```
*****************************
*                           *
*      MAP 80 SYSTEMS LTD   *
*                           *
*          VFC MANUAL       *
*                           *
*****************************
```

MAP 80 SYSTEMS LTD
Chertsey Computer Centre,
1 Windsor Street,
CHERTSEY,
Surrey. Tel 09328 64663
Issue 3 4th June 1983.

MODIFY

CONTENTS
========

SECTION 1
=========

INTRODUCTION
============

This manual has been written very carefully to introduce you to your new VFC. Read all sections and be sure you understand its operation before you proceed. If you have purchased the kit version, a seperate assembly manual will have been supplied. Read it carefully before you start assembly.

The MAP VFC has been designed so that it will work with all the existing Nasbus/80 Bus computers. For example it is possible to run a complete 64k CP/M computer using just this card and a Gemini G813. For the Nascom owner the card will allow him to reach the level of the Gemini computers whilst still retaining the compatibility with his old system. Unlike all other manufacturers MAP 80 Systems have designed a product that retains the old NAS-SYS monitor and will allow the user complete access to CP/M in the usual way. No mechanical switching or other hardware alterations are necessary to switch between systems. In fact, the user can even transfer some files from NAS-SYS into CP/M by leaving them in the memory whilst changing from system to system.

The complete VFC may be considered in four parts.
The first part consists of a memory mapped 80 column by 25 line display. This screen incorporates memory WAIT states thus avoiding all the problems of screen flash that were encountered on the early Nascoms. The screen may be mapped into the address space and then mapped out when it is not being updated. Two character generators allow the user a total of 512 different characters. One set of 256 is provided which also contains the normal alphanumeric set. The other set is left for expansion and could be a games set. As if that is not enough it is also possible to invert the lower 128 characters in each set.
The second part of the card is its on board floppy disc controller which will drive any 5 1/4" disc drive. The controller is completely Gemini G809 compatible.
The third feature of the card is its ability to interface with any ASCII encoded keyboard.
Finally a video switch allows the user to switch an external composite video signal onto the same monitor as the VFC. This external signal could be the old Nascom 48 column screen.
All the above features may be controlled by the on board software "VSOFT". This very powerful package may itself be mapped into the memory only when it is required.
A series of links allow the user total freedom for specialist applications. The following instructions give more detail on how to use your new MAP 80 product.

Happy Computing !!!!.

SECTION 2
==========

USING THE MAP 80 VFC

The MAP 80 VFC is a quad-functional card, firstly it is a 4k block of memory, the lower 2k is taken up by VSOFT a versatile re-locateable package for driving the VFC, and the upper 2k is the video RAM used by the display, this is usually configured as a 80 column 25 row display. Secondly it is a 5 1/4 Floppy Disc Controller, thirdly a keyboard port capable of reading almost any configuration of ASCII keyboard and finally a video switch to enable the selection of VFC or external video.

The VFC uses a total of 16 ports, these can be link selected to a 16 port boundary in the range 0x to Fx. As supplied VSOFT expects the port allocation of E0 to EF and this configuration is assumed in susequent notes. Versions of VSOFT at other port boundaries are available, if for instance you already have a 6809 FDC and wish to use the video portion only of the VFC. It is recommended that C0 to CF is used as the second choice, and that B0 to BF and F0 to FF are never used.

PORT ALLOCATION (Note all Port addresses are in HEX)

E0	Read	2797 Status register
E0	Write	2797 Command register
E1	Read or Write	2797 Track register
E2	Read or Write	2797 Sector register
E3	Read or Write	2797 Data register
*E4-E5	Read	Read 2797 pins DRQ,INTRQ and READY
*E4-E5	Write	Select drive
*E6-E7	Read	Read keyboard port
*E8-E9	Read or Write	Alarm trigger
EA	Write	6845 register select
EB	Read or Write	6845 data port
*EC-ED	Write only DO NOT READ	Video control port
EE	Read or Write	Select video 1
EF	Read or Write	Select video 2

Ports have been initially decoded in pairs and those marked * have not been furthur decoded, reading port E7, for example, will be exactly the same as reading port E6.

FDC PORTS E0 to E5

Ports E0 through E3 directly access the 2797 registers, the function of these are complex and data can be best obtained from Western Digital's application document. Port E4 when written to selects one of 4 drives, bit 0 set enables drive 0, bit 1 drive 1 etc. only one bit should be set at any time. Bit 4 is used to set the 2797 to either single density FM (set 1) or double density MFM (reset 0), bits 5,6 and 7 are unused. When read access is allowed to three 2797 pins, bit 0 holds the INTRQ (pin 39) line, bit 1 inverted READY (pin 32) and bit 7 DRQ (pin 40), all other bits are tied low. The WD2797 is an upgrade from the WD1797 but has on chip data separation and write pre-compensation, this has overcome many of the problems of write pre-compensation on the old 2143 whilst retaining full software compatability with the 1797.

KEYBOARD PORT E6 or E7

To obtain the current state of the keyboard port it is only necessary to execute an IN A,(0E6H). If the character loaded is 0FFH no key is pressed else the value loaded is the output from the keyboard with the strobe active. However if full programmable key, soft repeat and screen edit features are required use the VSOFT KBDST and KBDIN routines. Almost any keyboard can be used (see hardware notes for implementation), and depending on the keyboard characters 00H thru' 0FEH can be read.

ALARM PORT E8 or E9

Accessing port E8 with either an IN or OUT instruction will cause a negative going pulse to be generated at A01,(Auxilliary output 1) this is simply an output from the IO decoder and provides an LS TTL signal which can be used to trigger a beeper via some suitable external circuitry.

6845 REGISTERS PORTS EA and EB

The 6845 is a complex CRT controller, VSOFT is capable of looking after this device for you under normal circumstances, however if you wish to control it for your own special applications writing to port EA will select the 6845 register you wish to access and port EB will allow you to input/output to/from that register, you will need to obtain a copy of the Motorola MC6845 data sheet. Note there are two kinds of 6845, the 6845 and the 6845*1 the ordinary 6845 is used by MAP 80, the *1 version adds a few extra features which are not required by the VFC although it too can be used without modifications.

VIDEO CONTROL PORT EC and ED

Port EC controls the paging of the VFC in and out of the memory map as well as selecting various features. It is a write only port and as it is only partially de-coded reading it will corrupt the control data.

Bit allocation of port EC

		RESET	SET
0	Video RAM enable	Out of memory map	In memory map
1	VSOFT EPROM enable	see notes below	
2	Upper/inverse characters	Upper character set	Inverse video
3	Character generator	EPROM 1	EPROM 2
4			
5	VFC 4K boundary		
6	select		
7			

Whenever VSOFT or video RAM is selected a RAMDIS signal is generated, during RD cycles during VSOFT access and RD and WR cycles during VRAM access, it is therefore not possible to select the VFC over any other memory which produces RAMDIS i.e NAS-SYS on a NASCOM, VRAM and workspace RAM on a NASCOM, EPROM/ROM on a NASCOM, EPROM/ROM on an EPROM card. It should also be noted that if RAM underlying the VFCs VRAM when writing to VRAM is not to be corrupted you must be using a MAP 256k RAM or make the modifications detailed in SECTION 6 to your existing RAM card.

On power up or RESET all bits above are reset (0). This means that both the VSOFT EPROM and the video are deselected and do not appear in the memory map. It is possible however to use link 4 to invert bit 1. In doing this on power up or reset the VSOFT EPROM will be selected and will occupy locations 0000 to 07FF. If you select this option and your computer resets to address 0000 VSOFT will boot from disk Track 0 Sector 0 into address 0C00, using the on board FDC. Having sucessfully loaded the sector a check is made to ensure that the first 2 locations of the booted sector contain '00', if a valid sector is found VSOFT will execute the sector at 0C02. This sector will de-select VSOFT and then normally load a full operating system (e.g CP/M). With regard to RAMDIS mentioned above if this auto boot function is to be used NAS-SYS must be de-selected on the NASCOM. This is not a disadvantage, in fact it is a definite boon. Using the auto boot feature it will be possible to run a full 64k NASCOM and boot CP/M or NAS-SYS according to which disk you insert on RESET, and by using the VFC video switch, select either the normal NASCOM video or VFC video to your monitor without touching a single mechanical switch. Operating systems for all the above features plus much much more are available from MAP 00 SYSTEMS or your dealer.

BIT 1	VSOFT EPROM SELECT	
	RESET	SET
LINK 4 in auto boot mode	selected	deselected
LINK 4 in non boot mode	deselected	selected

To control the VFC you simply output to port 0EC a control byte, the upper 4 bits select the 4K page slot and the lower 4 bits select either, both or neither the EPROM and video RAM and the character set. For example with link 4 in auto boot mode to select the VFC at 8000 to 8FFF with VSOFT and VRAM selected, inverse video on characters with bit 7 set and character set one, output 85H to port 0ECH. The VFC must of course not be selected over the current program counter location or the current stack.

VSOFT -------- 'ON BOARD CONTROLLING SOFTWARE'

The VSOFT EPROM also contains comprehensive software for driving the VFCs video and keyboard, this software (with the exception of the boot routine) is fully re-locateable making it possible to locate the VFC at say 1000H and output a character to the screen by calling VSOFT and then locate it at 2000H and send another. VSOFT doesn't care where it is and looks after the calculations to ensure the correct relative position of the screen and cursor etc.

VSOFT has 5 entry points to 5 routines, the address of the routine entry will of course depend on where the VFC has been selected, the entries are therefore relative to the base address. The following notes detail the various routines.

BASE+0

BOOT NOT RELOCATEABLE this routine must only be used with the VFC selected at 0000H with link L4 linked to auto-boot, the routine loads track 0 sector 0 from drive A, using the VFC's 2797 Floppy disk contoller, into 0C00H.

BASE+3

VINIT VFC initialisation routine, used to initialise workspace and the 6845 CRT controller, this must be called before any calls are made to the video or keyboard routines as they expect a properly initialised workspace and no access should be attempted to VRAM until the 6845 is initialised as its power up state may cause a permanent WAIT state to be entered when VRAM is accessed!!!!. On calling VINIT the IX register must point to a block of RAM 23 bytes long which is used as VFC workspace, the A register must hold the state of the VFC control port, the VRAM and VSOFT bits (0 and 1) must be in the enabled position and the inverse/upper select (2) and character set select (3) bits set to your initial requirements, the A register will normally contain this as you will have just selected the VFC before calling VINIT.

BASE+5

KBDST KEYBOARD STATUS This routine polls the VFC keyboard and edit buffer (see EDIT) and returns with 0FFH in the A register if a character is available, or 00 if no character is available.

BASE+7

KBDIN KEYBOARD IN. This routine waits for and returns with a character from either the edit buffer or the keyboard. If the cursor is enabled (see ESC E) the cursor will blink. If you don't want to hang around waiting for a key to be pressed call KBDIN only after calling KBDST has returned with 0FFH. The keyboard port is capable of returning 8 bit characters excepting 0FFH which is used to indicate 'key not pressed' and 00 (changeable see ESCAPE "@") which is used to initiate the EDIT mode. The keyboard can also be accessed at any time by inputting port 0E6.

EDIT MODE

Pressing ^@ (00H) once will enter edit mode 1, the cursor will change to a full character non-blinking type (if the cursor is enabled see ESCAPE e), VSOFT will enter an internal keyboard read/video output, full use of cursor keys, control keys, and escape sequences can be made to provide full on screen editing. When a CR (0DH) is encountered editing ceases and the cursor line becomes an edit buffer and subsequent calls to KBDST and KBDIN will return this buffer until it is emptied, the keyboard is not read during this time and the entire line will be returned except . : * - if they appear in the first column (CP/M prompts),) if it appears in the second column and spaces after the last non-space character are ignored. After the entire line has been sent the edit mode will return a CR (0DH) and then return to normal cursor and input. If whilst in edit mode 1 you press ^@ a second time the cursor will change to blinking full character and you will enter a permanent edit mode, lines will be returned as with mode 1 but after being sent return will always be made into the edit mode, pressing ^@ a third time will return you to normal. (see also ESCAPE 'X' and ESCAPE 'Z').

BASE+12

VIDEO VIDEO OUTPUT This routine sends the character in C to be printed at the current cursor location, a host of characters and ESCAPE sequences are not printed but provide sophisticated screen management. (see SECTION 5).

KBDST, KBDIN and VIDEO are called with IX pointing to the VFC workspace and with DE holding the VFC location e.g if the VFC has been paged in at A000H DE will contain A000H.

All calling routines should save registers IX,HL,DE and BC as these will be corrupted. To ensure that the VFC is not selected over the current stack and does not cause stack overflow, a local stack should be used.

SECTION 3

PROGRAM EXAMPLE CP/M

The following program example shows how the VFC can be inserted into a CP/M BIOS but also serves as an example for any application. The VSOFT routines have been specifically designed with CP/M in mind and the control codes used closely follow the Gemini IVC and the Lear Siegler MD3A in order to make its use in CP/M applications easier, this necessarily uses different codes than those used by NASSYS but a simple conversion table can overcome these problems, a NASSYS interface example is also given. The source listing of VSOFT has not been included in order to keep costs to a minimum but is available along with data sheets on the 2797 FDC and 6845 CRTC for a nominal copying fee, and copies will be included on any systems disks purchased.

```
;          CPM DEMO
          .Z80

;Initial state of VFC, this single byte controls the VFCs position
;Bits 4 to 7 define the 4k slot, in this case 1000H, at which VFC
;will be selected for VSOFT access, this must not overlay EPROM,
;NASSYS VRAM/WORKSPACE, the calling BIOS routines, the VFC workspace
;or the local stack ALTSTK.
;                         bit 3 reset...character set 1
;                         bit 2 set.....inverse characters when bit 7 set
;                         bit 1 reset...VSOFT enabled (auto boot linked)
;                         bit 0 set.....VRAM enabled

INITV    DEFB 15H

;Workspace and local stack
VW:      DEFS 23                    ;VFC workspace
         DEFS 40                    ;Local stack for VSOFT
VFCSTK:  EQU $

EVMASK   EQU 00000011B              ;EPROM and VRAM enable bits
```

```
;VSOFT routine relative addresses
VINIT   EQU 3                   ;VFC initialisation
KBDST   EQU 5                   ;Keyboard status
KBDIN   EQU 7                   ;Keyboard in
VIDEO   EQU 12                  ;Video out

;VFC port locations
CNTRL   EQU 0ECH                ;VFC control port
VSWICH  EQU 0EFH                ;Videc switch for selectine video 2


;Initialise VFC
;Because it is an initialising routine registers are not preserved
INVFC:  CALL ALTSP              ;Ensure stack position
        LD IX,VW                ;Point to VFC workspace
        LD HL,INVRET            ;Make return
        PUSH HL
        LD A,(INITV)            ;VFC position + initial status
        PUSH AF
        AND 0F0H                ;Mask address
        LD H,A
        LD L,VINIT              ;Make call address
        POP AF
        DI
        OUT (CNTRL),A           ;Bring in the VFC
        JP (HL)                 ;Call VFC initialisation routine

;VFC returns here
INVRET: LD A,(IX)               ;Workspace now holds initial status
                                ;With VSOFT and VRAM disabled
                                ;The upper 4 bits are cleared
        OUT (CNTRL),A           ;Switch card out
        EI
        RET                     ;Return via RETSP to pick up
                                ;old stack

;Keyboard status
SCAN:   PUSH HL                 ;Save HL
        LD HL,KBDST             ;KBDST relative position
        JR VFCCOM               ;Common VFC entry routine

;Input keybord character
BLINK:  PUSH HL                 ;Save HL
        LD HL,KBDIN             ;KBDIN relative position
        JR VFCCOM               ;Common VFC entry routine

;Send character in C to CRT
CRT:    PUSH HL                 ;Save HL
        LD HL,VIDEO             ;VIDEO relative position

;Common VFC entry routine
VFCCOM: PUSH DE
        PUSH BC                 ;Save registers
        PUSH IX
```

```
        CALL GOVFC
        POP IX
        POP BC
        POP DE                  ;Recover registers
        POP HL
        RET

GOVFC:  CALL ALTSP              ;Make sure of stack position
        LD IX,VW                ;Initialise IX
        LD DE,VFCRET            ;Return address after VSOFT call
        PUSH DE                 ;Set up for return
        LD A,(INITV)            ;Location for paging in VFC
        AND 0F0H                ;Clear initial status
        LD D,A
        LD E,0                  ;DE = address offset
        ADD HL,DE               ;Add in VSOFT routine relative address
        OR (IX)                 ;Current VFC status
        XOR EVMASK              ;Set VRAM and EPROM enable bits
        DI                      ;Stop interrupts
        OUT (CNTRL),A           ;Bring in VFC
        AND 0F0H                ;Mask 4k SLOT
        JP (HL)                 ;Go to VSOFT routine
;DE holds address offset, C holds output char if video

;VSOFT returns here
VFCRET: LD B,(IX)               ;VFC status with EPROM and VRAM off
        LD C,CNTRL              ;VFC control port
        OUT (C),B               ;VFC out
        EI
        RET                     ;Via RETSP

;Use a local stack to avoid overflow and ensure it's position
;Note the Gemini and SYS BIOSes  already have this routine but
;it must not be used.
ALTSP:  LD (VFCSTK-6),HL        ;Save HL
        LD HL,RETSP             ;Set up return via RETSP to get old stack
        LD (VFCSTK-2),HL
        POP HL                  ;Get return address
        LD (VFCSTK-4),HL        ;Put at top of VFCSTK
        LD (VFCSP),SP           ;Save SP
        LD SP,VFCSTK-6          ;Set new one
        POP HL                  ;Recover HL
        RET                     ;Return

RETSP:  LD SP,(VFCSP)           ;Reset stack
        RET                     ;Done
        END
```

The above gives a common calling system for driving the VFC, using just one byte (INITV), which can be changed at will, to determine the location of the VFC during operation. As can be seen the format of only one VFC workspace byte need be known, this is the first byte VFCST, this byte shows the current status of port 0EC bits 2 and 3, the upper 4 bits are all reset (0) and the VSOFT and VRAM bits are in the de-selected state, this byte enables continuity of bits 2 and 3 when the VFC is paged in and out, a full example of the workspace follows.

```
;Sample video workspace
VW:
VFCST:   DEFB 0                    ;Status of VFC
LASVR:   DEFW 0                    ;Last screen address
CURSOR:  DEFW 0                    ;Cursor address
MLOCK:   DEFW 0                    ;Current top of screen
KPOS:    DEFW 0                    ;Current send position
PRGKEY:  DEFW 0                    ;Address of programable key table
CURTYP:  DEFW 0                    ;Normal cursor
STATE1:  DEFB 0                    ;Option bits
KCHR:    DEFB 0                    ;Key character store
SEND:    DEFB 0                    ;Number of characters during send
KCOUNT:  DEFW 0                    ;Key repeat counter
ESCST:   DEFB 0                    ;ESC status
ESCTYP:  DEFB 0                    ;ESC type
EDCHR:   DEFB 0                    ;Editing charater
ROW:     DEFB 0                    ;Row store
PIXEL:   DEFB 0                    ;Pixel for SET,RESET,TEST

;STATE1 bit allocation              RESET(default)      SET
INV      EQU 0   ;Invert video          OFF             ON
MCR      EQU 1   ;Master cursor state   ON              OFF
CSR      EQU 2   ;Cursor state          OFF             ON
AVA      EQU 3   ;Key available         NO              YES
PRG      EQU 4   ;Programable key state OFF             ON
EDM      EQU 5   ;Edit mode             1               2
ERR      EQU 6   ;Cursor/pixel error    OK              ERROR
SMD      EQU 7   ;Send mode             SCREEN          KEY TABLE
```

And now an example of how some of the VSOFT commands can be implemented, this example shows how a typical BIOS sign on message can be used to set specific user requirements, look at SECTION 5 to see the format of the ESCAPE sequences

```
        LD HL,SIGNON               ;Point to message
        LD B,SGNEND-SIGNON         ;Message length
                                   ;0 cannot be used as a terminator
                                   ;because message may contain 0s
SLOOP:  LD C,(HL)                  ;Get message byte
        CALL CRT                   ;Send it to VFC
        INC HL
        DJNZ SLOOP                 ;Send complete message
        program continues
                "
                "
                "
```

```
;Welcoming message
SIGNON:
;Clear the screen
        DEFB    1AH
;Set up prog key table at PRGTAB
        DEFB    ESCAPE,"P",LOW PRGTAB,HIGH PRGTAB
;Change cursor To just the bottom raster blinking
        DEFB    ESCAPE,"Y",48H,09
;Change edit key to C5h
        DEFB    ESCAPE,"@","E"+80H
        DEFB    CR,LF,LF,LF
        DEFB    "FRED BLOGGS' MAGIC BIOS"
        DEFB    CR,LF,LF
        DEFB    "00k CP/M vers 2.2"
        DEFB    CR,LF,LF
SGNEND:


;Sample Programable key table
PRGTAB:
;Switch shifted and unshifted ":*" key,  (some keyboards have : shifted
;(very annoying)
        DEFB ":","*",0FFH
        DEFB "*",":",0FFH
;Program   M such that each time it is pressed MAP 00 IS MAGIC followed
;by   CR   will   be   returned,   (this   is   not   a   very   practical
;application!!!!),
        DEFB "M","MAP 00 IS MAGIC",0DH,0FFH
;Use graphics key to provide strings of a more useful nature than above
;8 bit ASCII keyboards are ideal for programming, and function keys
;can be modified in a key table to provide anything useful according to
;application,  any  BIOS warm boot should always reset the key table to
;an internal BIOS table or kill it completely (ESC "@").

        DEFB "A"+80H,"DIR A:",0DH,0FFH
        DEFB "B"+80H,"DIR B:",0DH,0FFH
        DEFB "P"+80H,"DIR P:",0DH,0FFH
        DEFB 0                          ;Table terminator
        DEFS 40                         ;Space for more to be added
                                        ;in edit mode
```

^Z

SECTION 4
==========

PROGRAM EXAMPLE NAS-SYS

And now an example of how the VFC can be interfaced with NAS-SYS, there are very obvious shortcomings to the example below but the following listing gives a simple demonstration. Here a non auto-boot system is assumed and an example of patching the VFC video and keyboard to NAS-SYS is given. The program can be considered in three parts:-

NASRET This initialises the VFC and patches the NAS-SYS workspace to include the VFC video out and keyboard in routines and then returns to NAS-SYS having switched the video switch enabling the VFC video. Type T0 100 CR and as expected memory 0 to 100H will be tabulated, this works because the keyboard characters entered are also being put into the normal Nascom video RAM at the same time and the NAS-SYS RLIN routine can read the line. The tabulate will be considerably slower than normal as the output is now going through two video routines, three times as much RAM is being scrolled and the VFC is producing WAIT states during screen access. In normal program use ony the VFC video would be used.

VIDOUT Outputs a character in A to the VFC, certain NAS-SYS control codes are changed.

NAS-SYS	VFC	
CR (£0D)	CR,LF	(£0D,£0A)
DU (£17)	CR	(£0D)
CLS (£0C)	CLS	(£1A)
CUL (£11)	CUL	(£1C)
CUR (£12)	CUR	(£1D)
CUU (£13)	CUU	(£1E)
CUD (£14)	CUD	(£1F)
ESC (£1B)	CR,CEL	(£0D,£1B,£2A)
CSL (£15)	CSL	(£16)
CSR (£16)	CSR	(£17)
LF (£0A)	ESC	(£1B) ;Use instead of £1B when calling ;VFC ESC routines

LF (^J) has been used to send ESC (1BH) to the VFC so that escape routines can be sent, try typing ^J E.

KBDIN If fitted this routine will scan a VFC keyboard as well as the normal Nascom keyboard. To edit a line of characters on the VFC type ^@ (VFC keyboard only) to enter the edit mode and use the cursor keys etc to modify the line, when CR is pressed the entire line will be returned.

```
VFCLOC    EQU £A000           ;VFC working location (arbitary)
VINIT     EQU VFCLOC+3        ;VFC Initialisation routine
VIDEO     EQU VFCLOC+12       ;VFC video out routine
STAT      EQU VFCLOC+5        ;
KBD       EQU VFCLOC+7        ;VFC keyboard in routine
SCAL      EQU £10             ;
;
;
VSWICH    EQU £EF             ;Port for selecting video 2
CTRL      EQU £EC             ;VFC control port
EVMASK    EQU 03              ;CTRL port enable bits
INITV     EQU £A0             ;Initial VFC status with VSOFT and  VRAM
                              ;off and 256 character set 1 selected
                              ;page £A000
;
;
          ORG  £5000          ;Arbitary location
```

```
NASRET    CALL INVFC          ;Initialise VFC and return to NAS-SYS
          RST SCAL            ;
          DEFB £5B            ;
;
;
;Initialise the VFC and set NAS-SYS vectors to call the VFC
;video routines not the NAS-SYS ones
INVFC     LD IX,VW            ;Point to workspace
          LD A,INITV
          OR £A0              ;
          XOR EVMASK          ;
          OUT (CTRL),A        ;Select VFC
          CALL VINIT          ;Initialise VFC
          LD HL,VIDOUT
          LD (£0C78),HL       ;Set up user output vector
          LD HL,KBDIN         ;OMIT THIS LINE IF NO ASCII KEYBOARD
          LD (£0C7B),HL       ;OMIT THIS LINE IF NO ASCII KEYBOARD
          RST SCAL            ;
          DEFB £55            ;
          IN A,(VSWICH)       ;Switch to VFC screen
          LD A,£0C            ;NAS-SYS Clear screen
;Loop through and clear the VFC screen

;Video output patch, enter with character to be printed in A
VIDOUT    CALL NEWSP          ;Avoid stack overflow
          PUSH HL             ;Save registers
          PUSH DE
          PUSH BC
          PUSH IX
          PUSH AF
          LD IX,VW            ;Point IX to workspace
          LD HL,CHGTAB-1      ;Point HL to character change table
          LD C,A              ;Character to be matched into C
LOOP1     CALL TEST           ;Test for end
          JR NC,NOCHNG        ;End of table
          DEC A
          CP C
          JR Z,CHNG           ;Match found
;Get to next character
LOOP2     CALL TEST           ;Test for end of string
          JR NZ,LOOP2         ;Not at end
          JR NC,NOCHNG        ;End of table
          JR LOOP1
;Send string pointed to by HL
CHNG      INC HL
LOOP3     LD C,(HL)           ;New char into C
          PUSH HL             ;Save string pointer
          CALL GOVID          ;Print char
          POP HL
          CALL TEST           ;Test for end
          JR Z,DONE           ;String sent
          JR LOOP3            ;Print next
;Character sent unchanged
NOCHNG    CALL GOVID
;All sent
```

```
DONE    POP AF              ;Recover registers
        POP IX
        POP BC
        POP DE
        POP HL
        RET

;Test next location in table return Z if 00 or £FF, NC if 00
TEST    INC HL              ;Increment pointer
        LD A,(HL)           ;Get table character
        OR A                ;Test for table end
        RET Z               ;Z with NC
        INC A               ;Test for £FF
        SCF
        RET

;Set up and call VFC
SVID    LD DE,£A000         ;Offset in DE
        LD A,(IX)           ;Get VFC status
        XOR EVMASK          ;Enable bits
        OR £A0              ;
        OUT (£EC),A         ;Enable VFC
        CALL VIDEO          ;Go to VSOFT routine
        LD A,(IX)
        OUT (£EC),A         ;VFC out
        RET
;
;
;Table of characters requiring changing
;Each string is terminated by £FF
;Table is terminated by 00
CHGTAB  DEFB £0D  ,£0D,£0A      ,£FF
        DEFB £0C  ,£1A          ,£FF
        DEFB £17  ,£0D          ,£FF
        DEFB £11  ,£1C          ,£FF
        DEFB £12  ,£1D          ,£FF
        DEFB £13  ,£1E          ,£FF
        DEFB £14  ,£1F          ,£FF
        DEFB £1B  ,£0D,£1B ,£2A ,£FF
        DEFB £15  ,£16          ,£FF
        DEFB £16  ,£17          ,£FF
        DEFB £0A  ,£1B          ,£FF
        DEFB 0                   ;Terminator

;VFC workspace. Note CURSOR is at VW+3
VW      DEFS 23
;
;
NEWSP   LD (STACK-6),HL     ;Save HL
        LD HL,RETSP         ;Return address to pick up old stack
        LD (STACK-2),HL     ;
        POP HL              ;Return address
        LD (STACK-4),HL     ; to new stack
        LD (SPSAVE),SP      ;Save old stack
        LD SP,STACK-6       ;
```

```
            POP HL                  ;Recover HL
            RET                     ;
;
;
RETSP       LD SP,(SPSAVE)          ;Pick up old stack
            RET
;
;
            DEFS 50
STACK       EQU $                   ;
SPSAVE      DEFS 2                  ;
;
;
KBDIN       CALL NEWSP              ;Avoid over flow
            PUSH HL                 ;Save registers
            PUSH DE
            PUSH BC
            PUSH IX
            LD IX,VW                ;Point to workspace
            LD DE,VFCLOC            ;Tell VFC where it is
            PUSH DE                 ;
            LD A,(IX)               ;Get VFC status
            XOR EVMASK              ;Enable VSOFT and VRAM bits
            OR £A0                  ;Add location
            OUT (£EC),A             ;Bring in VFC
            CALL STAT               ;Get keyboard status
            POP DE                  ;
            JR NC,DONEK             ;No character available
            CALL KBD                ;Char available so get it
            SCF                     ;Signal got character
;
;
DONEK       PUSH AF                 ;
            LD A,(IX)               ;VFC status
            OUT (£EC),A             ;Send VFC into limbo
            JP DONE                 ;
```

SECTION 5
=========

CONTROL CODES
==============

^M 0DH CARRIAGE RETURN
The cursor is placed at the start of the cursor line

^J 0AH LINE FEED
The cursor is moved down one line, if the cursor is already on the bottom line the screen, starting one line below HOME (see ESC M) is scrolled up and the bottom line is cleared

^H 08H BACKSPACE
Destructive backspace, the cursor is moved one position left and a space is inserted, if the cursor is at HOME nothing happens.

^[1BH ESCAPE
Initiates command sequence, subsequent characters define the command.

^\ 1CH CURSOR LEFT
The cursor is moved left by one position, if it was on the first column of a line it will be moved to the last column of the line above but it will not be moved passed HOME, see ESC M.

^] 1DH CURSOR RIGHT
The cursor is moved right by one position, if it was on the last column of a line it will be moved to the first column of the line below but it will not be moved passed the last position on the screen.

^^ 1EH CURSOR UP
The cursor is moved up one line, remaining in the same column, the cursor will not be moved above the HOME line, see ESC M.

_ 1FH CURSOR DOWN
The cursor is moved down one line, remaining in the same column, the cursor will not be moved below the bottom line.

^K 0BH DELETE LINE
The cursor line is deleted and all lines below are scrolled up into it, the bottom line is cleared.

^N 0EH INSERT LINE
The cursor line and all below are scrolled down one line, the cursor line is cleared and the original bottom line is lost

^U 15H CURSOR HOME
The cursor is positioned to HOME, see ESC M.

^V 16H DELETE CHARACTER FROM LINE
The character under the cursor is deleted and all characters on that line to its right are moved one character left, the last column of the cursor line is cleared.

^W 17H INSERT CHARACTER IN LINE
The character under the cursor and all characters to its right are shifted one position right, the cursor position is cleared and the original character in the far right column is lost.

^Z 1AH CLEAR SCREEN
The screen from HOME (see ESC M) down is cleared.

^G 07H BELL
A negative pulse is generated at A01, which can be used to trigger an external alarm cicuit.

THE FOLLOWING ROUTINES ARE INITIATED BY SENDING THE ESCAPE CODE 1BH
ESCAPE 'k' KEYBOARD STATUS
Scans the keyboard and returns with the status in A, 00 no character available, FF character available. This routine only works to a keyboard plugged into the VFC keyboard socket.

ESCAPE 'K' KEYBOARD ENTRY

Wait until character is available from the keyboard, the cursor will blink according to state of ESC E/ESC 0.

ESCAPE ^V (16H) DELETE CHARACTER FROM SCREEN

The character under the cursor is deleted and all characters to its right and below are moved one character left, the first character on a line being moved to the last position of the line above, the last character position on the screen is cleared.

ESCAPE ^W (17H) INSERT CHARACTER IN SCREEN

The character under the cursor and all characters to its right and below are shifted one position right, the last character on a line is moved into the first position of the line below, the cursor position is cleared and the original character in the last character position is lost.

ESCAPE "=" R C POSITION CURSOR

The cursor is positioned to row R column C, the top left of the screen is row 0 column 0 and the arguments R and C are sent with an offset of 20H added to them, to position the cursor to the top left one would send 1BH,"=",20H,20H. It is possible to move the cursor into a screen location above HOME but it will not be moved if you try to move it completely off the screen.

ESCAPE 'R' X Y RESET PIXEL

The pixel at X,Y is reset, the top left of the screen is 0,0 and the arguments X and Y are sent with an offset of 20H added, if either argument is invalid the command is ignored.

ESCAPE 'S'X Y SET PIXEL

The pixel at X,Y is set, the top left of the screen is 0,0 and the arguments X and Y are sent with an offset of 20H added, if either argument is invalid the command is ignored.

ESCAPE 'T' X Y TEST PIXEL

The pixel at X,Y is examined, the top left of the screen is 0,0 and the arguments X and Y are sent with an offset of 20H added, if the pixel is set 01 is returned in A, if reset 00 is returned, if either of the X Y arguments is invalid 02 is returned.

ESCAPE "X" CLEAR TO END OF SCREEN

The character under the cursor and all characters below and to its right are cleared.

ESCAPE "*" CLEAR TO END OF LINE

The character under the cursor and all characters to its right on the same line are cleared.

ESCAPE "A" START ALTERNATE VIDEO

Subsequent characters printed will have bit 7 set, when inverse characters are selected (see ESCAPE I) characters will be printed in inverse video.

ESCAPE 'N' CANCEL ALTERNATE VIDEO
Cancels ESCAPE 'A'.

ESCAPE 'e' eNABLE CURSOR
Master cursor cursor enable this is used to enable the cursor when VSOFT is awaiting entry from the MAP VFC keyboard, the cursor will only be displayed when VSOFT is awaiting key entry unless ESC 'E' is active.

ESCAPE 'd' dISABLE CURSOR
Master cursor disable, the cursor is completely disabled.

ESCAPE 'E' ENABLE CURSOR
Enables the cursor to be displayed, the cursor will not show if the master cursor control ESC 'e' is not active.

ESCAPE 'D' DISABLE CURSOR
The cursor is disabled except when VSOFT is being used to obtain input from MAP VFC keyboard, and ESC 'e' is active.

ESCAPE 'I' SELECT INVERSE CHARACTERS
All characters in the video RAM having bit 7 set will be displayed as the inverse of the lower 128 character set. (see ESCAPE 'U').

ESCAPE 'U' SELECT UPPER CHARACTER SET
Selects 256 character set as opposed to inverse (see ESCAPE 'I').

ESCAPE 'M' MEMORY LOCK
Sets the HOME position to the start of the cursor line, this effectively becomes the top left of the screen and all lines above are locked out.

ESCAPE 'O' MEMORY LOCK OFF
The HOME position is restored to the top left of the screen.

ESCAPE 'P' AddressL AddressH SET PROGRAMABLE KEY TABLE AT Address
The table at Address, sent low byte first high byte second, will in future be scanned by KBDIN, if a key pressed matches one in the table the string in the table will be sent and not the actual key value. The table must be initialised before use, the format is

Key character to be changed,String (1 or more characters) to be sent,0FFH(string terminator). The last string terminator must be followed by a 00 which is table terminator.
e.g 01,'FRED',0FFH,02,00,0FFH,00
This would result in the string FRED being return when CTRL A was pressed and CTRL B would become 00 (BACKSPACE) the same as CTRL H.

ESCAPE 'Q' QUIT PROGRAMABLE KEY TABLE
Stop programable key table scan.

ESCAPE 'p' ENABLE PROGRAMABLE KEY TABLE

Re-start programable key scan at a table previously set up by ESCAPE 'P', this is used to restart a table that has been cancelled by ESCAPE '@'.

ESCAPE 'C' KEY STRING ESCAPE PROGRAM A KEY

Program a key using the table set up by ESCAPE 'P', if the key was previously programmed the original program will be cancelled and replaced. The following are special entries:-

ESCAPE 'C' ESCAPE
Clear table (00 is put into start of table)
ESCAPE 'C' KEY ESCAPE
Clear program of an individual key, the @ character is special and if this is sent a CR (0DH) will be entered, this enables strings followed by a CR to be entered to a table directly from the keyboard in EDIT mode.

ESCAPE '1' SELECT CHARACTER SET 1

Select character EPROM 1

ESCAPE '2' SELECT CHARACTER SET 2

Select character EPROM 2

ESCAPE '@' X RE-DEFINE EDIT ACTIVATION KEY

'@ (00) is normally used to activate the EDIT mode, this may be changed to any character X.

ESCAPE 'B' BLANK SCREEN

Video output ceases, but screen contents are not altered and access to video RAM is still allowed, no anti-flicker WAIT states are generated and high speed transfer to and from video RAM is possible e.g direct read/write to/from disk.

ESCAPE 'V' DISPLAY SCREEN

Re-display a screen previously blanked

ESCAPE 'X' GET A LINE ENTRY

Enter an internal keyboard entry mode, keys pressed will be acted upon directly enabling full on screen positioning and editing, when RETURN (0DH) is pressed the cursor line (excluding trailing spaces) will be made available and can be obtained by calling KBDST and KBDIN in the usual way.

ESCAPE 'Z' GET CURSOR LINE

The cursor line (excluding trailing spaces) is made available and can be obtained by calling KBDST and KBDIN in the usual way.

ESCAPE 'Y' 1 2 DEFINE CURSOR

The initial non-edit cursor is defined as the lower two rasters of the cursor position blinking, this may be changed to parameters 1 and 2 (see 6845 data sheet for details of cursor format).

SECTION 6
==========

IMPLEMENTATION NOTES

Only two links (L2 and L4) change with different computers.

A) NASCOM 1

Tell you later.

B) NASCOM 2 with VFC video and FDC auto-booting

De-select NAS-SYS on the main board by removing the link between 1 & 16 on the Nascom link block LKS1, the link between 2 & 15 enabling VRAM and VWRAM remains.

Link L2 a--d and b--c
Link L4 a--b

C) NASCOM 2 with VFC video non auto-booting

Link L2 a--b and c--d
Link L4 b--c

D) GEMINI G811 auto-booting

Link LKB1 on the G811 to reset at 0000H (see G811 manual)

Link L2 a--d and b--c
Link L4 a--b

E) GEMINI G813 auto-booting

Remove RPM 2.0 or 2.1 and insert BOOT813 (available from MAP 80 SYSTEMS)

Link L2 a--d and b--c
Link L4 b--c

F) GENERAL

L1 Video switch option

Link a-c if video switch is fitted
Link a-b if video switch is not fitted

L3 WD 2797 Clock frequency

Link b-c for 1 Mz. (Standard configuration for 5 1/4" Drives)
Link a-b for 2 Mz.

L5 Disk Ready line/Drive 4 selects etc. The following chart shows the connections for some commonly used drives.

DRIVE TYPE	LINK
Teac FD-55	a-b, c-d
Pertec FD250	a-b
Micropolis 1015	a-d, b-c

L6 Character generator PROM type selection

Link a-b if character generators are 2732 EPROMS
Link a-c if character generators are 2716 EPROMS

L7 Keyboard data selection

Link a-b if you are using a 7 bit keyboard
Link a-c if you are using an 8 bit keyboard

L8 FDC link

This link should be made ONLY if your board is populated as an FDC only.

L9 Write precompensation select

If you require that write precompensation is permanently enabled then no link is needed.
Link a-b if no write precompensation is required.
Link b-c if you require write precompensation on tracks 43 and above only.

L10 Clock selection

Link a-b to select the auxiliary clock
Link a-c to select the normal clock

L11 Ready signal enable

Link to use the ready signal from the disk drive

L12 Keyboard strobe polarity

Link a-b for Positive keyboard strobe
Link a-c for Negative keyboard strobe

L13 VFC Base Port selection

a 80-8FH
b 90-9FH
c A0-AFH
d B0-BFH Not recommended
e C0-CFH Recommended alternate
f D0-DFH
g E0-EFH Recommended
h F0-FFH Not recommended

L14 FDC link

This link should be made ONLY if your board is populated as an FDC only.

L15 NASIO (IOEXT)

This is only required by a Nascom 1 or 2 and should be made only if no other card is generating NASIO. The I/O link on either Nascom should be set to 'External'.

1) If you are running a CP/M system then the system size must not be larger than available ram, and must not overlay any EPROM.

2) Any EPROM card installed must be selected on page 2.

6) RAM MODIFICATIONS

1) If you are using a ram card other than the MAP 256k RAM and the VFC ram is underlying this ram then a modification to your ram board may be required if you wish to use the full complement of ram available. See below for Nascom Ram A, B and Gemini G802 modifications. (Nascom mods are standard for AVC).

Note : When using a G813 you will need to map out the on board 4k block from under the VFC during VSOFT or VFC VRAM access.

Ram A

Lift pin 1 of IC 35
Connect pin 1 of IC 35 to pin 10 of IC 35
Lift pin 13 of IC 21
Connect pin 13 of IC 21 to pin 2 of IC 35
Connect 10k ohm resistor from pin 2 of IC 35 to +5v
Note : No EPROM can now be decoded on this board.

Ram B

Remove link 5
Link test pin WREN to test pin RDEN

Note : Read and write planes are now inoperative but this should not cause any problems since the facility is rarely used.

8802

Disconnect IC48 pin 10 from +5v
Connect IC48 pin 10 to IC22 pin 4

Note : You must select page mode operation on your 8802

2) An alternative to modifying your ram board is to address VSOFT above your available ram. or in the case of a 64k ram board address VSOFT at F000.
With CP/M and a 64k ram card simlpy generate a 60k system and again address VSOFT at F000.

SECTION 7
=========

VFC PIN ASSIGNMENTS

A) VIN

Pin	
1	0v
2	Composite video signal in
3	0v

B) VOUT

Pin	
1	0v
2	Composite video signal out
3	0v

C) FDC (J2)

Pin	
6	See Note 1
8	Index pulse
10	Drive select 0/A
12	Drive select 1/B
14	Drive select 2/C
16	Motor on
18	Direction select
20	Step
22	Write data
24	Write gate
26	Track 0
28	Write protect
30	Read data
32	Side select
34	See Note 1

Note 1 - Pin assignments differ between drives for pin's 6 & 34. The chart below gives details for several commonly used drives.

DRIVE TYPE	PIN 6	PIN 34
Teac FD-55	Drive select 3/D	Ready
Mitsubishi FD250	Drive select 3/D	Spare
Micropolis 1015	Ready	Drive select 3/D

These drives can all be accomodated by appropriate linking of Link 5. See Section 7 for details.

Note 2 - All odd pins on 34 way connector are grounded.

D) KEYBOARD (J1)

Pin		Pin	
1	+5v	9	Spare
2	Strobe	10	D2
3	Spare	11	D3
4	0v	12	D0
5	D5	13	D1
6	D4	14	D7
7	D6	15	-12v
8	0v	16	Spare

SECTION 8
==========

COMPONENT LIST

IC's
====

1	D6208 ✓
2,19	74LS08 ✓✓
3	74LS86 ✓
4	74LS11 ✓
5,20,25	74LS32 ✓✓✓
6	7438 ✓
7	7406 ✓
8	74LS367 ✓
9,12,29	74LS74 ✓✓✓
11	74S74 ✓
10,13,32	74LS174 ✓✓✓
14	74LS465 (81LS95) ✓
15,31	74LS123 ✓✓
16	74LS165 ✓
[illegible]	74S04 ✓
18	74LS93 ✓
21	WD2797 ✓
22	2716/2732 (CHAR SET 2)
23,24	74LS257 ✓✓
26	74LS10 ✓
27	2716/2732 (CHAR SET 1) ✓ 1·1
28	6845 ✓
30	74LS14 ✓
33,42	74LS273 ✓✓
34,41	74LS138 ✓✓
35	4802/6116 ✓ 4802
36,43	74LS245 ✓✓
37	2716 (VSOFT) ✓
38,39,40	74LS157 ✓✓✓
44	74LS85 ✓
45,46	74LS244 ✓✓
47	74LS266 ✓

[illegible]C ONLY - 43,45,46,47,41,34,31,30,26,21,20,19,18,17,15,14,13,12,8,7
6,5

VIDEO ONLY - 2,3,4,5,9,10,11,12,16,17,18,19,20,25,26,29,30,27,28,32,33
34,35,36,37,38,39,40,41,42,43,44,45,46,47

UPGRADE A - 1,23,24

RESISTORS
==========

1	82R Grey R Blk ✓
2,3	680R Blue Grey Brn ✓
4,5	4K7 } Y V R ✓
22,23,24,25	4K7 }

6,7,8,9,10	150R Br Gn Br ✓
11	10R Br Blk Blk ✓
12	5K6 Gn Blk R ✓
13	100K Br Blk Y ✓
14,18	1K Br Blk R ✓
16,17	330R O O Br ✓
19	39K O W O ✓
20	270K R V ✓
21	560K Gn Blk O ✓
P1	100R POT / 150 LIN
P2	10K POT ✓
P3	50K POT ✓

FDC ONLY - 22,23,24,25,21,20,18,14,12,13,7,8,9,10,6,P2,P3

VIDEO ONLY - 1,2,3,4,5,11,16,17,19,22,23,24,25,P1

CAPACITORS
==========

1	220pf ✓
24 +l	1uf TANT ✓
9,25	33pf ✓
14	0.22uf
30	10uf TANT ✓
31,35,42	47uf TANT ✓ 6.3
43,44	2.2uf TANT 16v ✓
VC1	5-65pf TRIMMER ✓?

ALL OTHER CAPACITORS ARE 0.01uf

FDC ONLY - VC1,35,42,30,31,14,24,25,36,38,39,40,41,34,29,27,26,19,18
17,15,10,11,12,13,6,5,4

VIDEO ONLY - 1,16,9,35,42,2,3,4,7,8,10,15,17,18,22,23,26,27,28,29,32
33,34,36,37,38,39,40,41

UPGRADE A - 20,21,43,44

MISC
====

T1,T2	2N3904 ✓✓
XTAL	14.31818 Mhz CRYSTAL ✓
J1	16 WAY IDC ✓
J2	34 WAY IDC ✓
VIN,VOUT	3 PIN CONNECTOR ✓✓
D1	IN4148 ✓?

FDC ONLY - J2,D1

VIDEO ONLY - T1,T2,XTAL,VOUT,

UPGRADE - J1,VIN