO1320
400 L=2071:DATA199,248,231,252
410 FORT=1TO13:IFT/2=INT(T/2)THEN RESTORE
420 READ A,B:FORC=1TO28STEP4
430 IFT=1THEN460
440 POKEL,248:POKEL+29,199
450 POKEL+C,A:POKEL+C+3,B
460 NEXTC:L=L+64:NEXTT
470 N=36:FORC=1TO7:N(C)=N:N=N+1:NEXT
480 R=2069:C=4
490 S=S1+S2+S3:IFS/2<>INT(S/2)THEN770
500 M$="  MY MOVE":GOSUB1240
510 V=0:X=0:Y=0:Z=0
520 FORJ=1TO7:C=C+1:IFC>7THENC=1
530 P=R+C*4:IF N(C)>0THENGOSUB1000
540 IFFTHEN730
550 NEXT
```

```
560 IFXTHENC=X:GOTO730
570 IFYTHENC=Y:GOTO610
580 IFZTHENC=Z:GOTO610
590 C=INT(RND(1)*7+1):IFN(C)<1THEN590
600 V=V+1:IFC<>4ANDV<5THEN590
610 IFN(C)<8THENF=0:GOTO730
620 IFV<10THEN690
630 IFV/3=INT(V/3)THENMM=1:M$="NOW WHAT DO I DO"
640 GOSUB1260
650 IFV<21THEN690
660 FORA=1TO500:Q=INT(RND(1)*940+1993)
670 J=INT(RND(1)*256):POKEQ,J
680 NEXT:E=1
682 A7=1:B7=20:C7=.1:D7=50:GOSUB2000
684 GOTO980
690 N(C)=N(C)-7:Z=0:GOSUB1000:N(C)=N(C)+7
700 IFFTHENF=0:IFV<9THEN590
710 IFZTHENZ=0:IFV<6THEN590
720 IFXTHENX=0:GOTO590
730 X=R+63+C*4+128*(N(C)-C)/7
740 FORB=1TO2:POKEX+B,228:POKEX+B+64,201:NEXT
745 A7=1:B7=1000:C7=200:D7=10:GOSUB2000
750 S(N(C))=5:N(C)=N(C)-7
760 IFFTHENM$="    I WIN!":GOSUB1260:GOTO762
761 GOTO770
762 A7=1:B7=2000:C7=100:D7=10:GOSUB2000
764 S2=S2+1:GOTO280
770 M$="YOUR MOVE":GOSUB1240
780 P=R+C*4:POKEP,12:POKEP+1,12
790 GOSUB1400:Q=2057
800 IFPEEK(Q)=17ORPEEK(Q)=18THEN830
810 IFPEEK(Q)=32ORPEEK(Q)=49THEN830
820 GOTO790
830 IFPEEK(Q)=18THENT=18:GOTO900
840 T=17
850 IFPEEK(Q)=17THEN900
860 IFPEEK(Q)=49THENOO=1.5
870 IFOO=1.5THEN320
880 IFPEEK(Q)=32THEN940
890 GOTO830
900 IFT=18ANDC<7THENC=C+1
910 IFT=17ANDC>1THENC=C-1
920 POKEP,32:POKEP+1,32
925 A7=1:B7=10:C7=1:D7=1:GOSUB2000
930 GOTO780
940 L=(N(C)-C)/7:IFL<0THEN780
950 X=P+63+L*128
960 FORA=1TO2:POKEX+A,94 :POKEX+A+64,11 :NEXTA
965 A7=1:B7=100:C7=5:D7=10:GOSUB2000
970 S(N(C))=1:E=0:GOSUB1000
980 IFETHENM$="  YOU WIN!":GOSUB1240:GOTO982
981 GOTO990
982 A7=1:B7=5000:C7=100:D7=1:GOSUB2000
984 S1=S1+1:GOTO280
990 N(C)=N(C)-7:GOTO500
```

```
1000 P=R+C*4:POKEP,12:POKEP+1,12
1005 A7=1:B7=10:C7=1:D7=1:GOSUB2000
1010 FORA=0TO11:T(A)=0:NEXTA
1020 I=0:M=N(C)
1030 FORU=MTOM+21STEP7:IFU>42THEN1050
1040 T(I)=T(I)+S(U)
1050 NEXTU:I=I+1
1060 FORA=C-3TOC+3:IFA<1THENA=1
1070 IFA>4ORA>CTHEN1110
1080 FORB=ATOA+3:T(I)=T(I)+S(M-C+B):NEXTB:I=I+1
1090 N=M-(C-A)*8:IFN<1ORN>18THEN1110
1100 FORD=0TO3:T(I)=T(I)+S(N):N=N+8:NEXTD:I=I+1
1110 IFA>7THEN1150
1120 IFA<4ORA<CTHEN1150
1130 N=M+(C-A)*6:IFN<4ORN>21THEN1150
1140 FORD=0TO3:T(I)=T(I)+S(N):N=N+6:NEXTD:I=I+1
1150 NEXTA
1160 FORH=0TOI:D=T(H):IFD=4THENE=1
1170 IFD=15THENF=C
1180 IFD=10THENZ=C
1190 IFD=3THENX=C
1200 IFH=2ANDD=2THENY=C
1210 NEXT
1220 POKEP,32:POKEP+1,32
1230 RETURN
1240 FORA=1TO7:IFS(A)=0THEN1260
1250 NEXTA:M$="ITS A DRAW":G=1
1260 SCREEN1,14:PRINT"                    ":SCREEN1,14:PRINTM$
1270 IFMM=0THEN1290
1280 FORD=1TO600:NEXT:MM=0:M$="  ":GOTO1260
1290 POKE2953,32
1300 IFGTHENG=0:S3=S3+1:GOTO280
1310 RETURN
1320 PRINT" GIVEN UP SO SOON !!!":END
1330 IFZTHENC=Z:GOTO610
1340 RESTORE1500:FORZ=3200TO3246STEP2:READJ
1342 DOKEZ,J:NEXT
1344 FORZ=3328TO3364STEP2:READJ:DOKEZ,J:NEXT
1370 DOKE4100,3200:RESTORE:GOTO160
1380 IFUSR(0)<>0THENCLS:RETURN
1390 FD=RND(25):GOTO1380
1400 A=USR(0):IFA=0THEN1400
1410 POKE2057,A:RETURN
1500 DATA25311,312,18351,10927,-8179,233
1510 DATA23533,3330,19437,3328,8254,211,30731
1520 DATA8369,-4613,75,-20723,211,30731,8369
1530 DATA7163,-19590,-7648,201
1540 DATA-5871,8459,3347,4734,8979,12030,4810,-15603
1550 DATA3334,10441,10595,21024,26479,29285,17440
1560 DATA30575,26988,26478,46
2000 DOKE4100,3212
2010 DOKE3330,D7
2020 FORH8=A7TOB7STEPC7
2030 DOKE3328,H8:U5=USR(0):NEXTH8
2040 DOKE4100,3200:RETURN
Ok
```

DAVE HUNT'S RAMBLINGS

Correction to the last issue.

Regular readers will be well aware of my tendency to muck things about, and may well remember I had a go at a simple database handler by Mike Trim in the last issue. I made a note at the time that I thought my mods would do no harm, simply drawing a number of duplicated routines into a common subroutine. However, as is the luck of these things I forgot to change a line number and as I could not test it, being written using the extension to Nascom BASIC provided by Polydos, we published it as it was. Now the fact that I didn't know it was written for the Polydos extension BASIC was pointed out by a couple of readers - well I can't be expected to know everything now can I - Polydos is just one of those things that has passed my by and I've never tried it. Anyway, by publishing we seem to have caused Mr. Trim quite some trouble, as it would seem that a number of more resourceful readers have gone to the trouble of discovering his address (which was not published) and asking for corrections. This is of course something of an embarrassment to everyone, not the least Mr. Trim who has spent some sum on stamps putting right my mistake. To make ammends, I include Mr. Trims' database handler again in full (hopefully correct this time) and hope by so doing the 'powers that be' will see fit to make him a payment for the republication of his contribution. See program 1 on an adjacent page.

Non-returned disks and cassettes.

This brings me to other matters. My having the opportunity to play around with Mr. Trims' program came about because it was provided as hard copy, and the lot fell to me to type it in. In doing so I mentally rearranged it and then affected the mods. Now the reason it was supplied as hard copy was simply that the cost of a disk would have been equal to the payment made for the contribution, and this was good reason for not sending a disk. Point taken. In future we will try to ensure that when we receive copy on any sort of media we will try our best to return it to its rightful owner. This point has also been demonstrated by Chris (Dr. Dark) whose most recent contribution has arrived as hard copy for the same reasons. I only hope I don't get landed with the job of typing his lot in, he's almost as prolific as me when it comes to footage of printer paper. I know I haven't always returned media in the past because I have several tapes and a couple of disks belonging to other people. The only snag is that I don't know who they came from, as the names and addresses have either got lost or were not supplied in the first place.

Why we go on about CP/M.

Paul received a letter recently complaining bitterly that the magazine was becoming more and more CP/M and disk orientated and people like standard Nascom 1 owners were being left out in the cold. True. I know I've said it before, and I'm sure others have said it as well. We publish a fair balance of what we receive, and if contributions are tending in the direction of disks et al, then the magazine content goes that way as well. If you want more articles on the basic Nascom 1 then write them!!!

Machine code delays (with the ZX81?)

Which neatly brings me to the next two points. The first concerns a ZX81 owner (what's that I hear you ask). However, this person had got hold of a simple BASIC program which when loaded allowed the user to type in Z80 opcodes to any address above itself and then jump to the program. It sounded very like a stripped down NAS-SYS with no debugging facility. Here arose his problem. As there were no debugging facilities it was extremely difficult to understand what

was going on, and being a self confessed novice at machine code he admitted he didn't have a clue, and as far as he could see the program didn't work. What he had done was attach LEDs to the Sinclair I/O connector, which he believed to be some sort of I/O port, and was trying to switch the LEDs by machine code instructions which he took to be the machine code equivalent of POKEs to a given address. As this was his first attempt at machine code programming (I considered it a bit ambitious for a first attempt) he was somewhat disappointed when nothing particular happened.

Anyway, as I have about as much experience with the ZX81 as I have at piloting a 747 (a real one that is, no I can't land the natty Sinclair Spectrum Flight Simulation program either, and that's supposed to be a light plane although a couple of experienced pilots who have tried it assure me it's more like a 747), this was all Dutch to me [Ed. - apologies to our readers in Holland!]. It seemed to me somewhat unlikely that the ZX81 I/O connector was controlled by some sort of latched I/O device, and I guess it is nothing more than a partially decoded address/data bus. However, he assured me that he could flash the LEDs in sequence by poking a 2-4-8-.. sequence to a particular address in BASIC, so why could he not do the same thing in machine code. It was then that a glimmer dawned. The ZX81 in it's text display mode under its own BASIC is about as fast as a superannuated snail, so if anything was happening to the LEDs it was probably the speed (or rather the lack of) at which the program ran which allowed anything to be visible. I suggested that he take his BASIC program and introduce a FOR - NEXT delay loop to see if the rate of flashing the LEDs was reduced, and he went away to try it. A day or so later we came across each other again, and he confirmed that the FOR - NEXT delay did indeed slow down the LEDs, so how to do this in machine code.

I explained that the BASIC FOR - NEXT loop simply made the processor waste time by counting, and so it is the same in machine code. The only difference is that whereas the BASIC loop may only be counting a hundred or so for a delay of about a minute (see what I mean, brother is the ZX81 slow), at the machine code level, the few hundreds have to be considered as a few millions instead. All he wanted now was a machine code equivalent to the BASIC FOR - NEXT delay loop. So I gave him the very simple counting loop shown in program 2 (adjacent). Now I know it's simple, but there are many around who still haven't mastered (or even tried) the machine code end of the business, so I offer it here for all those who have been searching for a delay loop of between about a half second to about two minutes. (That includes the guy who wrote the time delay routine in the PLUTO colour card demo program, it's awfully complicated to do such a simple little thing.) I have since heard that the chap now has his LEDs flashing away in machine code and because of this modest success is now only too keen to progress to other things.

A Nascom 1 'feature'.

That story has very little to do with Nascom 1s but is at least the sort of level applicable to an unexpanded Nascom 1. The second story very definitely concerns the Nascom 1 and one of its shortcomings. Some months ago the shop supplied a Mr. Rudd Thornton of Largs with a Gemini GM812 IVC and a SYS7 to run it on his CP/M system. Well from that order it was easy to predict the system the gentleman has, a Nascom 2 with a Henelec/Gemini G805. Imagine our dismay when he phoned to say it didn't work because he has a Nascom 1. The problem was easily pinned down to the ambiguous I/O decoding of the Nascom 1 (a design fault or feature depending upon which way you look at it). Mr. Thornton has the G805 plugged into the PIO on the Nascom, and as soon as the IVC was plugged in the whole lot predictably stopped working. None of the easy alternatives were at first sight agreeable. Telling Mr. Thornton to scrap the idea and accept a refund was out as I can't stand the tears and reproachful looks from the boss when I ask him for a refund cheque. Similarly, it wouldn't be fair to ask Mr. Thornton to cough up another £60 odd quid to fit a Nascom I/O card which would

cure his problem. Not because I think that Mr. Thornton would have refused as a last resort, but because it was our fault that we supplied inappropriate gear through not enquiring too closely, and the best way to keep customers is to treat them nicely and not to trap them into spending more money than they need to.

Anyway Mr. Thornton was very nice about it, and we made several suggestions as to how to cure the problem for a few pence rather than pounds. I received a letter from Mr. Thornton a couple of weeks back and I reproduce his solution here.

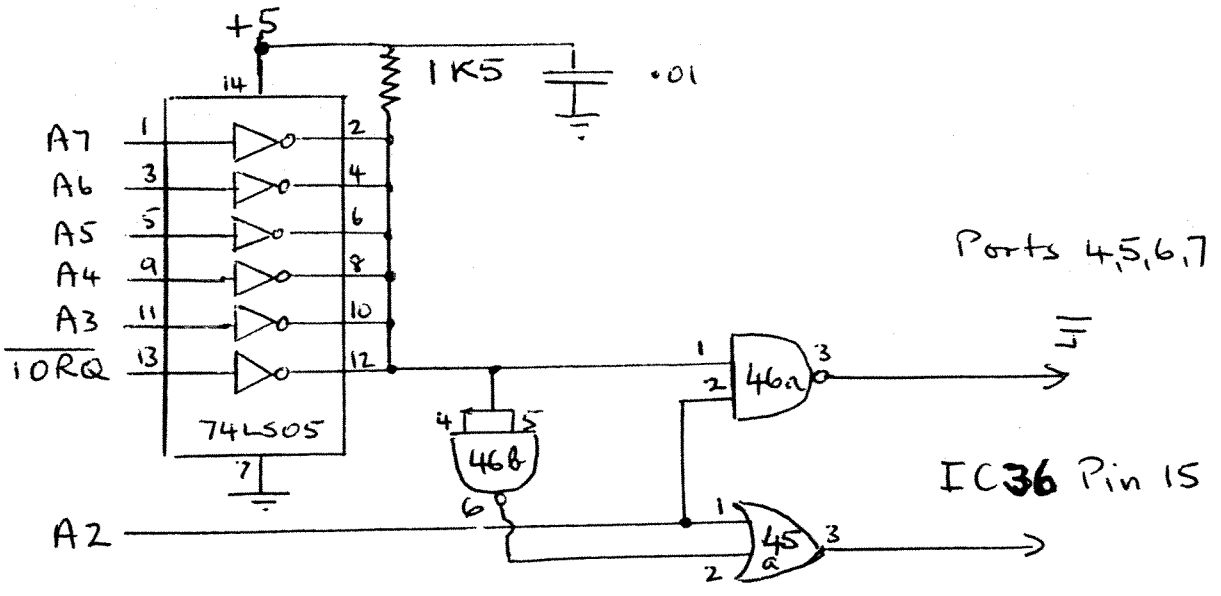
UNAMBIGUOUS I/O DECODE FOR NASCOM 1 by R. Thornton

The port decode on the Nascom 1 only looks at the bottom three address lines, giving a total of eight ports. Port 0 is used for the keyboard, ports 1 and 2 for the UART, ports 4, 5, 6, and 7 for the onboard PIO. Port 3 is unused.

No problem, just so long as you don't use a port address like 11H or 81H or B1H for something else. Any access to those ports will switch on the UART. This is because the onboard decode doesn't care what the high address lines are up to, if any of the bottom three address lines is active with IORQ, then something on the main board will be switched on as well. A link is provided on the board (IOEXT) to allow an external board to provide a full decode. Unfortunately someone at Nascom slipped up in this area of the design, and this link does not provide the correct function.

One can get away with some non critical applications combinations, for instance always writing to the conflicting ports, but one particular combination which can't be mixed is the Henelec/Gemini G805 disk controller and the Gemini GM812 intelligent video card. The IVC uses ports B1H, B2H and B3H, and consequently does fascinating and unpredictable things to the UART. What it doesn't do is put pictures on the screen.

There are several solutions; the easiest is to use the Nascom I/O board to feed the G805, but it is expensive. Further, a method was published which requires a bit of board surgery and allows a signal put on IOEXT to take control. (See D. Ritchies letter in INMC4). However, as any board running off the NASBUS will provide its own decode all that is necessary is to make the onboard decodes unambiguous. In other words to lock out the ports on the Nascom 1 when any higher address line is active. This can be done with one extra chip and without board surgery. If you have a dual monitor board the circuit can be wired on to a 24 pin header, picking up most of the lines required and leaving only one connection to be soldered to the main board.



The circuit is given in Fig. 1. To wire up, connect the 74LS05 to address line A3 through A7 and to IORQ. Common the collectors together using a 1K5 resistor to form a 'wired OR' gate, and connect this common output to pins 1, 4 and 5 of IC46 which are lifted out of their socket for the purpose. Pin 6 of IC46 is also lifted and connected to the (lifted) pin 2 of IC45. So far no snags have cropped up in testing, with or without the IVC in either Nascom or CP/M formats.

Well, as you see all ended well. Mr. Thornton is happy, I'm happy because that's one more problem solved, and my boss is no less happy than usual in that he knows nothing about this episode, and I have spared him the anguish and cramp of the wrist he gets when he signs cheques.

Turbo-charged N1

Whilst on the subject of Nascom 1's we have recently heard of a turbo charged N1 using a Z80B and running at 6MHz. Apparently the main snag of the conversion was not any defficiency in the design but the crazy cost of 16 150nS 2102 RAMs for the work space and video RAMs.

Centronics Interface Nasty

On the subject of problems solved, one OOO-nastie which cropped up recently which I may have mentioned before but is well worth repeating. This concerns the well known Centronics parallel printer connection. Now Centronics when they designed the interface very wisely put all the control signals on one side of the plug and made the pins on the other side all earths. This is so that when ribbon cable is used with insulation displacement type connectors, each signal line is separated from the next by an earth line thereby reducing noise. Now the Centronics plug has 36 pins, yet there are only thirteen signal lines, and appropriately enough, thirteen matching earths. This leaves ten unallocated lines. This means that other manufacturers who have adopted the Centronics plug convention are free to do as they will with the remaining lines, and believe me, they do. Herein lies the danger. It is tempting when wiring a Centronics socket to run a continuous earth across all the pins on one side, pins 19 through 36, as these are the signal earths as defined, or otherwise are unused. Well watch out. Epson in their wisdom have made the printer SELECT line pin 36. That's Ok, as it has to be earthed anyway to turn the printer on. But Seikosha have made pin 35 a +5 volt output for some inexplicable reason, so if it's earthed the least it will do is blow the fuses in the printer. NEC go even better, pin 31 is the INPUT PRIME, a sort of reset line, Ok it can be earthed, except it will hold the printer permanently at reset so it won't work, further pin 32 is the FAULT output which is a TTL level open collector output, so earthing it won't do it a lot of good. Pin 36 is the INPUT BUSY, a duplicate of pin 11, earth it and the handshake doesn't work. Another classic, but this time at the machine end is by the makers of that fire breathing Welsh creation (Dragon if you haven't worked it out). Because they use a 20 way IDS socket on the computer that means there are only 10 lines along the top edge, so what happens to the BUSY signal? Simple, they move it to the other edge so it means hand wiring the plug at one end or the other. Better still, the Dragon shoves +5 volts out of one of the other pins on the earthy side. Forget that and connect it to earth as you might expect, and you might get more smoke out of the thing than you bargained for. Brilliant isn't it!!!

A SYS 'feature'.

Now on to more mundain matters. At a reasonable guess there are a couple of hundred of Richard's SYS program out there and most of those must belong to readers of this magazine as otherwise how would you have heard about it? Well anyway Richard has built a somewhat undesirable feature into the SYSB versions

Program 1 Data base handler by M. L. Trim.

```
100 CLEAR 10000 : DIM A$(6,200) : LN=200
110 DATA 27085,14336,-13564,6399,18178,10927
120 DATA -8179,233
130 DATA 31711,1080,-53,536,-20665,3370,-5664,0
140 DATA 4100,3340 : FOR I=3340 TO 3354 STEP 2
150 READ J : DOKE I,J : NEXT
160 GOTO 720
170 REM
180 REM  1. Input list routine
190 PRINT"To abort I/P type 0 on request for customer."
200 FOR I=1 TO LN
210 GOSUB 1110
220 IF A$(0,I)="0" THEN 240
230 NEXT I
240 PRINT"Press R to return to menu"
250 Z=USR(0) : IF Z=82 THEN 720
260 GOTO 250
270 REM
280 REM  2. Display list
290 CLS : A=0
300 FOR I=1 TO LN
310 GOSUB 1220
320 A=A+1 : IF I=LN THEN 420
330 IF A=2 THEN 350
340 GOTO 410
350 T$="Press C to continue"
360 FOR T=1 TO LEN(T$) : POKE 3030+ASC(MID$(T$,T,1))
370 NEXT T : A=0
380 Z=USR(0) : IF Z=67 THEN CLS : GOTO 410
390 Z=USR(0) : IF Z=82 THEN I=LN : GOTO 720
400 GOTO 380
410 NEXT I
420 T$="Press R to return to menu"
430 FOR T=1 TO LEN(T$) : POKE 3030+ASC(MID$(T$,T,1))
440 NEXT T
450 Z=USR(0) : IF Z=82 THEN 720
460 GOTO 450
470 REM
480 REM  4. Save list
490 INPUT"Do you wish to save on disk (Y/N) ";C$
500 IF C$="Y" THEN 520
510 GOTO 720
520 SETCLS(1) : SETNEW(1),"DATA.TX",S : FOR I=1 TO LN
530 FOR J=0 TO 6 : SETOUT(1),A$(J,I) : NEXT J : NEXT I
540 SETCLS(1) : GOTO 720
550 REM
560 REM  5. Load list
570 SETCLS(1) : SETNEW(1),"DATA.TX" : FOR I=1 TO LN
580 FOR J=0 TO 6 : SETINP(1),A$(J,I) : NEXT J : NEXT I
590 GOTO 720
600 CLS
610 PRINT"                              Thank you and goodbye."
620 END
630 REM
640 REM  3. Change list
650 INPUT"Enter number of the item to change ";I
660 GOSUB 1110
670 PRINT"Press R to return to menu"
680 Z=USR(0) : IF Z=82 THEN 720
690 GOTO 680
700 REM
710 REM  Menu routine
720 CLS
730 PRINT TAB(15);"1. Input list"
740 PRINT TAB(15);"2. Display list"
750 PRINT TAB(15);"3. Change list"
760 PRINT TAB(15);"4. Save list"
770 PRINT TAB(15);"5. Load list"
780 PRINT TAB(15);"6. Print list"
790 PRINT TAB(15);"7. Find customer"
800 PRINT TAB(15);"8. End program"
810 PRINT
820 INPUT"Enter number of function required ";N
830 IF (N<1) + (N>8) THEN 720
840 ON N GOTO 190,290,650,490,570,870,940,600
850 REM
860 REM  6. Print list
870 CLS : INPUT"Number of customers to list ";LN
880 WIDTH 255 : LINES 2000 : SETPRON
890 FOR I=1 TO LN
900 GOSUB 1220 : NEXT I
910 WIDTH 48 : LINES 5 : SETPROFF : GOTO 450
920 REM
930 REM  7. Find customer
940 CLS
950 PRINT"   Type 1st 4 letters customer to be "
960 INPUT"    located";H$
970 FOR I=1 TO LN : IF H$=LEFT$(A$(0,I),4) THEN 1030
980 NEXT I
990 FOR I=1 TO LN : IF H$=A$(0,I) THEN 1030
1000 NEXT I
1010 PRINT"Customer not on file" : FOR T=1 TO 2000 : NEXT T
1020 GOTO 720
1030 GOSUB 1220
1040 INPUT"Do you wish to copy to printer ";K$
1050 IF K$="Y" THEN 1070
1060 I=LN : GOTO 720
1070 SETPRON : GOSUB 1220
1080 I=LN : SETPROFF : GOTO 720
1090 REM
1100 REM  Subroutine to get name and address input
1110 PRINT"Customer's name" : INPUT A$(0,I)
1120 IF A$(0,I)="0" THEN I=LN : RETURN
1130 INPUT"and address 1";A$(1,I)
1140 INPUT"and address 2";A$(2,I)
1150 INPUT"and address 3";A$(3,I)
1160 INPUT"and address 4";A$(4,I)
1170 INPUT"and address 5";A$(5,I)
1180 INPUT"and telephone ";A$(6,1)
1190 RETURN
1200 REM
1210 REM  Subroutine to print name and address
1220 PRINT I " ";A$(0,I)
1230 FOR J=1 TO 6 : PRINT "   ";A$(J,I) : NEXT J
1240 RETURN
```

```
Program 2       Simple machine code delay by D. R. Hunt.
VARIABLE DELAY APPROX. 0.5 - 120 SECS   M-80    1 Apr 1983      17:37   PAGE 1


                                ; Simple delay routine D. R. Hunt

0000    06 00                   LD      B,0       ; Load B with time delay

0002    C5              LOOP:   PUSH    BC        ; Save B during inner loop
0003    01 0000                 LD      BC,0      ; Prime inner loop counter
0006    0B              INLOOP: DEC     BC        ; Count down one
0007    78                      LD      A,B       ; Copy B to A
0008    B1                      OR      C         ; Z set only if A and C = 00
0009    20 FB                   JR      NZ,INLOOP ; If not 00 go round again
000B    C1                      POP     BC        ; Get B back
000C    10 F4                   DJNZ    LOOP      ; Dec B, if not Z, round again
000E    C9                      RET               ; All done, so go home
```

```
Program 3    Ammendment to SYSB15 and SYSB16 module SYSB5.MAC
;       ***************************************
;       ******* STATUS OF LIST DEVICE ********
;       ***************************************

; Used by DESPOOL to improve console performance
; If list device is busy, return A set to 0
liststst: ld      a,(iobyte)      ; IOBYTE
        and     0c0h            ; Printer type
        cp      40h             ; CRT:
        jr      z,lfree
        cp      0c0h            ; UL1:
        jr      z,lstcen

; TTY: or LPT:
        in      a,(uarts)       ; Check serial port
        if      gemini
        bit     5,a
        else
        bit     6,a
        endif
        jr      z,lbusy

        if      lhand
        in      a,(phand)       ; Check handshake
        if      gemini
        bit     4,a             ; Test CTS
        jr      z,lbusy
        else
        rla                     ; Test bit 7
        jr      nc,lbusy
        endif
        endif
        jr      lfree

; UL1:
lstcen: if      ul1cen
        in      a,(cenpad)      ; Check parallel printer
        rra
        jr      c,lbusy
        endif

lfree:  ld      a,0ffh          ; Free
        ret
lbusy:  ld      a,0             ; Busy
        ret
```

which has only recently come to light. I know this affects all SYSB versions from SYSB11 onwards, although it was correct in versions up to SYSN7 for the Henelec/Gemini GM805 system (I haven't got copies of the earlier versions so it may or may not be relevant). Now to allow the CBIOS to cope with spooler type software CP/M 2.2 allows a call to the LST: device status to be made whilst polling the keyboard to check if the LST: device is ready to accept further data. In this way a file can be shoved out to the printer whilst keyboard input is taking place. Hence allowing printer despooling to take place. If the LST: status is busy then no printer action can take place. SYS of course allows this with a separate LST: status routine, unfortunately by an oversight, it checks both serial and parallel routines together and if either is busy it reports busy. Now most users don't have two printers attached so it is likely that either the serial or parallel output will be floating. By the fact that it's floating, it could either return busy or not busy as it feels, Murphy's Law says it will return busy, so the likelihood is that spooler operation will be either erratic or non existence, depending ... No-one has reported this, but I feel that several people have tried and wondered what is wrong with their despooling software and then given up, alternatively if they have tried their despooling software under Gemini's CP/Ms they should have had no problem. In SYSB15 and SYSB16 a simple change to the end of module SYSB5 will cure this problem and is listed as program 3. Whilst on the subject of SYS, SYS fetishists may be upset to know that SYSB16.1 already exists. If you have SYSB15 don't worry, SYSB16 only incorporated a printer margin facility and a software tidy up, whilst SYSB16.1 fixes the aforementioned 'feature'. [Late insert: Richard tells me SYSB17 is on the way with a reported bug in the extended screen edit routine fixed.]

Super Diskpen.

Another 'feature' built in to the response to an advert in the last issue regards the new Super Diskpen - those who ordered it didn't get any documentation!!! Not too disastrous for those who know how to drive PEN, or those who discovered what the '?' and '??' commands do. However, documentation was promised and the advertiser had a file of names and addresses of those who had purchased PEN for the purposes of sending on the documentation. Don't ask me how it happened, but the file has got lost. So if you're sitting there staring at a PEN command line and wondering what to do next drop the advertiser a line and let him know your name and address.

Another command in DISKPEN/GEMPEN version 3 is the '&' command. Some late versions may have been supplied with an overlay called MENU.OVL, and you may be wondering what this is for. When the '&' command is fired up, the menu tells you about a spooling device which allows PEN to have an intelligent printer despooling function built in. This is the first of the PEN overlays which are becoming available, and incedently, how the funny in the spooling software in SYS came to light.

A second overlay for PEN has just been sent to me for evaluation and this one is really clever. It gives PEN the ability to function as a free field data handler. A sort of mini-database controller which whilst not as powerful as the big business type database controllers has a number of advantages over the way the data is formatted, and like all PEN 'bolt-on' goodies, it is cheap. About one twentieth the price of dBASE II for instance for about one quarter the power. If you are worrying about not having MENU.OVL - don't!! When a new overlay program is purchased a new MENU.OVL is supplied with it covering all the previous issues. But don't forget that these overlays can only be used with version 3 of PEN, and the signon will be VG:3x, VN:3x or VS:3x, where x is the type of printer output configured.

A User Group.

Now onto news about a computing club, the Nascom - Thames User Group, or words to that effect that they somehow abbreviate to NAS-TUG. They meet every

other Thursday at the Frogmore Hotel, Alma Road, Windsor. Anyway, the Thames Rip-off Society, sorry, NAS-TUG have what they call an organised evening on one Thursday followed by what they call a disorganised workshop evening the following Thursday fortnight. They attract members from a very wide catchment area (I've even been known to go along myself and I don't exactly live in Windsor) because they're a friendly bunch. Membership is not restricted to Nascom owners - if their latest newsletter is to be believed, a couple of members are currently 'down-grading their equipment to Gemini' and one member admits to owning an Osbourne but comes along to find out what real computing is about. Details of who to contact can be sought through The Chertsey Computer Centre or Henry's Radio. Sorry to mention shops, but they are the only places I can remember who would know the relevant numbers. (There you are Mark, a plug, now how about bumping our circulation up a bit by enlightening many of YOUR members who are unaware of OUR existence.)

CCPZ Revisited.

To follow up my piece about CCPZ elsewhere in this mag, I made comment about having a go at persuading CCPZ to accept something like .13 as a command line, that is using the full stop in place of the command word USER because it's three letters less to type. Well during a chat with Mike Waters of Norwich, who was using CCPZ and had fixed a bug concerning the way CCPZ generates form feeds, I mentioned my problem, and he has since kindly sent me a fixed version with the form feed bug fixed and the full stop fully implemented. In fact he notes that it was quite surprising that the command line parser did what it did when I tried to make it go as, of course, a full stop is the delimiter between the file name and the file type. So of course, it should have gone away and looked for a file called " .13 " or some such. I guess the strange behaviour I experienced was something to do with the way I patched the full stop command into the command line parser, oh well .... I can't have been thinking at the time, 'cos I should have twigged that .13, or such like, would in fact be interpreted as a file name. Whether I would have come up with a solution quite as elegant as Mikes and got it in the remaining space .... well that's a different matter.

Databases.

And so at long last it should now come round to the DH bit about databases. I say 'should', because I've run out of time to write it. Paul has been nagging me for this copy for the last week and this edition is well behind time as it is. So it looks as if there is going to be no discourse on databases from me this episode (do I hear cries of shame??). Any way it is perhaps as well, as with the arrival of MAXiFILE the data handler for PEN, it forms a very relevant transition from the simple to the all laughing dancing databases (future episodes to come). So this 'lay-off' for a few weeks will give me a chance to master MAXiFILE and see exactly how good it is.

Coming Soon!

Future writings from the DH PEN are also planned. I've recently had the opportunity to spend some considerable time playing with all three of the popular colour cards for 80-BUS/NASBUS - the Nascom AVC, the Climax MV256 and the Pluto card (that's why this article is incomplete this time). Anyway I'll be writing at length about colour cards over the next couple of weeks. Whilst I'm at it, I'll also be talking to a colleague at Paddington Tech who is having a go at interfacing a BBC model B to a Gemini as an intelligent terminal. He thinks it will provide him with all the colour and hi-res (about 320 x 256) he wants and save him the cost of buying a keyboard and a video card in to the bargain. Should be fun if he ever gets that beastly thing called the 'Tube' to work. I don't think Acorn know how it works, so who else do you ask?

RANDOM RUMOURS (& TRUTHS)

by S. Monger

By all accounts Gemini have been busy on the software front lately, taking out licenses with a number of 'the big names'. These include Microsoft (MBASIC), MicroPro (WordStar, SpellStar and Mailmerge), Sapphire Systems (MARS) and Ashton Tate (dBASE II). Yes, these have all been available on Gemini disk formats from various suppliers for some time now, but at least you will now know that those that need implementing for particular screens etc. have already had that work done for the Gemini IVC. Mind you, Anthian Software, a small northern (Hull?) company have always implemented any software that they have sold on Gemini formats, unlike Lifeboat and MPI.

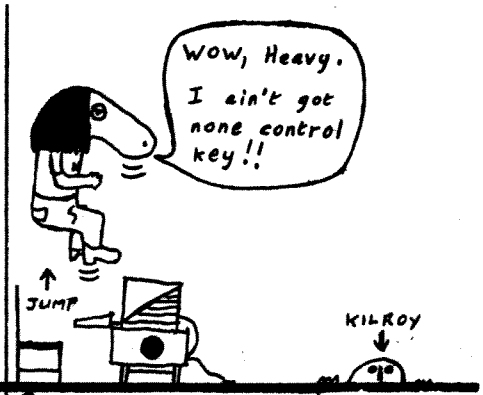
Readers of magazines other than this, although recognising that they are nowhere near as good, must have noticed the adverts that they contain for the Lucas/Nascom LX printer. No, this is not a revamp of the Nascom IMP, but it would appear to be an import from Japan under some sort of exclusive deal. One dealer, wanting to know the reputation of the manufacturer before committing himself (or should that read 'being committed'?) asked a member of the Lucas/Nascom staff who the manufacturer was. "It's a Lucas printer", "Yes, but who makes it?", "It's a Lucas printer", "Yes, I know that's what the badge says, but who makes it?", "It's a Lucas printer", CLUNK - phone hung up. So, for all of you with insatiable curiosity, one of my usually reliable sources tells me that he believes it may be a Citizen, and that there is at least one other company about to start importing it. Now, if I've got my facts right, Citizen is owned by that small watch-making company called Seiko, who also own that other small printer company Epson, and if I worked for Lucas and a dealer asked me who made their printer I would reply, "Well, I really shouldn't tell you this, but ....."

There are a number of jokes doing the rounds of the dealers about the possibly unfortunate choice of name of Climax Computers. Now, this page is probably read by a number of innocent youngsters, and so I won't bring up the jokes that have come my way. However, it would be fair to say that their colour board is now coming out of the door in reasonable numbers, after an initial delay, and it appears to be getting a good reception. Good luck to this young company.

Gemini have recently started shipment of their network system, MultiNet. By all accounts it is quite impressive, allowing upto thirty or so workstations to be connected to a single Galaxy Winchester system with printer. To all users it appears as though they are running their own separate twin disk CP/M system, one of the drives being private, and the other a common read-only drive. It is also possible to have Passwording, Auto-program execution at any station, shared read/write files using a special file-locking technique, and all printer output handled centrally, or at local printers. I'm wondering if I can install one in my house so that I can type this lot in in the kitchen, bathroom, bedroom etc. - what fun! Nascom seem to have had quite a reasonable success with their NasDos based network, and they have been advertising it quite heavily lately. Unlike IO Research, whose IO-Net seems to have disappeared into (temporary?) obscurity.

That friendly Irishman, Rory O'Farrell was spotted recently in England doing the rounds of Amersham Computer Centre, Chertsey Computers (MAP to you), and Henry's Radio. All astute observers commented that he didn't have any books under his arms. How can he possibly walk, talk and eat without writing a book review at the same time! Maybe the answer is in the fact that he was stocking up with a colour graphics board and A DISK SYSTEM. Now he'll be able to save the book reviews to magnetic media umpteen times faster! Does this mean more free time for Rory, or more book reviews?

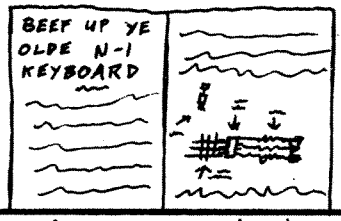
The story so far.... Laurence has bought POLYDOS for his beefed up NASCOM, all is well untill he tries to use it, and....

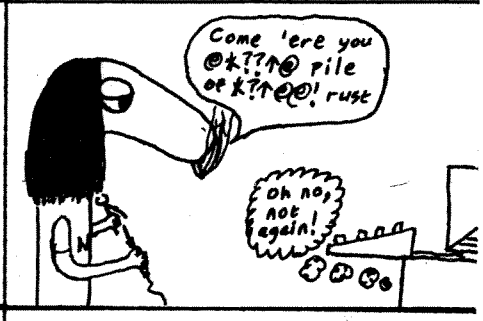


A quick flick thru the back issues of 'THE OTHER' mag, and....

.... an answer is found.

Out come the tools and Laurence's Nascom gets 'fixed'....

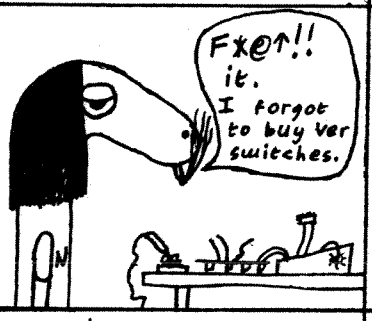


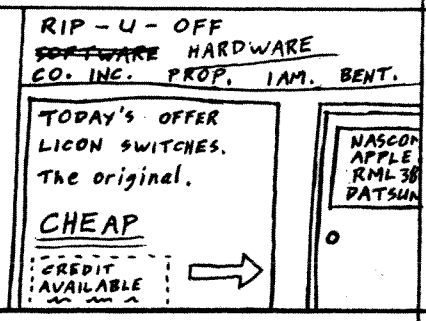


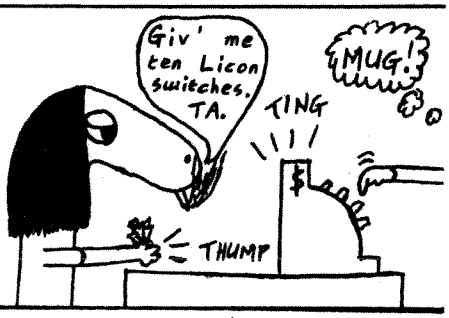
.... but not in the way he expected!

A trip to the local store may solve the problem....

.... it does.



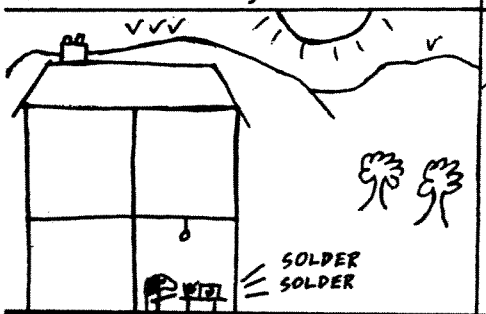


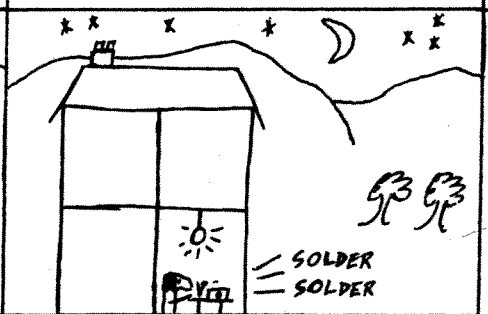


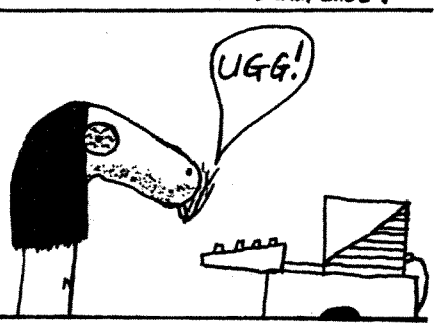
Laurence's domicile becomes a hive of activity....

.... all night!

Morning comes, and the job is finished. So is Lawrance!



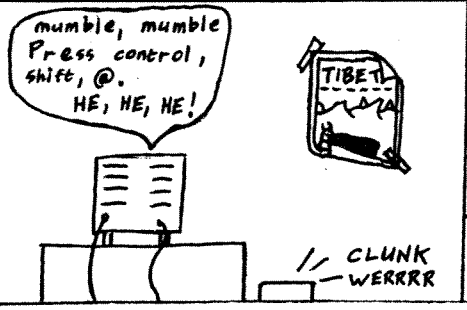




The big moment arrives and....

.... it works!

Before laying out heavy bread on POLYDOS for your N-1 take heed, it needs an N-2 type keyboard to work properly.



BY D.G. RICHARDS. TONYREFAIL.
MID. GLAMORGAN.
1983

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

AMERSHAM COMPUTER CENTRE

80-BUS HARDWARE

GM811	Z80A CPU Board	£125.00
GM813	Z80A CPU + 64K RAM Board	£225.00
GM812	Z80A Video Controller Board	£125.00
GM802K	16KRAM Kit (64K Board)	£ 80.00
GM802	64K Dynamic RAM Board	£125.00
GM829	FDC/SASI Controller	£145.00
LUCAS LOGIC	Floppy Disk Controller	£145.00
GM803K	Eprom/Rom Board (Kit)	£ 55.00
GM816	I/O Board (RTC CTC 3*PIO)	£125.00
EV814	IEEE 488 Controller Board	£140.00
RB32KC	32K Cmos Batt/Backed RAM	£170.00
IO824	8ch - 8bit A/D Board	£120.00
PLUTO	High Res Colour Graphics board, 2 Screen Memories (640h*288v) 3 bit Pixels	£339.00
BABY PLUTO	(320h*288v) 3 bit Pixels	£299.00
CC837	Colour Graphics Board, (256*256)16 colour pixel display. (PAL) (PAL + RGB)	£199.00 £220.00
NAS AVC	Colour Graphics Board 80col (320h*256v) Pixels	£185.00
GM810	8 Slot Motherboard 5A PSU	£ 69.50

DISK SYSTEMS

GM825-1S	Single Drive Disk Unit (400k)	£350.00
GM825-2S	Double Drive Disk Unit (800k)	£575.00
GM825-1D	Single Drive Disk Unit (800k)	£435.00
GM825-2D	Double Drive Disk Unit (1600k)	£745.00
LUCAS LOGIC	Single Drive Unit (inc FDC)	£470.00
LUCAS LOGIC	Double Drive Unit (inc FDC)	£685.00
GM835	5.4 Meg Winchester Sub-System	£1450.00
GM835/10	10.8 Meg " " "	£1850.00

DISK OPERATING SYSTEMS

GM512	CP/M for Multiboard & GM815	£105.00
GM532	CP/M for Multiboard & GM825	£105.00
GM513	CP/M for Nascom & GM809/GM815	£125.00
GM556	CP/M for Nascom & GM829/GM825	£125.00
LUCAS LOGIC	CP/M for LUCAS Disk System	£100.00
GM515	POLYDOS 1 for Nascom & GM805	£ 90.00
GM516	POLYDOS 2 for Nascom & GM815	£ 90.00
GM533	POLYDOS 3 for Nascom & Lucas	£ 90.00
GM534	POLYDOS 4 for Nascom & GM825	£ 90.00
LUCAS LOGIC	NASDOS for Nascom & Lucas	£ 60.00
LEVEL 9	Q-DOS for Nascom & GM805	£ 40.00

MONITORS

9"AVT	High Res, Green or Amber	£ 99.00
12"P12	High Res, Green or Amber	£110.00
12"M12	Metal cased,90 Deg,High Res	£150.00
14" LUXOR	Hi-Res Colour R G B Monitor	£600.00

PRINTERS

EPSON		
FX-80	160cps Dotmat/Friction/Trac	£436.00
NEC		
PC8023BC	100cps DotMat/Tractor	£330.00
SMITH CORONA		
TP1	Daisywheel/12cps	£485.00
OLIVETTI		
PRAXIS 41	Daisywheel Typwriter/Printer	£660.00

NASCOM SOFTWARE TAPES

LEVEL 9	Extension Basic Tape	£ 15.00
CCsoft	Nas-Graphpac Tape	£ 20.00
NASCOM	Pascal Compiler Tape	£ 45.00
L-Soft	Logic soft Relocator Tape	£ 13.00

LEVEL 9 NASCOM GAMES TAPES

5 GAMES	(Gunner/Wumpus etc)	£ 6.00
Missile Defence		£ 8.00
Asteroids		£ 8.00
Space Invasion		£ 7.00
Bomber		£ 5.00
Fantasy		£ 6.00
Nightmare Pork		£ 5.00
Colossal Adventure	(32K)	£ 10.00

NASCOM FIRWARE

LEVEL 9	Extension Basic (4*2708)	£ 25.00
NASCOM	Pascal Compiler	£ 75.00
POLYDATA	Polytext Text Editor	£ 35.00
GEMINI	Naspen Text Editor	£ 20.00
GEMINI	Nas-Sys 3 (2708/Nascom 1)	£ 20.00
GEMINI	Nas-Sys 3 (2716/Nascom 2)	£ 20.00
BITS & PC's	Programmers Aid (N/Sys1)	£ 20.00
BITS & PC's	Programmers Aid (N/Sys3)	£ 20.00

GEMINI RPM/CPM SOFTWARE

COMAL80 BASIC	(Includes GEMzap/pen/debug)	£100.00
*GEM PEN	Text Editor / Formatter	£ 45.00
*GEM ZAP	Assembler (screen editing)	£ 45.00
*GEM DEBUG	Debug/Disassembler	£ 30.00
COPY SB	Superbrain to Gemini (DDDS)	£ 30.00
LIST/REPAIR	Recovers Lost Data Etc	£ 25.00
DATAFLOW	Information Processor	£125.00
*G BASIC	Graphics Basic (IVC)	£ 25.00
GEM GRAPHPAC	Links to Mbasic (IVC)	£ 25.00
COM-PAS	Pascal Generates M/Code	£120.00

*Tape also available.
When ordering disks please specify the format.

MEDIA

		EACH	BOX 10
SCOTCH C10	Cassettes	£ .60	£ 6.00
SCOTCH C30	" "	£ .70	£ 7.00
DYSAN 1042D	D/S disk (M or P)	£5.00	£ 43.00
DYSAN 1041D	S/S disk (M)	£4.75	£ 40.00

(M) Suitable for Micropolis Drives
(P) Suitable for Pertec Drives

80-BUS SYSTEMS

QUANTUM	QM 2000 System (2.4M Bytes)	£2250.00
GM903	GALAXY 2 System,2 Drives	£1495.00
GM904	GALAXY 2 System,1 Drive	£1275.00
GM905	GALAXY 2/2 drives(1.6M Bytes)	£1695.00
GM907	GALAXY 3 (1 X 5.4M+1 X 800k)	£2500.00
NASCOM 3	48K with Nas Sys 3 & Graphics	£ 549.00

Access

WE ARE OPEN MONDAY TO SATURDAY 9-5.30. PERSONAL CALLERS WELCOME.

AMERSHAM COMPUTER CENTRE LTD.
18, WOODSIDE ROAD, AMERSHAM, BUCKS. HP7 OBH
Telephone: 02403 22307 Telex: 837788

ORDERS ACCEPTED BY MAIL AND PHONE.
ACCESS AND VISA CARD HOLDERS WELCOME.
Prices subject to the addition of relevant VAT charge + P&P

VISA